sensors

Special Issue Reprint

Vision Sensors for Object Detection and Tracking

Edited by
ByoungChul Ko

mdpi.com/journal/sensors

MDPI

Vision Sensors for Object Detection and Tracking

Vision Sensors for Object Detection and Tracking

Guest Editor

ByoungChul Ko

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editor
ByoungChul Ko
Department of Computer
Engineering
Keimyung University
Daegu
Republic of Korea

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Sensors* (ISSN 1424-8220), freely accessible at: https://www.mdpi.com/journal/sensors/special_issues/03ESJ2194W.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-9115-3 (Hbk)
ISBN 978-3-7258-9116-0 (PDF)
https://doi.org/10.3390/books978-3-7258-9116-0

© 2026 by the authors. Articles in this reprint are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The reprint as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (https://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

Ching-Chang Wong, Tai-Ting Tsai and Can-Kun Ou
Integrating Heuristic Methods with Deep Reinforcement Learning for Online 3D Bin-Packing Optimization
Reprinted from: *Sensors* **2024**, *24*, 5370, https://doi.org/10.3390/s24165370 . . . . . **1**

Mumuxin Cai, Xupeng Wang, Ferdous Sohel and Hang Lei
Diffusion Models-Based Purification for Common Corruptions on Robust 3D Object Detection
Reprinted from: *Sensors* **2024**, *24*, 5440, https://doi.org/10.3390/s24165440 . . . . . **18**

Yongsoo Kim, Jaehyeon So, Chanwook Hwang, Wencan Cheng and Jong Hwan Ko
An Energy-Efficient Dynamic Feedback Image Signal Processor for Three-Dimensional Time-of-Flight Sensors
Reprinted from: *Sensors* **2024**, *24*, 6918, https://doi.org/10.3390/s24216918 . . . . . **38**

Md Tanzil Shahria and Mohammad H. Rahman
Activities of Daily Living Object Dataset: Advancing Assistive Robotic Manipulation with a Tailored Dataset
Reprinted from: *Sensors* **2024**, *24*, 7566, https://doi.org/10.3390/s24237566 . . . . . **59**

Yoram Furth and Stanley R. Rotman
Efficacy of Segmentation for Hyperspectral Target Detection
Reprinted from: *Sensors* **2025**, *25*, 272, https://doi.org/10.3390/s25010272 . . . . . **79**

Jaehoon Kim and Byoung Chul Ko
Scene Graph and Natural Language-Based Semantic Image Retrieval Using Vision Sensor Data
Reprinted from: *Sensors* **2025**, *25*, 3252, https://doi.org/10.3390/s25113252 . . . . . **104**

Tianhang Weng and Xiaopeng Niu
Enhancing UAV Object Detection in Low-Light Conditions with ELS-YOLO: A Lightweight Model Based on Improved YOLOv11
Reprinted from: *Sensors* **2025**, *25*, 4463, https://doi.org/10.3390/s25144463 . . . . . **123**

Jianying Yuan, Hao Li, Chen Cheng, Zugui Liu, Sidong Wu and Dequan Guo
Enhanced SOLOv2: An Effective Instance Segmentation Algorithm for Densely Overlapping Silkworms
Reprinted from: *Sensors* **2025**, *25*, 5703, https://doi.org/10.3390/s25185703 . . . . . **143**

Hye Rim Kim and Byoung Chul Ko
Keyword-Conditioned Image Segmentation via the Cross-Attentive Alignment of Language and Vision Sensor Data
Reprinted from: *Sensors* **2025**, *25*, 6353, https://doi.org/10.3390/s25206353 . . . . . **162**

Dong Feng, Hengyuan Liu and Zhiyu Liu
Lightweight 3D Multi-Object Tracking via Collaborative Camera and LiDAR Sensors
Reprinted from: *Sensors* **2025**, *25*, 7351, https://doi.org/10.3390/s25237351 . . . . . **181**

Jae Kwan Park and Ji-Hyeong Han
MemLoTrack: Enhancing TIR Anti-UAV Tracking with Memory-Integrated Low-Rank Adaptation
Reprinted from: *Sensors* **2025**, *25*, 7359, https://doi.org/10.3390/s25237359 . . . . . **207**

Pierluigi Rossi, Elisa Cioccolo, Maurizio Cutini, Danilo Monarca, Daniele Puri, Davide Gattamelata and Leonardo Vita
Outdoor Characterization and Geometry-Aware Error Modelling of an RGB-D Stereo Camera for Safety-Related Obstacle Detection
Reprinted from: *Sensors* **2025**, *25*, 7495, https://doi.org/10.3390/s25247495 . . . . . **232**

sensors

MDPI

Article

Integrating Heuristic Methods with Deep Reinforcement Learning for Online 3D Bin-Packing Optimization

Ching-Chang Wong *, Tai-Ting Tsai and Can-Kun Ou

Department of Electrical and Computer Engineering, Tamkang University, New Taipei City 25137, Taiwan; 609470124@gms.tku.edu.tw (T.-T.T.); 611470070@o365.tku.edu.tw (C.-K.O.)
* Correspondence: wong@ee.tku.edu.tw

Abstract: This study proposes a method named Hybrid Heuristic Proximal Policy Optimization (HHPPO) to implement online 3D bin-packing tasks. Some heuristic algorithms for bin-packing and the Proximal Policy Optimization (PPO) algorithm of deep reinforcement learning are integrated to implement this method. In the heuristic algorithms for bin-packing, an extreme point priority sorting method is proposed to sort the generated extreme points according to their waste spaces to improve space utilization. In addition, a 3D grid representation of the space status of the container is used, and some partial support constraints are proposed to increase the possibilities for stacking objects and enhance overall space utilization. In the PPO algorithm, some heuristic algorithms are integrated, and the reward function and the action space of the policy network are designed so that the proposed method can effectively complete the online 3D bin-packing task. Some experimental results illustrate that the proposed method has good results in achieving online 3D bin-packing tasks in some simulation environments. In addition, an environment with image vision is constructed to show that the proposed method indeed enables an actual robot manipulator to successfully and effectively complete the bin-packing task in a real environment.

Keywords: 3D bin-packing; deep reinforcement learning; proximal policy optimization; heuristic algorithms

1. Introduction

The three-dimensional bin-packing (3D-BP) problem is a classic combinatorial optimization problem aimed at arranging objects within container boxes to maximize space utilization [1,2]. Historically, the manual arrangement of items for packing has been labor-intensive and often yields suboptimal packing configurations. The 3D-BP problem is widely regarded as an NP-hard problem and can be categorized into offline and online. In the offline 3D-BP problem, the numbers and sizes of objects are known in advance, while the online 3D-BP problem is more challenging as it requires packing objects neatly into containers without prior knowledge of their quantities or sizes.

In the realm of packing problems, the one-dimensional bin-packing (1D-BP) problem stands out as a typical combinatorial optimization challenge, aiming to fit numerous items into the fewest number of boxes with predetermined capacities [3,4]. A review of relevant algorithms addressing the 1D-BP problem over the past two decades reveals its complexity and provides insights into its solution [5]. As dimensions increase, from two dimensions (2D) to three dimensions (3D), packing complexity escalates accordingly.

Early studies on 3D packing tasks primarily relied on well-designed heuristic algorithms [2,6–8], often inspired by manual packing experiences. For instance, research [9] introduced the layer-based approach three-dimensional bin-packing problem, dividing boxes into layers along two possible directions to maximize item arrangements. Expanding upon this approach, research [10] has proposed a three-stage layer-based heuristic to tackle the three-dimensional single-bin-size bin-packing problem. Additionally, research [11]

presented a multi-objective 3D packing algorithm optimizing container usage and balancing weights across containers. The research in [12] combined research [9,11] to devise a multi-stage optimization algorithm for solving the 3D-BP problem.

Despite the effectiveness of heuristic algorithms, inherent limitations exist, such as lack of learning capabilities [13] and exponential growth in solution time with increasing problem sizes [14], rendering reliance solely on heuristic algorithms impractical for achieving optimal packing. Recently, deep reinforcement learning (DRL) has demonstrated promising performance across various domains, prompting its application in 3D packing problems [15–20]. Leveraging DRL enables continuous learning from manual packing experiences, aligning packing outcomes with human preferences. Particularly, addressing online 3D-BP using DRL has become a recent research trend [21,22]. Methods such as model-based reinforcement learning proposed by the research in [23], based on research [24], optimize container space utilization through DRL. Effective and easily implementable constrained DRL methods within the actor–critic framework were introduced by research [25]. Moreover, research [26] proposed an adjustable robust reinforcement learning approach to tackle online 3D-BP, effectively adjusting robustness weights to achieve an ideal balance of policy performance across various environmental conditions.

Given the NP-hard nature of the 3D-BP problem and the limitations of heuristic algorithms, there is a clear research gap in developing effective and efficient methods for online 3D-BP. This study aims to address this gap by combining reasonable constraints and heuristic algorithms with DRL to improve space utilization in practical scenarios involving robot manipulators for packing. These constraints include placement direction constraints and partial support constraints, which are necessary due to the limitations of robot manipulators without specially designed end effectors. The placement direction constraints prohibit the rotation of objects along the x and y axes, while partial support constraints increase the possibilities for stacking objects and enhance overall container space utilization.

This study is organized into four sections. Section 1 introduces the background and research objectives. Section 2 details the methodology for solving the 3D-BP problem using heuristic algorithms in conjunction with a DRL system. Section 3 describes the experimental environment established for the online 3D-BP task, along with the presentation of simulation and actual experimental results. Finally, Section 4 provides the conclusions and future work.

2. Methodology

This study adopts the classic and mainstream extreme point heuristic algorithm [27] within the realm of heuristic algorithms and enhances it with custom-designed modifications tailored to the overall online 3D-BP tasks.

2.1. Extreme Point Sorting Method

In the packing task, the current placement point of each item utilizes the extreme point method proposed by research [27], where each item's stacking generates three extreme points: upper, left-front, and right-rear. Placing items at the upper extreme point may result in residual space due to varying item sizes. This residual space cannot accommodate all items, leading to space wastage. To minimize space wastage, we devise the extreme point priority sorting method to place items in positions with minimal wasted space. The operation flow is shown in Figure 1.

The method calculates the residual space generated after placing the current item at all placement points. If the residual space cannot accommodate all previously placed items, it is considered wasted space. If the residual space can accommodate one item, it is considered available space. All placement points are sorted based on wasted space, with those producing less wasted space prioritized. The prioritization sequence for placement points is as follows: points with no residual space, points with available space, and finally,

points with wasted space, arranged from lowest to highest wasted space. Algorithm 1 provides the pseudocode for the extreme point priority sorting method.

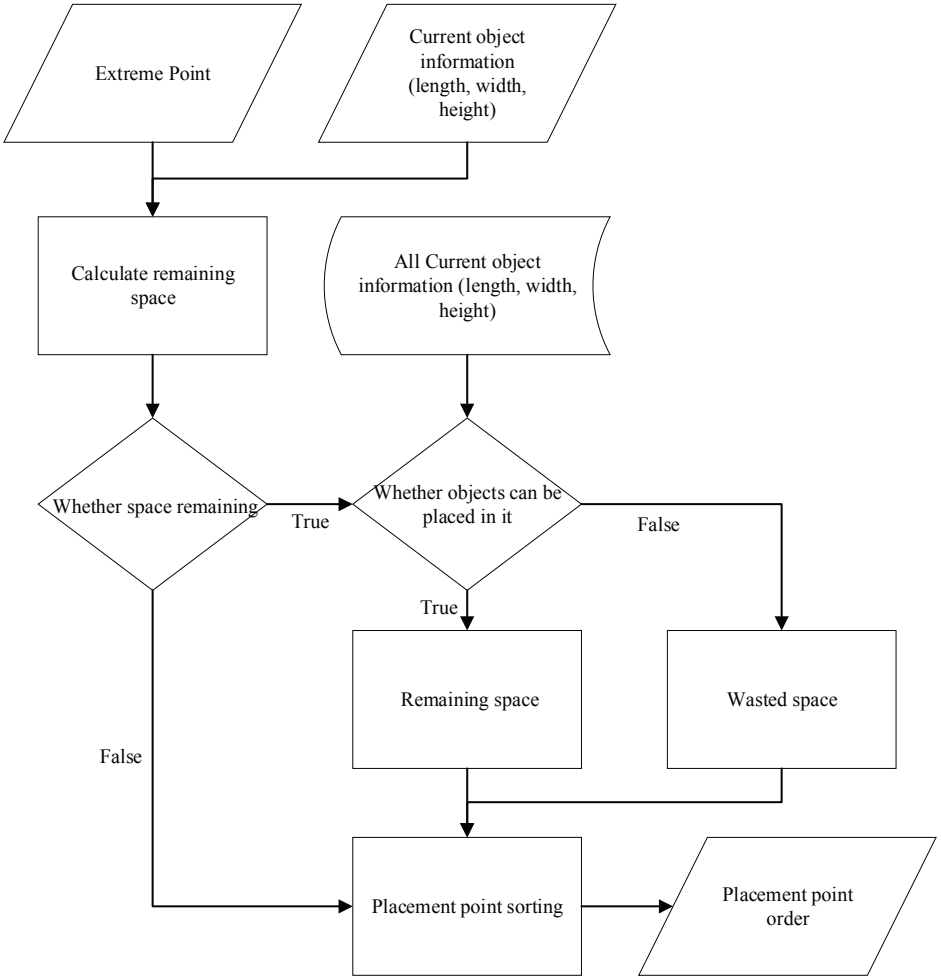


Figure 1. Flowchart of the extreme points priority sorted.

Algorithm 1: Extreme Point Priority Sorting Method

```
Input EP: List of Extreme Points
Input I: Item to be added to the 3D bin
Input K: List of items already in the 3D bin
function Calculate_remain_space(I, EP):
    return space remaining
function Dec_waste_space(rem_space_i):
    if space remaining < for all object ∈ K
        return True
function Max_score(EP, score_list):
    return EP list ← score from high to low
for all i ∈ EP do
    rem_space_i = Cauculate_remain_space(I, i)
    if Dec_waste_space(rem_space_i) then
        score_list ← - rem_space_i
    else
        score_list ← 0
    end if
end for
return Max_score(EP, score_list)
```

When placing objects of different sizes, L-shaped residual spaces can be formed. Before calculating the residual space, the space generated after placement is divided into two residual spaces: one along the x-axis and one along the y-axis. The space division method follows the approach mentioned in the research of [5]. The outermost vertices of the object cut the residual space along the x-axis and y-axis. To utilize the space more efficiently, the direction with the longer residual axis is given a larger space. The overlapping space is defined as transferable space, which is allocated to the residual space along the longer axis.

2.2. Packing Constraints

Packing constraints are rules derived from manual packing experiences, designed to optimize the packing task. This study uses container space state representation to implement the constraints of the heuristic algorithm and packing constraints, including a simple non-complete support constraint.

2.2.1. Container Space State Representation

To enable the heuristic algorithm and related constraints to function effectively in the packing task, the state of the container space must first be represented. There are various methods for handling container space. For instance, the method used by study [28] involves cutting the remaining container space after placing an item and using an algorithm to determine whether the item can be placed to fit the remaining space. In contrast, study [25] proposed creating a 2D height map for container space projection, where the values filled in the 2D map represent the current height. However, the height map overlooks three-dimensional information, potentially causing discrepancies in some situations. To address this, study [13] proposed using a 3D grid to represent three different space states, wasted space, available space, and unavailable space, using the 3D grid as input to neural networks.

Building on the method by study [13], this study proposes creating a 3D grid to represent container space states, but with consideration for the support state of the space. The container space is divided into grids with a volume of 1, where each grid's value indicates the current space state. The three defined space states, as shown in Figure 2, are space occupied, empty space without item support, and empty space with item support. This container space state representation allows for the evaluation of the rationality of item stacking within the container.

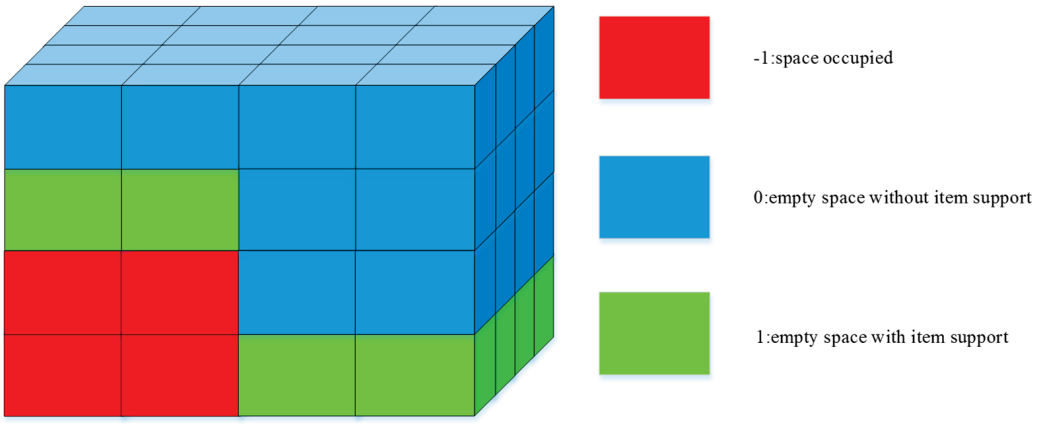


Figure 2. Three-dimensional grid state diagram.

2.2.2. Partial Support Constraints

In traditional methods for stacking objects, complete support constraints are used to maintain static stability. This requires that the bottom of the object be fully supported by other objects or the container itself, prohibiting any overhanging placements. However, complete support constraints overly restrict space utilization and limit the heuristic algorithm's flexibility.

In this study, we introduce partial support constraints to allow for more effective space use while maintaining stability. This is achieved through the object's area and the center of the bottom of the object, with three main constraints: half the length of the object L_{object} must be less than the length of the placement space L_{space}, half the width of the object W_{object} must be less than the width of the placement space W_{space}, and the position directly below the center of the object $object_{pos}$ must be supported by another object or the container. These constraints are mathematically represented by

$$\frac{1}{2}L_{object} < L_{space} \tag{1}$$

$$\frac{1}{2}W_{object} < W_{space} \tag{2}$$

$$object_{pos} = (x_{center}, y_{center}, z_{center}) \tag{3}$$

By adhering to these three constraints, objects achieve partial support, ensuring stability and preventing collapse due to inadequate support. Combining the extreme point method with partial support constraints increases the possibilities for item placement, thereby enhancing space utilization.

2.3. Integration of the Heuristic Algorithm with Deep Reinforcement Learning

In online 3D-BP tasks, training solely with DRL can result in poor training outcomes due to significant discrepancies in the definition of strategy quality. Therefore, heuristic algorithms are typically integrated into the training process. This section explains how heuristic algorithms and constraints are integrated into DRL, divided into three parts: extreme points and extreme point priority sorting constraints, deepest bottom left with fill, and reward function design.

2.3.1. Extreme Points and Extreme Point Priority Sorting Constraints

This study integrates extreme points and the extreme point priority sorting method into the action strategy of DRL. When placing objects into the target container, the current extreme point is identified using the extreme point algorithm, which becomes the chosen placement for the action strategy. As illustrated in Figure 3, the currently placed object generates three extreme points: upper, right, and front. The extreme points generated by previously placed objects are categorized as other extreme points. The upper extreme point is classified as another extreme point in the strategy selection. Thus, the action strategy offers three placement positions and two placement directions. Among the other extreme points, many placement points exist. Using the designed extreme point priority sorting method, the wasted space caused by each placement is calculated and sorted, arranging the extreme points from least to most wasted space. To determine whether an object can be placed in the container, the 3D grid state of the container is used to prevent unreasonable placement scenarios.

2.3.2. Deepest Bottom Left with Fill

The Deepest Bottom Left with Fill (DBLF) heuristic algorithm [29] aims to place objects as close to the target container's corners and edges as possible. Building on the method by the research in [13], this study integrates DBLF into deep reinforcement learning by creating a 3D grid-based placement score space, similar to the 3D grid state method for container space. The container space is divided into grids with a volume of 1, where each grid's value represents the placement score of an object.

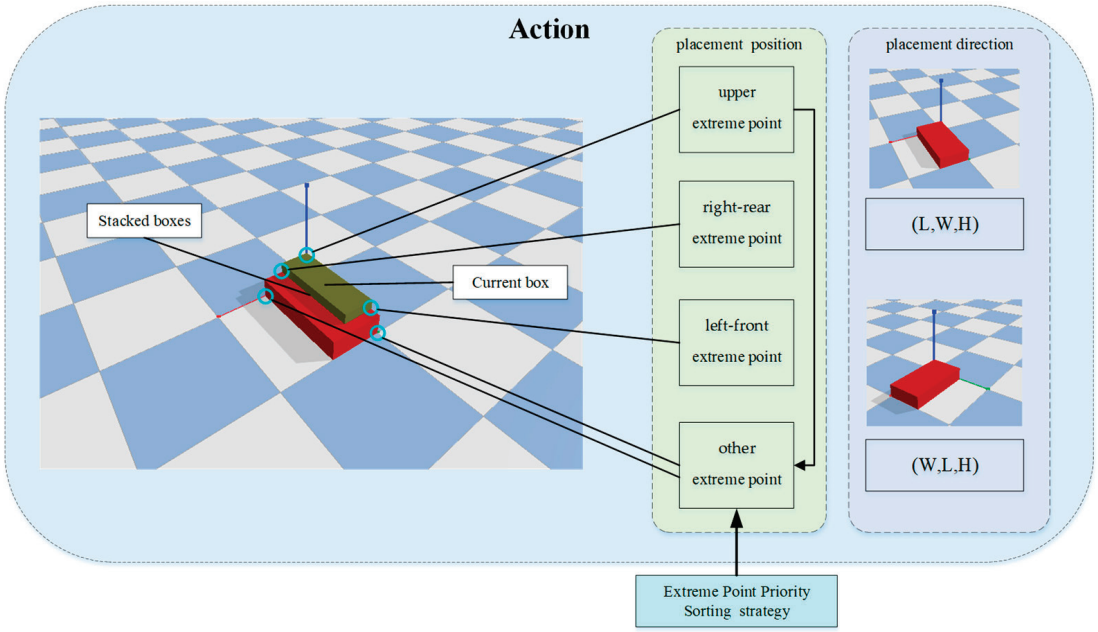


Figure 3. Action strategy diagram.

Initially, the space score values are set to decrease from the bottom-left corner to the top-right corner, as illustrated in Figure 4. The figure shows a 2D score distribution, and each layer follows the depicted distribution. Stacking these distributions forms the initial 3D score space. The scoring is primarily based on whether there are objects near the space. As shown in Figure 5, if a space is already occupied, its grid score is removed; if a space is unoccupied but has neighboring objects, its grid score is increased by 3 points; if a space is unoccupied and has no neighboring objects, its grid score remains unchanged.

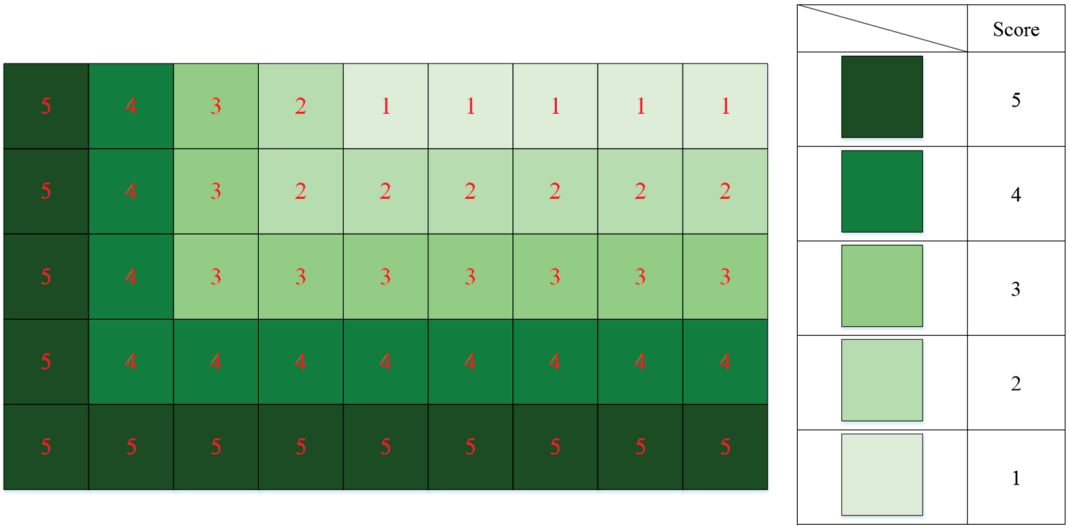


Figure 4. Diagram of initial space score.

The reward function is designed to encourage the placement of objects in higher-scoring spaces, achieving the DBLF algorithm's effect. This method ensures that objects are placed efficiently, maximizing space utilization by guiding the placement towards optimal positions within the container.

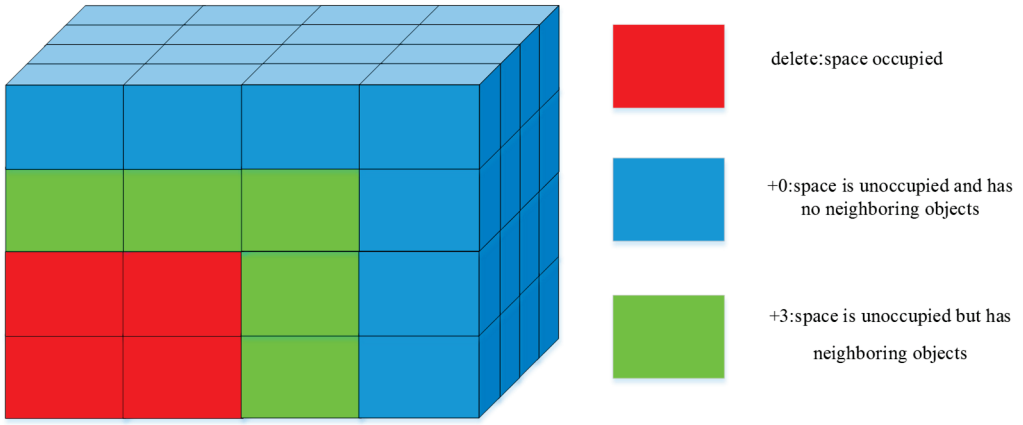


Figure 5. Diagram of space score.

2.3.3. Reward Function Design

This study uses Proximal Policy Optimization (PPO) [30] combined with heuristic algorithms to train the online 3D-BP task. The reward function, a critical component in integrating heuristic algorithms, is primarily based on the placement score of objects. The reward score is derived from the placement space score, supplemented by the wasted space generated after placing the object. The less wasted space produced, the higher the reward score.

The feasibility of the chosen placement position and direction also affects the reward score. If an object cannot be placed, a penalty is applied by deducting an appropriate amount from the reward score. The final total reward score considers the current space utilization rate for overall scoring. The reward function in the online 3D-BP environment uses a proportional addition format. When an object cannot be placed, it signifies the end of an episode. The total reward for that episode, R_{total}, can be expressed by

$$R_{total} = \gamma \cdot \frac{\sum_{i=0}^{n} S_i}{n} + \beta \cdot waste_{first} + \beta \cdot (\sigma - S_{ratio}) \tag{4}$$

where S_i is the score obtained from placing the i object, and n is the total number of objects placed in that episode. The first term represents the average placement score for the episode. $waste_{first}$ is the area of the unused space at the bottom of the container at the end of the episode. σ is a standard for space utilization, which is not a standard deviation but an average value of the standard measure for space utilization, adjusted after running the bin-packing task multiple times. The exact number of runs can vary. S_{ratio} is the overall space utilization ratio for the episode, which can be expressed by

$$S_{ratio} = \frac{\sum_{i=0}^{n} v_i}{L \cdot W \cdot H} \cdot 100 \tag{5}$$

where v_i is the volume of the current object, and L, W, H are the length, width, and height of the container, respectively. The third term represents the difference between the space utilization and the standard value. β and γ are constant proportions that must satisfy the condition $2\beta + \gamma = 1$. In Equation (4), S_i refers to the score obtained from placing the i object, which is also the reward R_{step} for each placement step. The formula can be expressed by

$$R_{step} = S_i = \begin{cases} \rho \cdot S_{score}(pos_i, ori_i) - \mu \cdot waste_i, & if\ action_i = 0\ or\ action_i = 1 \\ \omega \cdot S_{i-1} + \tau \cdot S_{score}(pos_i, ori_i), & if\ action_i = 2 \end{cases} \tag{6}$$

where S_{score} refers to the grid score, which is the sum of the scores of the grids occupied by the placed object. pos$i$ represents the position of the i placement, ori_i i denotes the

orientation of the i placement, and $waste_i$ is the wasted space generated during the i object placement. S_{i-1} is the score obtained from the previous placement, and $action_i$ is the action taken for the i placement. The constants ρ, μ, τ, and ω are predefined ratios. When $action_i = 0,\ 1$, it indicates that the action strategy selects the front extreme point or the right extreme point. In these cases, S_i is calculated by subtracting the ratio of wasted space from the placement score. When $action_i = 2$, it indicates that the action strategy selects other extreme points, which have been filtered through the extreme point priority sorting method. These selected points are generally the best current positions. To prevent the heuristic algorithm from overly interfering with the learning of the action strategy, Si is set to the sum of the placement score from the $i-1$ placement and a proportion of the grid score from the current placement. The reward parameters used in the function are listed in Table 1.

Table 1. Parameter table of reward function.

Parameters	Definition	Value
σ	Standard for space utilization	80
β	Constant	0.2
γ	Constant	0.6
L	Length of container	400 (cm)
W	Width of container	300 (cm)
H	Height of container	200 (cm)
ρ	Constant	0.7
μ	Constant	0.3
τ	Constant	0.2
ω	Constant	0.8

During the bin-packing process, two situations may arise: (i) The currently chosen placement position is not feasible, while other positions are feasible. (ii) The currently chosen orientation is not feasible, while another orientation is feasible. For these situations, two penalty mechanisms are designed to deduct placement scores and described by

$$S_i = \begin{cases} S_i - 10,\ if\ D_{stack}(pos_i,\ ori_i) = False \\ \quad and\ D_{stack}(pos_n,\ ori_i) = True \\ S_i - 5,\ if\ D_{stack}(pos_i,\ ori_i) = False \\ \quad and\ (D_{stack}(pos_i,\ ori_n) = True \\ \quad or\ D_{stack}(pos_n,\ ori_n) = True) \end{cases} \tag{7}$$

where D_{stack} is used to determine if the placement is feasible through the grid state S_{state}. If the grid states occupied by the placement position comply with the constraints, the D_{stack} outputs *True*; otherwise, it outputs *False*. The variables pos_n and ori_n represent the new placement position and orientation, respectively. If either of the two situations occur, the episode continues, but an appropriate placement score S_i is deducted as a penalty for making an incorrect choice. By integrating these penalties and constraints into the reward function, the deep reinforcement learning model is guided to improve its decision-making process, leading to more efficient and optimal packing solutions.

3. Results and Discussion

3.1. System Architecture and Experimental Environment

The overall system architecture of this study is shown in Figure 6. In the motion control section of the robot manipulator, the MoveIt! motion planning module is utilized to plan the movements of the robot manipulator. This ensures that the manipulator can avoid all obstacles in its path and move quickly and accurately to the target position. In the visual assistance part, RGB-D images from a depth camera are used to obtain the three-dimensional length information and position of the target object. The obtained target object position is then converted into coordinates so that the robot manipulator can reach

the target object to retrieve it. The information about the target object obtained is one of the inputs for the deep reinforcement learning system. Finally, the Robot Operating System (ROS) is used to transmit information between various modules and components, allowing the robot manipulator to complete the online 3D packing task. An actual experimental environment of this study is shown in Figure 7. The depth camera is mounted above the inspection area on the workbench to obtain the position and information of the target objects. The robot manipulator is installed on the right side of the workbench, with its end effector being a suction cup used for picking up target objects.

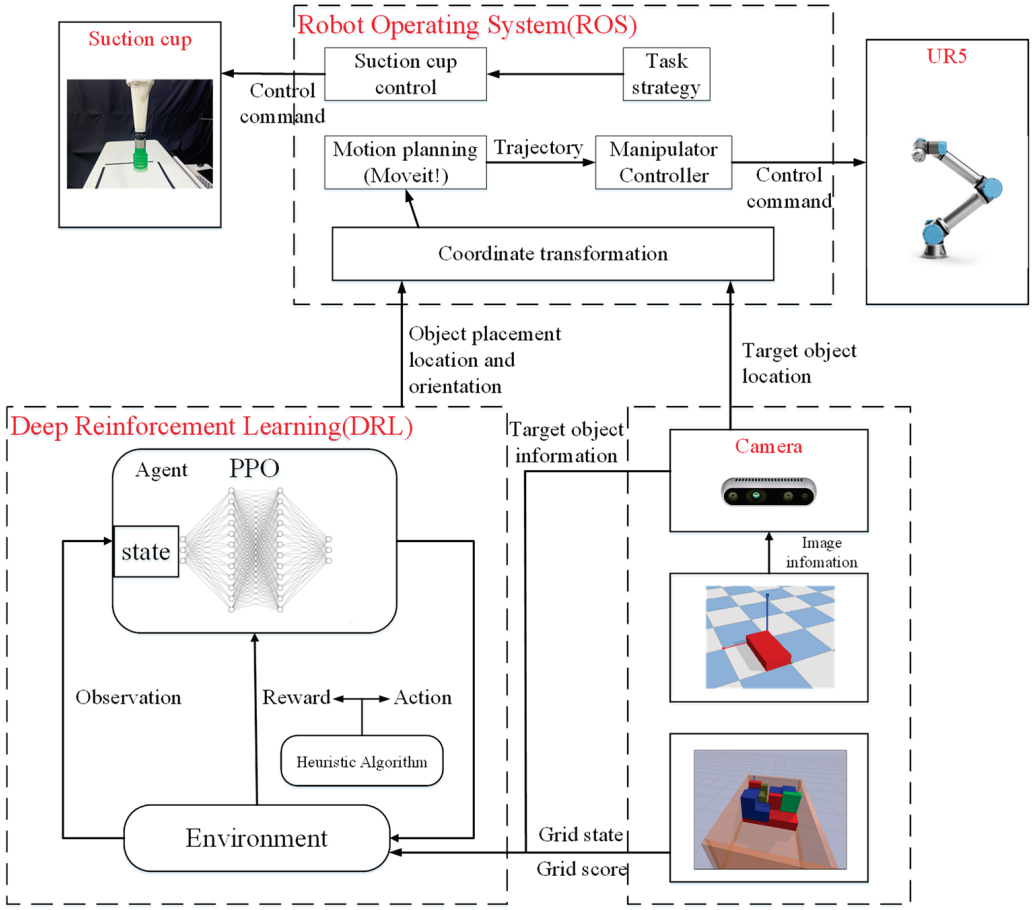


Figure 6. System architecture of the proposed method for 3D-BP tasks.

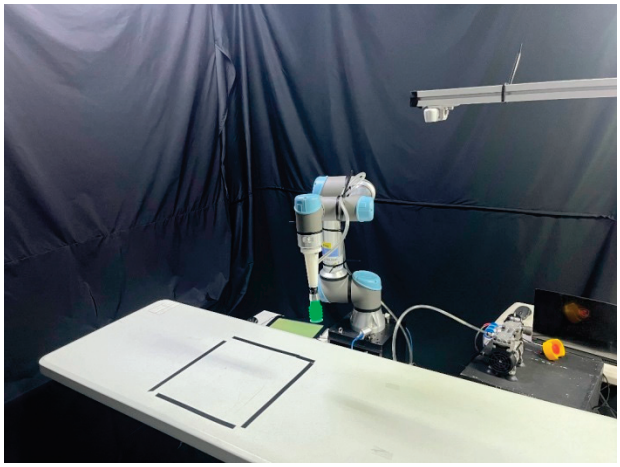


Figure 7. Experimental environment of a real 3D-BP task using a robot manipulator.

3.2. Result of Model Training

The model is based on the one provided by research [31], which shares similar constraints with this online 3D-BP task, considering only two orientations for the current object, $ori(0,\ 0,\ 0)$ and $ori(0,\ 0,\ 90)$. The dimensions of the objects and the container box are shown in Table 2.

Table 2. Object information of Model 1.

Object	Size (cm)	Color
Object 1	$30 \times 40 \times 20$	Orange
Object 2	$30 \times 50 \times 20$	Blue
Object 3	$40 \times 50 \times 20$	Purple
Object 4	$30 \times 50 \times 40$	Green
Object 5	$40 \times 50 \times 30$	Light blue
Container	$400 \times 300 \times 200$	Wood color (transparent)

To evaluate the proposed method, 100 test runs were conducted using the same model. The results are illustrated in Figure 8. Figure 8a,b show the packing results with space utilization rates of 90.08% and 90.06%, respectively. Figure 8c depicts the space utilization rate for each round of the 100 tests The proposed method achieved an average space utilization rate of approximately 83%, with the highest utilization reaching about 92% and the lowest around 74%. Figure 8d shows the number of objects placed in each of the 100 test tests. The average number of objects placed per round is around 455, with the maximum being 510 and the minimum being 405.

To compare the proposed HHPPO method with the model provided by research [31], Table 3 shows an improvement in the same model, with an average space utilization increase of 3%. The top 5% space utilization increased by up to 6%, and the bottom 5% space utilization improved by 3.4%. Additionally, Table 4 shows that the average number of objects placed increased by approximately 15. The test results confirm the effectiveness of the proposed method.

Table 3. Space utilization in each round between research [31] and the proposed method.

Space Utilization	Research [31]	HHPPO	Comparison
Highest	85%	92%	Increase 7%
Top 5%	83.2%	89.2%	Increase 6%
Average	80%	83%	Increase 3%
Bottom 5%	72.4%	75.8%	Increase 3.4%
Lowest	70%	74%	Increase 4%

Table 4. Number of objects placed in each round between research [31] and the proposed method.

Space Utilization	Research [31]	HHPPO	Comparison
Highest	475	505	Increase 30
Top 5%	473	500	Increase 27
Average	440	455	Increase 15
Bottom 5%	410	424	Increase 14
Lowest	395	405	Increase 10

We also performed predictions on Model 2. The dimensions of the objects and the container box are shown in Table 5. We conducted 100 tests, and the results are illustrated in Figure 9. Figure 9a,b show the object placement results for a space utilization rate of 85.07%, from top and bottom views, respectively. Figure 9c presents the space utilization rate for each round of the 100 tests. It shows that the proposed method achieves an average space utilization rate of approximately 65% for Model 2. The highest space utilization rate is

around 80%, and the lowest is about 48%. Figure 9d displays the number of objects placed in each of the 100 tests. The average number of objects placed per round is approximately 80, with the maximum being 105 and the minimum being 63. The result presents more challenges in achieving high space utilization compared to Model 1. This is because the sizes of all objects are closer to the size of the target container, making it difficult to completely fill the container. Moreover, the varying lengths and widths of the objects add to the complexity of stacking them neatly.

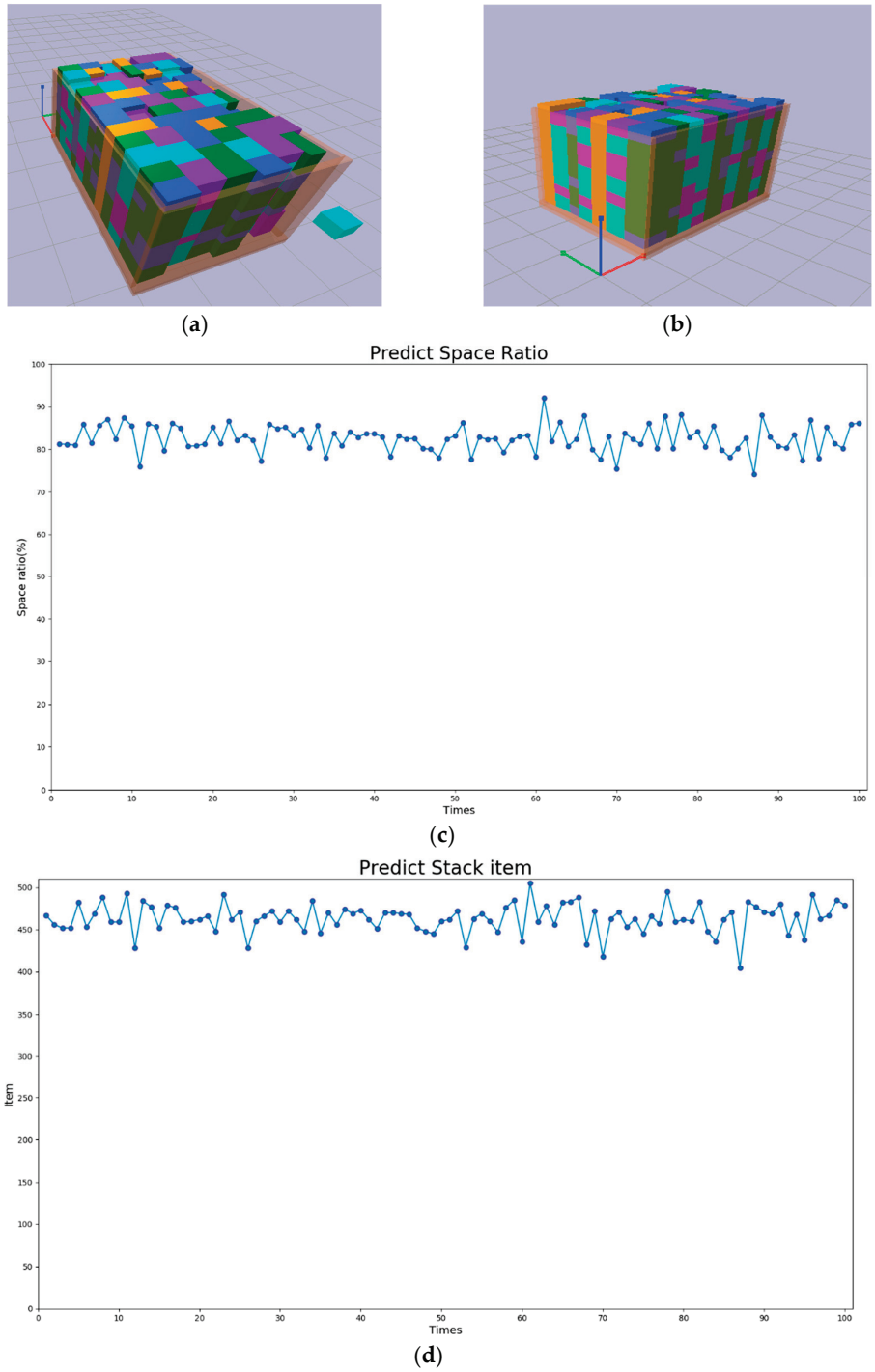
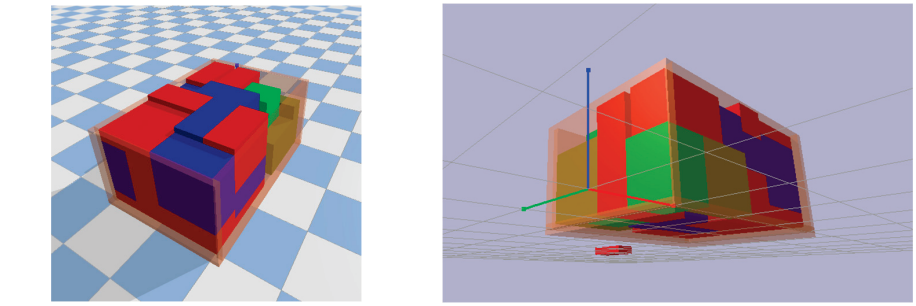
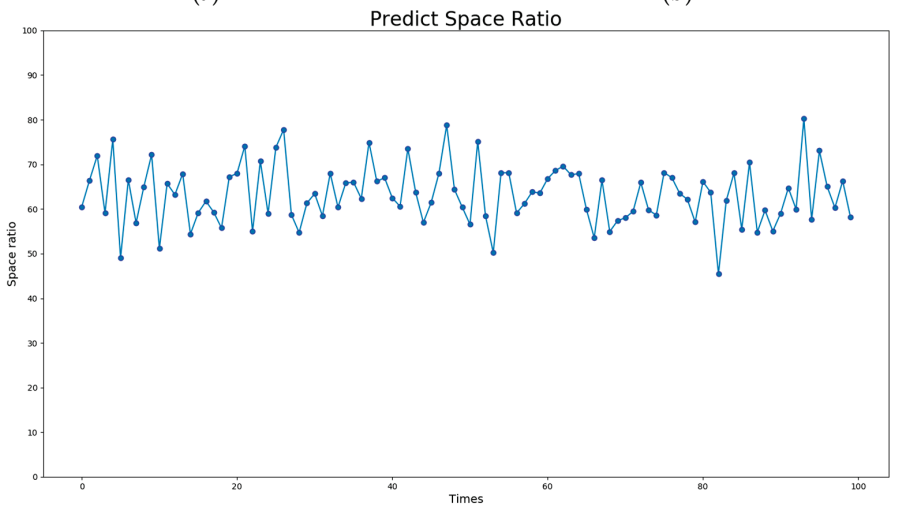


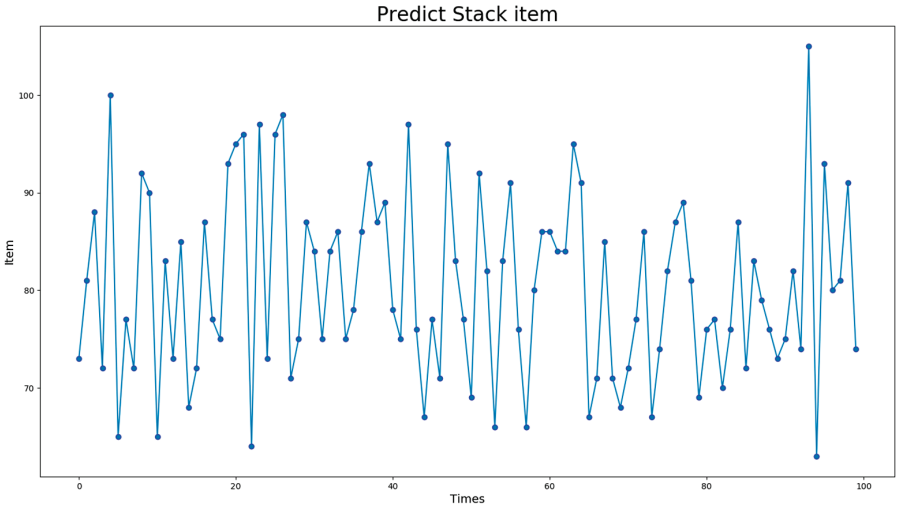
Figure 8. Test results of Model 1. (**a**) Top view of the packed result. (**b**) Bottom view of the packed result. (**c**) Space utilization rate for each round of the 100 tests. (**d**) Number of objects placed in each of the 100 tests.



(**a**) (**b**)



(**c**)



(**d**)

Figure 9. Test results of Model 2. (**a**) Top view of the packed result. (**b**) Bottom view of the packed result. (**c**) Space utilization rate for each round of the 100 tests. (**d**) Number of objects placed in each of the 100 tests.

Table 5. Object information of Model 2.

Object	Size (cm)	Color
Object 1	50 × 100 × 20	Red
Object 2	30 × 90 × 10	Brown
Object 3	50 × 50 × 50	Blue
Object 4	60 × 60 × 10	Green
Container	300 × 200 × 150	Wood color (transparent)

3.3. Experimental Results of Model Implement

The information on the objects used in the experiment is shown in Table 6. The experiments in this study are divided into two parts: online 3D-BP tasks simulated in Gazebo and tasks executed by a real robot in a real environment. Figure 10 shows the online 3D-BP task in the Gazebo simulation. Figure 11 depicts the storyboard of the online 3D-BP task in the real environment, and Figure 12 shows the front and rear views of the completed online 3D-BP task.

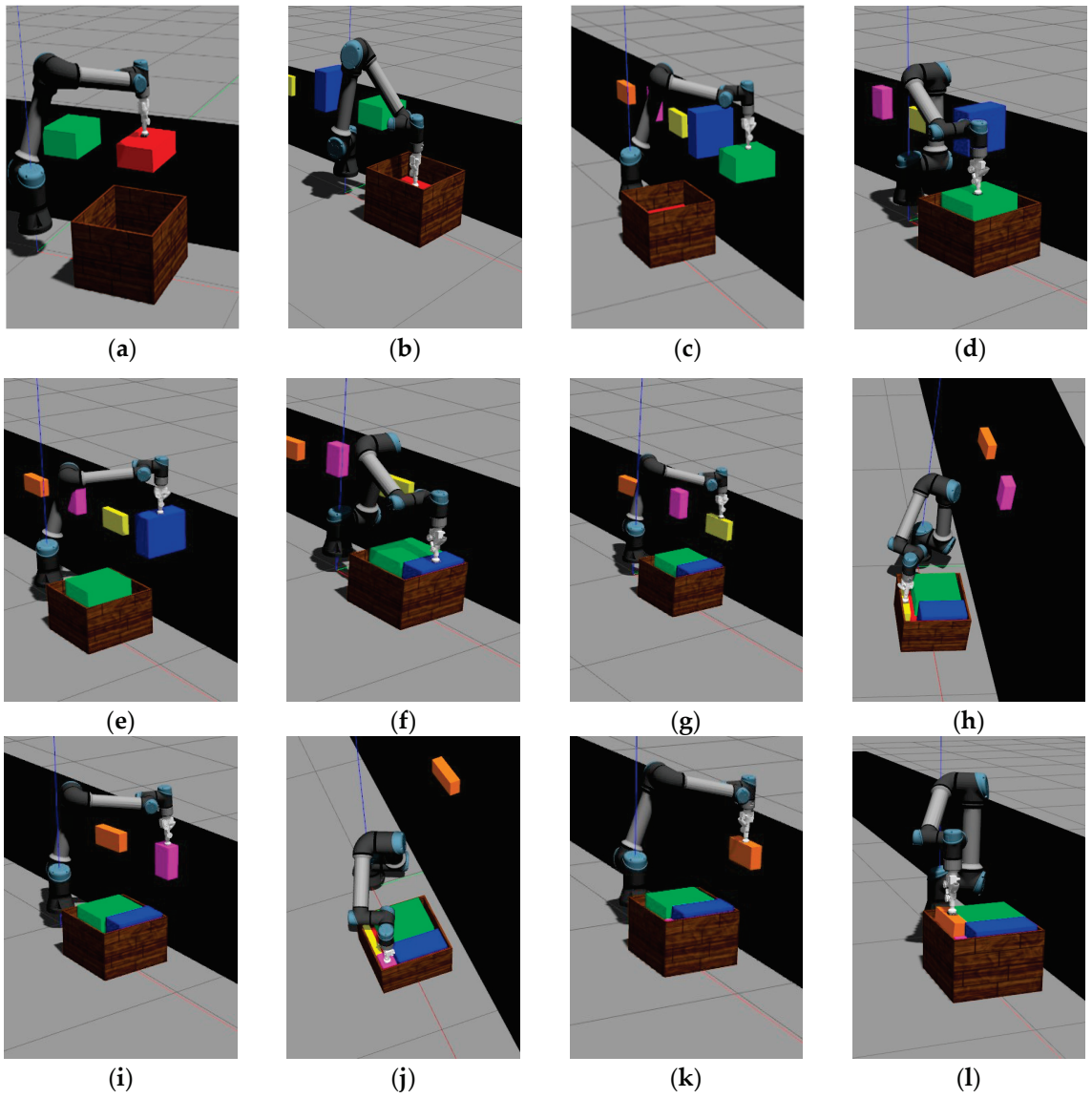


Figure 10. Snapshot of the online 3D-BP task simulation in Gazebo based on the proposed method. (**a**) The robot sucks the first object and approaches the packing bin. (**b**) The robot moves the first object towards the pre-determined position inside the bin. (**c**) The robot sucks the second object and approaches the packing bin. (**d**) The robot moves the second object towards the pre-determined position inside the bin. (**e**) The robot sucks the third object and approaches the packing bin. (**f**) The robot moves the third object towards the pre-determined position inside the bin. (**g**) The robot sucks the fourth object and approaches the packing bin. (**h**) The robot moves the fourth object towards the pre-determined position inside the bin. (**i**) The robot sucks the fifth object and approaches the packing bin. (**j**) The robot moves the fifth object towards the pre-determined position inside the bin. (**k**) The robot sucks the sixth object and approaches the packing bin. (**l**) The robot moves the sixth object towards the pre-determined position inside the bin.

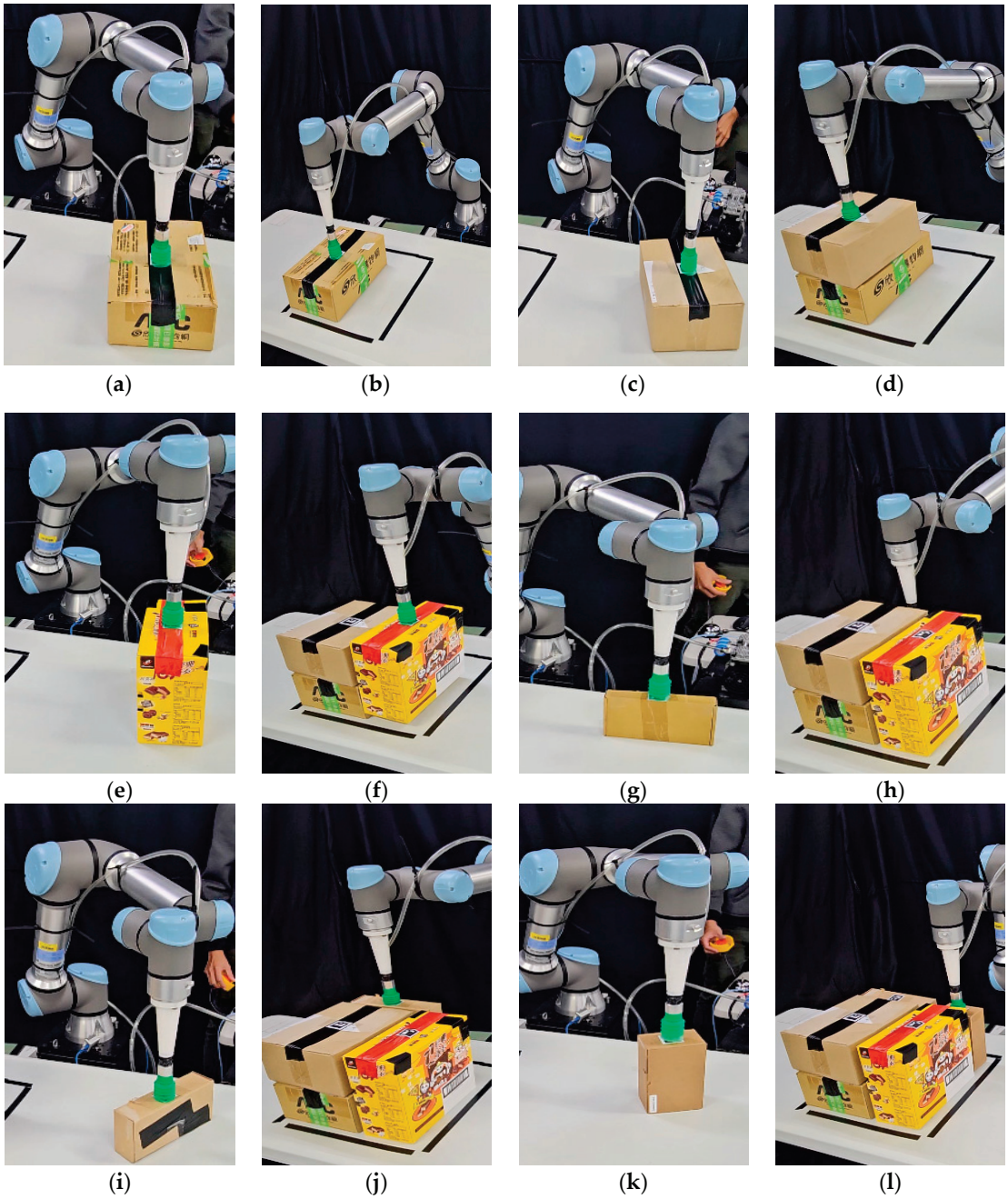


Figure 11. Snapshot of the online 3D-BP task with real robot manipulator based on the proposed method. (**a**) The robot sucks the first object and approaches the packing bin. (**b**) The robot moves the first object towards the pre-determined position inside the bin. (**c**) The robot sucks the second object and approaches the packing bin. (**d**) The robot moves the second object towards the pre-determined position inside the bin. (**e**) The robot sucks the third object and approaches the packing bin. (**f**) The robot moves the third object towards the pre-determined position inside the bin. (**g**) The robot sucks the fourth object and approaches the packing bin. (**h**) The robot moves the fourth object towards the pre-determined position inside the bin. (**i**) The robot sucks the fifth object and l approaches the packing bin. (**j**) The robot moves the fifth object towards the pre-determined position inside the bin. (**k**) The robot sucks the sixth object and approaches the packing bin. (**l**) The robot moves the sixth object towards the pre-determined position inside the bin.

Table 6. Object information of implement model.

Object	Size (cm)	Color
Object 1	21 × 28 × 10	Red
Object 2	21 × 25 × 12	Green
Object 3	12 × 25 × 23	Blue
Object 4	21 × 4 × 12	Yellow
Object 5	11 × 7 × 18	Pink
Object 6	22 × 6 × 11	Orange
Container	35 × 35 × 24	Wood color



(**a**) (**b**)

Figure 12. Results of an online 3D-BP task by the proposed method for a real robot manipulator. (**a**) Top-front view of the bin. (**b**) Side view of the bin.

4. Conclusions

This study proposes the use of a PPO algorithm combined with heuristic algorithms to achieve online 3D-BP tasks. By leveraging visual assistance, the task can be completed in simulated and real-world environments using an actual robot manipulator. The main contributions comprise two parts: (i) In the heuristic algorithm for bin-packing, the proposed extreme point priority sorting method enhances space utilization by ranking the extreme points generated by the extreme point method based on wasted space. A 3D grid representation of container space is introduced, along with some partial support constraints designed according to the methods outlined in this study. (ii) In the deep reinforcement learning integration, a method combining heuristic algorithms with deep reinforcement learning is presented. The PPO algorithm is used, incorporating reward function design and the action space of the policy network. This combination integrates the extreme point method, DBLF algorithm, and the designed extreme point priority sorting heuristic algorithms, demonstrating promising results in the experiments. These contributions collectively advance the effectiveness and efficiency of the online 3D-BP task, providing an approach that can be applied in both simulated and real-world environments. In future work, unpacking algorithms will be used to create a more robust and efficient method to achieve efficient online 3D-BP tasks in dynamic environments. The objectives are to improve space utilization by minimizing the gaps between boxes through an advanced unpacking algorithm.

Author Contributions: Conceptualization, C.-C.W.; methodology, C.-C.W., T.-T.T. and C.-K.O.; software, T.-T.T. and C.-K.O.; validation, T.-T.T. and C.-K.O.; formal analysis, C.-C.W., T.-T.T. and C.-K.O.; investigation, T.-T.T. and C.-K.O.; resources, C.-C.W.; data curation, T.-T.T. and C.-K.O.; writing—original draft preparation, T.-T.T. and C.-K.O.; writing—review and editing, C.-C.W.; visualization, T.-T.T. and C.-K.O.; supervision, C.-C.W.; project administration, C.-C.W.; funding acquisition, C.-C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly supported by the National Science and Technology Council (NSTC) of Taiwan, R.O.C. under grant number NSTC 112-2221-E-032-035-MY2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Martello, S.; Pisinger, D.; Vigo, D. The three-dimensional bin packing problem. *Oper. Res.* **2000**, *48*, 256–267. [CrossRef]
2. Li, T.-H.S.; Liu, C.Y.; Kuo, P.H.; Fang, N.C.; Li, C.H.; Cheng, C.W.; Hsieh, C.Y.; Wu, L.F.; Liang, J.J.; Chen, C.Y. A three-dimensional adaptive PSO-based packing algorithm for an IoT-based automated e-fulfillment packaging system. *IEEE Access* **2017**, *5*, 9188–9205. [CrossRef]
3. Fleszar, K.; Hindi, K.S. New heuristics for one-dimensional bin-packing. *Comput. Oper. Res.* **2002**, *29*, 821–839. [CrossRef]
4. González-San-Martín, J.; Cruz-Reyes, L.; Gómez-Santillán, C.; Fraire, H.; Rangel-Valdez, N.; Dorronsoro, B.; Quiroz-Castellanos, M. Comparative Study of Heuristics for the One-Dimensional Bin Packing Problem. In *Hybrid Intelligent Systems Based on Extensions of Fuzzy Logic, Neural Networks and Metaheuristics*; Castillo, O., Melin, P., Eds.; Springer: Cham, Switzerland, 2023; pp. 293–305. [CrossRef]
5. Zhao, H.; Zhu, C.; Xu, X.; Huang, H.; Xu, K. Learning practically feasible policies for online 3D bin packing. *Sci. China Inf. Sci.* **2022**, *65*, 112105. [CrossRef]
6. Ma, Y.; Hong, X.; Dong, S.; Cheng, C.K. 3D CBL: An efficient algorithm for general 3D packing problems. In Proceedings of the 48th Midwest Symposium on Circuits and Systems, Covington, KY, USA, 7–10 August 2005; pp. 1079–1082.
7. Gonçalves, J.F.; Resende, M.G. A biased random key genetic algorithm for 2D and 3D bin packing problems. *Int. J. Prod. Econ.* **2013**, *145*, 500–510. [CrossRef]
8. Chen, C.-S.; Lee, S.-M.; Shen, Q. An analytical model for the container loading problem. *Eur. J. Oper. Res.* **1995**, *80*, 68–76. [CrossRef]
9. Hasan, J.; Kaabi, J.; Harrath, Y. Multi-objective 3D bin-packing problem. In Proceedings of the 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), Manama, Bahrain, 15–17 April 2019; pp. 1–5.
10. Harrath, Y. A three-stage layer-based heuristic to solve the 3D bin-packing problem under balancing constraint. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 6425–6431. [CrossRef]
11. Kaabi, J.; Harrath, Y.; Bououdina, H.E.; Qasim, A.T. Toward smart logistics: A new algorithm for a multi-objective 3D bin packing problem. In Proceedings of the Smart Cities Symposium, Zallaq, Bahrain, 22–23 April 2018; pp. 1–5.
12. Hsayri, M.; Tlili, M. A novel multi stage optimization algorithm for a 3D-BPP resolution. In Proceedings of the 17th International Conference on Innovations in Intelligent Systems and Applications (INISTA), Hammamet, Tunisia, 20–23 September 2023; pp. 1–5.
13. Yang, S.; Song, S.; Chu, S.; Song, R.; Cheng, J.; Li, Y.; Zhang, W. Heuristics Integrated deep reinforcement learning for online 3D bin packing. *IEEE Trans. Autom. Sci. Eng.* **2023**, *21*, 939–950. [CrossRef]
14. Zhu, W.; Chen, S.; Dai, M.; Tao, J. Solving a 3D bin packing problem with stacking constraints. *Comput. Ind. Eng.* **2024**, *188*, 109814. [CrossRef]
15. Duan, L.; Hu, H.; Qian, Y.; Gong, Y.; Zhang, X.; Xu, Y.; Wei, J. A multi-task selected learning approach for solving 3D flexible bin packing problem. *arXiv* **2018**, arXiv:1804.06896.
16. Que, Q.; Yang, F.; Zhang, D. Solving 3D packing problem using transformer network and reinforcement learning. *Expert Syst. Appl.* **2023**, *214*, 119153. [CrossRef]
17. Murdivien, S.A.; Um, J. Boxstacker: Deep reinforcement learning for 3D bin packing problem in virtual environment of logistics systems. *Sensors* **2023**, *23*, 6928. [CrossRef] [PubMed]
18. Hu, H.; Zhang, X.; Yan, X.; Wang, L.; Xu, Y. Solving a new 3D bin packing problem with deep reinforcement learning method. *arXiv* **2017**, arXiv:1708.05930.
19. Nguyen, T.-H.; Nguyen, X.-T. Space splitting and merging technique for online 3-D bin packing. *Mathematics* **2023**, *11*, 1912. [CrossRef]

20. Fang, J.; Rao, Y.; Zhao, X.; Du, B. A hybrid reinforcement learning algorithm for 2D irregular packing problems. *Mathematics* **2023**, *11*, 327. [CrossRef]
21. Qi, M.; Zhang, L. Online 3D packing problem based on bi-value guidance. *J. Comput. Commun.* **2023**, *11*, 156–173. [CrossRef]
22. Verma, R.; Singhal, A.; Khadilkar, H.; Basumatary, A.; Nayak, S.; Singh, H.V.; Kumar, S.; Sinha, R. A generalized reinforcement learning algorithm for online 3D bin-packing. *arXiv* **2020**, arXiv:2007.00463.
23. Puche, A.V.; Lee, S. Online 3D bin packing reinforcement learning solution with buffer. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 8902–8909.
24. Wu, Y.; Mansimov, E.; Grosse, R.B.; Liao, S.; Ba, J. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–10.
25. Zhao, H.; She, Q.; Zhu, C.; Yang, Y.; Xu, K. Online 3D bin packing with constrained deep reinforcement learning. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 741–749. [CrossRef]
26. Pan, Y.; Chen, Y.; Lin, F. Adjustable robust reinforcement learning for online 3D bin packing. In Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023), New Orleans, LA, USA, 10 December 2023; pp. 1–29. Available online: https://proceedings.neurips.cc/paper_files/paper/2023/file/a345ed605675c7c484e740a8ceaa6b45-Paper-Conference.pdf (accessed on 8 July 2024).
27. Crainic, T.G.; Perboli, G.; Tadei, R. Extreme point-based heuristics for three-dimensional bin packing. *Informs J. Comput.* **2008**, *20*, 368–384. [CrossRef]
28. Liu, K.; Lv, X.-f. Research on palletizing and packing based on heuristic algorithm. *J. Phys. Conf. Ser.* **2023**, *2449*, 012040. [CrossRef]
29. Karabulut, K.; İnceoğlu, M.M. A hybrid genetic algorithm for packing in 3D with deepest bottom left with fill method. In Proceedings of the International Conference on Advances in Information Systems, Izmir, Turkey, 20–22 October 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 441–450.
30. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
31. Nguyen, T.-H.; Tran, V.-T.; Doan, P.-Q.; Mac, T.-T. A novel heuristic algorithm for online 3D bin packing. In Proceedings of the 21st International Conference on Control, Automation and Systems (ICCAS), Jeju, Republic of Korea, 12–15 October 2021; pp. 1993–1997.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Diffusion Models-Based Purification for Common Corruptions on Robust 3D Object Detection

Mumuxin Cai [1], Xupeng Wang [2,*], Ferdous Sohel [3] and Hang Lei [1]

[1] School of Information and Software Engineering, The University of Electronic Science and Technology of China, Chengdu 610054, China; caimumuxin@std.uestc.edu.cn (M.C.)
[2] Laboratory Of Intelligent Collaborative Computing, The University of Electronic Science and Technology of China, Chengdu 610054, China
[3] School of Information Technology, Murdoch University, Perth, WA 6150, Australia; f.sohel@murdoch.edu.au
* Correspondence: xupeng.wang@uestc.edu.cn

Abstract: LiDAR sensors have been shown to generate data with various common corruptions, which seriously affect their applications in 3D vision tasks, particularly object detection. At the same time, it has been demonstrated that traditional defense strategies, including adversarial training, are prone to suffering from gradient confusion during training. Moreover, they can only improve their robustness against specific types of data corruption. In this work, we propose LiDARPure, which leverages the powerful generation ability of diffusion models to purify corruption in the LiDAR scene data. By dividing the entire scene into voxels to facilitate the processes of diffusion and reverse diffusion, LiDARPure overcomes challenges induced from adversarial training, such as sparse point clouds in large-scale LiDAR data and gradient confusion. In addition, we utilize the latent geometric features of a scene as a condition to assist the generation of diffusion models. Detailed experiments show that LiDARPure can effectively purify 19 common types of LiDAR data corruption. Further evaluation results demonstrate that it can improve the average precision of 3D object detectors to an extent of 20% in the face of data corruption, much higher than existing defence strategies.

Keywords: 3D object detection; LiDAR scene data; point cloud; diffusion models; defence strategy; adversarial robustness

1. Introduction

With the rapid development of Light Detection and Ranging (LiDAR) sensors, their ability to capture large-scale point cloud data has been extensively utilized. LiDAR sensors are advanced remote sensing devices that utilize laser technology to precisely measure distances and acquire detailed 3D spatial information. The process begins with the emission of laser pulses from the sensor, which are then directed towards the target area. Upon striking an object, the pulses reflect back and are captured using the sensor's receiver. By measuring the time it takes for the laser pulses to travel to and from the object, combined with the precise angular measurements of the laser's direction, the LiDAR sensor calculates the object's 3D position. This process is repeated rapidly, scanning across the entire target area, generating a dense point cloud of millions of 3D points that accurately represent the environment's geometry and topology. The resulting data are then processed and analyzed to extract meaningful information for applications such as topographic mapping, urban planning, environmental monitoring, and autonomous navigation systems. In particular, LiDAR-based 3D object detection models have found wide applications in various fields, such as autonomous driving [1]. However, these neural network-based deep models are susceptible to adversarial attacks that can easily deceive models and make incorrect predictions [2,3]. Notably, 3D object detection models have also been proven to be very vulnerable to adversarial attacks [4,5]. The data collected using LiDAR sensors present some common corruptions, such as noise and low density [6]. Possible data corruption

can be exploited as adversarial attacks, posing a significant threat to security critical applications. Therefore, there is a need for innovative techniques to purify corrupted data and enhance the robustness of 3D object detection-based systems.

There are two main types of defense strategies for adversarial attacks, namely adversarial training [7,8] and adversarial purification [9,10]. Adversarial training utilizes adversarial samples to train neural network models, making them robust against the same type of adversarial attacks. This is a commonly used standard defense strategy, as it is very effective in dealing with fixed types of adversarial attacks and can directly improve the robustness of the model. Adversarial training is usually aimed at defending against specific adversarial attacks, which leads to a lack of flexibility and difficulty in coping with multiple types of simultaneous adversarial attacks. Deep models that have undergone specific adversarial training are often very fragile and inflexible, when faced with other types of adversarial attacks. This requires exhausting adversarial training to compensate. In addition, typical adversarial training techniques suffer from a much higher training cost and computational complexity in contrast to standard training methods. On the other hand, adversarial purification relies on generative models to purify input adversarial samples, making them harmless and unable to interfere with deep models. Compared to adversarial training, adversarial purification is more flexible and can handle a wider range of adversarial threats, even performing well in the face of unknown adversarial attacks. In addition, adversarial purification is typically trained and operated independently of task models, while adversarial training must bound to neural networks for training together. Adversarial training can be seen as an improvement to the detector, while adversarial purification is not a problem that should be considered for detectors. The detector improvement and data purification are two separate research areas. Data purification focuses on purifying adversarial samples, whereas the goal of a detector is to predict normal samples as correctly as possible. If the detector itself considers data purification, it will inevitably affect its performance on normal samples. However, the existing adversarial purification method is limited by the ability of generative models, and often lags behind adversarial training in improving robustness performance [11]. The capabilities of the generative models used, such as generative adversarial networks (GANs) [12] and variational autoencoder (VAE) [13], cannot meet the needs of adversarial purification, resulting in their limited robustness performance.

The technique of diffusion models has emerged with its powerful generative ability surpassing GANs and VAE [14,15]. In addition to the 2D images, diffusion models have also demonstrated superior performance in the field of 3D point cloud generation [16] and purification [17]. Therefore, we consider utilizing the technique in large-scale LiDAR scenes to purify corrupted samples. The diffusion model mainly consists of forward diffusion process and reverse diffusion process. The forward diffusion process follows a Markov chain, gradually adding noise to the input sample. When there are enough steps, the input sample can be transformed into an ideal noise model. The reverse diffusion process, on the other hand, involves continuously removing noise from noisy images, which is usually conducted using neural networks and ultimately restores the noise to a clear original image. The diffusion model's ability to recover from noise introduces significant randomness. This makes it more flexible when used in adversarial purification defense strategies and it is able to cope with a wider variety of unknown adversarial attacks.

In this paper, we propose an adversarial purification method to purify common corruptions in large-scale LiDAR scene data, named LiDARPure. The proposed purification method is detector independent and can be equipped to any detector. It can be used for low frame rate 3D object detection applications with existing hardware, such as robotics. With the improvement of computational power, it can also be applied to higher real-time 3D object detection applications in the future. Compared with adversarial training, LiDARPure leverages the generative ability of diffusion models, enabling it to handle various types of data corruption without being hindered by data sparsity and gradient confusion. The corrupted LiDAR scene data are transformed into random noise through the forward diffusion process of the diffusion model, in which a large amount of randomness

is introduced to smooth out the corruption in the data. Subsequently, the noise is restored to a purified LiDAR scene using the reverse diffusion process. It should be noted that we extract latent representations in units of voxels and use them as conditions to help the denoising in the reverse diffusion process. In addition, we use the proposed denoising recovery loss to supervise the reverse diffusion process, making the purified sample closer to the pure data.

The main contributions of this article can be summarized as follows:

- We demonstrate that existing standard adversarial training has significant limitations for 3D object detection models because of the large-scale space, sparsity, variety of corruptions, and gradient confusion.
- We propose LiDARPure, which leverages the powerful generative ability of diffusion models to purify common corruptions on LiDAR scenes and works independently of the 3D object detector.
- We propose a denoising recovery loss term, which utilizes the latent features and geometry information to supervise the reverse diffusion process.
- The experiments demonstrate that the proposed method achieves superior robustness to corruptions and outperforms existing 3D point cloud purification methods.

2. Related Work

2.1. LiDAR-Based 3D Object Detection

In recent years, 3D object detection with LiDAR sensors has witnessed significant advancements, enabling practical perception in autonomous driving and robotics. Notably, VoxelNet [18] pioneered the use of voxel feature encoding to effectively represent point clouds, setting the foundation for subsequent works. SECOND [19] adopted a sparse 3D convolutional network to process the voxelized point clouds, significantly reducing computational complexity and achieving faster inference speeds while maintaining high detection accuracy. PointPillars [20] projected point clouds onto a 2D grid in a bird's eye view, forming pillars, and then processed these pillars using 2D convolutions. It further simplified the representation by encoding point clouds into pillars, striking a balance between performance and efficiency. PointRCNN [21] introduced a two-stage approach, which first generated 3D proposals from raw point clouds and then refined them, enhancing the recall rate of point cloud segmentation and detection accuracy simultaneously. PV-RCNN [22] extended PointRCNN by incorporating voxel-based features into its refinement stage, demonstrating the effectiveness of combining voxel-based efficiency with point-based accuracy. CenterPoint [23] and CenterFormer [24] adopted a center-based paradigm, which directly predicted object centers and their attributes, and demonstrated state-of-the-art results, simplifying the detection pipeline. These models have significantly contributed to the field, advancing the capabilities of 3D object detection from LiDAR data.

2.2. Adversarial Attack against 3D Object Detection

LiDAR sensors are widely used in 3D vision tasks, but the data they collect poses many challenges, such as large-scale range and sparsity. The existing adversarial attacks [4,5,25,26] have proven that 3D object detection is highly vulnerable to malicious samples. Cao et al. [25] conducted the first security study of LiDAR-based perception in autonomous vehicles, exploring LiDAR spoofing attacks by adding malicious vehicle point clouds to areas obscured by vehicles, achieving attack success rates of around 75%. Tu et al. [4] deceived LiDAR detectors by placing the adversarial mesh on vehicle rooftop, which is disguised as cargo carried by vehicles. Wang et al. [5] used optimized adversarial perturbations to attack the entire scene, significantly reducing the detector's performance to detect all targets. The points collected using LiDAR often suffer from data corruption, which can lead to instability and pose significant safety challenges for 3D object detection tasks. Shin et al. [27] proved that additional light may cause LiDAR to make incorrect distance judgments on objects. LISA [28] has demonstrated through simulation that adverse weather is also an important cause of data corruption in LiDAR. Heavy fog [29,30] and snowfall [31] have been shown to cause a

significant amount of noise and point shift in the scene. In addition to data corruption caused by adverse weather, LiDAR may also spontaneously have local and global noise, density degradation, shearing, and object rotation [32]. Dong et al. [6] summarized 27 common data corruptions in autonomous driving scenarios and tested the robustness of mainstream detectors. Among them, 19 types of data corruption completely occur on LiDAR sensors. These data corruption types are different, making it difficult to completely eliminate through adversarial training. This indicates the need for more universal and flexible methods to eliminate the impact of LiDAR data corruption on 3D object detection models.

2.3. Defend Strategies for 3D Computer Vision Methods

Defense strategies tailored for 3D computer vision tasks have attracted attentions from the research community with the extensive studies on their adversarial counterparts. Generally, existing defense methods can be divided into to adversarial training and purification. Hahner et al. [29,31] simulated the data captured in adverse weather and used them for adversarial training, which greatly enhances the robustness of 3D object detection models to heavy fog and snowfall weather. Teufel et al. [33] used data simulation and data augmentation methods to train a 3D object detector in three types of severe weather conditions, enabling it to have high adversarial robustness to rain, snow, and fog weather simultaneously. 3D-Vfield [26] improves the performance of the model by adding possible unknown objects to the scene data trained in the model. However, these methods of adversarial training can only be effective for specific types of attacks. In 3D recognition tasks, DUP-Net [34] purified the samples by denoising and upsampling, removing the outliers and noise points from adversarial samples. Dong et al. [35] filtered the vector directions of various regions in the point cloud and fed the purified features into the classifier to avoid adversarial attacks. SUN et al. [36] designed a self supervised mechanism for 3D recognition models, which enhanced the robustness of the model during training. IF-Defense [37] and LPC [38] purify adversarial samples through implicit functions and gradient rules. Currently, adversarial purification is mainly used for defense against point cloud object recognition, while adversarial training is usually used for large-scale LiDAR scenes. The 3D object detection model requires more efficient adversarial purification methods to cope with the corruption of large-scale LiDAR data.

2.4. Diffusion Models for 3D Data

Both discrete-time diffusion models [14] and continuous time diffusion models [15] have shown their potential in the field of two-dimensional image generation, and some studies [16,39–41] have also begun to use diffusion models to generate and complete three-dimensional point clouds. Lee et al. [42] and Nunes et al. [43] also demonstrated the ability of diffusion models to generate and complete large-scale real-world scenes. In addition, DiffPure [44] demonstrates the adversarial purification ability of diffusion models when used in defense strategies in the field of images. Subsequently, the diffusion model was also introduced into the adversarial purification of point clouds. PointDP [17] used the purification method derived from diffusion models to improve the robust precision of 3D recognition models, surpassing existing advanced IF-Defense [37] and LPC [38] methods. This proves the effectiveness of diffusion models in combating adversarial attacks on 3D data. The above research indicates that using diffusion models and adversarial purification to process large-scale LiDAR scene data and their common data corruptions is promising. However, these defense strategies based on diffusion models for 3D data are only applicable to dense point clouds of individual objects, focusing on 3D recognition models, which cannot be well applied to large-scale LiDAR scenes. The LiDARPure proposed in this paper can handle sparser large-scale spaces and purify corrupted scenes simultaneously.

3. Limitations of Adversarial Training on 3D Object Detection

Adversarial training, as a training method used in machine learning and deep learning to improve model robustness and generalization ability, has been widely used in the field

of 2D images for a long time. However, in the task of 3D point cloud processing, its performance always has certain limitations. As Sun et al. [17] found, in 3D point cloud recognition tasks, the gradient of the loss function is difficult to backpropagate to the input data during adversarial training due to a large number of *k*NN layers, top-*k* operations, and max pooling operations. There is a serious gradient confusion among them, which can lead to non-differentiability in some cases and greatly increases the difficulty of training. Compared to the fixed resolution of 2D images, the disorder and arbitrary number of points in point cloud data, as well as the sampling algorithm used, will increase a lot of randomness during the training process. The use of sampling algorithms is reasonable for standard training, but the randomness introduced will greatly increase the workload and computational cost of adversarial training.

In addition to the limitations of adversarial training in the 3D point cloud recognition tasks mentioned above, we focus more on the performance of adversarial training in large-scale LiDAR data and 3D object detection models. The number of points in large-scale LiDAR point cloud scenes is usually much larger than that of a single point cloud object. Three-dimensional object detection models will lose more information and introduce more randomness during data sampling. Compared to 3D point cloud recognition models, the tasks completed using 3D object detection models are more complex, requiring simultaneous localization and a classification of objects. This increases the complexity of its deep neural network model and may use more top-*k* and max pooling operations that can cause gradient confusion. In fact, a large number of 3D object detection models use PointNet [45] or PointNet++ [46] as part of their feature extraction structures, such as PointPillars [20], PointRCNN [21], and CenterPoint [23]. In adversarial attack work [5], we have found that the Non-Maximum Suppression (NMS) operation introduced by existing 3D object detection models in object localization also introduces significant randomness. This hinders not only the optimization of the gradient against adversarial attacks, but also the fitting against adversarial training.

In addition to the challenges posed by the increase in data size and network complexity for adversarial training, 3D object detection faces more possible data corruptions compared to point cloud recognition tasks. The main data corruptions that may be encountered in point cloud recognition tasks is local density reduction and noises. In the recognition task, a single object point cloud does not need to consider the impact of adverse weather such as heavy fog and snowfall, which will add a large number of additional noise points to the scene. In addition, LiDAR point cloud scenes are also susceptible to unique data corruption such as shearing and spatiotemporal misalignment. Adversarial training usually only deals with specific adversarial attacks or data corruption; Hahner et al. [29,31] demonstrated that simulating and adversarial training against adverse weather can improve the robustness of 3D object detection models. However, it is unreasonable to conduct adversarial training on data corruption in all common types of LiDAR scenarios. Firstly, Dong et al. [6] proposed 27 common data corruptions issue faced by LiDAR point cloud data. There are so many types of them that adversarial training requires significant computational cost, and deep networks need to consider all the corruptions simultaneously, making it difficult to fit. Secondly, adversarial training is not a perfect solution. Lechner et al. [47] demonstrated that adversarial training targeting specific adversarial attacks may reduce the performance of the models and make it more vulnerable to other types of adversarial attacks. We selected three adverse weather corruptions simulation methods in LiDAR scenarios, namely snow [31], rain [28], and fog [29], and studied their impact with related adversarial training on the KITTI dataset. The three simulation methods for adverse weather conditions generated corrupted samples on the training set of the KITTI dataset were used for the adversarial training of the PointPillars detector. The malicious samples simulated using adverse weather on the KITTI validation set are used to evaluate the robustness of adversarial training against an unknown corruption. In Table 1, the performance of the PointPillars detector is demonstrated in the form of AP when it has undergone different adversarial training and encountered data corruptions, respectively.

The results presented in Table 1 indicate that adversarial training can improve the robustness of detectors against the same type of data corruption. But at the same time, adversarial training may reduce the standard performance of 3D object detection models and make them more vulnerable to other data corruptions. This also proves that for LiDAR point cloud data with a large number of common types of corruption, it is not applicable to use adversarial training to deal with all data corruptions.

Table 1. Average precision (AP) of PointPillars detector with different adversarial training strategies when experiencing data corruptions.

Corruptions	Adversarial Training			
	None	Snow	Rain	Fog
None	**78.4**	78.1	77.5	77.2
Snow	36.5	39.5	34.0	33.1
Rain	36.2	35.2	38.8	35.7
Fog	64.3	64.1	63.8	66.3

4. LiDARPure: Purification for LiDAR Scenes

In this section, we first present the fundamentals of the diffusion models and their applications in 3D point cloud data. Next, the problem studied in this paper is formulated in Section 4.2. We describe the proposed LiDARPure method for purifying LiDAR scenes in detail in Section 4.3. The loss functions used for the training of the diffusion model-based purification are then described in Section 4.4.

4.1. Preliminaries

Diffusion models can be classified into discrete diffusion models and continuous diffusion models. In this paper, we mainly utilize the discrete diffusion model and one of its solutions, Denoising Diffusion Probabilistic Models (DDPM) [14]. According to DDPM, the discrete-time conditional diffusion model can be formulated as follows:

Given a sample $x_0 \sim q(x)$ sampled from the data distribution $q(x)$, the forward diffusion process is defined as a Markov chain that gradually adds Gaussian noise ϵ to the sample over T steps, thus generating a series of noisy samples $X = \{x_1, \ldots, x_T\}$. The noise added in each step is controlled by forward variances $\beta = \{\beta_1, \ldots, \beta_T\} \in (0, 1)$. Thus, x_t can be represented as follows:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}, \quad \epsilon_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{1}$$

The conditional distribution of $q(x_t)$ can then be represented as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{2}$$

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \tag{3}$$

Given $\alpha_t = 1 - \beta_t$, x_t and reverse diffusion process can be written as follows:

$$x_t = \sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, \tag{4}$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)\mathbf{I}). \tag{5}$$

During the diffusion process, the information of the data x_0 is gradually destroyed. As $T \to \infty$, the entire scene is imposed with added Gaussian noise and $q(x_T) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$.

The reverse denoising diffusion process aims at solving the inverse process $q(x_{t-1}|x_t)$ for the forward diffusion process $q(x_t|x_{t-1})$. An existing study [15] on the theory of stochastic differential equations proves that $q(x_{t-1}|x_t)$ also approximately obeys a Gaussian distribution when β_t is sufficiently small. Thus, it is only necessary to obtain the noise

$\epsilon_\theta(x_t, t)$ at step t to calculate x_{t-1}. The reverse diffusion process is difficult to compute directly, so the DDPM uses the neural network $p_\theta(x_{t-1}|x_t)$ to fit the $q(x_{t-1}|x_t)$. The goal is to make the predicted noise and the actual added noise as identical as possible. The conditional probability distribution of the neural network $p_\theta(x_{t-1}|x_t)$ can be represented as follows:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \boldsymbol{\Sigma}_\theta(x_t, t)), \tag{6}$$

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t). \tag{7}$$

Although the probability distribution $q(x_{t-1}|x_t)$ of the reverse diffusion process is difficult to solve, one can derive $q(x_{t-1}|x_t, x_0)$, given the condition x_0:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_\theta(x_t, x_0), \tilde{\beta}_t \mathbf{I}). \tag{8}$$

With the variance and mean above, the analytic form of $q(x_{t-1}|x_t, x_0)$ is obtained. The objective function of the training diffusion models is to minimize the negative log-likelihood of $p_\theta(x_0)$ by constructing a variational bound.

In the reverse diffusion process, additional information about the data can be input to assist data generation, which is called the conditional diffusion model. In the diffusion model-based 3D point cloud generation methods, due to the lack of texture information and the sparsity of point cloud data, the latent geometric feature z of point clouds are often used as conditions in the reverse diffusion process to help restore the original point cloud [16]:

$$p_\theta(x_{0:T}|z) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, z), \tag{9}$$

where z is the latent feature extracted using encoder E from point cloud data x. Usually, the encoder E and noise predictor $\epsilon_\theta(x_t, t, z)$ are trained together. The diffusion model has demonstrated a strong generation capability on point cloud data, and also has great potential in large-scale LiDAR scenes, making it possible to purify common corruptions in LiDAR scenes.

4.2. Problem Formulation

Given a real-world scenario P, a LiDAR sensor collects it and ideally obtains a pure LiDAR point cloud scene S. The scene S contains m objects $B = \{b_1, \ldots, b_m\}$ and their corresponding bounding boxes $b_m = (x, y, z, l, w, h, \theta)$, where (x, y, z) represents the center point coordinates of the objects, (l, w, h) represents the length, width, and height of the bounding box, and θ represents the orientation of the objects in the bird's eye view. The goal of 3D object detector D is to output predictions that match the bounding boxes of the objects, which means $D(S) = B$. However, during the data collection process, due to sensor defects or adverse weather conditions, it often leads to data corruption such as noise n. A pure LiDAR scene S becomes a corrupted LiDAR scene $\hat{S}$ when these corruptions n are imposed, which can be written as $\hat{S} = S + n$. The corruption of data causes errors in the detection result $D(\hat{S})$ of the detector D, which is different from the detection result $D(S)$ of pure LiDAR scenes, i.e., $D(\hat{S}) \neq D(S)$. The goal of this study is to purify the corrupted LiDAR scene S based on the powerful generation ability of diffusion models. By adding noise through the forward diffusion process of T steps, the corrupted scene is transformed into Gaussian noise S^T, and then gradually denoised through the reverse diffusion process to obtain the purified point cloud scene S', which needs to be as similar as the pure scene S as possible. At this point, detector D has the same prediction results for them, which means $D(S') = D(S)$.

4.3. Design of LiDARPure

As shown in Figure 1, the corrupted LiDAR scene data are progressively added with noise through a diffusion process to obtain a diffuse scene, and a reparameterized noise

predictor is utilized in the denoising process of reverse diffusion to remove both the added noise and the corruption in the scene. Based on the LiDAR scene-scale DDPM in [43], we optimize the denoising process so that the local features at each point are involved in the noise predictor. In addition, we use a conditional diffusion model based on latent features to assist in the noise predictor.

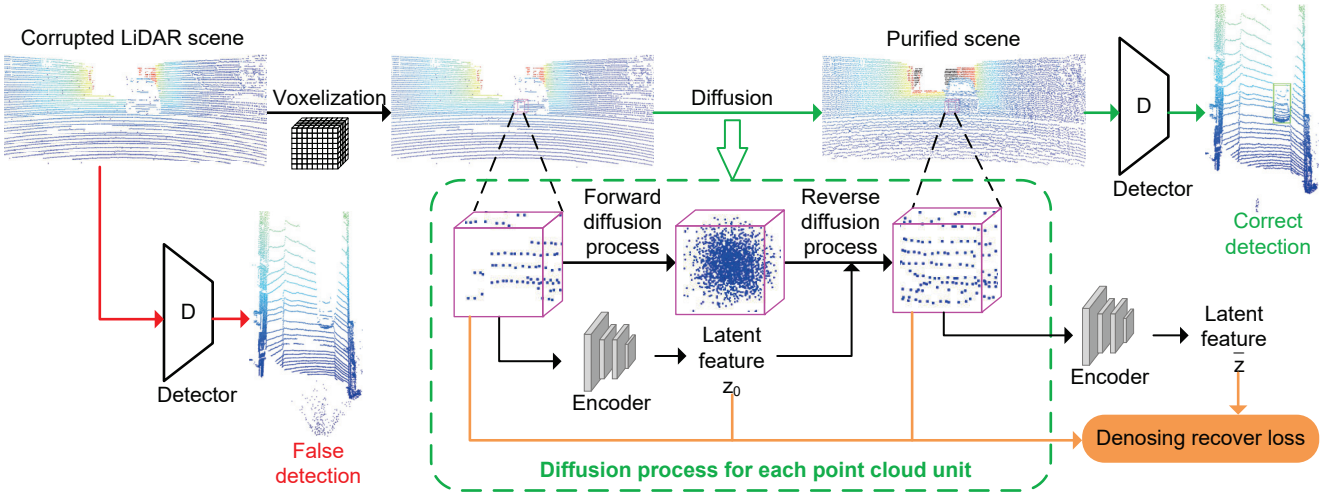


Figure 1. Overview of the proposed LiDARPure method for purifying the corrupted LiDAR scene. Given the corrupted LiDAR scene, it is divided into independent point cloud units through voxelization. LiDARPure utilizes a diffusion model to purify the point cloud within each unit. The corrupted points within the unit are converted into noise through a forward diffusion process, and then restored to purified point clouds through reverse diffusion process. The latent feature of the point cloud within the unit are extracted using the encoder and participate as the condition in the reverse diffusion process. Each unit and its latent features before (z_0) and after ($\bar{z}$) purification are used to calculate the denoising recovery loss. All units complete the diffusion process to form a purified scene.

We choose to use the conditional diffusion model to add noise and denoise from the scene point cloud, smoothing out corrupt information in the data. Different from single object points in 3D recognition tasks, large-scale LiDAR has a larger scene range and more sparse point cloud data. Therefore, referring to the 3D detector CenterPoint [23], we divided the LiDAR scene into several units using voxels in VoxelNet [18]. Additionally, utilizing pillars in PointPillars [20] instead of voxels is optional. If the corrupted scene $\hat{S}$ is divided into N units, it can be represented as $\hat{S} = (u^1, \ldots, u^N)$. Unlike the approach of Nunes et al. [43], we perform the diffusion process on a per unit basis. Therefore, for the reverse diffusion process, its output is $S' = (u^{1'}, \ldots, u^{N'})$. By using the latent feature extractor E for each unit in VoxelNet or PointPillars, the latent features of all units in the corrupted scene can be represented as $Z = (z^1, \ldots, z^N) = E(\hat{S})$.

According to Formula (4), the forward diffusion process of unit u^n can be described as follows:

$$u_t^n = \sqrt{\overline{\alpha}_t} u_0^n + \sqrt{1 - \overline{\alpha}_t}\epsilon, \quad \forall u_n \in \hat{S}, \tag{10}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\overline{\alpha}_t = \prod_{i=1}^{t}(1 - \beta_i)$. During the forward diffusion process, the points within each unit and the entire LiDAR scene gradually transform into Gaussian noise. The corrupt information is gradually smoothed out, and the geometric information of the scene is also blurred. Therefore, in the process of reverse diffusion, we utilize the latent features of each unit to assist the generation and restoration of the scene. We calculate the reverse diffusion process for the n-th unit using Formula (9), which can be rewritten as follows:

$$p_\theta(u_{0:T}^n | z^n) = p(u_T^n) \prod_{t=1}^{T} p_\theta\left(u_{t-1}^n | u_t^n, z^n\right), \tag{11}$$

$$p_\theta\left(u_{t-1}^n | u_t^n, z^n\right) = \mathcal{N}(u_{t-1}^n | \mu_\theta^n(u_t^n, t, z^n), \beta^n \mathbf{I}), \tag{12}$$

where $\mu_\theta^n(u_t^n, t, z^n)$ represents the approximate mean value of neural network parameterization. With the introduction of conditional latent features z_n, $\mu_\theta^n(u_t^n, t, z^n)$ can be further written according the reparameterization method similar to DDPM:

$$\mu_\theta^n(u_t^n, t, z^n) = \frac{1}{\sqrt{1-\beta_t}}(u_t^n - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta^n(u_t^n, t, z^n)). \tag{13}$$

Based on Formula (11), the reverse diffusion process of n-th unit in the scene can be approximated using a neural network, which can be expressed as follows in step t:

$$u_{t-1}^n = u_t^n - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta^n(u_t^n, t, z^n) + \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{14}$$

where $\epsilon_\theta^n(u_t^n, t, z^n)$ is the noise predictor at time t about u_t^n. The noise predictor ϵ_θ^n is combined with the encoder E in the unit for training to ensure that the diffusion model better utilizes the extracted latent features. We independently perform forward diffusion and reverse diffusion processes for each unit, and utilize their geometric latent features as conditions in the reverse denoising process. Based on the above formula and derivation, we can use the conditional diffusion model to purify each unit in the scene. It is important to note that the flexibility of LiDARPure allows it to select arbitrary spatial processing and point cloud feature extraction units. Voxels and pillars are optional large-scale scene feature extraction units that are completely independent, and the performance analysis on these two units is provided in the experiments section.

4.4. Loss Function

The noise predictor ϵ_θ^n is trained to predict the noise at each step of the denoising process. Based on this noise predictor, the difference between the prediction and Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ can be calculated. For the n-th voxel or pillar unit, the noise difference at step t can be computed as the $l2$-norm distance:

$$dis^n = ||\epsilon - \epsilon_\theta^n(u_t^n, t, z^n)||_2. \tag{15}$$

The above distance measurement is used for each step of the reverse diffusion process of the independent unit, making the output of the noise predictor more accurate and restoring the most realistic unit during the denoising process. For the entire scene, the reverse diffusion process of all units is synchronous, so the single step denoising distance loss of the entire LiDAR scene is formulated as follows:

$$\mathcal{L}_{dis} = \sum_{n=0}^{N} dis^n. \tag{16}$$

Considering that the latent features of the scene serve as conditions to supervise the entire reverse diffusion process and play an important role in 3D object detection, it is crucial to pay attention to the changes in latent features after the reverse diffusion process. Therefore, the latent feature recovery measurement is proposed to measure the latent feature differences between the scene purified using the diffusion process and the pure scene. Similar to the distance loss $\mathcal{L}_{dis}$, we independently calculate the difference in the latent features of each voxel or pillar unit's point cloud after the diffusion process:

$$diff^n = ||\overline{z}^n - z_0^n||_2, \tag{17}$$

where $\overline{z}^n$ represents the latent feature extracted using the n-th unit of the pure sample. z_0^n represents the corresponding latent feature of the purified sample after the t-step reverse diffusion process. In addition to the differences between abstract latent features, we believe that maintaining the geometric position of the point clouds in each unit is equally important.

We use the Hausdorff distance to control the geometric information of point clouds within each unit:

$$pos^n = Hausdorff(\overline{u}^n, u^{n'}), \tag{18}$$

where $\overline{u}$ represents the point cloud units in the pure sample. The Hausdorff distance is often used to measure the maximum difference in geometric information between two point cloud sets.

The latent features and geometric information of each unit after purification are controlled based on $diff^n$ and pos^n. Therefore, the denoising recovery loss of the entire scene can be represented as follows:

$$\mathcal{L}_{rec} = \sum_{n=0}^{N} (diff^n + pos^n). \tag{19}$$

The denoising recovery loss ensures that the latent features and geometric information of the purified scene after the reverse diffusion process are closer to the pure scene, which makes it perform better as the input in subsequent 3D object detection tasks.

5. Experiments

In this section, the experimental results of the proposed LiDARPure method are reported. First, we introduce our experimental setup in Section 5.1. The quantitative results and qualitative results are presented in Sections 5.2 and 5.3. In addition, ablation studies are discussed in detail in Section 5.4.

5.1. Experimental Setup

Datasets and Detectors. In this part, we describe the experimental setup focusing on the dataset and the victim 3D object detectors employed for evaluation. KITTI [48], a seminal dataset for autonomous driving research, boasts a wealth of dedicated LiDAR scene data, comprising 3712 training samples and 3769 validation samples, among others. These samples encapsulate a diverse range of real-world driving scenarios, from bustling urban streets to open highways, ensuring a comprehensive evaluation of 3D object detection methods.

As for the 3D object detectors utilized in our experiments, we selected three prominent approaches: PointPillars [20], PointRCNN [21], and PV-RCNN [22]. PointPillars revolutionizes point cloud representation through pillars, achieving a balance between efficiency and accuracy. PointRCNN adopts a two-stage architecture, generating and refining 3D proposals, and is adept at handling sparse and noisy LiDAR data. PV-RCNN combines voxel-based and point-based features, enhancing global and local context understanding. Together, these detectors showcase the diversity and progress in 3D object detection from LiDAR point clouds.

Common corruptions. Our study builds upon the seminal work presented by Dong et al. [6], which introduces an extensive benchmark of 27 common data corruptions and their corresponding simulation methods, tailored to mimic the diverse challenges encountered by LiDAR sensors and 3D object detection. To address these corruptions, we utilized the proposed LiDARPure method to purify the damage in the data through a diffusion model. Our study employs the comprehensive corruption framework proposed in the aforementioned paper as a testbed and conducted detailed adversarial purification experiments on 19 types of LiDAR data corruption. Dong et al. divided each type of corruptions into five levels, and we chose level three with moderate corruption for experiments that are closer to real scenarios. The detection performance with LiDARPure purified scene data can demonstrate their effectiveness in enhancing the robustness of 3D object detection systems against these challenging conditions.

Evaluation Metrics. The commonly used evaluation metrics in 3D object detection are average precision and mean average precision. Considering the most common scenario, we use robust average precision to measure the performance of the proposed method,

which represents the AP of defense strategies in responding to adversarial attacks. In addition, the first column in the table represents its standard AP without applying any defense strategy on pure data for comparison. We choose vehicles with moderate difficulty in the dataset to calculate robust AP. For each detector, the robust AP without LiDARPure applied for adversarial purification is also included to show the impact of LiDARPure on the performance.

Implementation details. During the training process, we selected the Adam [49] optimizer and set the learning rate to 1×10^{-4}. A total of 80 epochs were trained, halving the learning rate every 20 epochs. For the parameters of the diffusion model, we referred to the method of Nunes et al. [43] and set β_0 and β_T to 3.5×10^{-5} and 5×10^{-3}, respectively. The number of forward diffusion steps T was set to 1000. In the process of reverse diffusion, we used DPMSolver [50] to accelerate denoising, which can reduce the denoising steps T to 50. When voxels were selected for feature extraction, their resolution was set to 0.2 m. For pillars, the resolution was 0.15 m. A more detailed discussion on resolution will be provided in ablation studies (Section 5.4). After the diffusion process, redundant scene points were sampled to the same number of points before being fed into the detector. The experiments were conducted on a personal computer equipped with an i7 13700 CPU, 64 GB of RAM, and an RTX 4090 graphics processing unit.

5.2. Comparison with State-of-the-Art Methods

In this section, we demonstrate LiDARPure's adversarial purification performance against common data corruptions and compare it with state-of-the-art defense strategies. The methods of adversarial training mainly include LISA [28] and adversarial simulation by Hahner et al. [29,31]. In addition, Teufel et al.'s adversarial training based on data augmentation method [33] is also used for performance comparison. To our knowledge, there is currently no adversarial purification method specifically designed for LiDAR sensors and 3D object detection. Therefore, we chose the purification methods DUP-Net [34] and IF-Defense [37] in 3D point cloud recognition for comparison. LPC [38] is not chosen because its structural design can only be used for the defense of point cloud classification models, making it difficult to apply to 3D object detection models.

The average precision of the PointPillars detector under data corruptions with defense strategy is shown in Table 2. The voxel and pillar in bracket denote the units utilised to perform the potential feature extraction and diffusion processes, respectively. At the same time, the standard average precision of the PointPillars detector is displayed in the unit corresponding to "No Corruption" and "standard AP". From Table 2, we can see that existing adversarial training methods [28,29,31,33] for LiDAR scene data are effective for specific data corruption, but they will reduce the performance of the detector when exposed to other data corruptions. And adversarial training for LiDAR data slightly reduces the standard average precision of the detector in a pure environment. In addition, the adversarial purification methods DUP-Net and IF-Defense for 3D point cloud recognition significantly reduce the average precision of the target detector in all situations. Intuitively, this is because these methods focus on removing outliers and maintaining surface consistency, resulting in distant target objects in the scene being removed due to fewer points. This also proves that adversarial purification methods for 3D point cloud recognition are not suitable for LiDAR data corruption and 3D object detection models. In contrast, the proposed LiDARPure method effectively purifies scene data without targeting specific data corruption and outperforms existing adversarial training methods by more than 2% of robust AP in most cases. It should be noted that both voxel-based and pillar-based LiDARPure have almost no impact on the performance of the detector in clean scenes, maintaining the standard average precision of the detector as much as possible.

Table 2. The standard AP (%) and robust AP (%) of various defense strategies when exposed to 19 types of data corruption on PointPillars detector. The basic unit for diffusion process is represented in bracket. The best results are shown in bold and previous best results are underlined.

Corruptions	(Standard AP)	Defense Strategies (Robust AP)							
		Snowfall Simulation [31]	LISA [28]	Fog Simulation [29]	Teufel et al. [33]	DUP-Net [34]	IF-Defense [37]	Ours (Voxel)	Ours (Pillar)
No Corruption	78.4	78.1	77.5	77.9	78.0	54.2	59.4	78.3	**78.4**
Snow	36.5	39.5	34.0	33.1	37.8	26.2	28.2	**44.6**	44.0
Rain	36.2	35.2	38.8	35.7	39.6	25.3	26.2	**43.9**	43.1
Fog	64.3	64.1	63.8	66.3	64.3	43.4	47.4	**71.7**	71.5
Sunlight	62.3	61.4	61.7	61.5	61.7	45.0	48.7	**66.8**	66.4
Density	76.5	76.2	75.9	76.2	75.4	54.0	56.9	77.2	**77.6**
Cutout	70.3	70.1	70.0	70.4	69.8	50.0	54.1	73.0	**73.2**
Crosstalk	70.9	68.7	69.3	68.8	69.0	48.9	50.5	71.6	**72.5**
Gaussian	74.7	74.0	73.8	74.2	73.8	50.4	59.3	**77.1**	76.5
Uniform	77.3	77.2	76.8	77.2	76.4	56.3	56.1	**78.0**	77.7
Impulse	78.2	78.2	78.1	78.3	78.1	54.1	60.9	78.2	78.2
Moving object	50.2	50.3	50.0	52.1	52.5	33.8	37.4	48.4	50.2
Local Density	69.6	69.4	69.5	69.7	69.9	48.8	52.4	**73.6**	73.4
Local Cutout	61.8	60.6	61.6	60.5	60.4	41.9	44.3	**66.2**	65.7
Local Gaussian	76.7	76.7	76.4	77.0	76.5	52.8	60.7	**77.7**	77.6
Local Uniform	78.0	78.0	77.9	78.2	78.0	54.2	59.5	**78.2**	77.9
Local Impulse	78.4	78.4	78.3	78.4	78.4	53.3	56.5	**78.5**	78.4
Shear	39.6	36.9	39.0	37.8	36.6	28.7	31.2	39.9	**40.4**
Scale	70.3	70.2	69.9	70.6	69.4	47.7	51.4	**73.3**	73.2
Rotation	72.7	72.8	72.3	72.7	72.6	50.5	57.9	**73.4**	72.2
Average	65.5	65.2	65.1	65.2	65.3	45.5	49.5	**68.0**	67.9

Furthermore, the LiDARPure method was also launched on PointRCNN and PV-RCNN detector. The quantitative results are presented in Tables 3 and Table 4, respectively. From Tables 3 and 4, we can see that the proposed LiDARPure method improves the average robustness precision of 3D object detectors using various point cloud representations. The higher the standard averaging precision, the higher the robust AP of the detector after it has been cleaned up from corruption by LiDARPure. This fully demonstrates the effectiveness of LiDARPure's purification and generation strategies based on diffusion models for various types of corruption in LiDAR scene data. In addition, we can also observe that the voxel-based diffusion solution generally performs better than the pillar-based diffusion solution. This is because when the data corruption in the scene is mainly additional noise, the voxel-based solution removes the noise in each smaller unit. Adverse weather conditions essentially add a large amount of noise to the close range space of sensors, so voxel-based solutions are also effective in addressing data corruptions under adverse weather. Meanwhile, the pillar-based approach performs better in handling large-scale corruptions such as point density decrease and shear in the scene. Overall, the quantitative results in the three tables demonstrate that LiDARPure has excellent purification ability for common data corruptions in LiDAR point cloud scenes, which can significantly improve the average adversarial precision of 3D object detectors and surpass existing adversarial training strategies.

Table 3. The standard AP (%) and robust AP (%) of various defense strategies when exposed to 19 types of data corruption on PointRCNN detector. The basic unit for diffusion process is represented in bracket. The best results are shown in bold and previous best results are underlined.

Corruptions	(Standard AP)	Defense Strategies (Robust AP)							
		Snowfall Simulation [31]	LISA [28]	Fog Simulation [29]	Teufel et al. [33]	DUP-Net [34]	IF-Defense [37]	Ours (Voxel)	Ours (Pillar)
No Corruption	80.6	80.5	80.1	79.4	80.2	55.9	58.9	80.3	80.0
Snow	50.4	54.3	49.8	49.9	52.7	36.0	36.3	**55.9**	54.5
Rain	51.3	51.0	53.4	50.6	51.9	36.4	36.8	**54.2**	53.8
Fog	72.1	71.6	71.9	72.7	72.8	51.6	54.7	**73.7**	73.1
Sunlight	62.8	61.6	62.1	62.0	62.2	42.5	48.3	**64.3**	63.6

Table 3. *Cont.*

Corruptions	(Standard AP)	Defense Strategies (Robust AP)							
		Snowfall Simulation [31]	LISA [28]	Fog Simulation [29]	Teufel et al. [33]	DUP-Net [34]	IF-Defense [37]	Ours (Voxel)	Ours (Pillar)
Density	80.4	78.2	80.5	79.7	79.4	56.0	62.3	80.3	**80.5**
Cutout	73.9	73.6	71.6	73.2	72.5	52.1	53.4	74.7	**75.1**
Crosstalk	71.5	69.8	71.2	71.5	71.3	51.6	52.0	73.5	**73.7**
Gaussian	61.2	60.9	60.0	60.1	60.3	42.5	47.6	**64.1**	63.6
Uniform	76.4	76.5	74.9	76.3	76.6	52.1	55.5	**77.5**	77.2
Impulse	79.8	78.9	78.8	77.8	79.1	56.9	57.0	**80.2**	80.0
Moving object	50.5	50.2	50.5	50.3	50.2	34.7	37.8	51.9	**52.6**
Local Density	74.2	72.8	72.7	73.1	73.2	52.4	54.9	**77.7**	76.8
Local Cutout	67.9	66.2	66.6	67.4	66.7	45.8	48.9	**69.9**	69.7
Local Gaussian	69.8	68.7	67.5	69.2	69.0	47.5	54.3	**72.4**	72.1
Local Uniform	77.7	77.5	76.1	76.5	77.4	53.9	58.9	**79.6**	79.3
Local Impulse	80.3	80.1	78.9	79.7	79.8	54.6	61.5	80.2	**80.3**
Shear	39.8	39.7	39.8	39.1	39.5	27.0	30.7	40.9	**41.4**
Scale	71.5	71.6	70.3	71.3	71.1	49.8	54.5	**74.1**	73.5
Rotation	75.6	75.4	75.4	74.4	75.3	51.6	54.6	**77.2**	77.0
Average	67.7	67.3	66.9	67.1	67.4	47.1	50.5	**69.6**	69.3

Table 4. The standard AP (%) and robust AP (%) of various defense strategies when exposed to 19 types of data corruption on PV-RCNN detector. The basic unit for the diffusion process is represented in the brackets. The best results are shown in bold and previous best results are underlined.

Corruptions	(Standard AP)	Defense Strategies (Robust AP)							
		Snowfall Simulation [31]	LISA [28]	Fog Simulation [29]	Teufel et al. [33]	DUP-Net [34]	IF-Defense [37]	Ours (Voxel)	Ours (Pillar)
No Corruption	84.4	84.0	83.9	83.8	84.0	58.0	65.5	84.2	84.1
Snow	52.4	55.7	51.6	51.2	54.6	37.7	39.2	**56.5**	56.0
Rain	51.6	51.3	53.1	50.8	52.8	36.0	39.6	**54.1**	53.7
Fog	79.5	79.0	79.2	80.7	80.4	54.7	57.8	**82.1**	81.6
Sunlight	79.9	79.4	78.2	79.1	79.7	53.6	60.6	**82.0**	81.3
Density	82.8	82.6	81.0	79.5	81.7	57.8	63.1	83.2	**83.5**
Cutout	76.1	74.3	74.3	76.2	75.6	53.1	55.3	77.5	**78.0**
Crosstalk	82.3	80.4	79.6	82.4	82.6	56.9	58.7	82.6	**82.8**
Gaussian	65.1	63.4	63.7	63.0	64.0	43.9	50.1	**68.6**	68.4
Uniform	81.2	80.6	79.2	78.4	80.4	58.4	62.8	82.1	**82.2**
Impulse	82.8	81.6	81.4	80.4	81.1	57.4	62.1	**83.3**	83.0
Moving object	54.6	53.7	53.0	53.4	54.0	39.0	42.1	**55.8**	55.5
Local Density	77.6	75.3	76.6	75.6	76.9	56.2	55.2	**80.3**	79.8
Local Cutout	72.3	70.6	70.7	71.3	71.4	50.4	51.9	75.8	**75.9**
Local Gaussian	70.4	68.5	69.5	67.9	69.4	50.3	51.4	**74.1**	73.4
Local Uniform	82.1	80.3	80.9	79.4	80.5	58.0	63.6	**83.4**	83.2
Local Impulse	84.0	81.7	81.1	83.2	82.6	58.7	61.0	84.0	**84.1**
Shear	47.7	47.0	46.7	47.4	47.0	33.0	36.2	48.6	**49.3**
Scale	76.8	75.0	75.9	74.3	75.7	54.8	55.0	78.7	**78.8**
Rotation	79.9	77.6	77.9	79.7	79.0	55.1	62.0	**80.4**	80.0
Average	72.6	71.5	71.2	71.3	72.1	50.8	54.1	**74.4**	74.2

5.3. Visualization

In this section, we present the qualitative results of the proposed LiDARPure method. Snowfall is a representative severe weather condition, and we used the LiDARPure method to purify the LiDAR scene that was corrupted by simulated snowfall. The qualitative results are shown in Figure 2. The first line in the figure is a image captured using camera. The bird's eye view of the scene and an enlarged view of the vehicles are shown in the second and third lines.

From the first line of Figure 2, we can see that snowfall generates a lot of noise at close range on the LiDAR and obscures the scene behind, resulting in a large number of voids in the rear space. In the corrupted scene, both the ground and target vehicles have some missing points, making it difficult to identify the points of the vehicles. After purification

using LiDARPure, a large amount of noise generated by snowfall at close range has been removed from the LiDAR. And from the second line, we can learn that although it cannot recover point clouds in completely blocked void areas, more points are generated around a small number of points of the target vehicle. This greatly increases the probability of the vehicle being perceived by the detector.

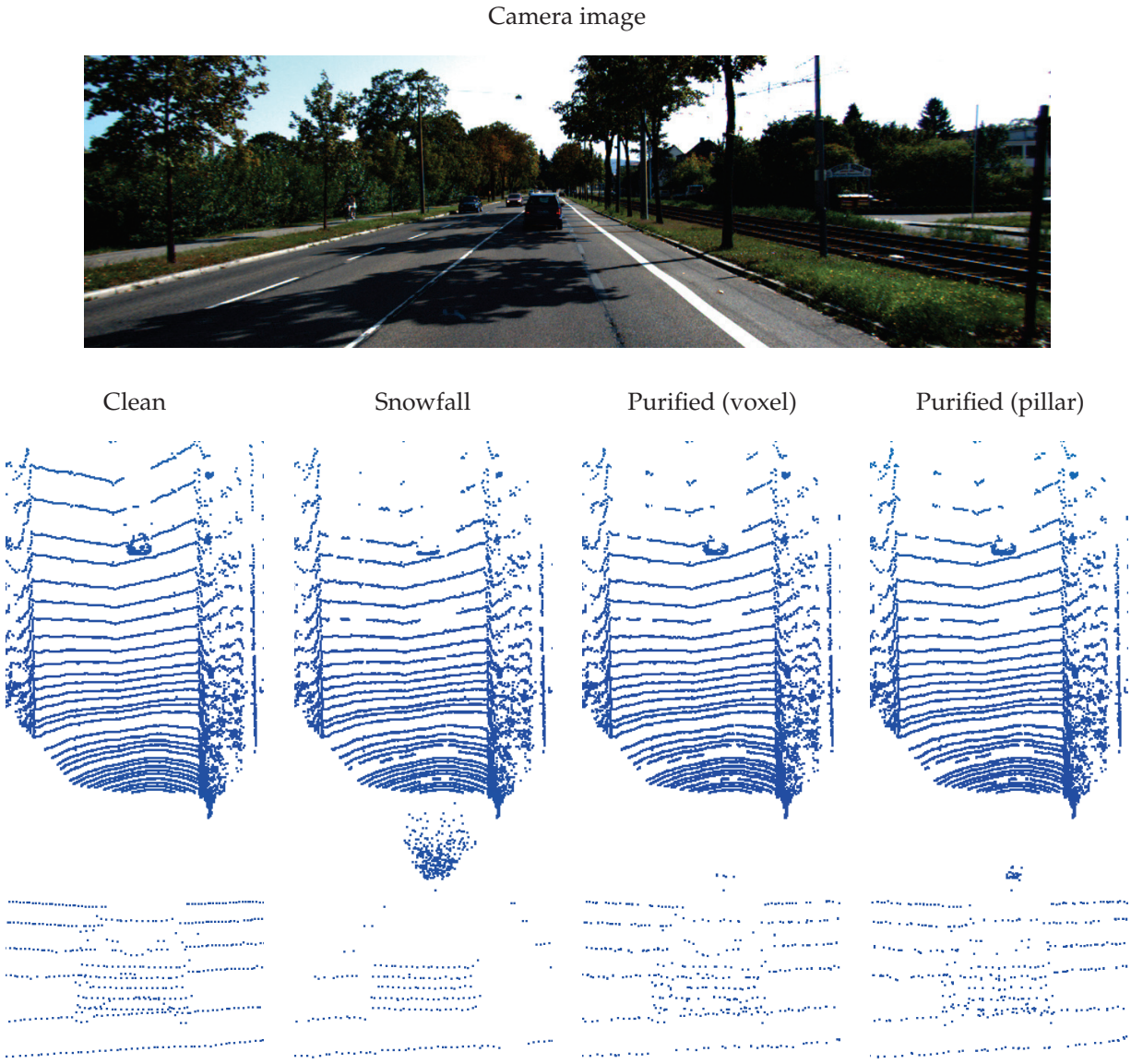


Figure 2. The qualitative results of the proposed LiDARPure method purify the LiDAR scene corrupted by snowfall. The units used are indicated in the brackets.

Gaussian noise is a typical global data corruption in LiDAR sensor. In Figure 3, we present the qualitative results of using the LiDARPure method to purify global Gaussian noise. From the figure, we can see that although the LiDARPure method cannot fully restore the regularity of the arrangement of points in the scene and objects; it can smooth out the level of noise and make each point deviate from its original position by a smaller margin. A lower noise level makes the detection results output by the detector closer to the ground truth. In addition to improving the performance of the detector, compared to the chaotic points after corruption, the points in the purified scene appear more regular and shaped visually. However, this also indicates that diffusion models targeting each voxel or pillar unit alone are difficult to learn the arrangement properties of LiDAR point cloud data.

In the process of purifying each unit, its neighborhood and global geometric information should also be considered in order to better restore the LiDAR scene.

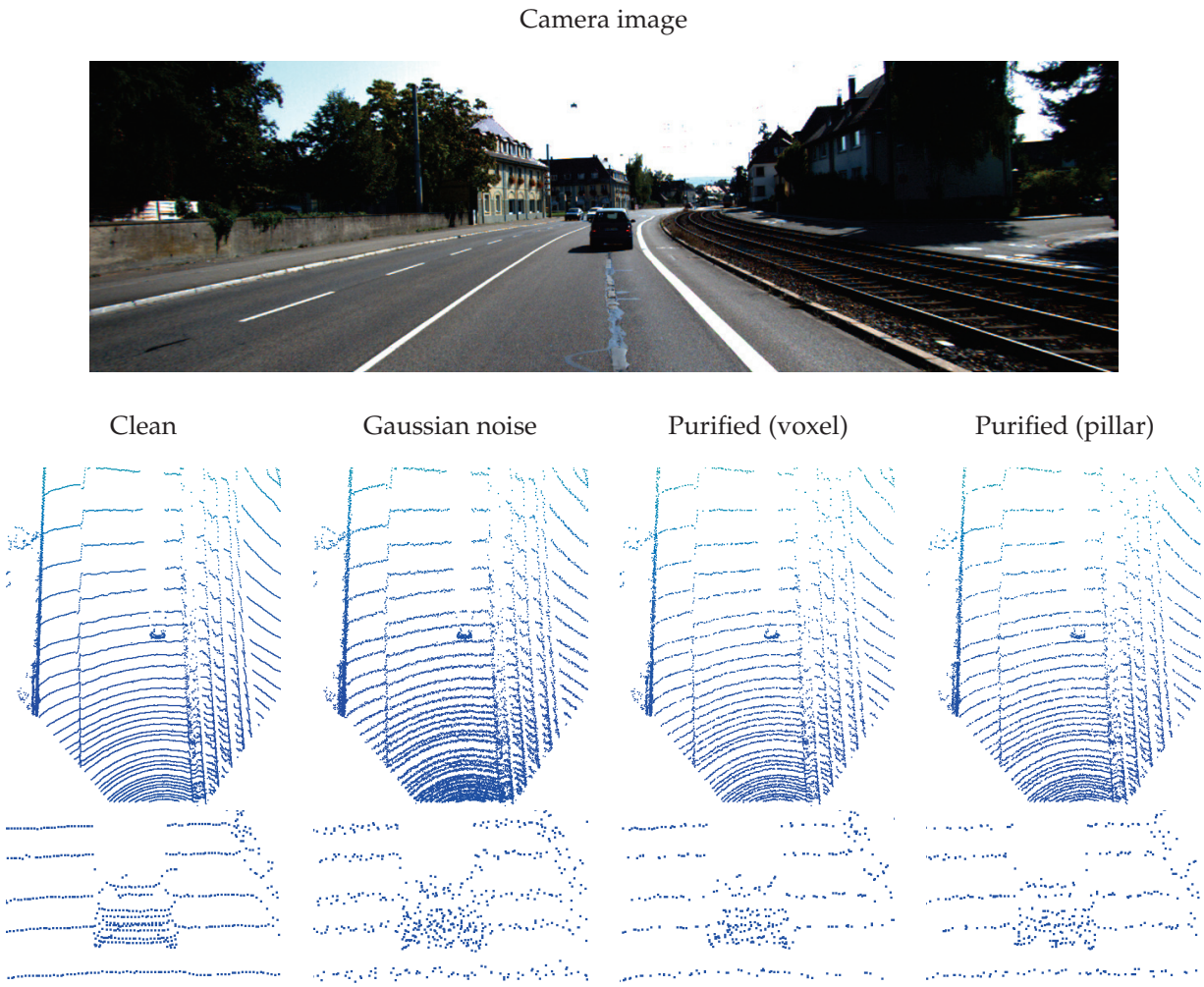


Figure 3. The qualitative results of proposed LiDARPure method purifies the LiDAR scene corrupted by Gaussian noise. The units used are indicated in bracket.

5.4. Ablation Study

Denoising recovery loss $\mathcal{L}_{rec}$. The denoising recovery loss $\mathcal{L}_{rec}$ has been proposed to guide the regeneration of point clouds during the reverse diffusion process. And it mainly consists of two parts, namely $diff$ for constraining latent features differences and pos for controlling the geometric information of point clouds. We ablate them separately to demonstrate their effectiveness. In addition, we also replaced the proposed denoising recovery loss $\mathcal{L}_{rec}$ with traditional l_2-$norm$ loss and Chamfer distance loss, and compared their effects on the performance of the reverse diffusion process. It is worth noting that using l_2-$norm$ loss and Chamfer distance loss requires downsampling the number of points within the unit to be the same as before the diffusion process. The detailed comparison of the loss function is presented in Table 5 and the performance of the denoising recovery loss is based on voxel units.

It can be learnt from the Table 5 that the proposed denoising recovery loss performs better than the l_2-$norm$ loss and Chamfer distance loss on all detectors. And both $diff$ and pos in $\mathcal{L}_{rec}$ help the diffusion model complete the reverse diffusion process. The performance of $\mathcal{L}_{rec}$ may not be as good as the Chamfer distance loss when one of them is used alone, whereas the huge improvement in the mean robust AP when they are used together reflects the effectiveness of their combination.

Table 5. The mean robust AP (%) of all data corruptions under different loss function on three victim detectors.

Denoising Recovery Loss		l_2-*norm*	Chamfer Distance	Mean Robust AP		
diff	*pos*			PointPillars	PointRCNN	PV-RCNN
-	-	✓	×	64.1	67.4	69.7
-	-	×	✓	65.9	68.3	71.3
✓	×	-	-	65.6	68.8	72.8
×	✓	-	-	66.4	68.2	71.5
✓	✓	-	-	68.0	69.6	74.4

Impact of diffusion steps T**.** We investigated the impact of diffusion steps T on the detector's corruption robustness during the reverse diffusion process. The ablation results on the PV-RCNN detector are shown in Figure 4. From the figure, we can see that the robustness of the detector to various common data corruptions increases with T. Fog can cause a small number of noise points in the scene, which has a low impact on the performance of the detector. Therefore, when T is equal to 5, the detector also achieves high robust AP. Due to the fact that rain and snow generate a lot of noise in areas close to the sensor and block objects that are behind, its initial robust AP only exceeds 40. Because the blocked void space cannot be fully restored, its final performance is lower than 60. The global Gaussian noise and shear have a significant impact on the performance of the detector, so its robust AP is below 20 when $T = 5$. However, as the reverse diffusion process progresses, the data corrupted by Gaussian noise can eventually be restored to a robust AP of nearly 70. And the sheared scenes can also be well restored, with a robust AP four times higher than the initial state. For adverse weather corruptions, the detector can achieve high robustness in early steps T and the performance stabilizes when T is greater than 30. However, for severe corruptions such as global Gaussian noise and object shear, the robustness average precision tends to converge only after T exceeds 40. This is because the noise close to LiDAR sensor generated by adverse weather can be quickly eliminated, while global point perturbation and noise require a repeated diffusion process to smooth out. Therefore, in order to make a balance between performance and computational costs, we choose $T = 50$ to balance performance and expenses.

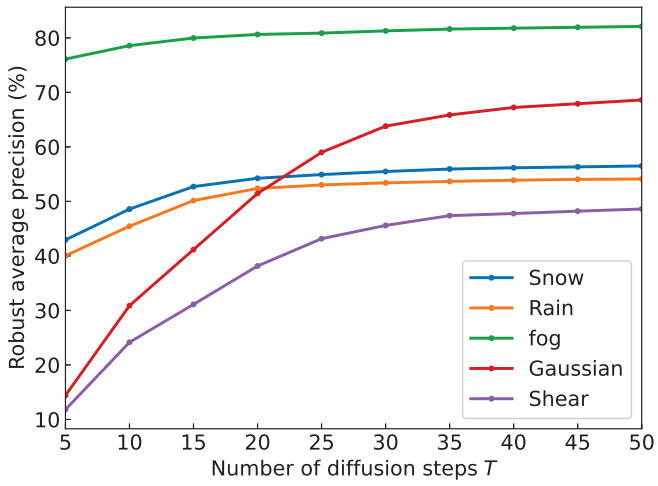


Figure 4. Robust AP affected by different steps T on PV-RCNN detector in the reverse diffusion process.

Impact of unit resolution r**.** Resolution affects the size of each unit in space and the number of its points. Due to the structural differences between voxel and pillar, it is necessary to choose different resolutions for them. We study the performance of LiDARPure at different resolutions. All types of corruption have been taken into account, so we calculated the mean of robust AP and time consumption to better select the resolution r. The results of voxel units and pillar units are shown in Table 6.

Table 6. The mean robust AP (%) of all data corruptions and time consumption (s) under different resolutions r.

	r for Voxel					r for Pillar				
	0.1	**0.15**	**0.2**	**0.25**	**0.3**	**0.1**	**0.15**	**0.2**	**0.25**	**0.3**
PointPillars	64.1	67.8	68	65.2	58.9	68.1	67.9	64.2	60.6	52.3
PointRCNN	64.0	68.3	69.6	65.0	59.7	67.8	69.3	67.8	62.0	57.1
PV-RCNN	67.4	72.6	74.4	69.2	62.2	72.9	74.2	72.0	69.5	64.8
time consumption	2.74	2.16	1.31	0.85	0.64	1.67	1.29	0.94	0.56	0.33

From the table, we can see that the optimal resolutions for voxel and pillar units are different. The excessive number or quantity of points within a unit is not conducive to the purification of diffusion processes. A pillar has an infinite vertical space with the same base area as a voxel. For a LiDAR scene with a limited number of point clouds, in order to make the number of points in the pillar and voxel units similar, it is necessary to lower the bottom resolution of the pillar. Furthermore, the resolution r greatly affects the time consumption of the proposed method. For both types of units, the smaller r leads to more units to be subjected to diffusion and denoising processes. From the table, we can understand that the single sample time consumption reaches more than 1.5 s for both units when r is equal to 0.1 m and the time consumption decreases as r increases. In order to balance the performance and time consumption, we choose $r = 0.2$ m for voxel and $r = 0.15$ m for pillar.

6. Conclusions

In this work, we propose LiDARPure, a novel approach utilizing diffusion models to effectively purify common corruptions in LiDAR data for 3D object detection. By conducting independent diffusion processes for each unit, LiDARPure overcomes the challenge of gradient confusion in adversarial training for LiDAR scenes. The utilization of the latent features of the scene as conditions for the diffusion models further improved the restoration process, resulting in a high-fidelity reconstruction of the original LiDAR scene. Our experiments have demonstrated that LiDARPure can effectively purify 19 common types of LiDAR data corruption, significantly improving the robustness of 3D object detection models against corrupted LiDAR data. In addition, the proposed method has two main limitations. First, the method fails to recover the original point cloud in a completely empty region due to severer information loss induced by LiDAR capture. Second, the computational consumption of the whole diffusion process is high, and limits its application in real-time scenarios. However, the flexibility and scalability of the proposed approach make it applicable to a wide range of LiDAR sensors and datasets. The modular design of LiDARPure allows for easy integration into existing 3D object detection pipelines, facilitating its adoption in real-world applications.

Author Contributions: Conceptualization, M.C. and X.W.; methodology, M.C.; software, M.C.; validation, M.C. and X.W.; formal analysis, M.C.; investigation, M.C.; resources, M.C.; data curation, M.C.; writing—original draft preparation, M.C.; writing—review and editing, X.W., F.S. and H.L.; visualization, M.C.; supervision, X.W., F.S. and H.L.; project administration, M.C.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China (No. 62072076), Sichuan Provincial Research Plan Project (No. 2022ZDZX0005).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LiDAR	Light Detection and Ranging
GANs	Generative adversarial networks
VAE	Variational autoencode
NMS	Non-Maximum Suppression
AP	Average Precision
DDPM	Denoising Diffusion Probabilistic Models

References

1. Fernandes, D.; Silva, A.; Névoa, R.; Simões, C.; Gonzalez, D.; Guevara, M.; Novais, P.; Monteiro, J.; Melo-Pinto, P. Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Inf. Fusion* **2021**, *68*, 161–191. [CrossRef]
2. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.J.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
3. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE Computer Society: Washington, DC, USA, 2016; pp. 2574–2582. [CrossRef]
4. Tu, J.; Ren, M.; Manivasagam, S.; Liang, M.; Yang, B.; Du, R.; Cheng, F.; Urtasun, R. Physically Realizable Adversarial Examples for LiDAR Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; Computer Vision Foundation/IEEE: Piscataway, NJ, USA, 2020; pp. 13713–13722. [CrossRef]
5. Wang, X.; Cai, M.; Sohel, F.; Sang, N.; Chang, Z. Adversarial point cloud perturbations against 3D object detection in autonomous driving systems. *Neurocomputing* **2021**, *466*, 27–36. [CrossRef]
6. Dong, Y.; Kang, C.; Zhang, J.; Zhu, Z.; Wang, Y.; Yang, X.; Su, H.; Wei, X.; Zhu, J. Benchmarking Robustness of 3D Object Detection to Common Corruptions in Autonomous Driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1022–1032. [CrossRef]
7. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; OpenReview.net: Amherst, MA, USA, 2018.
8. Gowal, S.; Rebuffi, S.A.; Wiles, O.; Stimberg, F.; Calian, D.A.; Mann, T.A. Improving Robustness using Generated Data. In Proceedings of the Annual Conference on Neural Information Processing Systems, Virtual, 6–14 December 2021; pp. 4218–4233.
9. Shi, C.; Holtz, C.; Mishne, G. Online Adversarial Purification based on Self-supervised Learning. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021; OpenReview.net: Amherst, MA, USA, 2021.
10. Yoon, J.; Hwang, S.J.; Lee, J. Adversarial Purification with Score-based Generative Models. In Proceedings of the International Conference on Machine Learning—PMLR, Virtual Event, 18–24 July 2021; Volume 139, pp. 12062–12072.
11. Croce, F.; Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Proceedings of the International Conference on Machine Learning—PMLR, Virtual Event, 13–18 July 2020; Volume 119, pp. 2206–2216.
12. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.C.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
13. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
14. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In Proceedings of the Annual Conference on Neural Information Processing Systems, Virtual, 6–12 December 2020.
15. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021; OpenReview.net: Amherst, MA, USA, 2021.
16. Luo, S.; Hu, W. Diffusion Probabilistic Models for 3D Point Cloud Generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 20–25 June 2021; Computer Vision Foundation/IEEE: Piscatway, NJ, USA, 2021; pp. 2837–2845. [CrossRef]
17. Sun, J.; Wang, J.; Nie, W.; Yu, Z.; Mao, Z.; Xiao, C. A Critical Revisit of Adversarial Robustness in 3D Point Cloud Recognition with Diffusion-Driven Purification. In Proceedings of the International Conference on Machine Learning—PMLR, Honolulu, HI, USA, 1 July 2023; Volume 202, pp. 33100–33114.
18. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; Computer Vision Foundation/IEEE Computer Society: Washington, DC, USA, 2018; pp. 4490–4499. [CrossRef]
19. Yan, Y.; Mao, Y.; Li, B. SECOND: Sparsely Embedded Convolutional Detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]

20. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. PointPillars: Fast Encoders for Object Detection From Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; Computer Vision Foundation/IEEE: Piscataway, NJ, USA, 2019; pp. 12697–12705. [CrossRef]
21. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; Computer Vision Foundation/IEEE: Piscataway, NJ, USA, 2019; pp. 770–779. [CrossRef]
22. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020. Computer Vision Foundation/IEEE: Piscataway, NJ, USA, 2020; pp. 10526–10535. [CrossRef]
23. Yin, T.; Zhou, X.; Krähenbühl, P. Center-Based 3D Object Detection and Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 20–25 June 2021; Computer Vision Foundation/IEEE: Piscataway, NJ, USA, 2021; pp. 11784–11793. [CrossRef]
24. Zhou, Z.; Zhao, X.; Wang, Y.; Wang, P.; Foroosh, H. CenterFormer: Center-Based Transformer for 3D Object Detection. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Lecture Notes in Computer Science; Springer: Berlin, Germany, 2022; Volume 13698, pp. 496–513. [CrossRef]
25. Cao, Y.; Xiao, C.; Cyr, B.; Zhou, Y.; Park, W.; Rampazzi, S.; Chen, Q.A.; Fu, K.; Mao, Z.M. Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, London, UK, 6 November 2019; ACM: New York, NY, USA, 2019; pp. 2267–2281. [CrossRef]
26. Lehner, A.; Gasperini, S.; Marcos-Ramiro, A.; Schmidt, M.; Mahani, M.A.N.; Navab, N.; Busam, B.; Tombari, F. 3D-VField: Adversarial Augmentation of Point Clouds for Domain Generalization in 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 17274–17283. [CrossRef]
27. Shin, H.; Kim, D.; Kwon, Y.; Kim, Y. Illusion and Dazzle: Adversarial Optical Channel Exploits Against Lidars for Automotive Applications. In Proceedings of the Cryptographic Hardware and Embedded Systems, Taipei, Taiwan, 25–28 September 2017; Lecture Notes in Computer Science; Springer: Berlin, Germany, 2017; Volume 10529, pp. 445–467. _22 [CrossRef]
28. Kilic, V.; Hegde, D.; Sindagi, V.; Cooper, A.B.; Foster, M.A.; Patel, V.M. Lidar Light Scattering Augmentation (LISA): Physics-based Simulation of Adverse Weather Conditions for 3D Object Detection. *arXiv* **2021**, arXiv: 2107.07004.
29. Hahner, M.; Sakaridis, C.; Dai, D.; Gool, L.V. Fog Simulation on Real LiDAR Point Clouds for 3D Object Detection in Adverse Weather. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 15263–15272. [CrossRef]
30. Bijelic, M.; Gruber, T.; Mannan, F.; Kraus, F.; Ritter, W.; Dietmayer, K.; Heide, F. Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 13–19 June 2020; Computer Vision Foundation/IEEE: Piscataway, NJ, USA, 2020; pp. 11679–11689. [CrossRef]
31. Hahner, M.; Sakaridis, C.; Bijelic, M.; Heide, F.; Yu, F.; Dai, D.; Gool, L.V. LiDAR Snowfall Simulation for Robust 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 16343–16353. [CrossRef]
32. Yu, K.; Tao, T.; Xie, H.; Lin, Z.; Liang, T.; Wang, B.; Chen, P.; Hao, D.; Wang, Y.; Liang, X. Benchmarking the Robustness of LiDAR-Camera Fusion for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Vancouver, BC, Canada, 17–24 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 3188–3198. [CrossRef]
33. Teufel, S.; Gamerdinger, J.; Volk, G.; Gerum, C.; Bringmann, O. Enhancing Robustness of LiDAR-Based Perception in Adverse Weather using Point Cloud Augmentations. In Proceedings of the IEEE Intelligent Vehicles Symposium, Anchorage, AK, USA, 3–7 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6. [CrossRef]
34. Zhou, H.; Chen, K.; Zhang, W.; Fang, H.; Zhou, W.; Yu, N. DUP-Net: Denoiser and Upsampler Network for 3D Adversarial Point Clouds Defense. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1961–1970. [CrossRef]
35. Dong, X.; Chen, D.; Zhou, H.; Hua, G.; Zhang, W.; Yu, N. Self-Robust 3D Point Recognition via Gather-Vector Guidance. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; Computer Vision Foundation/IEEE: Piscataway, NJ, USA, 2020; pp. 11513–11521. [CrossRef]
36. Sun, J.; Cao, Y.; Choy, C.B.; Yu, Z.; Anandkumar, A.; Mao, Z.M.; Xiao, C. Adversarially Robust 3D Point Cloud Recognition Using Self-Supervisions. In Proceedings of the Annual Conference on Neural Information Processing Systems, Virtual, 6–14 December 2021; pp. 15498–15512.
37. Wu, Z.; Duan, Y.; Wang, H.; Fan, Q.; Guibas, L.J. IF-Defense: 3D Adversarial Point Cloud Defense via Implicit Function based Restoration. *arXiv* **2020**, arXiv: 2010.05272.
38. Li, K.; Zhang, Z.; Zhong, C.; Wang, G. Robust Structured Declarative Classifiers for 3D Point Clouds: Defending Adversarial Attacks with Implicit Gradients. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 15273–15283. [CrossRef]

39. Lyu, Z.; Kong, Z.; Xu, X.; Pan, L.; Lin, D. A Conditional Point Diffusion-Refinement Paradigm for 3D Point Cloud Completion. In Proceedings of the International Conference on Learning Representations, Virtual Event, 25–29 April 2022; OpenReview.net: Amherst, MA, USA, 2022.
40. Zeng, X.; Vahdat, A.; Williams, F.; Gojcic, Z.; Litany, O.; Fidler, S.; Kreis, K. LION: Latent Point Diffusion Models for 3D Shape Generation. In Proceedings of the Annual Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November 2022.
41. Xu, J.; Wang, X.; Cheng, W.; Cao, Y.P.; Shan, Y.; Qie, X.; Gao, S. Dream3D: Zero-Shot Text-to-3D Synthesis Using 3D Shape Prior and Text-to-Image Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 20908–20918. [CrossRef]
42. Lee, J.; Im, W.; Lee, S.; Yoon, S.E. Diffusion Probabilistic Models for Scene-Scale 3D Categorical Data. *arXiv* **2023**, arXiv: 2301.00527. https://doi.org/10.48550/ARXIV.2301.00527.
43. Nunes, L.; Marcuzzi, R.; Mersch, B.; Behley, J.; Stachniss, C. Scaling Diffusion Models to Real-World 3D LiDAR Scene Completion. *arXiv* **2024**, arXiv: 2403.13470. https://doi.org/10.48550/ARXIV.2403.13470.
44. Nie, W.; Guo, B.; Huang, Y.; Xiao, C.; Vahdat, A.; Anandkumar, A. Diffusion Models for Adversarial Purification. In Proceedings of the International Conference on Machine Learning—PMLR, Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 16805–16827.
45. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5099–5108.
46. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 77–85. [CrossRef]
47. Lechner, M.; Hasani, R.M.; Grosu, R.; Rus, D.; Henzinger, T.A. Adversarial Training is Not Ready for Robot Learning. In Proceedings of the IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May–5 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 4140–4147. [CrossRef]
48. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
49. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; Bengio, Y., LeCun, Y., Eds.
50. Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; Zhu, J. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. In Proceedings of the Annual Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

An Energy-Efficient Dynamic Feedback Image Signal Processor for Three-Dimensional Time-of-Flight Sensors

Yongsoo Kim [1,2], Jaehyeon So [3], Chanwook Hwang [3], Wencan Cheng [4] and Jong Hwan Ko [3,*]

[1] Department of Semiconductor and Display Engineering, Sungkyunkwan University, Suwon 16419, Republic of Korea; yongsookim@g.skku.edu
[2] Samsung Electronics, Hwaseong 18448, Republic of Korea
[3] Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon 16419, Republic of Korea; judenause@g.skku.edu (J.S.); ghkdcks12@g.skku.edu (C.H.)
[4] Department of Computer Science, National University of Singapore, Singapore 117417, Singapore; wccheng@nus.edu.sg
* Correspondence: jhko@skku.edu

Abstract: With the recent prominence of artificial intelligence (AI) technology, various research outcomes and applications in the field of image recognition and processing utilizing AI have been continuously emerging. In particular, the domain of object recognition using 3D time-of-flight (ToF) sensors has been actively researched, often in conjunction with augmented reality (AR) and virtual reality (VR). However, for more precise analysis, high-quality images are required, necessitating significantly larger parameters and computations. These requirements can pose challenges, especially in developing AR and VR technologies for low-power portable devices. Therefore, we propose a dynamic feedback configuration image signal processor (ISP) for 3D ToF sensors. The ISP achieves both accuracy and energy efficiency through dynamic feedback. The proposed ISP employs dynamic area extraction to perform computations and post-processing only for pixels within the valid area used by the application in each frame. Additionally, it uses dynamic resolution to determine and apply the appropriate resolution for each frame. This approach enhances energy efficiency by avoiding the processing of all sensor data while maintaining or surpassing accuracy levels. Furthermore, These functionalities are designed for hardware-efficient implementation, improving processing speed and minimizing power consumption. The results show a maximum performance of 178 fps and a high energy efficiency of up to 123.15 fps/W. When connected to the hand pose estimation (HPE) accelerator, it demonstrates an average mean squared error (MSE) of 10.03 mm, surpassing the baseline ISP value of 20.25 mm. Therefore, the proposed ISP can be effectively utilized in low-power, small form-factor devices.

Keywords: image signal processor; ToF sensor; dynamic feedback; energy efficiency; 3D depth image; point cloud

1. Introduction

In recent years, various cutting edge technologies and devices based on artificial intelligence (AI) have been developed. This has led to increased attention towards image and video information processing techniques for their utilization. Among these, object recognition and detection are key research topics in the field of computer vision. The primary objective is to accurately identify objects within images and videos [1,2]. In this domain, various types of sensor can be employed, and time-of-flight (ToF) sensors have emerged as one of the most widely used sensor types in recent years [3,4]. ToF sensors are generally more precise in providing accurate 3D geometric data and work well in a variety of ambient lighting situations compared to other methods [5]. These characteristics have raised interest in the field of 3D image technology using 3D ToF sensors, leading to various research efforts [6,7]. Recently, these sensors have been integrated into small devices. They

are being applied in various research areas such as human pose estimation, hand pose estimation (HPE) , and facial recognition. As a result, there is an increasing demand for ToF sensors that prioritize energy efficiency, as well as accuracy and reliability [8–11]. However, 3D ToF sensors generate large amounts of raw data. The hardware required to process these complex 3D images in real time demands significant computational resources and high power consumption. Furthermore, the domains that primarily utilize ToF sensors, such as augmented reality (AR), virtual reality (VR), and various biometric recognition fields, are predominantly deployed on low-power, small form-factor devices, such as mobile and edge devices [12]. Therefore, optimizing resource utilization in 3D image processing is of paramount importance to enhance energy efficiency in these devices.

In previous research, the focus was primarily on improving the analog circuits of 3D ToF sensors, with limited attention given to the digital circuits responsible for post-processing, such as the image signal processor (ISP) [13–15]. However, with the increasing complexity of 3D video processing tasks and the growing volume of 3D data, there has been a heightened interest in efficient data management and post-processing. Generally, processes or algorithms for data processing are complex, making implementation with analog circuits difficult and often inefficient. For example, the process of calculating depth using phase signals from a ToF sensor and converting the results into a point-cloud format requires complex computations. This makes it very difficult and inefficient to implement with analog circuits. Therefore, by implementing these functions through a digital circuit-based ISP and integrating it with the sensor, it is possible to provide various and efficient post-processing techniques and algorithm solutions based on flexible design methods, thereby improving the quality and efficiency of 3D image processing [16–19].

In this paper, we propose an ISP that supports a dynamic feedback configuration to maximize the performance of 3D ToF sensors and enhance energy efficiency. In applications using 3D ToF sensors, it is common to receive various forms of unpredictable 3D images depending on the surrounding environment. In such situations, conventional ISPs can degrade the performance of the application by receiving and processing unnecessary object information from unwanted areas and delivering it to the application. In addition, when using a sensor that operates at a fixed resolution, the ISP must perform numerous calculations (e.g., depth computation and point-cloud conversion) for every pixel in each frame, even when high-resolution images are not required. These situations ultimately increase system latency and power consumption, reducing energy efficiency.

The proposed dynamic feedback ISP has a wide range of practical applications, especially in fields where efficient real-time processing and energy-saving capabilities are crucial. For instance, AR and VR devices, such as smart glasses, can greatly benefit from the ISP's ability to process only the necessary regions with optimal resolution, thereby extending battery life. Additionally, autonomous vehicles and robotics rely on ToF sensors for navigation and object detection. In such scenarios, our ISP's dynamic resolution adjustment ensures that only the relevant areas are processed in high resolution, enhancing reaction speed and reducing latency. Moreover, medical applications, including hand gesture recognition for surgical assistance, demand precise and fast real-time processing. The proposed ISP can optimize performance by dynamically focusing on relevant areas, thereby ensuring low latency and high accuracy. Similarly, security and surveillance systems can utilize this ISP to operate continuously in power-constrained environments. For example, only areas with detected motion can be processed in high resolution, while other regions remain idle, leading to significant energy savings.

These practical use cases demonstrate that the proposed ISP is well-suited for modern applications requiring low power consumption, high performance, and precise processing. The ability to dynamically adjust resolution and extract valid areas in real time not only reduces computational overhead but also ensures that the system can adapt to changing environments, making it highly applicable to a variety of edge devices.

The overall contributions of this paper can be summarized as follows:

(1) We propose a dynamic area extraction technique that continuously extracts the required valid area from sensor images based on real-time feedback from the application. This reduces computation for unnecessary areas, thereby increasing energy efficiency, reducing processing time, and providing more reliable data to the application.
(2) We introduce a dynamic resolution technique that adjusts resolution based on feedback from the application and the previous frame's valid object information, using pixel grouping for logical resolution changes. This efficiently manages processing time, power consumption, and resources according to system operating conditions.
(3) We propose an ISP that efficiently manages resources by using optimized pipeline design and operational control methods, maximizing energy efficiency and minimizing power consumption.

The rest of this paper is organized as follows. In Section 2, we introduce and review research related to energy-efficient ISPs. In Section 3, we provide a detailed explanation of the techniques applied to the dynamic feedback ISP proposed in this paper. Then, in Section 4, we introduce an application system that combines an energy-efficient dynamic feedback image signal processor with a real-time hand pose estimation accelerator. The experimental results of the proposed ISP are discussed in Section 5, and finally, the paper is concluded in Section 6.

2. Related Work

Technologies such as AR, VR, and autonomous driving are rapidly evolving and becoming more prevalent in daily life. As a result, there has been an increase in the use of high-resolution, high-frame-rate images based on 3D depth image sensors. This substantially increases the volume of pixel data that must be processed by sensors and ISPs, resulting in higher power consumption and greater resource use. However, 3D depth image sensors are primarily used in mobile and edge devices, which typically have limited resources. As a result, addressing these challenges is crucial. Consequently, various studies are underway to mitigate these issues and efficiently process image information.

Recently, studies have been published that implement depth information processing capabilities on hardware such as FPGAs to directly interface with 3D depth image sensors in order to improve performance and efficiency. F. Chen [16] notes the increasing use of 3D time-of-flight (ToF) sensors and the corresponding shortage of 3D image signal processors (ISPs) capable of supporting these sensors. And they also introduce the implementation of an energy-efficient real-time ISP for 3D ToF sensors on an FPGA. J. Seiter [20] has also developed a depth calculator system for 3D ToF sensors on FPGA, which utilizes minimal resources and power consumption. N. Druml [21] proposes a platform that processes raw data from 3D ToF sensors to calculate depth using hardware implemented on an FPGA, which is then integrated into an environment that can leverage software for diverse data transformations to improve usability. These prior studies have focused on efficiently implementing features for processing 3D depth images on performance-enhancing hardware.

However, enhancing performance through hardware implementation has its limitations, and eventually, additional functionalities may be necessary for better performance. Among these additional features, research into identifying regions of interest (ROIs) or utilizing them in future frames is actively ongoing. These studies improve energy efficiency and reduce operating time by identifying the ROI and only activating or post-processing the pixels within this area. M. Casares et al. [22] reduced energy consumption and processing time by utilizing tracking information from previous frames. This information was used to identify ROI areas in subsequent frames and process only those areas. Additionally, the camera was switched to an idle state based on the speed of changes or movements in the foreground objects. Iqbal et al. [23] verified the performance of an FPGA-based implementation by adopting an adaptive subsampling algorithm to improve the energy efficiency of embedded vision systems. Although this algorithm is limited to a single ROI,

it predicts the location of the ROI in future frames and turns off the pixels outside this area, enabling vision applications to utilize the ROI programming feature.

In addition, recent developments have introduced multitasking and multimodality techniques into ISPs, making them highly useful for handling various tasks. Yu et al. [24,25] proposed a multitasking model using infrared thermal imaging to simultaneously perform identity recognition and contact time estimation, demonstrating potential applications in crime investigation and security. Nguyen et al. [26] developed a few-shot learning-based multitasking model for detecting and segmenting camouflaged animals, achieving strong performance, even with limited data. Similarly, the ISP we propose performs multitasking by applying both the dynamic area extraction technique and the dynamic resolution technique. This approach enables efficient processing of both tasks simultaneously.

Such studies have explored various additional functionalities to enhance sensor operation efficiency using ROI techniques. However, implementing these additional ROI detection/tracking features requires more resources, which can lead to increased power consumption and operational time. Thus, implementing hardware with minimal overhead for additional performance-enhancing features can offer significant benefits. In particular, if the hardware can dynamically process and provide the optimal image form required by an application in real-time through feedback, it allows for efficient hardware configuration and operation. In addition, since the application already detects or tracks the ROI to perform its task, efficient hardware configuration can be achieved by utilizing this information as feedback without adding additional functionality. This paper proposes an image signal processor (ISP) for 3D depth image sensors that utilizes feedback information in the form of point clouds provided by the application to dynamically adjust the area or resolution of images while minimizing the implementation of additional features. Through this approach, we demonstrate that implementing all functions on an FPGA optimized for hardware operation using minimal resources can reduce operational time and enhance energy efficiency.

3. Method

In this section, we introduce the dynamic area extraction technique and the dynamic resolution technique proposed in this paper. The dynamic area extraction technique identifies the area containing valid objects based on feedback information received from the application. The dynamic resolution technique dynamically adjusts the resolution of the valid area based on feedback information from the application. This approach allows the ISP to process only the information corresponding to the ROI in the entire input image based on the application's feedback and optimally adjust the resolution for that area. Consequently, the ISP can process images efficiently, and the application can receive the necessary information more accurately.

As seen in the example of the conventional ISP in Figure 1, typically, the raw image obtained from the sensor may include objects unrelated to the target object, or the ratio of the valid object in the entire image may be low, complicating the identification of the target object. Therefore, this paper proposes a method to extract valid areas along the X, Y, and Z axes from the raw 3D images of each frame based on feedback received from the application in order to minimize computations on unnecessary areas.

Additionally, with a fixed resolution, accurately identifying valid objects becomes difficult when they are mixed with unnecessary ones, as it is impossible to increase the resolution for more precision. Conversely, when valid objects are already clearly identified and a lower resolution would suffice, a fixed resolution can lead to unnecessarily high resolution and reduced operational efficiency. Therefore, as shown in the example of the dynamic feedback ISP in Figure 1, a method is proposed to dynamically adjust the resolution through pixel grouping based on the application's feedback. This approach ensures optimal image quality suitable for the application's operation in each frame.

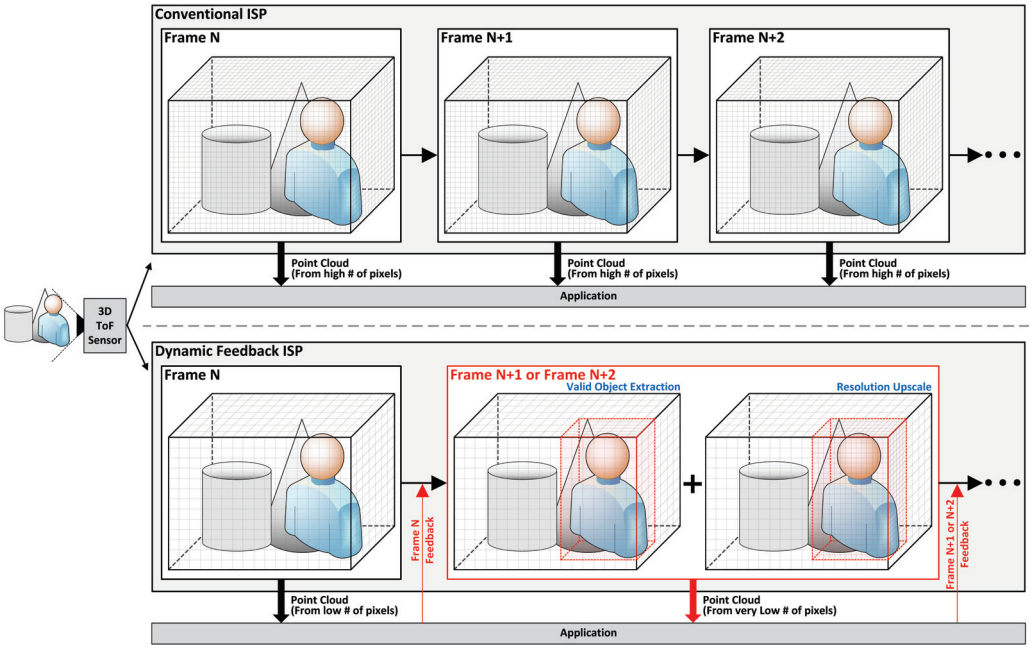
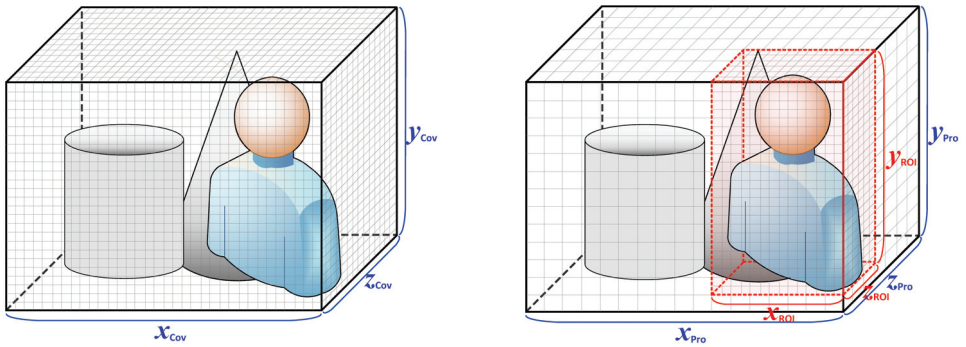


Figure 1. A general overview of the operation comparison between a conventional ISP and the proposed dynamic feedback ISP. The proposed dynamic feedback ISP can reduce the number of pixels used compared to the conventional ISP, resulting in reduced computation and power consumption, thus enhancing energy efficiency. (#: Number).

The effectiveness of the proposed approach can be illustrated with the following example. As shown in Figure 2a, the traditional method processes 1280, 960, and 400 pixels along the x_{Cov}, y_{Cov}, and z_{Cov} axes, respectively, requiring the same amount of pixel data to be processed for every frame. However, as demonstrated in Figure 2b, with the method proposed in this paper, the first frame can be processed with reduced resolution, such as 320, 240, and 100 along the x_{Pro}, y_{Pro}, and z_{Pro} axes, enabling rough processing. For subsequent frames, the application feedback can be used to identify ROI information, and only the ROI can be processed at a higher resolution, such as 640, 480, and 200 along the x_{ROI}, y_{ROI}, and z_{ROI} axes. This approach enables more efficient processing of the ROI compared to the traditional method. This example is provided for illustrative purposes, and the efficiency may vary dynamically depending on different environments and conditions. Equation (1) presents a formula that calculates the efficiency ratio based on the resolution levels used.

$$\frac{x_{Cov}}{x_{\mathrm{ROI}}} \times \frac{y_{Cov}}{y_{\mathrm{ROI}}} \times \frac{z_{Cov}}{z_{\mathrm{ROI}}} = Efficiency \tag{1}$$



(a) (b)

Figure 2. Comparison of pixel usage between the conventional method and the proposed method. (**a**) Example of conventional method. (**b**) Example of proposed method.

Additionally, the proposed method can help improve the performance of applications compared to traditional methods. For the hand pose estimation application presented as an example system in this paper, it can enhance the accuracy of hand pose estimation. This is because, instead of randomly sampling a fixed number of points from the entire image, the proposed ISP extracts points solely from the ROI. As a result, the number of valid points corresponding to meaningful objects increases, which directly improves the accuracy of the application. Equation (4) demonstrates the improvement in accuracy achieved by the proposed method shown in Equation (3) compared to the traditional method described in Equation (2).

$$N_{\text{valid}} = \frac{A_{\text{valid}}}{A_{\text{total}}} \times N_{\text{total}} \tag{2}$$

$$N_{\text{valid}}^{\text{ROI}} = N_{\text{ROI}} \quad (A_{\text{valid}} \subseteq A_{\text{ROI}}) \tag{3}$$

$$\frac{N_{\text{valid}}^{\text{ROI}}}{N_{\text{valid}}} > 1 \tag{4}$$

Notations:
- N_{valid}: The number of points containing valid objects
- A_{valid}: The area containing valid objects
- A_{total}: The total area of the image
- N_{total}: The number of points sampled from the entire image
- $N_{\text{valid}}^{\text{ROI}}$: The number of valid points within the ROI
- N_{ROI}: The number of points sampled from the ROI

In the proposed method, the accuracy of the application is improved by focusing on extracting valid points from the ROI instead of sampling from the entire image. In conventional approaches, a fixed number of points (N_{total}) are sampled from the entire image, which often includes a large portion of irrelevant or background areas. As a result, the number of valid points corresponding to meaningful objects (N_{valid}) is limited by the ratio of the valid area to the total image area, $\frac{A_{\text{valid}}}{A_{\text{total}}}$. However, in the proposed ISP, the sampling is restricted to the ROI, ensuring that the points correspond more accurately to the objects of interest. Since the ROI is selected to encompass the valid area ($A_{\text{valid}} \subseteq A_{\text{ROI}}$), the number of valid points in the ROI sampling ($N_{\text{valid}}^{\text{ROI}}$) increases compared to the conventional approach.

This increase in valid points enhances the accuracy of the application, as it allows the system to focus its computational resources on the meaningful portions of the data, rather than on irrelevant areas. The effectiveness of the proposed method can be quantified with Equation (4), indicating that the ROI-based sampling results in a higher proportion of relevant points, thereby improving the overall accuracy and performance of the application.

Through these proposed methods, the application can obtain more suitable preprocessed 3D images rather than using raw 3D images, thereby reducing unnecessary tasks and computations, decreasing power consumption, and enabling more efficient operation. Figure 3 presents the overall operation flow of the proposed feedback operation in a flowchart.

3.1. Dynamic Area Extraction Technique

3D images from sensors usually contain unprocessed raw data, which often lack suitability for direct use by applications. For example, the raw 3D image received from the sensor may include background or surrounding elements that are not of interest to the application, which could interfere with the functional operations, such as inference performed by the application, potentially leading to accuracy losses and ultimately affecting the overall performance. Additionally, the additional computations required for the unnecessary parts can cause increased operation time and power consumption.

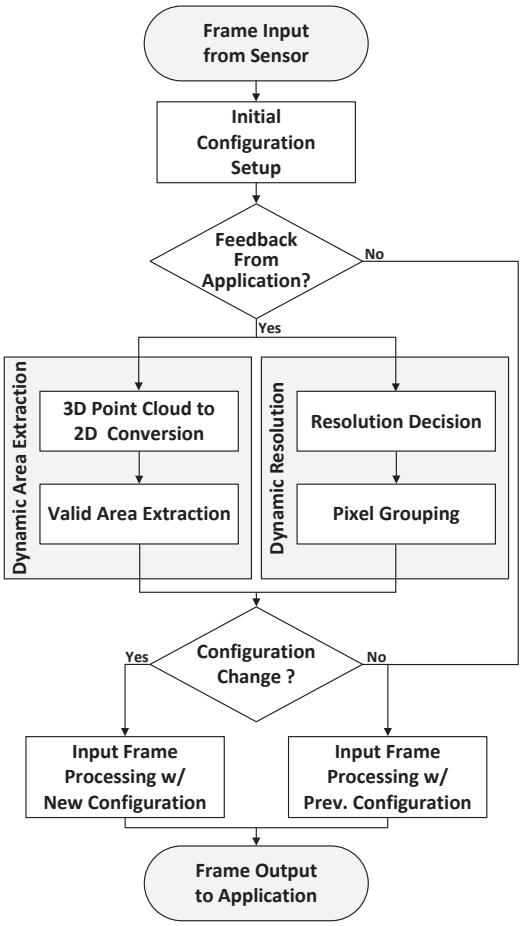


Figure 3. Overall flow chart for dynamic feedback operation.

Therefore, we propose a dynamic area extraction technique to identify core areas required by the application and make them available for utilization within the application. Figure 4a shows an example of applying valid area extraction to a 3D image. Applying this method to ISPs allows the processing of raw data only for the areas used in the actual application. The ISP receives feedback in the form of point-cloud information representing the actual and valid areas used by the application. Based on this feedback, only the actual valid areas from the entire data of the next frame input from the ToF sensor is computed and processed, and then passed to the application. The processing of this feedback data is carried out in two stages, as described below. The first method is the 3D point cloud to 2D conversion method. It converts 3D point-cloud feedback into 2D information, mapping it to the sensor's input image to identify the valid area used by the application. The second method is the valid area extraction method. This method uses the valid area information obtained from the first method to extract and process only the data in that area. Algorithm 1 explains the detailed operation algorithm of the dynamic area extraction technique.

3.1.1. 3D Point Cloud to 2D Conversion

Our work utilizes an application that provides feedback on areas of interest in the form of a 3D point cloud. This feedback is generated based on the application's own task-specific criteria and depth information from the previous frame produced by the ToF sensor. We then interpret and map these areas as 'valid' regions using our proposed method for further processing. However, the majority of sensors output raw images in RGB-D format, rather than providing actual 3D coordinates. Therefore, it is essential to process these 2D raw images to accurately identify and localize valid objects. To achieve this, we must first reverse transform the 3D point-cloud data of the valid areas, received as feedback from the application, into 2D pixel information. This information is then mapped onto the

raw 2D image and incorporated into the subsequent input frame. To generate bounding boxes through the processing of these subsequent input frames, we have employed several algorithms used for coordinate transformations, which are explained below.

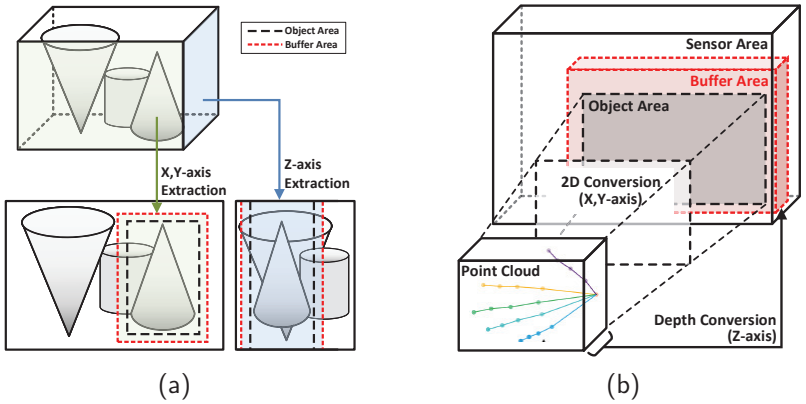


Figure 4. Dynamic area extraction technique. (**a**) Example of valid area extraction. (**b**) Valid area extraction mechanism.

Algorithm 1 Dynamic Area Extraction Technique

```
Initialize: curr_area = init_value

while (Dynamic_Area_Extraction_En == 1) do
    if (user_reset == 1) then
        next_area = init_value
    else if (b_box <= curr_area) then
        next_area = b_box
        up_res_flag = 1
    else if (b_box > curr_area) then
        if ((b_box - curr_area) > limit) then
            next_area = b_box
            down_res_flag = 1
        else
            next_area = b_box
            up_res_flag = 1
        end if
    else
        next_area = curr_area
    end if
    curr_area = next_area
end while
```

Abbreviations:
- `init_value`: Initial area configured by user
- `curr_area`: Current area
- `b_box`: Bounding box area
- `limit`: Threshold value for area change
- `down_res_flag`: Flag to decrease resolution
- `up_res_flag`: Flag to increase resolution

To convert a 3D point cloud into 2D pixel coordinates, it is essential to understand the transformation of depth data from a 2D depth image into a 3D point cloud, as shown in Algorithm 2 [27]. Thisalgorithm leverages transformation processes to map depth data to spatial coordinates. The algorithm incorporates critical camera parameters, such as the principal point coordinates (c_x, c_y), focal lengths along the x and y axes (f_x, f_y), and the

dimensions of the depth image (*height* and *width*). These parameters facilitate the geometric transformations required to convert pixel-based depth values into their corresponding 3D coordinates. The algorithm iterates through each pixel in the depth image, extracts the depth value, and computes the 3D coordinates (x, y, z) by adjusting the pixel positions based on the principal point and scaling by the focal lengths.

Algorithm 2 Depth to point clouds

Require: Depth_image, $c_x, c_y, f_x, f_y, height, width$
Ensure: x, y, z
1: **for** i in range *height* **do**
2: **for** j in range *width* **do**
3: $z = \text{depth\_image}[i][j]$
4: $x = (j - c_x) * z / f_x$
5: $y = (i - c_y) * z / f_y$
6: **end for**
7: **end for**

We used Algorithm 2 to derive Algorithm 3, which projects 3D coordinates back onto the 2D image plane. Using projection equations, it calculates the 2D pixel coordinates (px_x, px_y) by scaling the 3D coordinates with the focal lengths and adjusting for the principal point. This process effectively translates the 3D spatial location back to its corresponding position in the 2D image. Through this algorithm, we obtain the coordinates of pixels used in the valid area extractor via pipelining.

Algorithm 3 Point clouds to 2D pixel coordinates

Require: $x, y, z, c_x, c_y, f_x, f_y$
Ensure: px_x, px_y
1: $px_x = (x * f_x / z) + c_x$
2: $px_y = (y * f_y / z) + c_y$

3.1.2. Valid Area Extraction

The valid area extraction method defines bounding boxes for extracting valid areas in order to accurately determine the area where objects exist in 3D images. Figure 4b illustrates how the valid area extraction method works. It uses the 2D positions of valid objects from the previous frame, obtained through the previously described process of converting point clouds into 2D, to determine the valid pixel areas in the X and Y directions in the 3D image of the next frame. Simultaneously, it determines the valid range in the Z direction based on the depth information of valid objects from the previous frame. At this time, to account for potential changes in object size, movement, and environmental conditions between frames, users are allowed to define the valid object area by adding a certain margin level around the area received as feedback from the application in each direction.

The prevalent approach to accomplish the valid area extraction is to project the 3D point-cloud representations onto the 2D image plane, and subsequently delineate the boundaries by performing coordinate-wise comparisons between the projected 2D pixel coordinates. However, the method of converting all 3D point clouds into pixels involves numerous calculations and comparisons, which can result in latency and increased power consumption. Therefore, to address the computational complexity and inefficiency associated with this conversion process, we propose a simple approximation method based on the maximum and minimum values of the coordinates x and y of the point cloud. To begin the method, the minimum and maximum values of both x and y coordinates within the point cloud must be determined. This process yields four ordered pairs: $\{Min(x), Z_{min(x)}\}$, $\{Max(x), Z_{max(x)}\}$, $\{Min(y), Z_{min(y)}\}$, and $\{Max(y), Z_{max(y)}\}$. Here, $Min(x)$ and $Max(x)$ represent the minimum and maximum values of the x coordinates in the point cloud, respectively, with $Z_{min(x)}$ and $Z_{max(x)}$ being

the corresponding z coordinate values at these x coordinates. The y coordinates follow similarly, with $Min(y)$ and $Max(y)$ denoting the minimum and maximum values, and $Z_{min(y)}$ and $Z_{max(y)}$ as the corresponding z coordinate values.

Figure 5 is an example of projecting a 3D point-cloud area onto a sensor-based 2D plane. In the figure, the blue lines represent the 3D points, and their projections on the sensor plane form the bounding box indicated by red dots. This bounding box accurately captures the spatial extent of the point cloud in the 2D image, compensating for the depth-based discrepancies in the pixel coordinates.

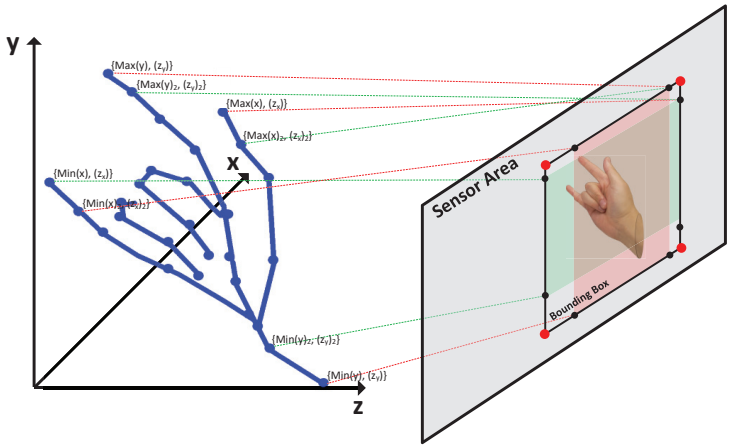


Figure 5. Example of bounding box decision.

However, considering Algorithm 3, which calculates pixel coordinates based on depth, it becomes evident that the maximum and minimum values of the point cloud may not align with those of the pixel coordinates. For instance, consider two points, A(3,y_A,4) and B(2,y_B,2), within the point cloud, where the camera parameters are $f_x = 2$ and $c_x = 4$. When applying Algorithm 3, we obtain pixel coordinates $px_x(A) = 5$ and $px_x(B) = 6$. Despite $x_A > x_B$ in the point cloud, the pixel coordinate $px_x(A) < px_x(B)$. To address this discrepancy, we extend our consideration to the second-largest and second-smallest alignment pairs. This results in eight ordered pairs: $\{Min(x), Z_{min(x)}\}$, $\{Max(x), Z_{max(x)}\}$, $\{Min(y), Z_{min(y)}\}$, $\{Max(y), Z_{max(y)}\}$, and their respective second-largest and second-smallest pairs. This allows us to define a reasonably accurate bounding box with just eight transformation and comparison operations, resulting in advantages in terms of latency and power consumption.

3.2. Dynamic Resolution Technique

The resolution of images captured through the depth sensor is determined by the number of pixels supported by the sensor. By utilizing all available pixels from the sensor, we can obtain more precise depth image data by using the maximum resolution supported by the sensor. However, this approach increases the amount of raw data that the ISP needs to process, resulting in higher computational demands and higher power consumption. Therefore, this paper proposes a dynamic resolution technique that allows for logical resolution control depending on the situation.

The dynamic resolution technique uses a pixel grouping method that consolidates multiple pixels into a single pixel or utilizes each pixel individually. This method adjusts the resolution of the next frame higher or lower based on the application's feedback information for the current frame and the results of the dynamic area extraction technique, supporting the efficient operation of the application. Algorithm 4 explains the detailed operation algorithm of the dynamic resolution technique.

Algorithm 4 Dynamic Resolution Technique

```
Initialize: c_p_group = init_value

while (Dynamic_Resolution_En == 1) do
    if (user_reset == 1) then
        n_p_group = init_value
    else if (up_res_flag == 1) then
        if (c_p_group == 1) then
            n_p_group = c_p_group
        else
            n_p_group = c_p_group - user_def
        end if
    else if (down_res_flag == 1) then
        if (c_p_group == 6) then
            n_p_group = c_p_group
        else
            n_p_group = c_p_group + user_def
        end if
    else
        n_p_group = c_p_group
    end if
    c_p_group = n_p_group
end while
```

Abbreviations:
-`init_value`: Initial pixel group configured by user
-`c_p_group`: Current pixel group
-`n_p_group`: Next pixel group
-`user_def`: User define value for pixel group change
-`down_res_flag`: Flag to decrease resolution
-`up_res_flag`: Flag to increase resolution

3.2.1. Resolution Decision

The resolution decision method derives the resolution to be applied to the next frame based on the feedback information from the application. The ISP detects areas containing valid objects in each frame received from the sensor and adjusts the resolution based on the application's feedback. Figure 6 illustrates an example of the resolution decision.

At the initial operation, the ISP starts with a low resolution, delivering rough, low-resolution overall image information to the application. Based on this, the application performs its initial operations and generates information about the actual necessary valid area within the overall image, providing this feedback to the ISP. When processing the next frame, the ISP increases the resolution only for the extracted valid area based on this feedback, allowing the application to receive more precise valid area images. By increasing the resolution only for the valid area instead of the entire image, unnecessary computations are reduced, while the application can generate the needed images more accurately, thus processing images much more efficiently.

Conversely, if the valid area information is reset by the user's settings or if the position or size of the valid area changes beyond the user-defined threshold, the system will lower the resolution to the initial state. This indicates that the previously set valid area has significantly changed, suggesting a substantial alteration in the target or surrounding environment. Consequently, the system repeats the aforementioned process to accurately identify the valid area and adjust the resolution accordingly. Through this mechanism, the ISP operates at the optimal resolution depending on the situation, maximizing efficiency.

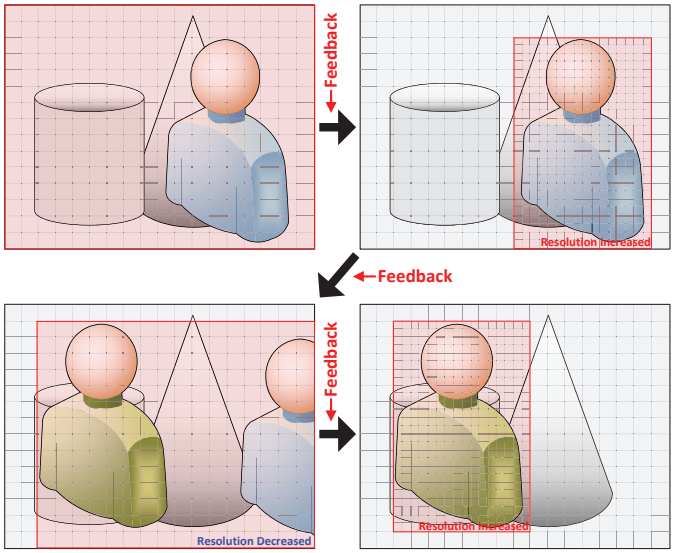


Figure 6. Example of resolution decision. Red box: valid area.

3.2.2. Pixel Grouping

As the resolution of the input image increases, the amount of computation required to process that image also increases. If the resolution of the image could be adjusted flexibly according to the situation, the amount of data to be processed would change accordingly, enabling efficient operation. However, if the sensor itself does not have real-time resolution control functionality, changing the resolution in real time is not easy. Therefore, a method to adjust the image resolution in real time without modifying the existing sensor would be very useful. Therefore, we propose a pixel grouping method that logically groups a fixed number of pixels supported by the sensor to effectively adjust the resolution of the entire image or specific areas. This grouping method provides the flexibility to adjust the resolution to an appropriate level depending on the content of the images received by the sensor.

Resolution adjustment is implemented through the pixel grouping method based on the results of the resolution decision method described above. Table 1 shows the example of pixel grouping approach integrated into the ISP proposed in this paper. As seen in Algorithm 4, based on the feedback information and image characteristics, one of the four pixel grouping methods (representative pixel data, maximum pixel data, minimum pixel data, or average pixel data) is selected, and the system adjusts the settings in the direction of the selected pixel group using the user_def value. The ISP then logically groups the pixels according to this setting or the user's choice to adjust the resolution. For example, when grouping four pixels in a 2 × 2 configuration from a ToF sensor that supports a maximum resolution of 1280 × 960, selecting one representative pixel can reduce the overall image resolution to a maximum of 640 × 480. At this point, the number of pixels selected in each group is fixed, but the specific positions of the selected pixels are determined based on the user's settings. This approach significantly reduces the pixel information that the ISP needs to process, lowering the computational load, reducing power consumption, and conserving resources while greatly improving the processing speed.

Table 1. Example of pixel grouping based on representative pixel selection.

	Grouping	Valid Pixel	Resolution	Computational Ratio
1	Base Line	■	1280 × 960	100%
2	4 Pixels Grouping		1280 × 480	50%
3			640 × 960	50%
4			640 × 480	25%
5	16 Pixels Grouping		320 × 480	12.5%
6			320 × 240	6.25%

4. Proposed System Architecture

In this section, we propose an example hardware architecture of an application system based on the dynamic feedback ISP. The proposed ISP is designed to operate at optimal performance when utilizing feedback information from the application. Therefore, we have developed a system that uses the 3D ISP for a hand pose estimation accelerator application. Figure 7 illustrates the block diagram of the entire example system. The dynamic area controller extracts the valid areas in each frame based on feedback from the application. Then, the dynamic pixel controller and dynamic depth controller extract pixel data from these valid areas for use in the application. Additionally, the dynamic resolution controller analyzes the quality level of the images needed for each frame, and based on this analysis, the dynamic pixel controller can dynamically adjusts the resolution for the next frame by controlling the pixel information. In other words, all feedback information is comprehensively analyzed and evaluated by the feedback controller, then applied to the next frame by the main controller for use in the application.

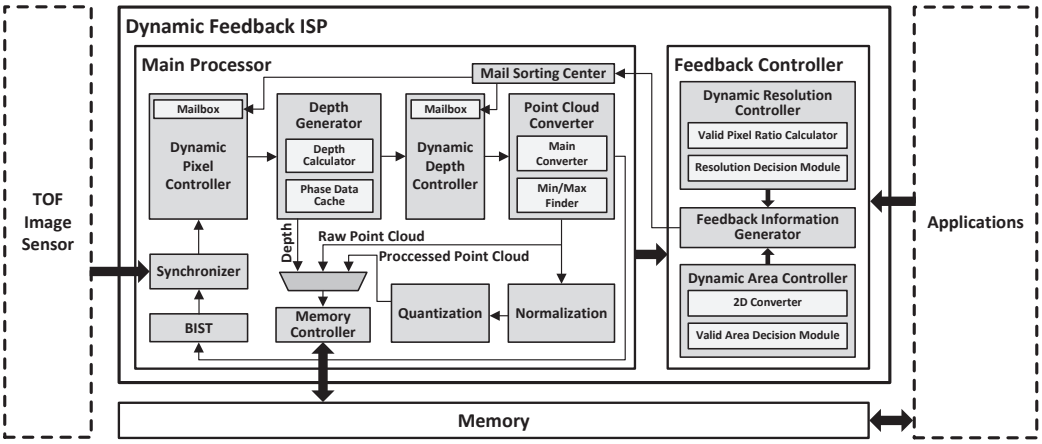


Figure 7. Overall block diagram of proposed example hardware based on a dynamic feedback ISP.

4.1. Main Processor

The main processor plays a crucial role in processing the raw data received from the ToF sensor into a suitable format for use in applications. It consists of a dynamic pixel controller that processes pixel data received from the sensor based on feedback information, a depth generator for computing depth information, a dynamic depth controller for selective depth direction filtering based on feedback information, a point-cloud converter for transforming depth information into point-cloud format, and a quantization and normalization module for post-processing point-cloud data.

The dynamic pixel controller can finely control raw pixel data from ToF sensors to extract data from valid areas or adjust the resolution as needed. Figure 8a illustrates the details of the dynamic pixel controller, and Table 2 provides a summary of the functions. Based on the information received from the feedback controller, row and column filters are

used to selectively pass only the information of pixels within the valid area of the incoming frame to the next stage. The selected pixels are grouped by the grouping controller based on the resolution information obtained from the feedback. Depending on the configuration, the minimum, maximum, or average values of the grouped pixels are computed, followed by precision adjustment before output. All these operations are designed to be processed serially and continuously as soon as they are inputted from the sensor, ensuring that there is no delay in real-time operation.

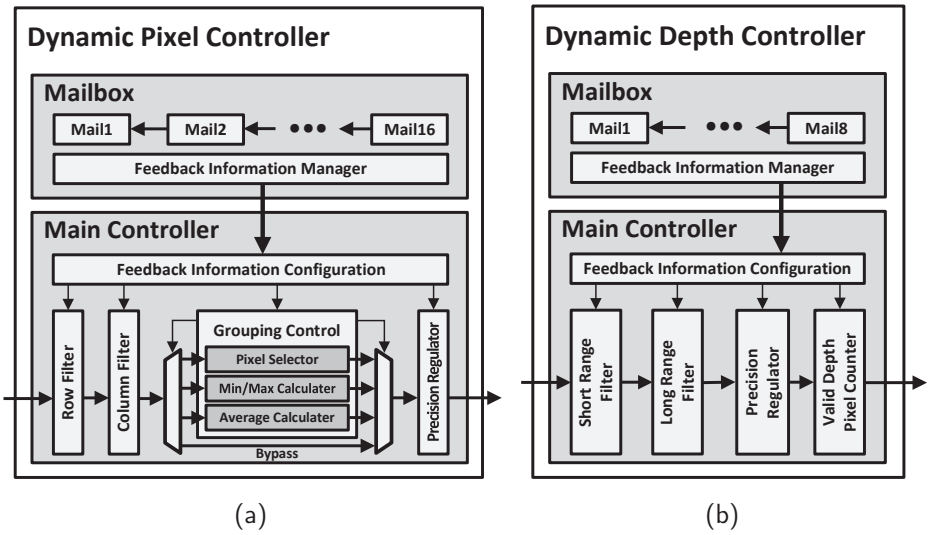


Figure 8. Hardware details in main processor. (**a**) Dynamic pixel controller. (**b**) Dynamic depth controller.

Table 2. Summary of the functions of the modules within the dynamic pixel controller and dynamic depth controller.

Module	Function	Functionality
Common	Mailbox	Manage and schedule the feedback received from the application.
Dynamic pixel controller	Row filter	Filter pixels in the sensor's row direction based on feedback.
	Column filter	Filter pixels in the sensor's column direction based on feedback.
	Grouping control	Group pixels and perform calculations such as min/max/average on the selected pixels.
	Precision regulator	Adjust the precision of the final calculation results to be usable by the ISP.
Dynamic depth controller	Short-range filter	Filter pixels that fall outside the valid depth range in the direction close to the sensor.
	Long-range filter	Filter pixels that fall outside the valid depth range in the direction far from the sensor.
	Precision regulator	Adjust the depth precision of the selected pixels to be usable by the ISP.
	Valid depth pixel counter	Count the number of pixels that remain after the filtering process.

The dynamic depth controller is responsible for extracting valid areas in the depth direction. Figure 8b illustrates the details of the dynamic depth controller, and Table 2 provides a summary of the functions. First, the pixel information transmitted from the previous step is selectively filtered using short-range and long-range filters to retain only those pixels that fall within the valid depth range. Through this process, information such as out-of-focus sensor data or unnecessary background can be effectively filtered. These filtered pixels are then transformed by the point-cloud converter and transmitted to the application. This allows the application to perform computations only on the pixels that belong to the required areas, thereby enhancing the efficiency of the hardware operation. Simultaneously, the valid depth pixel counter measures the number of pixels within the valid depth range, providing auxiliary information for resolution determination to the dynamic resolution controller.

The feedback information received from the application may arrive at irregular intervals, depending on the type of application and its operating environment. Therefore, the mailboxes within the dynamic pixel controller and dynamic depth controller are designed to continuously receive and manage feedback delivered irregularly from the application via the feedback controller. Additionally, they verify the validity of the feedback information and control its sequential processing through scheduling.

4.2. Feedback Controller

The feedback controller receives feedback information from the application, processes it into a suitable format for feedback operation, and generates the configuration information that is applied for the next frame. The feedback controller consists of a dynamic area controller, a dynamic resolution controller, and a feedback information generator.

The dynamic area controller generates information for valid area extraction based on feedback information. This enables the main processor to extract core areas more accurately for delivery to the application. The 2D converter translates the 3D valid area information received as feedback from the application into 2D information. The valid area decision module utilizes this converted information to match it with the sensor's pixel areas and generate information about the valid areas used by the application from the input data received from the sensor. The size, shape, or position of objects in the valid area may vary across input frames depending on the environmental conditions. Taking these factors into account, the system determines the expanded final valid area by additionally reflecting the margin information input by the user.

The dynamic resolution controller generates information to dynamically adjust the resolution of each frame based on the quality of the 3D images obtained from the ToF sensor and feedback information from the application. The valid pixel ratio calculator calculates the ratio of valid depth pixels computed by the dynamic depth calculator and forwards this ratio to the resolution decision module.

The resolution decision module evaluates the quality of the 3D image based on the depth information analyzed by the valid pixel ratio calculator using the user-input valid pixel threshold and determines the resolution for the next frame.

The feedback information generated by the dynamic area controller and dynamic resolution controller is consolidated through the feedback information generator. This integrated feedback information is transmitted to the main processor, where it is utilized in processing the pixels received from the ToF sensor.

4.3. Pipelined Architecture

The proposed ISP employs pipeline architecture throughout its operation to enhance hardware efficiency. We modularized the functions required for ISP operation to align with the pipeline structure, then arranged these modules to optimize performance in hardware. Figure 9 illustrates the operation of the proposed ISP when processing a 640 × 480 resolution image, comparing the sequential architecture with the pipeline architecture. In the sequential structure, each function is executed step-by-step, resulting in increased processing time proportional to the number of frames. In contrast, the pipeline architecture is structured to maximize module utilization, making its benefits more pronounced as the number of frames increases. This approach achieves approximately 2.48× higher performance when processing the same number of frames compared to the sequential structure. Additionally, each module is designed to operate only when required, with power shut off during idle states, optimizing power consumption by carefully managing the timing of each pipeline stage.

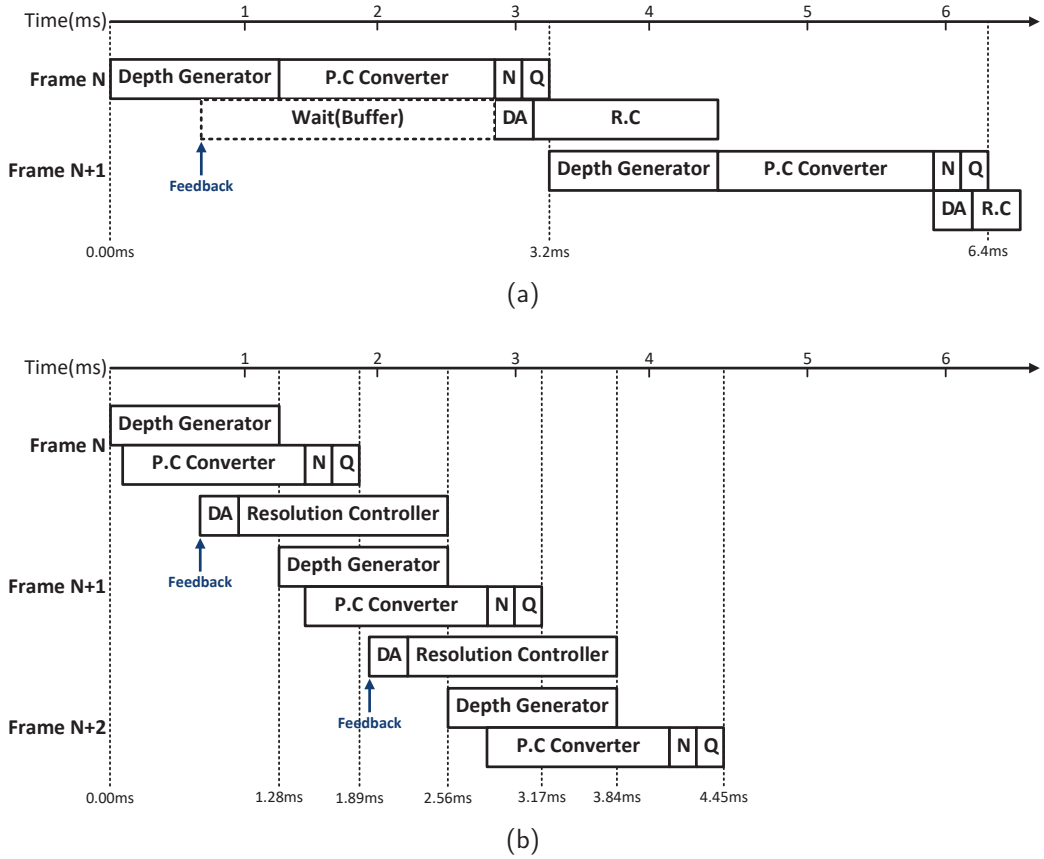


Figure 9. Comparison of sequential and pipelined operation flow. (**a**) Example of sequential operation flow. (**b**) Example of pipelined operation flow. P.C: Point Cloud, N: Normalization, Q: Quantization, DA: Dynamic area controller, R.C: Resolution controller.

5. Experimental Results

5.1. Implementation Results

The proposed ISP with dynamic feedback was implemented on an FPGA to assess its performance and efficiency. The baseline ISP processes all the pixel data from the 3D ToF sensor for each frame according to the conventional method and delivers them to the application. On the other hand, the proposed ISP optimizes and delivers each frame's data to the application using the dynamic feedback technique.

The proposed ISP was synthesized on the Xilinx Kintex UltraScale KU115 FPGA evaluation platform using Xilinx Vivado 2022.2, with a clock frequency of 100 MHz. Table 3 shows the resource utilization and performance of the baseline ISP and the proposed ISP with the applied technique. The proposed ISP, compared to the baseline ISP, showed an increase of approximately 39% in LUT usage, approximately 34% in FF usage, and an additional 2 DSPs. The increase in resources was a result of adding a feedback controller, dynamic pixel controller, and dynamic depth controller to the proposed ISP. Despite this increase, the efficiency of the proposed ISP improved by approximately 6.4×. Additionally, power consumption was reduced by 10%, indicating a significant performance enhancement compared to the additional resources used.

Table 3. Resource utilization and performance comparison with baseline ISP.

	Implementation				Performance				
	LUT	**FF**	**DSP**	**BRAM**	**Resolution**	**Frame Rate (fps)**	**Power (W)**	**Efficiency (fps/W)**	**Error** [a]
Baseline	17,736	23,036	0	4	1280 × 960 (fixed)	31 (@100 Mhz)	1.61	19.25	20.25 mm
Proposed	24,657	30,975	2	4	1280 × 960 (variable) [b]	178 (@100 Mhz)	1.45	123.15	10.03 mm

[a] This is measured using the HandFoldingNet model application with NYU datasets. [b] The resolution changes dynamically, as proposed in this paper.

5.2. Performance Evaluation

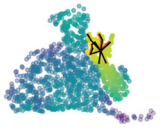
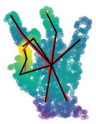
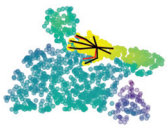
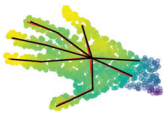
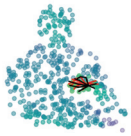
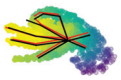
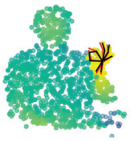
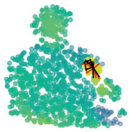
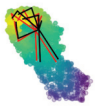
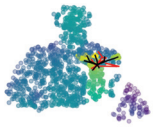
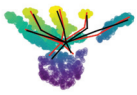
5.2.1. Performance Evaluation of Hand Pose Estimation

A sample system, using a hand pose estimation application, was set up to evaluate the proposed ISP's performance. For this purpose, we implemented a hand pose estimation accelerator based on the recently introduced HandFoldingNet [28] model to construct the example system. This setup allowed us to assess the performance of the proposed ISP and analyze its impact on the performance improvement of the application using it. The proposed sample system operates by post-processing the 3D image data input from the ToF sensor in the ISP. This processed data is then used by the accelerator to estimate hand poses. However, since 3D images directly captured using the ToF sensor do not provide ground truth for evaluating the results of hand pose estimation, quantitative analysis and evaluation are not feasible.

Therefore, we used the NYU dataset [29] to quantitatively analyze and evaluate the impact of the functional differences between the proposed ISP with dynamic feedback features and the baseline ISP on the final performance of the hand pose estimation accelerator. The NYU dataset includes hand pose information for evaluating the performance of various models and algorithms. It uses continuous video-format images to observe feedback changes frame-by-frame. It also contains other surrounding backgrounds along with hand poses, making it suitable for evaluating the performance of the proposed ISP. However, since the NYU dataset is provided in RGBD format, it cannot be directly used with an ISP that uses phase information as input. Therefore, we directly input depth information into the ISP's point-cloud converter to analyze and evaluate the performance of the hand pose estimation accelerator. Additionally, since the NYU dataset shows little variation in hand shape, movement, and environmental changes, the margin applied to the valid area was set to the system default of 10%, and the resolution decision threshold was set to the system default of 85%. The proposed example system is implemented on an FPGA to perform hand pose estimation.

Table 4 shows the results of performing the hand pose estimation accelerator based on the baseline ISP and the proposed ISP using the original images from the NYU dataset. Essentially, all original images contain not only the hand but also people and background objects. When using the baseline ISP for such images, as seen in the results, all information in the image, including the background, is transmitted to the hand pose estimation accelerator. Additionally, because the area occupied by the hand is very small due to the unnecessary background, the accelerator has to distinguish this and perform hand pose estimation, resulting in higher errors and lower accuracy. As seen in Table 4, the unnecessary background information is removed by feedback and only actual hand information is transmitted in the proposed ISP. Therefore, the hand pose estimation errors are decreased, and more accurate results are achieved. Consequently, the baseline ISP showed an average distance error of 20.25 mm, while the proposed ISP showed an average distance error of 10.03 mm, indicating a reduction of more than 50%.

Table 4. Hand pose estimation results of baseline and proposed system based on HandFoldingNet model on NYU dataset.

Original	Baseline	Proposed	Original	Baseline	Proposed
					
	Error = 7.09 mm	Error = 4.39 mm		Error = 8.89 mm	Error = 7.44 mm
					
	Error = 39.07 mm	Error = 12.3 mm		Error = 14.15 mm	Error = 7.86 mm
					
	Error = 22.16 mm	Error = 10.06 mm		Error = 22.10 mm	Error = 4.39 mm

5.2.2. Hardware Performance Evaluation

The proposed ISP demonstrates superior processing speed and lower power consumption compared to the baseline ISP, resulting in higher energy efficiency. As shown in Table 3, the baseline ISP supports a maximum resolution of 1280 × 960 but operates at a fixed resolution at all times, whereas the proposed ISP enables dynamic resolution adjustment, ranging from 1280 × 960 to 320 × 240. Therefore, while the baseline ISP achieves a maximum frame rate of 31 fps at a clock frequency of 100 MHz, the proposed ISP achieves a faster frame rate of approximately 178 fps as the resolution and image size dynamically change depending on the situation. Additionally, when performing the hand pose estimation application based on the NYU dataset used in this paper, it achieved a power consumption of 1.45 W and a high energy efficiency of 123.15 fps/W. Furthermore, the proposed ISP arbitrarily selects only 1024 pixels to convert to a point cloud and delivers this to the application to reduce the computational load required to convert all incoming pixels to a point cloud. This reduction is possible because the proposed ISP operates with the valid area and optimal resolution through feedback.

We also conducted a comparative analysis with previous research. It was challenging to find directly related studies controlling ISP operation based on feedback from 3D ToF sensors. However, we have compiled similar related research in Table 5. M. Yoshimura et al. [19] proposed a dynamic ISP similar to our study, where the parameters for each frame can be controlled based on the recognition results of the previous frame. They experimented using a GPU with software implementation using RGB sensor input data. They evaluated five parameters: auto gain (AG), denoiser (DN), sharpener (SN), gamma tone mapping (GM), and contrast stretcher (CS). Their results indicate that their proposed approach outperforms the others. However, these functions are all implemented in software and run on a GPU, which significantly limits their practicality and energy efficiency when used as an ISP that needs to operate closely with the sensor. F. Chen et al. [16], Z. Chen et al. [30], and I. Gyongy et al. [31] proposed ISPs based on 3D ToF sensors with additional features and implemented them in hardware. In summary, these studies focused solely on implementing ISP functions and improving their performance. They did not propose any special features, such as utilizing feedback, as we have in our study. Compared to these approaches, our proposed method not only provides fundamental ISP functions but also incorporates application feedback. This addition leads to lower power consumption, superior performance,

and higher energy efficiency, demonstrating the advantages of our approach. Furthermore, this allows efficient hardware configuration and operation without compromising the performance of the application.

Table 5. Performance comparison with related work.

Work	Sensor Type	Implementation	Performance	Major Components
[19]	RGB	Software (GPU:RTX2080)	Latency: 27.7 ms, 36 fps, 49.6% (AP@0.5:0.95) [a]	Latent update style ISP controller, End-to-end optimization method
[30]	3D ToF	FPGA	256 × 256, 90 fps (50 Mhz), Error: 1 mm	Depth calculation(CORDIC algorithm), Gaussian filter
[16]	3D ToF	FPGA (XCZU9EG) 21,002 LUTs, 20,032 FFs, 39 BRAMs , 19 DSPs	1280 × 960, 60 fps (100 Mhz), 2.1W, Precision: 0.09 mm	Depth and intensity caculation, Depth calibration, Multi-frequency fusion, Point cloud conversion
[31]	3D ToF	ASIC (40 nm)	64 × 32, 50 fps (100 Mhz), 70 mW, Accuracy: 5 cm	In-pixel peak detection, On-chip depth computation, Pixel with embedded histogramming
This Work	3D ToF	FPGA (KU115) 24,657 LUTs, 30,975 FFs, 4 BRAMs , 2 DSPs	1280 × 960, 178 fps (100 Mhz), 1.45 W, MSE [b] (HPE) [c]: 10.03 mm	Dynamic feedback controller (Dynamic area control, Dynamic resolution control), Point-cloud conversion

[a] Average Precision, [b] Mean squared error, [c] Hand Pose Estimation

6. Conclusions

In this paper, we proposed an ISP specific to 3D ToF sensors that can optimize their functionality by dynamically changing settings based on feedback from the application. The proposed ISP is fully compatible with applications that can provide point-cloud feedback. It has been validated for hardware implementation and performance on the Xilinx KU115 FPGA evaluation board. The first technique that we introduce is the dynamic area extraction technique, which uses feedback information from the application. It calculates and post-processes only the valid areas necessary for the application to operate for each frame of input from the sensor. This approach reduces unnecessary computations while utilizing valid area information, thereby maintaining accuracy and increasing operational efficiency. The second proposed technique is the dynamic resolution technique, which determines the resolution based on changes in the valid area due to the dynamic area extraction technique or feedback from the application. It evaluates valid pixels by considering feedback and valid area information and then adjusts the resolution based on those results. These two techniques work together in a complementary manner. They allow the application to operate efficiently by improving performance, preserving accuracy, reducing power consumption, and decreasing operation time. Furthermore, when implemented in the FPGA, we apply the pipeline architecture and a low-power scheme to achieve even more efficient operation. In the results, we confirm that the proposed method supports achieving an accuracy equal to or greater than that of the conventional approach for the application. Additionally, we observed outstanding energy efficiency, reaching 123.15 fps/W.

In conclusion, the proposed dynamic feedback ISP not only improves energy efficiency and processing speed but also shows great potential for real-world applications such as AR/VR devices, autonomous vehicles, medical systems, and security solutions. By enabling efficient operation with minimal resources, the ISP is well-suited for use in edge devices that require both high performance and low power consumption.

Nonetheless, the proposed ISP is optimized for ToF sensors and is currently designed to handle a single ROI using only point-cloud feedback, which presents certain limitations. In future research, we plan to enhance and extend the feedback functionality to process multiple ROIs simultaneously and to accommodate various types of feedback information. Additionally, we aim to expand the ISP's capabilities to support a wider range of sensors, improve compatibility, enhance performance, and optimize power consumption.

Author Contributions: Conceptualization, Y.K. and W.C.; methodology, Y.K.; validation, J.S., C.H., and W.C.; formal analysis, Y.K., J.S. and C.H.; investigation, Y.K. and J.S.; writing—original draft preparation, Y.K.; writing—review and editing, Y.K., J.S., C.H., W.C. and J.H.K.; visualization, C.H.; supervision, J.H.K.; project administration, J.H.K.; funding acquisition, J.H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (No. 2022R1A4A3032913, No. 2020M3H2A1076786, No. RS-2024-00345732); the Ministry of Science and ICT (MSIT), Korea, under the ICT Creative Consilience program (IITP-2023-2020-0-01821), the ITRC (Information Technology Research Center) support program (IITP-2021-0-02052), and the Artificial Intelligence Innovation Hub (RS-2021-II212068) supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP); the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190421, AI Graduate School Support Program (Sungkyunkwan University)); and the Technology Innovation Program (Industrial Strategic Technology Development Program) (RS-2023-00235718, 23040-15FC) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea) (1415187505).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Acknowledgments: The EDA tool was supported by the IC Design Education Center(IDEC), Korea.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kim, J.; Ro, Y. Attentive Layer Separation for Object Classification and Object Localization in Object Detection. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019.
2. Tao, Y.; Zongyang, Z.; Jun, Z.; Xinghua, C.; Fuqiang, Z. Low-altitude small-sized object detection using lightweight feature-enhanced convolutional neural network. *J. Syst. Eng. Electron.* **2021**, *32*, 841–853. [CrossRef]
3. Bergh, M.; Gool, L. Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In Proceedings of the 2011 IEEE Workshop on Applications of Computer Vision (WACV), Kona, HI, USA, 5–7 January 2011; pp. 66–72.
4. Memo, A.; Zanuttigh, P. Head-Mounted Gesture Controlled Interface for Human-Computer Interaction. *Multimed. Tools Appl.* **2018**, *77*, 27–53. [CrossRef]
5. Foix, S.; Alenya, G.; Torras, C. Lock-in Time-of-Flight (ToF) Cameras: A Survey. *IEEE Sens. J.* **2011**, *11*, 1917–1926. [CrossRef]
6. Bhandari, A.; Raskar, R. Signal Processing for Time-of-Flight Imaging Sensors: An introduction to inverse problems in computational 3-D imaging. *IEEE Signal Process. Mag.* **2016**, *33*, 45–58. [CrossRef]
7. Chen, Y.; Ren, J.; Cheng, X.; Qian, K.; Wang, L.; Gu, J. Very Power Efficient Neural Time-of-Flight. In Proceedings of the 2010 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 1–5 March 2020; pp. 2257–2266.
8. Piao, C.; Ahn, Y.; Kim, D.; Park, J.; Kang, J.; Kim, S.; Chun, J.; Choi, J. A Low-Power Indirect Time-of-Flight CMOS Image Sensor with Fixed Depth Noise Compensation and Dual-Mode Imaging for Depth Dynamic Range Enhancement. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 3989–3999. [CrossRef]
9. Dai, S.; Estrada, C.; Xiong, A.; Xu, C.; Yuan, J.; Chan, M. A CMOS-Compatible Photonic Demodulator With Low-Power Consumption for Time-of-Flight Image Sensor. *IEEE Trans. Electron Devices* **2022**, *69*, 6178–6183. [CrossRef]
10. Noraky, J.; Sze, V. Low power depth estimation for time-of-flight imaging. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 2114–2118.
11. Nicola, M.; David, S.; Lucio, P. A low-power pixel-level circuit for high dynamic range Time-of-Flight camera. In Proceedings of the 2014 IEEE SENSORS, Valencia, Spain, 2–5 November 2014; pp. 625–628.
12. Jans, R.; Green, A.; Koerner, L. Characterization of a Miniaturized IR Depth Sensor With a Programmable Region-of-Interest That Enables Hazard Mapping Applications. *IEEE Sens. J.* **2020**, *20*, 5213–5220. [CrossRef]
13. Kim, D.; Lee, S.; Park, D.; Piao, C.; Park, J.; Ahn, Y.; Cho, K.; Shin, J.; Song, S.; Kim, S.; et al. Indirect Time-of-Flight CMOS Image Sensor with On-Chip Background Light Cancelling and Pseudo-Four-Tap/Two-Tap Hybrid Imaging for Motion Artifact Suppression. *IEEE J. Solid-State Circuits* **2020**, *55*, 2849–2865. [CrossRef]
14. Keel, M.; Jin, Y.; Kim, Y.; Kim, D.; Kim, Y.; Bae, M.; Chung, B.; Son, S.; Kim, H.; An, H.; Choi, S.; et al. A VGA Indirect Time-of-Flight CMOS Image Sensor With 4-Tap 7-μm Global-Shutter Pixel and Fixed-Pattern Phase Noise Self-Compensation. *IEEE J. Solid-State Circuits* **2020**, *55*, 889–897. [CrossRef]
15. Zhao, Z.; Xu, J.; Guo, S.; Nie, K. A Novel Imaging Method for Two-Tap Pulsed-Based Indirect Time-of-Flight Sensor. *IEEE Sens. J.* **2023**, *23*, 7017–7030. [CrossRef]

16. Chen, F.; Ying, R.; Xue, J.; Wen, F.; Liu, P. A Configurable and Real-Time Multi-Frequency 3D Image Signal Processor for Indirect Time-of-Flight Sensors. *IEEE Sens. J.* **2022**, *22*, 7834–7845. [CrossRef]
17. Arif, O.; Daley, W.; Vela, P.; Teizer, J.; Stewart, J. Visual tracking and segmentation using Time-of-Flight sensor. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 2241–2244.
18. Yu, K.; Li, Z.; Peng, Y.; Loy, C.; Gu, J. ReconfigISP: Reconfigurable Camera Image Processing Pipeline. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 4228–4237.
19. Yoshimura, M.; Otsuka, J.; Irie, A.; Ohashi, T. Dynamicisp: Dynamically controlled image signal processor for image recognition. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 12866–12876.
20. Seiter, J.; Hofbauer, M.; Davidovic, M.; Zimmermann, H. FPGA based time-of-flight 3D camera characterization system. In Proceedings of the IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), Karlovy Vary, Czech Republic, 8–10 April 2013; pp. 240–245.
21. Druml, N.; Ehrenhoefer, C.; Bell, W.; Gailer, C.; Plank, H.; Herndl, T.; Holweg, G. A fast and flexible HW/SW co-processing framework for Time-of-Flight 3D imaging. *Microelectron. Reliab.* **2018**, *81*, 64–76. [CrossRef]
22. Casares, M.; Velipasalar, S. Adaptive Methodologies for Energy-Efficient Object Detection and Tracking With Battery-Powered Embedded Smart Cameras. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 1438–1452. [CrossRef]
23. Iqbal, O.; Muro V.; Katoch, S.; Spanias, A.; Jayasuriya, S. Adaptive Subsampling for ROI-Based Visual Tracking: Algorithms and FPGA Implementation. *IEEE Access* **2022**, *10*, 90507–90522. [CrossRef]
24. Yu, X.; Liang, X.; Zhou, Z.; Zhang, B. Multi-task learning for hand heat trace time estimation and identity recognition. *Expert Syst. Appl.* **2024**, *255*, 124551. [CrossRef]
25. Yu, X.; Liang, X.; Zhou, Z.; Zhang, B.; Xue, H. Deep soft threshold feature separation network for infrared handprint identity recognition and time estimation. *Infrared Phys. Technol.* **2024**, *138*, 105223. [CrossRef]
26. Nguyen, T.-D.; Vu, A.-K.N.; Nguyen, N.-D.; Nguyen, V.-T.; Ngo, T.D.; Do, T.-T.; Tran, M.-T.; Nguyen, T.V. The Art of Camouflage: Few-Shot Learning for Animal Detection and Segmentation. *IEEE Access* **2024**, *12*, 103488–103503. [CrossRef]
27. Zhou, X. *A Study of Microsoft Kinect Calibration*; Department of Computer Science, George Mason University: Fairfax, VA , USA, 2012.
28. Cheng, W.; Park, J.; Ko, J. HandFoldingNet: A 3D hand pose estimation network using multiscale-feature guided folding of a 2D hand skeleton. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 11260–11269.
29. Tompson, J.; Stein, M.; Lecun, Y.; Perlin, K. Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. *ACM Trans. Graph.* **2014**, *33*, 1–10. [CrossRef]
30. Chen, Z.; Liu, L.; Wu, N. Calculating depth image with pixel-parallel processor for a TOF image sensing system. In Proceedings of the 2015 IEEE SENSORS, Busan, Republic of Korea, 1–4 November 2015; pp. 1–4.
31. Gyongy, I.; Erdogan, A.; Dutton, N.; Martín, G.; Gorman, A.; Mai, H.; Rocca, F.; Henderson, R. A direct time-of-flight image sensor with in-pixel surface detection and dynamic vision. *IEEE J. Sel. Top. Quantum Electron.* **2023**, *30*, 1–11. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Activities of Daily Living Object Dataset: Advancing Assistive Robotic Manipulation with a Tailored Dataset

Md Tanzil Shahria [1,*,†] and Mohammad H. Rahman [1,2,†]

[1] Computer Science, University of Wisconsin-Milwaukee, Milwaukee, WI 53211, USA; rahmanmh@uwm.edu
[2] Mechanical Engineering, University of Wisconsin-Milwaukee, Milwaukee, WI 53211, USA
* Correspondence: mshahria@uwm.edu
† Current address: Biorobotics Laboratory, University of Wisconsin-Milwaukee, 115 East Reindl Way, USR 281, Milwaukee, WI 53212, USA.

Abstract: The increasing number of individuals with disabilities—over 61 million adults in the United States alone—underscores the urgent need for technologies that enhance autonomy and independence. Among these individuals, millions rely on wheelchairs and often require assistance from another person with activities of daily living (ADLs), such as eating, grooming, and dressing. Wheelchair-mounted assistive robotic arms offer a promising solution to enhance independence, but their complex control interfaces can be challenging for users. Automating control through deep learning-based object detection models presents a viable pathway to simplify operation, yet progress is impeded by the absence of specialized datasets tailored for ADL objects suitable for robotic manipulation in home environments. To bridge this gap, we present a novel ADL object dataset explicitly designed for training deep learning models in assistive robotic applications. We curated over 112,000 high-quality images from four major open-source datasets—COCO, Open Images, LVIS, and Roboflow Universe—focusing on objects pertinent to daily living tasks. Annotations were standardized to the YOLO Darknet format, and data quality was enhanced through a rigorous filtering process involving a pre-trained YOLOv5x model and manual validation. Our dataset provides a valuable resource that facilitates the development of more effective and user-friendly semi-autonomous control systems for assistive robots. By offering a focused collection of ADL-related objects, we aim to advance assistive technologies that empower individuals with mobility impairments, addressing a pressing societal need and laying the foundation for future innovations in human–robot interaction within home settings.

Keywords: image dataset; activities of daily living; assistive robotics; object detection; deep learning

1. Introduction

The global demographic shift towards an aging population and the rising prevalence of disabilities have intensified the need for advanced assistive technologies that promote autonomy and improve the quality of life for individuals with mobility impairments. In the United States alone, over 61 million adults live with a disability [1], representing approximately 26% of the adult population. Mobility disabilities are among the most common, affecting millions who rely on wheelchairs or other assistive devices for daily movement. Moreover, almost one-third of mobility device users need assistance from another person in one or more of the activities of daily living (ADLs), such as eating, grooming, dressing, and personal hygiene [2]. Dependence on caregivers for these fundamental activities can impact the independence and psychological well-being of individuals with disabilities.

Assistive robotic technologies, particularly wheelchair-mounted robotic arms, have emerged as promising tools to enhance the independence of individuals with mobility impairments. These robotic systems are designed to assist users in performing ADLs by extending their physical capabilities and enabling them to interact with objects in their

environment. However, the practical deployment of such assistive robots faces challenges, primarily due to the complexity of control interfaces and the cognitive load required for manual operation. For many users, especially those with severe disabilities, controlling a multi-degree-of-freedom robotic arm can be daunting and may not lead to the desired level of independence.

Automation of robotic control through advanced perception systems offers a viable solution to these challenges. Deep learning-based object detection models have revolutionized computer vision [3], enabling robots to perceive and interpret complex environments with remarkable accuracy. By integrating such models, assistive robots can autonomously identify, locate, and manipulate objects relevant to ADLs, thereby reducing the operational burden on users. However, the success of these models is heavily dependent on the availability of large, high-quality datasets that accurately represent the target domain [4].

Existing large-scale image datasets, such as the Common Objects in Context (COCO) dataset [5], Open Images [6], the Large Vocabulary Instance Segmentation (LVIS) dataset [7], and collections from platforms like Roboflow Universe [8], have been instrumental in advancing object detection and segmentation algorithms. These datasets provide extensive collections of labeled images, covering a wide variety of objects and scenes. Nonetheless, they are general-purpose datasets not specifically tailored for assistive robotics applications. They often include a broad spectrum of objects, many of which are irrelevant or unsuitable for manipulation by assistive robots in home environments. Additionally, these datasets contain mislabeled or low-quality images, which can adversely affect the training and performance of deep learning models in real-world scenarios [9].

The lack of specialized datasets focusing on manipulable ADL objects presents a significant barrier to the development of effective perception systems for assistive robots. Domain-specific datasets have been shown to improve model performance in specialized applications by providing relevant contextual information and reducing the model's exposure to irrelevant data. For assistive robotics, a dataset that captures the diversity of objects encountered in daily living tasks, annotated with high precision, is essential for training models that can operate reliably in home settings. In this study, we aim to address these limitations by creating a specialized, high-quality dataset tailored for ADL object detection in assistive robotics. The primary objectives of this research are:

- Dataset curation and compilation: to curate and compile a comprehensive dataset of manipulable ADL objects from existing large-scale datasets, focusing on items relevant to assistive robots in home environments.
- Annotation standardization: to standardize the annotations of the collected images into a uniform YOLO (You Only Look Once) Darknet annotation format [10], facilitating seamless integration with state-of-the-art object detection models.
- Data quality enhancement: to filter and enhance the dataset by removing mislabeled and low-quality images through a combination of automated filtering using a pre-trained YOLOv5x [11] model and manual review processes.

Our contributions include the development of a novel dataset of ADL objects specifically curated for assistive robotics applications, which we will make publicly available as an open-source resource. This dataset addresses the critical need for high-quality, task-specific data that can significantly improve the performance of object detection models in assistive scenarios. By providing a focused collection of manipulable objects commonly encountered in daily living tasks, we aim to bridge the data gap in this domain and facilitate advancements in assistive technologies.

This work addresses a pressing societal need and lays the foundation for future innovations in human–robot interaction within home settings. By enhancing the perceptual capabilities of assistive robots, we can empower individuals with mobility impairments to achieve greater independence and improve their overall quality of life. The results of this study underscore the importance of targeted datasets in driving advancements in deep learning and robotics, highlighting the potential for specialized data to unlock new levels of performance and reliability in assistive technologies.

2. Related Work

The advancement of object detection has been significantly propelled by the development of large-scale, annotated datasets that serve as benchmarks for training and evaluating deep learning models. The Common Objects in Context (COCO) dataset [5] is one of the most influential, providing over 200,000 images with detailed annotations for object detection, segmentation, and captioning tasks. COCO's diverse range of everyday objects in natural contexts has made it a cornerstone for research in computer vision. Similarly, the Open Images dataset [6] offers an extensive collection of approximately 9 million images annotated with image-level labels, bounding boxes, and visual relationships, encompassing a vast array of object classes. The Large Vocabulary Instance Segmentation (LVIS) dataset [7] extends these efforts by focusing on instance segmentation with a large vocabulary of over 1000 object categories, aiming to address the long-tail distribution of object frequencies in the real world.

While these general-purpose datasets have been instrumental in advancing object detection techniques, their applicability to robotic manipulation—especially in assistive robotics—is limited. The objects included often do not represent the manipulable items commonly encountered in activities of daily living (ADLs), and the images may not reflect the specific environmental contexts of home settings where assistive robots operate. Also, the annotations in these datasets are not tailored to the needs of robotic perception systems, which require precise and relevant information for object manipulation tasks.

Recognizing the need for specialized datasets, several efforts have been made to create collections more suited to robotic applications. The Yale-CMU-Berkeley (YCB) Object and Model Set [12] is a notable example, providing physical objects along with high-resolution 3D scans for benchmarking robotic manipulation algorithms. This dataset includes everyday items like kitchenware and tools selected for their relevance to manipulation tasks and variability in shape, size, and texture. The Washington RGB-D Object dataset [13] offers RGB-D images of household objects from multiple viewpoints, facilitating research in object recognition and pose estimation using depth information, which is valuable for robotic grasping and manipulation.

In the context of robotic grasping, the Cornell Grasping dataset [14] provides images annotated with positive and negative grasp rectangles, aiding the development of algorithms for detecting graspable regions on objects. While these datasets have advanced specific areas of robotic manipulation, they often lack the breadth and diversity needed for comprehensive ADL object detection in assistive robotics. They may focus on a limited set of objects or specific tasks like grasp detection without encompassing the full range of items and scenarios encountered in daily living environments.

Assistive robotics has seen growing interest, with research focusing on developing robotic systems that aid individuals with disabilities in performing ADLs. Smarr et al. [15] explored robotic assistance in dressing, utilizing computer vision to detect clothing items and plan manipulation strategies. This study highlights the potential of assistive robots to enhance independence and underscore the challenges posed by complex control interfaces and the need for reliable perception systems.

Deep learning techniques have been increasingly adopted in assistive robotics to enhance autonomy and adaptability. For instance, Shahria et al. [16] developed a vision-based assistive robot capable of recognizing and manipulating objects for picking up an object task, employing the YOLOv5 [11] model for object detection. However, these approaches often rely on general-purpose datasets or limited custom datasets that do not fully capture the diversity of objects and contexts relevant to assistive tasks in home environments. This reliance can lead to performance issues when models encounter unfamiliar objects or settings not represented in their training data.

The limitations of existing datasets in addressing the specific needs of assistive robotics have been acknowledged in recent studies. Ruiz-del-Solar et al. [17] emphasized the importance of domain-specific datasets for improving the performance of robotic perception systems in assistive applications. They noted that datasets tailored to the objects and envi-

ronments typical of home settings could significantly enhance the accuracy and reliability of object detection models used in assistive robots.

To address these challenges, some researchers have created specialized datasets focusing on specific aspects of assistive robotics. For example, the Assistive Grasping dataset [18] includes a collection of objects commonly found in household environments, annotated for grasp planning tasks. While valuable, these datasets are often limited in scope and do not provide the comprehensive coverage needed for general ADL object detection.

Our work seeks to fill this gap by creating a large-scale, high-quality dataset specifically curated for manipulable ADL objects in home environments. By systematically collecting and annotating images from existing large-scale datasets—focusing on objects pertinent to daily living tasks—we aim to provide a resource that addresses the specific challenges faced in assistive robotics. This contribution is intended to facilitate the development of advanced perception systems that can empower individuals with disabilities through more autonomous and effective assistive technologies.

3. Dataset Creation

The development of a specialized dataset tailored for ADL object detection in assistive robotics necessitates meticulous data collection and rigorous quality assurance processes. This section outlines the methodology employed in creating our dataset, comprising two primary phases: data collection and data filtering. The goal was to assemble a high-quality dataset of manipulable ADL objects with precise annotations, suitable for training deep learning models in assistive robotic applications.

3.1. Data Collection

The initial step involved identifying a comprehensive list of ADL objects relevant to assistive robotic manipulation in home environments. Building upon our previous study [19], which identified approximately 90 objects that assistive robots need to interact with to provide effective assistance, we curated a focused subset of 54 ADL objects for this study.

The previous study [19] was conducted under a National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR)-funded project. A multidisciplinary team comprising experts in assistive technology, occupational therapy, and human–robot interaction collaboratively evaluated the initial list. Through expert review and iterative feedback from end-users and caregivers, we ensured that the selected objects are both relevant and practical for enhancing user independence. The refinement from 90 to 54 objects was focused on objects that are autonomously or semi-autonomously manipulable by assistive robots and essential for performing ADLs, considering factors such as size, shape, and frequency of use in daily tasks. The selected objects encompass a range of items commonly encountered in daily activities. This targeted selection ensures that the dataset is directly applicable to the contexts in which assistive robots operate.

To compile the dataset, we sourced images from four major open-source datasets known for their extensive collections and diversity:

- COCO dataset: We downloaded over 80,000 images covering 24 classes from the COCO dataset. COCO is renowned for its high-quality annotations and variety of everyday objects in context, making it a valuable resource for our purposes.
- Open Images dataset: From the Open Images dataset, we acquired over 31,000 images encompassing 39 classes. This dataset provides a vast collection of images with rich annotations, including object bounding boxes and class labels.
- LVIS dataset: The LVIS dataset contributed over 27,000 images across 36 classes. LVIS focuses on a large vocabulary of object categories and instance segmentation, offering detailed annotations suitable for object detection tasks.
- Roboflow Universe: For classes not sufficiently represented in the aforementioned datasets, we searched the Roboflow Universe platform, obtaining over 19,000 images

spanning 14 classes. Roboflow Universe aggregates datasets from various contributors, providing access to specialized datasets that complement our requirements.

After collecting the images, we standardized all annotations by converting them to the YOLO Darknet format. This involved transforming existing annotation formats into a unified structure specifying object class labels and bounding box coordinates relative to image dimensions. Class IDs were mapped according to our predefined list of 54 objects, ensuring consistency across the dataset. Uniformity in annotation format and class labeling is critical for effective model training and broader adoption by the research community. The list of 54 object classes is presented in Table 1.

Table 1. The list of 54 object classes along with their class id.

0: Apple	1: Handbag	2: Fork	3: Carrot	4: Donut	5: Backpack
6: Books	7: Cellphone/Mobile	8: Bottles	9: Remote	10: Pizza	11: Keyboard
12: Mouse	13: Knife	14: Glasses	15: Cup	16: Spoon	17: Bowl
18: Banana	19: Orange	20: Broccoli	21: Hot Dog	22: Clock	23: Laptop
24: Eraser	25: Pencil Box	26: Notebook	27: Pen	28: Ruler	29: Sharpener
30: Lipstick	31: Sock	32: Table Lamp	33: Pill Bottle	34: Lotion Bottle	35: Hat
36: Food Can	37: Wallet	38: Perfume	39: Camera	40: Shoe	41: Toilet Tissue
42: Scissor	43: Highlighter	44: Plum	45: Avocado	46: Peach	47: Tomato
48: Key	49: Bag of chips	50: Wrist Watch	51: Door Knob	52: Milk Gallon	53: Pencil

In this study, we focused on curating and filtering existing image data to create a high-quality dataset tailored for assistive robotic applications. We did not alter or augment the original images; instead, we meticulously selected images based on the specific needs of our application. By leveraging existing datasets and applying rigorous filtering criteria, we ensured that the dataset is both relevant and of high quality without introducing any modified or artificially generated content.

3.2. Data Sources and Ethical Considerations

All images and annotations obtained from the mentioned datasets are released under the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits the sharing and adaptation of the material in any medium or format for any purpose, even commercially, provided that appropriate credit is given, a link to the license is provided, and any changes made are indicated. This license explicitly allows for the redistribution of the images and annotations, enabling us to legally incorporate and redistribute these data within our newly created dataset. In compliance with the license terms, we have provided appropriate attribution to the original creators of each dataset, included license information in our documentation, and indicated any modifications performed. By adhering to these requirements, we ensure that our dataset respects the legal and ethical standards set forth by the original data providers, facilitating its use and redistribution by the broader research community and fostering advancements in assistive robotics.

3.3. Data Filtering

Ensuring the quality and reliability of the dataset is paramount for training effective deep-learning models. To this end, we implemented a two-phase filtering process comprising automated filtering and manual verification. This rigorous approach was designed to eliminate mislabeled data, incorrect annotations, and low-quality images that could adversely affect model performance.

3.3.1. Automated Filtering

The automated filtering phase utilized a pre-trained YOLOv5x [11] model, known for its high accuracy and efficiency in object detection tasks. The selection of YOLOv5x was based on several key considerations:

- Balance of speed and accuracy: YOLOv5x offers a superior balance between detection speed and accuracy, making it well-suited for processing large volumes of images efficiently without compromising on performance. This balance is crucial for maintaining the feasibility of the automated filtering process within our resource constraints.
- Proven effectiveness on the COCO dataset: Because YOLOv5x is trained on the COCO dataset, it demonstrates proficiency in detecting a wide range of common objects that overlap significantly with our selected ADL classes. This pre-existing competence reduces the likelihood of false negatives and enhances the reliability of the automated filtering process.
- Ease of integration and customization: YOLOv5x is highly adaptable and can be easily fine-tuned to specific datasets. Its compatibility with popular deep learning frameworks and availability of extensive documentation facilitated seamless integration into our data processing pipeline.
- Community and support: YOLOv5x benefits from a strong community and active support, providing access to numerous resources, updates, and best practices that informed our implementation strategy.

While alternative object detection models such as Faster R-CNN, EfficientDet, and Mask R-CNN offer distinct architectural advantages, the scope of our study prioritized efficiency and practicality in the data filtering phase. Conducting comprehensive comparisons with these models would require significant additional computational resources and time, which were beyond the constraints of our current project. Also, YOLOv5x's established performance on datasets similar to ours provided a reliable foundation for initial data quality assurance. However, we acknowledge that exploring other models could provide valuable comparative insights and may be considered in future work to further validate and potentially enhance the filtering process.

Automated filtering process: The automated filtering process involved the following steps:

- Model inference: for each image, the YOLOv5x model was used to perform object detection, generating predictions that include detected classes, bounding box coordinates, and confidence scores.
- Annotation comparison: the model's predictions were compared against the annotations in our dataset. If the predicted classes matched the annotated classes, and if the confidence scores exceeded a predefined threshold (set at 0.5), the image and its annotations were considered valid and retained.
- Mismatch identification: images where the model's predictions did not align with the annotations were flagged as potentially mislabeled or poorly annotated. These images were moved to a separate directory for discarding.
- Unknown classes handling: for annotated classes not recognized by the pre-trained model (due to the model's limited class set), images were set aside for manual verification to prevent the inadvertent removal of valid data.

The utilization of a pre-trained YOLOv5x model in automated filtering offers several advantages:

- Efficiency: automated filtering accelerates the initial data cleansing process, enabling rapid processing of a large volume of images.
- Consistency: the model applies uniform validation criteria across all images, reducing human error and bias.
- Quality assurance: filtering out images with incorrect annotations or mislabeled objects enhances the overall dataset quality, which is crucial for training reliable models.

The automated filtering process is exemplified by a Python script. The script systematically loads images and annotations, performs model inference, and categorizes images based on the comparison results, ensuring that only high-quality data are retained.

3.3.2. Manual Verification

Following automated filtering, manual verification was conducted to refine the dataset further. This phase involved human reviewers using a Python script to visualize images with annotations, allowing for meticulous inspection of annotation accuracy and image quality. Criteria for excluding images during manual verification included:

- Incorrect labels: annotations where the class labels did not correspond to the objects present in the image.
- Poorly labeled images: inaccurate bounding boxes, missing annotations, or annotations that only partially covered the object.
- Annotation inconsistencies: discrepancies between the number of objects present and the number of annotations (e.g., an image with n objects but n+m or n-m annotations).
- Incorrect bounding box sizes: bounding boxes that were disproportionately large or small relative to the object could impair the model's ability to learn precise localization.

Images meeting quality standards were retained, while those failing to comply were removed. Special consideration was given to images containing multiple objects. If an image contained n instances of object A and m instances of object B, and the annotations accurately reflected all instances of at least one object class, the image was retained for that class. The images in Figures 1–3 illustrate examples of mislabeled or ignored data excluded during the filtering process from three major datasets: COCO, LVIS, and Open Images. These examples highlight instances where data quality issues, such as incorrect annotations or irrelevant object labels and annotation missing for multiple objects, were identified and subsequently filtered out to ensure the integrity and accuracy of the final dataset.

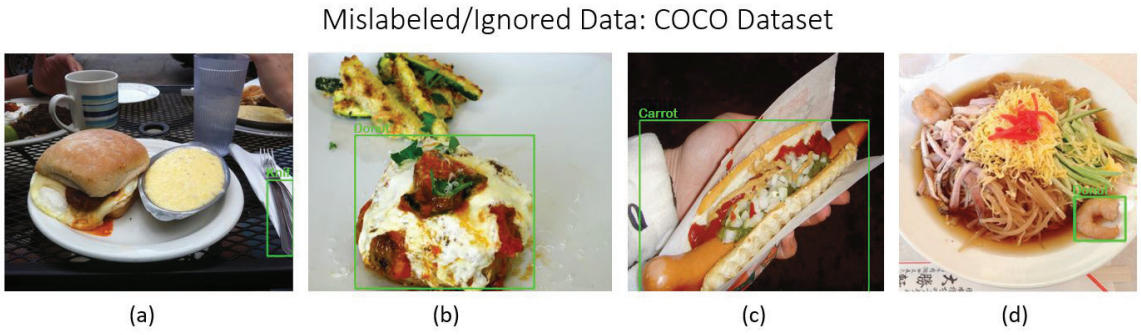


Figure 1. Examples of mislabeled or ignored data from the COCO dataset: (**a**) incorrect class label; (**b**) incorrect class label; (**c**) incorrect class label; (**d**) incorrect class label.

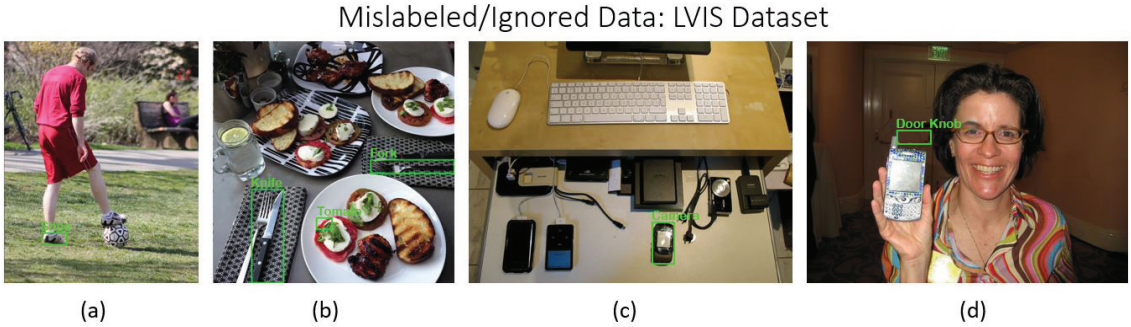


Figure 2. Examples of mislabeled or ignored data from the LVIS dataset: (**a**) missing annotation for one class; (**b**) missing annotations for multiple classes; (**c**) incorrect class label; (**d**) incorrect bounding box information.

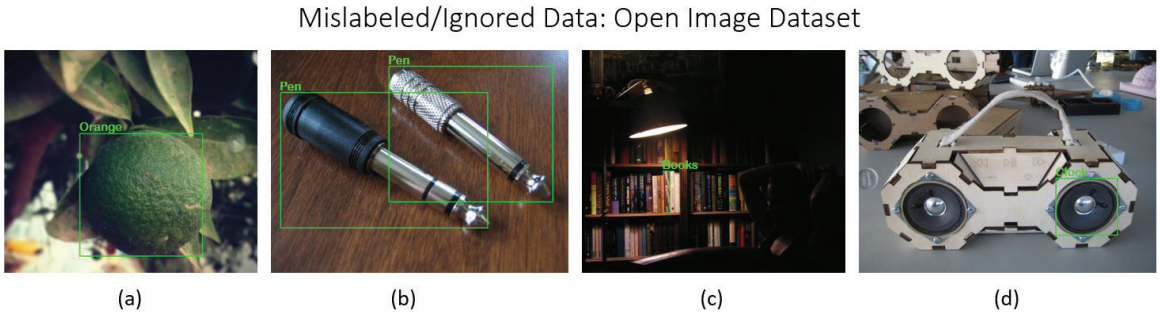


Figure 3. Examples of mislabeled or ignored data from the Open Images dataset: (**a**) incorrect class label; (**b**) incorrect class label; (**c**) missing annotations; (**d**) incorrect class label.

3.3.3. Error Analysis and Class Imbalance

Error analysis: during the automated filtering and manual verification processes, we identified several common types of data issues within the source datasets. The most frequent issues included:

1. Incomplete annotations for multiple objects: Many images contained multiple instances of the same object class, but annotations were provided for only a single instance. This resulted in underrepresented object instances within the dataset.
2. Inaccurate bounding box information: A significant number of annotations had bounding boxes that were either too large, too small, or improperly aligned with the actual object boundaries, leading to imprecise localization.
3. Incorrect class labels: Mislabeling of object classes was prevalent, where the annotated class did not correspond to the object present in the image. This included both overgeneralized and overly specific class labels.
4. Poor image quality: Low-resolution images, excessive noise, and poor lighting conditions were common, affecting the clarity and detectability of objects.

Class imbalance: Addressing class imbalance is crucial for training models that perform uniformly across all object classes. In our dataset, certain classes were significantly underrepresented compared with others. To mitigate this, we primarily focused on the following strategies:

- Selective data collection: We prioritized collecting additional images for underrepresented classes from the Roboflow Universe platform and other supplementary sources to balance the class distribution.
- No data augmentation: Importantly, we did not apply data augmentation techniques, such as oversampling or the use of generative adversarial networks (GANs), to generate synthetic data. This decision was made to preserve the authenticity and integrity of the data, ensuring that all samples represent real-world instances without introducing potential biases or artifacts that synthetic data might introduce.

3.4. Dataset Integrity and Scope

It is important to note that our dataset is not intended to supplant existing open-source datasets nor to suggest that they lack quality. Instead, we aim to provide a specialized dataset specifically designed for ADL object manipulation in assistive robotics. By focusing on objects pertinent to daily living tasks and ensuring high annotation quality through cross-verification, our dataset addresses a niche yet critical area in the field.

The final dataset comprises a curated collection of high-quality images with accurate annotations, ready for training deep learning models to enhance the perceptual capabilities of assistive robots. Table 2 presents the object count statistics from various image sources, comparing downloaded and filtered datasets across all sources. This provides a comprehensive overview of the data distribution for each object category.

Table 2. Object count statistics from various image sources (downloaded and filtered data).

ID	Object Name	Downloaded Data				Filtered Data				Total
		COCO	Open Image	LVIS	Roboflow	COCO	Open Image	LVIS	Roboflow	
0	Apple	1586	389	1207		1191	376	717		2284
1	Handbag	6841	652	1859		4253	216	1107		5576
2	Fork	3555		1861		2602		1177		3779
3	Carrot	1683	499	1222		1370	441	777		2588
4	Donut	1523				1365				1365
5	Backpack	5528	165	1905		3345	90	1188		4623
6	Books	5332	1791	1903		3283	938	1004		5225
7	Cellphone	4803	1778		80	3477	1486		48	5011
8	Bottles	8501	3808			6967	3567			10,534
9	Remote	3078				2491				2491
10	Pizza	3166	588	1881		2908	547	1407		4862
11	Keyboard	2115	1299			1627	739			2366
12	Mouse	1876	512			1671	438			2109
13	Knife	4326	537	1868		2928	324	1170		4422
14	Glass	2533				2233				2233
15	Cup	9189	1117	1521		7698	1055	1007		9760
16	Spoon	3529	383	1127		2207	331	682		3220
17	Bowl	7111	689	1922		5355	455	1244		7054
18	Banana	2243	589	1787		1974	536	1344		3854
19	Orange	1699	883			1324	797			2121
20	Broccoli	1939	210	1309		1627	201	970		2798
21	Hot Dog	1222	72			997	64			1061
22	Clock	4659	884	1844		3840	830	1457		6127
23	Laptop	3524	5123		79	3245	4788		42	8075
24	Eraser		18	318			6	126		132
25	Pencil Box				258				110	110
26	Notebook		189	310			30	72		102
27	Pen		536	339	36		415	60	10	485
28	Ruler		36		540		30		107	137
29	Sharpener				846				120	120
30	Lipstick				190				94	94
31	Sock		231	1945	758		120	12	62	194
32	Table lamp				946				220	220
33	Pill Bottle				188				93	93
34	Lotion Bottle				651				194	194
35	Hat		6292	1932		5223	347			5570
36	Food Can		445	1051			83	70		153
37	Wallet		113	1041			38	127		165

Table 2. *Cont.*

ID	Object Name	Downloaded Data				Filtered Data				Total
		COCO	Open Image	LVIS	Roboflow	COCO	Open Image	LVIS	Roboflow	
38	Perfume		13	930			7	190		197
39	Camera		1721	1391		1178	111			1289
40	Shoe		1915	1237			25	218		243
41	Toilet Tissue		156		931		121		284	405
42	Scissors		172	707	829		170	497	10	677
43	Highlighter		128	494			11	264		275
44	Plum			1089				165		165
45	Avocado		117	1723			3	102		105
46	Peach		99	71	655		42	2	78	122
47	Tomato		1213	1470			18	111		129
48	Keys		195	941			24	165		189
49	Bag of Chips			575				213		213
50	Wrist Watch			1089				240		240
51	Door Knob			1710				590		590
52	Milk Gallon			224				47		47
53	Pencil		153	475			19	169		188

4. ADL Dataset

The final dataset resulting from our meticulous data collection and rigorous filtering processes is a comprehensive resource tailored specifically for assistive robotic manipulation in activities of daily living (ADLs). This section provides an in-depth overview of the dataset's characteristics, including the distribution of images across object classes, the diversity of sources, and the quality enhancements achieved through our filtering methods.

4.1. Dataset Composition

Our dataset encompasses 112,925 images after filtering, covering 54 ADL object classes crucial for assistive robots operating in home environments. The images were sourced from four prominent open-source datasets: COCO, Open Images, LVIS, and Roboflow Universe. The diversity of these sources ensures a wide range of visual contexts, object appearances, and environmental variations, enhancing the robustness of models trained on this dataset.

To facilitate effective training and validation, the total dataset was randomly split into two directories: train and valid. Approximately 85% of the data is kept in the train directory for model training, while the remaining 15% is stored in the valid directory for validation purposes. This random splitting strategy helps ensure that both sets are representative of the overall data distribution, contributing to the development of models with strong generalization capabilities.

Class Distribution

The dataset includes objects commonly encountered in daily tasks, such as utensils, personal care items, electronic devices, and household goods. Table 3 provides a comprehensive summary of the class-wise distribution of images and instances across the proposed dataset, including detailed counts for each class as well as aggregate totals for the number of images and instances processed. The class distribution is balanced to the extent possible, with particular attention paid to objects of high relevance to assistive tasks. Some key observations from the dataset include:

- High-frequency classes: Objects like Cup, Bottle, and Hat have a higher number of images, reflecting their frequent use and importance in daily activities. For instance, the Cup class comprises a total of 9760 images after filtering, sourced primarily from COCO and Open Images.
- Specialized classes: Less common but essential objects, such as Pill Bottle, Door Knob, and Milk Gallon, are included to cover a comprehensive range of assistive needs. While smaller in image count, these classes are critical for developing models that can handle specialized tasks.
- Diverse representations: By integrating data from multiple sources, the dataset captures variations in object appearance due to different angles, lighting conditions, backgrounds, and occlusions, essential for training robust models capable of generalizing to real-world scenarios.

Table 3. Class-wise statistics of the final dataset, including the number of images and instances for each class.

Class ID	Number of Images	Total Instances	Class ID	Number of Images	Total Instances
0	2284	6176	27	485	607
1	5576	9937	28	137	165
2	3779	5343	29	120	143
3	2588	8438	30	94	188
4	1365	6789	31	194	600
5	4623	7132	32	220	242
6	5225	18,384	33	93	106
7	5011	6977	34	194	273
8	10,534	28,694	35	5570	6990
9	2491	4934	36	153	155
10	4862	7828	37	165	184
11	2366	3172	38	197	198
12	2109	2541	39	1289	1409
13	4422	6995	40	243	442
14	2233	7255	41	405	465
15	9760	20,414	42	677	709
16	3220	5283	43	275	275
17	7054	14,059	44	165	543
18	3854	10,768	45	105	143
19	2121	10,728	46	122	149
20	2798	7896	47	129	164
21	1061	2669	48	189	301
22	6127	7980	49	213	261
23	8075	9941	50	240	288
24	132	178	51	590	590

Table 3. *Cont.*

Class ID	Number of Images	Total Instances	Class ID	Number of Images	Total Instances
25	110	110	52	47	155
26	102	126	53	188	276
Summary					
Total number of images with only one class				110,202	
Total number of images with more than one class				2723	
Total number of images in the dataset				112,925	
Total number of instances in the dataset				236,768	

4.2. Data Quality Enhancements

The filtering processes significantly improved the dataset's quality by removing mislabeled, poorly annotated, or low-quality images. The overall reduction in the number of images from the initial download to the final dataset underscores the stringency of our quality assurance measures.

- Automated filtering impact: The automated filtering phase eliminated images with incorrect or inconsistent annotations, as identified by discrepancies between model predictions and annotations. For example, after filtering, the Bottle class was reduced from 8501 images to 6967 images from the COCO dataset.
- Manual verification impact: Manual inspection further refined the dataset, ensuring that annotations precisely matched the objects in the images. This step was crucial for classes where automated filtering might overlook nuanced errors, particularly in images with multiple objects or complex scenes.

4.3. Data Source Contributions

The contributions of each data source to the final dataset are as follows:

- COCO dataset: This dataset provided the majority of images, with 69,956 images across various classes after filtering. COCO's high-quality annotations and contextual diversity make it a valuable foundation for the dataset.
- Open Images dataset: This dataset contributed 25,488 images after filtering. Open Images enriched the dataset with additional classes and varied visual representations.
- LVIS dataset: This dataset added 13,993 images post-filtering. LVIS's focus on a large vocabulary of objects enhanced the diversity of object categories.
- Roboflow Universe: This dataset supplied 3488 images after filtering, particularly for classes underrepresented in other datasets. This inclusion ensured coverage of specialized objects necessary for assistive tasks.

4.4. Dataset Accessibility and Usage

The dataset is publicly available (under CC BY 4.0) as an open-source resource to encourage collaboration and advancement in the field. Researchers and developers can leverage this dataset to train and evaluate object detection models specifically tailored for ADL tasks, accelerating the development of semi-autonomous control systems for assistive robots. The dataset can be accessed/download from the following link: https://doi.org/10.6084/m9.figshare.27263424 (accessed on 24 November 2024)

5. Experimental Evaluation

To assess the effectiveness of our ADL dataset, we conducted experiments using the YOLOv8n deep learning model. This evaluation aimed to demonstrate the dataset's capability to enhance object detection performance in assistive robotics applications. Also,

we performed a comparative analysis with models trained on existing public datasets to highlight the advantages and unique characteristics of our ADL dataset.

5.1. Model Validation and Performance Evaluation

5.1.1. Experimental Setup

We selected YOLOv8n for its optimal balance between accuracy and computational efficiency, which is crucial for real-time applications in assistive robotics. The model was fine-tuned on our ADL dataset for 50 epochs using a learning rate of 0.01, a batch size of 32, and an input image size of 640 × 640 pixels. Training was conducted on a system equipped with an NVIDIA RTX A2000 GPU (manufactured by NVIDIA Corporation, headquartered in Santa Clara, CA, USA) to ensure efficient processing.

5.1.2. Evaluation Metrics

The model was evaluated using standard object detection metrics to provide a comprehensive understanding of its performance:

- Mean average precision (mAP@0.5): measures the average precision across all classes at an intersection over union (IoU) threshold of 0.5.
- Mean average precision (mAP@0.5:0.95): averages the mAP over multiple IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05.
- Precision: the ratio of true positive detections to the total number of positive detections.
- Recall: the ratio of true positive detections to the total number of actual positives in the dataset.

5.1.3. Results

Table 4 presents the performance metrics of the fine-tuned YOLOv8n model on the validation set of our ADL dataset. The model achieved a mAP@0.5 of 0.592 and a mAP@0.5:0.95 of 0.458. Additionally, the precision and recall metrics of the model are 0.572 and 0.608, respectively.

Table 4. Performance metrics of fine-tuned models.

Metric	YOLOv8n (Fine-Tuned)
mAP@0.5	0.592
mAP@0.5:0.95	0.458
Precision	0.572
Recall	0.608

5.1.4. Inference Speed

To ensure that the fine-tuned model remains suitable for real-time applications in assistive robotics, we evaluated its inference speed. The average inference times per image are presented in Table 5. The inference times remained consistent before and after fine-tuning, with the YOLOv8n model averaging 18.45 ms post-fine-tuning compared with 18.10 ms pre-fine-tuning. This consistency confirms that the enhanced performance does not incur additional computational overhead, maintaining the model's suitability for real-time robotic operations.

Table 5. Average inference times (milliseconds per image) of YOLOv8n models.

Model	Inference Time (ms)
Fine-tuned YOLOv8n	18.45
Pre-trained YOLOv8n	18.10

5.1.5. Discussion

The performance of the model demonstrates the effectiveness of our ADL dataset in enhancing model performance for assistive robotics applications. Fine-tuning allows the YOLOv8n model to learn specific features and contexts relevant to ADL objects, which are not adequately represented in general-purpose datasets. The maintenance of real-time inference speeds ensures that these enhancements are practical for deployment in assistive robotic systems.

5.2. Comparative Analysis with Existing Public Datasets

To demonstrate the advantages and unique characteristics of our ADL dataset in assistive robotics applications, we conducted a comparative analysis with existing public datasets commonly used for object detection tasks, such as COCO and Open Images. This analysis focuses on the relevance of the datasets to assistive robotics, coverage of ADL-related objects, annotation quality, and their suitability for training models in assistive applications.

5.2.1. Relevance to Assistive Robotics

Our ADL dataset is specifically curated to include 54 object classes that are directly relevant to activities of daily living and suitable for manipulation by assistive robots in home environments. In contrast, while the COCO and Open Images datasets contain a wide variety of object classes, they are general purpose and not tailored to the needs of assistive robotics. Many ADL-relevant objects are underrepresented or absent in these datasets, limiting their effectiveness for training models intended for assistive applications.

As illustrated in Table 6, our dataset offers comprehensive coverage of ADL-relevant objects, whereas COCO and Open Images include only a subset of these classes. This extensive coverage ensures that models trained on our dataset are exposed to a broader range of objects pertinent to assistive robotics, thereby enhancing their applicability and performance in real-world assistive tasks.

Table 6. Comparison of ADL-relevant object classes across datasets.

Dataset	Total Classes	ADL-Relevant Classes
ADL dataset	54	54
COCO dataset	80	24
Open Images	600	39

5.2.2. Annotation Quality and Consistency

Our dataset emphasizes high-quality annotations, achieved through rigorous filtering and manual verification processes detailed in Section 3.3. This ensures accurate bounding boxes and consistent class labels, which are critical for the precision required in assistive robotics. High-quality annotations contribute to better model performance and reliability, essential factors for deployment in real-world assistive applications.

In contrast, existing datasets like COCO and Open Images, while extensive, may contain mislabeled data or inconsistent annotations, particularly for objects that are less common or have ambiguous definitions. Such issues can adversely affect the performance of models in critical applications like assistive robotics, where accuracy and reliability are paramount. By providing meticulously curated and annotated data, our ADL dataset mitigates these challenges, offering a more dependable foundation for training robust object detection models tailored to assistive robotics.

6. Discussion

The creation of the specialized ADL dataset presented in this study addresses a critical gap in assistive robotics, particularly in enhancing the autonomy and effectiveness of assis-

tive robots. Integrating deep learning-based object detection models into assistive robots can revolutionize how individuals with mobility impairments interact with their environment, reducing dependence on caregivers and improving quality of life. This discussion elaborates on how our dataset supplements existing work, the technical advancements it enables, and its implications for future research and development.

6.1. Contribution to the Field

Our dataset supplements existing large-scale datasets like COCO, Open Images, and LVIS by providing a focused collection of images specifically tailored for ADL object detection in assistive robotics. While these general-purpose datasets have been invaluable for advancing computer vision algorithms, they lack the specificity required for assistive applications. By curating a dataset with 54 manipulable ADL objects, we ensure that the training data align closely with the operational environment of assistive robots. This alignment is crucial for developing models that perform reliably in home settings, where the variability of objects and contexts can significantly affect model performance.

Also, the rigorous filtering and annotation processes enhance the dataset's quality, addressing common issues in existing datasets such as mislabeled data and inconsistent annotations. High-quality annotations are essential for training deep learning models that require precise localization and classification capabilities, particularly when assistive robots must manipulate objects safely and effectively.

Enhancement of robotic perception and control systems: Beyond the development of a specialized dataset, our work significantly contributes to the advancement of robotic perception and control systems in assistive robotics. By providing high-quality, task-specific data, we enable the training of more accurate and reliable object detection models tailored to the unique demands of assistive tasks. This leads to improved autonomy and efficiency in robotic manipulation, reducing the cognitive and operational burdens on users. Additionally, our dataset serves as a benchmark for evaluating and comparing different deep learning architectures in the context of assistive robotics, fostering further research and innovation in this domain. The standardized annotation format and rigorous quality assurance processes we established can be adopted by future studies, promoting consistency and reproducibility in assistive robotics research.

6.2. Technical Advancements

The availability of this specialized dataset enables several technical advancements in assistive robotics. Models trained on the dataset can achieve higher accuracy in detecting and localizing objects used in ADLs as they are exposed to various instances and contexts specific to assistive tasks, reducing the domain gap and improving generalization. Accurate object detection is crucial for effective robotic manipulation as it enables perception systems to handle complex scenes, recognize objects in cluttered environments, and adapt to variations in appearance. Additionally, automating object detection reduces cognitive load on users, allowing them to issue high-level commands instead of managing intricate robotic movements. The dataset provides a benchmark for evaluating object detection models, encouraging innovation and standardization in assistive robotics.

6.3. Tangible Outcomes and Validation

The deployment of our ADL dataset is expected to yield significant tangible outcomes in the realm of assistive robotics, particularly in enhancing the autonomy and effectiveness of robotic systems in supporting activities of daily living. By enabling the development of more accurate and reliable object detection models, our dataset facilitates autonomous robotic manipulation of essential ADL objects, thereby increasing user independence and reducing reliance on human caregivers. The anticipated benefits include improved quality of life for individuals with mobility impairments, greater economic efficiency through the reduction of caregiving costs, and accelerated innovation in assistive technologies. To confirm that these benefits meet expectations, we will establish a robust scientific

validation framework that combines empirical performance evaluations, user studies, and longitudinal assessments. This framework will involve quantitative metrics such as mean average precision (mAP), precision, and recall to objectively measure model performance and qualitative feedback from end-users and caregivers to assess usability and real-world impact. Also, real-world deployment tests with assistive robotic systems will be conducted to observe and validate the practical effectiveness of the trained models in dynamic home environments. Through this comprehensive validation approach, we aim to ensure that the deployment of our dataset delivers the intended benefits and contributes meaningfully to the advancement of assistive robotics.

6.4. Limitations and Future Work

Despite the significant contributions of our dataset, some limitations present opportunities for future research:

1. Class imbalance: While efforts were made to balance the dataset, some classes have fewer images due to data availability in source datasets. This imbalance could affect model performance on underrepresented classes. Future work may involve collecting additional data or employing data augmentation techniques to address this issue.
2. Limited viewpoint diversity: The current dataset may not fully capture objects from multiple angles and backgrounds, which is important for simulating complex real-world environments. Future work will involve incorporating images of objects from various perspectives and settings to enhance model robustness against variations in object appearance.
3. Dynamic interactions: Considering the actual usage scenarios of assistive robots, the dataset lacks images of objects in use or in motion, such as door handles at various stages of operation. Including such dynamic representations in future datasets could improve the models' ability to handle real-world manipulation tasks involving moving objects.
4. Three-dimensional perception: Assistive robots often rely on depth information for accurate manipulation. Integrating RGB-D data or 3D models could improve object detection and localization, facilitating more precise interactions.
5. Incorporation of time series data: Future research will focus on extending the dataset to include time series data, capturing dynamic interactions and temporal variations. This enhancement aims to better simulate real-world environments where objects may be in motion or subject to changing conditions, thereby improving the models' ability to handle dynamic manipulation tasks.
6. Interactive learning and user feedback mechanisms: We plan to explore the application of the dataset in interactive learning frameworks for robots. Implementing user feedback mechanisms will allow assistive robots to optimize model performance in real time, adapting to individual user preferences and specific contextual requirements. This approach aims to enhance the adaptability and responsiveness of assistive robotic systems, ensuring they meet the diverse needs of users in dynamic settings.

7. Conclusions

This study addressed a critical gap in assistive robotics by creating a specialized dataset of manipulable activity of daily living (ADL) objects tailored for assistive robots. Recognizing the limitations of existing general-purpose datasets, we curated a focused collection encompassing 54 essential ADL objects. We ensured high-quality annotations and image integrity suitable for training deep learning models through meticulous data collection and rigorous filtering—both automated using a pre-trained YOLOv5x model and manual verification. Our dataset stands as a significant contribution to the field, providing a valuable resource that enhances the perceptual capabilities of assistive robots. By enabling more accurate object detection and localization, it facilitates the development of semi-autonomous control systems that reduce user cognitive load and increase independence for individuals with mobility impairments. The dataset's alignment with real-world

assistive tasks ensures that models trained on it are better equipped to operate reliably in home environments.

The implications of this work extend beyond immediate applications, laying the groundwork for future research in human–robot interaction, contextual understanding, and personalized assistive technologies. By making the dataset publicly available, we invite the research community to build upon our efforts, fostering innovation that can lead to more advanced, user-friendly assistive robots. In conclusion, our specialized ADL dataset fills a crucial void in assistive robotics and opens avenues for significant advancements in deep learning applications within this domain. It represents a step forward in empowering individuals with disabilities, contributing to enhanced autonomy and improved quality of life.

Author Contributions: Conceptualization, methodology, investigation, validation, formal analysis, data curation, visualization, writing—original draft preparation, M.T.S.; resources, supervision, project administration, M.H.R.; writing—review and editing M.T.S. and M.H.R. All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work partially supported by the National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR grant numbers 90DPGE0018-01-00 and 90DPKT0015-01-00) and Northwestern Mutual Data Science Institute (NMDSI) Student Scholar Program. NIDILRR is a Center within the Administration for Community Living (ACL), Department of Health and Human Services (HHS). The contents of this paper do not necessarily represent the policy of NIDILRR, ACL, HHS, or NMDSI, and you should not assume endorsement by the Federal Government.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Our dataset is released under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. This license allows for sharing, adaptation, and distribution of the dataset, provided appropriate credit is given to the original creators. The dataset can be accessed from the following link: https://doi.org/10.6084/m9.figshare.27263424 (accessed on 24 November 2024). Full license terms are available at https://creativecommons.org/licenses/by/4.0/ (accessed on 24 November 2024)

Acknowledgments: We gratefully acknowledge the use of data from the following open-source datasets, which played a pivotal role in the development of our specialized ADL object dataset: COCO dataset: We extend our gratitude to the creators and contributors of the COCO dataset for making their images and annotations publicly available under the CC BY 4.0 license [5]. Open Images dataset: We thank the Open Images team for providing a comprehensive dataset of annotated images under the CC BY 4.0 license [6]. LVIS dataset: We appreciate the efforts of the LVIS dataset creators in releasing their extensive dataset under the CC BY 4.0 license [7]. Roboflow Universe: We acknowledge the Roboflow Universe community for sharing their datasets under the CC BY 4.0 license [20–56]. Their contributions have been invaluable to our research, and we thank them for their commitment to open science and data sharing. The license information for these datasets can be found at: https://creativecommons.org/licenses/by/4.0/ (accessed on 24 November 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADLs	Activities of daily living
COCO	Common Objects in Context
LVIS	Large Vocabulary Instance Segmentation
YOLO	You Only Look Once
YCB	Yale-CMU-Berkeley
RGB-D	Red, green, and blue plus depth

References

1. Disability Impacts All of Us Infographic|CDC. 2024. Available online: https://www.cdc.gov/ncbddd/disabilityandhealth/infographic-disability-impacts-all.html (accessed on 9 October 2024).
2. The University of California—Disability Statistics Center Mobility Device Statistics: United States. Disabled World. 2022. Available online: https://www.disabled-world.com/disability/statistics/mobility-stats.php (accessed on 9 October 2024).
3. Keylabs. Unveiling the Power of Deep Learning in AI Image Recognition. *Keylabs: Latest News and Updates*. 2024. Available online: https://keylabs.ai/blog/the-brains-behind-the-eyes-unveiling-the-power-of-deep-learning-in-ai-image-recognition/ (accessed on 9 October 2024).
4. Sarker, I.H. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Comput. Sci.* **2021**, *2*, 420. [CrossRef] [PubMed]
5. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
6. Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *Int. J. Comput. Vis.* **2020**, *128*, 1956–1981. [CrossRef]
7. Gupta, A.; Dollar, P.; Girshick, R. Lvis: A dataset for large vocabulary instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5356–5364.
8. Roboflow Universe: Open Source Computer Vision Community. 2024. Available online: https://universe.roboflow.com/ (accessed on 9 October 2024).
9. Vasudevan, V.; Caine, B.; Gontijo Lopes, R.; Fridovich-Keil, S.; Roelofs, R. When does dough become a bagel? analyzing the remaining mistakes on imagenet. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 6720–6734.
10. YOLO Darknet TXT Annotation Format, 2024. Available online: https://roboflow.com/formats/yolo-darknet-txt (accessed on 9 October 2024).
11. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Yifu, Z.; Wong, C.; Montes, D.; et al. ultralytics/yolov5: V7. 0-yolov5 Sota Realtime Instance Segmentation. *Zenodo* 2022. Available online: https://zenodo.org/records/7347926 (accessed on 9 October 2024).
12. Calli, B.; Singh, A.; Walsman, A.; Srinivasa, S.; Abbeel, P.; Dollar, A.M. The ycb object and model set: Towards common benchmarks for manipulation research. In Proceedings of the 2015 International Conference on Advanced Robotics (ICAR), Istanbul, Turkey, 27–31 July 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 510–517.
13. Lai, K.; Bo, L.; Ren, X.; Fox, D. A large-scale hierarchical multi-view rgb-d object dataset. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1817–1824.
14. Jiang, Y.; Moseson, S.; Saxena, A. Efficient grasping from rgbd images: Learning using a new rectangle representation. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 3304–3311.
15. Smarr, C.A.; Mitzner, T.L.; Beer, J.M.; Prakash, A.; Chen, T.L.; Kemp, C.C.; Rogers, W.A. Domestic robots for older adults: Attitudes, preferences, and potential. *Int. J. Soc. Robot.* **2014**, *6*, 229–247. [CrossRef] [PubMed]
16. Shahria, M.T.; Ghommam, J.; Fareh, R.; Rahman, M.H. Vision-Based Object Manipulation for Activities of Daily Living Assistance Using Assistive Robot. *Automation* **2024**, *5*, 68–89. [CrossRef]
17. Ruiz-del Solar, J.; Loncomilla, P.; Soto, N. A survey on deep learning methods for robot vision. *arXiv* **2018**, arXiv:1803.10862.
18. Mahler, J.; Pokorny, F.T.; Hou, B.; Roderick, M.; Laskey, M.; Aubry, M.; Kohlhoff, K.; Kröger, T.; Kuffner, J.; Goldberg, K. Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1957–1964.
19. Rahman, M.M.; Shahria, M.T.; Sunny, M.S.H.; Khan, M.M.R.; Islam, E.; Swapnil, A.A.Z.; Bedolla-Martínez, D.; Rahman, M.H. Development of a Three-Finger Adaptive Robotic Gripper to Assist Activities of Daily Living. *Designs* **2024**, *8*, 35. [CrossRef] [PubMed]
20. Work, U. AI Based Automatic Stationery Billing System Data Dataset. 2022. Available online: https://universe.roboflow.com/university-work/ai-based-automatic-stationery-billing-system-data (accessed on 11 October 2024).
21. Destruction, P.M. Pencilcase Dataset. 2023. Available online: https://universe.roboflow.com/project-mental-destruction/pencilcase-se7nb (accessed on 11 October 2024).
22. Destruction, P.M. Final Project Dataset. 2023. Available online: https://universe.roboflow.com/project-mental-destruction/final-project-wsuvj (accessed on 11 October 2024).
23. Personal. CSST106 Dataset. 2024. Available online: https://universe.roboflow.com/personal-pgkq6/csst106 (accessed on 11 October 2024).
24. New-Workspace-kubz3. Pencilcase Dataset. 2022. Available online: https://universe.roboflow.com/new-workspace-kubz3/pencilcase-s9ag9 (accessed on 11 October 2024).

25. Finespiralnotebook. Spiral Notebook Dataset. 2024. Available online: https://universe.roboflow.com/finespiralnotebook/spiral_notebook (accessed on 11 October 2024).
26. Dairymilk. Classmate Dataset. 2024. Available online: https://universe.roboflow.com/dairymilk/classmate (accessed on 11 October 2024).
27. Dziubatyi, M. Domace Zadanie Notebook Dataset. 2023. Available online: https://universe.roboflow.com/maksym-dziubatyi/domace-zadanie-notebook (accessed on 11 October 2024).
28. One. Stationery Dataset. 2024. Available online: https://universe.roboflow.com/one-vrmjr/stationery-mxtt2 (accessed on 11 October 2024).
29. jk001226. Liplip Dataset. 2024. Available online: https://universe.roboflow.com/jk001226/liplip (accessed on 11 October 2024).
30. jk001226. Lip Dataset. 2024. Available online: https://universe.roboflow.com/jk001226/lip-uteep (accessed on 11 October 2024).
31. Upwork5. Socks3 Dataset. 2022. Available online: https://universe.roboflow.com/upwork5/socks3 (accessed on 11 October 2024).
32. Book. DeskTableLamps Material Dataset. 2024. Available online: https://universe.roboflow.com/book-mxasl/desktablelamps-material-rjbgd (accessed on 11 October 2024).
33. Gary. Medicine Jar Dataset. 2024. Available online: https://universe.roboflow.com/gary-ofgwc/medicine-jar (accessed on 11 October 2024).
34. TEST. Kolmarbnh Dataset. 2023. Available online: https://universe.roboflow.com/test-wj4qi/kolmarbnh (accessed on 11 October 2024).
35. Tube. Tube Dataset. 2024. Available online: https://universe.roboflow.com/tube-nv2vt/tube-9ah9t (accessed on 11 October 2024).
36. Staj. Canned Goods Dataset. 2024. Available online: https://universe.roboflow.com/staj-2ipmz/canned-goods-isxbi (accessed on 11 October 2024).
37. Hussam, M. Wallet Dataset. 2024. Available online: https://universe.roboflow.com/mohamed-hussam-cq81o/wallet-sn9n2 (accessed on 14 October 2024).
38. Training, K. Perfume Dataset. 2022. Available online: https://universe.roboflow.com/kdigital-training/perfume (accessed on 14 October 2024).
39. Keyboards. Shoe-Walking Dataset. 2024. Available online: https://universe.roboflow.com/keyboards-tjtri/shoe-walking (accessed on 14 October 2024).
40. MOMO. Toilet Paper Dataset. 2024. Available online: https://universe.roboflow.com/momo-nutwk/toilet-paper-wehrw (accessed on 14 October 2024).
41. Project-zlrja. Toilet Paper Detection Dataset. 2024. Available online: https://universe.roboflow.com/project-zlrja/toilet-paper-detection (accessed on 14 October 2024).
42. Govorkov, Y. Highlighter Detection Dataset. 2023. Available online: https://universe.roboflow.com/yuriy-govorkov-j9qrv/highlighter_detection (accessed on 14 October 2024).
43. Stock. Plum Dataset. 2024. Available online: https://universe.roboflow.com/stock-qxdzf/plum-kdznw (accessed on 14 October 2024).
44. Ibnu. Avocado Dataset. 2024. Available online: https://universe.roboflow.com/ibnu-h3cda/avocado-g9fsl (accessed on 14 October 2024).
45. Molina, N. Detection Avocado Dataset. 2024. Available online: https://universe.roboflow.com/norberto-molina-zakki/detection-avocado (accessed on 14 October 2024).
46. VietNam Fruit in Lab. Peach Dataset. 2023. Available online: https://universe.roboflow.com/vietnam-fruit-in-lab/peach-ejdry (accessed on 14 October 2024).
47. kkabs Group. Tomato Detection 4 Dataset. 2023. Available online: https://universe.roboflow.com/kkabs-group-dkcni/tomato-detection-4 (accessed on 14 October 2024).
48. Money Detection. Tomato Checker Dataset. 2024. Available online: https://universe.roboflow.com/money-detection-xez0r/tomato-checker (accessed on 14 October 2024).
49. Ain Shams University. Smart Cam V1 Dataset. 2023. Available online: https://universe.roboflow.com/ain-shams-university-byja6/smart_cam_v1 (accessed on 14 October 2024).
50. SHEHAB EMAD. Keysdetection Dataset. 2023. Available online: https://universe.roboflow.com/shehab-emad-n2q9i/keysdetection (accessed on 14 October 2024).
51. Roads. Chips Dataset. 2024. Available online: https://universe.roboflow.com/roads-rvmaq/chips-a0us5 (accessed on 14 October 2024).
52. New-Workspace-Bgkzo. Object Dataset. 2021. Available online: https://universe.roboflow.com/new-workspace-bgkzo/object-eidim (accessed on 14 October 2024).
53. Watch, W. Wrist Watch Dataset. 2024. Available online: https://universe.roboflow.com/wrist-watch/wrist-watch-0l25c (accessed on 14 October 2024).
54. WYZUP. Milk Dataset. 2024. Available online: https://universe.roboflow.com/wyzup/milk-onbxt (accessed on 14 October 2024).

55. AussieStuff. Food Dataset. 2024. Available online: https://universe.roboflow.com/aussiestuff/food-al9wr (accessed on 14 October 2024).
56. Almukhametov, A. Pencils Color Dataset. 2023. Available online: https://universe.roboflow.com/almas-almukhametov-hs5jk/pencils-color (accessed on 14 October 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Efficacy of Segmentation for Hyperspectral Target Detection

Yoram Furth * and Stanley R. Rotman

Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva blvd 1, Beer-Sheva 84105, Israel; srotman@bgu.ac.il
* Correspondence: yoramf@post.bgu.ac.il

Abstract: Algorithms for detecting point targets in hyperspectral imaging commonly employ the spectral inverse covariance matrix to whiten inherent image noise. Since data cubes often lack stationarity, segmentation appears to be an attractive preprocessing operation. Surprisingly, the literature reports both successful and unsuccessful segmentation cases, with no clear explanations for these divergent outcomes. This paper elucidates the conditions under which segmentation might improve detector performance. Focusing on a representative algorithm and assuming a target additive model, the study examines all influential factors through theoretical analysis and extensive simulations. The findings offer fundamental insights and practical guidelines for characterizing segmented datasets, enabling a thorough evaluation of segmentation's utility for detector performance. They outline the range of target scenarios and parameters where segmentation may prove beneficial and help assess the potential impact of proposed segmentation strategies on detection outcomes.

Keywords: hyperspectral image; point target detection; Segmented Matched Filter; segmentation

1. Introduction

Assuming the presence of a subpixel target within a hyperspectral image, traditional algorithms frequently fail in detection despite matching the target spectral signature. These algorithms typically employ the inverse covariance matrix for noise reduction. Our inquiry is directed towards elucidating the specific conditions under which using image segmentation can enhance the detector performance.

Historically, there have been conflicting reports on how beneficial segmentation is for detection. For instance, the generally positive results of [1–3] contrast with the negative findings reported by Pieper et al. [4]. These inconsistencies raised questions about the true value of segmentation. However, recent studies [5–7] have demonstrated that segmentation preprocessing can significantly improve detection performance, emphasizing that its success often depends on specific methods and conditions. These findings, combined with the earlier conflicting results, highlight the need for a deeper understanding of the factors that influence segmentation's efficacy.

Focusing on the Normalized Segmented Matched Filter (NSMF) with a target additive model, this study explores segmentation's role from foundational concepts to practical applications. It determines the conditions under which segmentation is likely to enhance detection efficacy for a given algorithm and data properties. We demonstrate how various parameters—such as the target's strength, spectral signature, and the acceptable false alarm rate—impact segmentation's contribution. Building on these insights, we aim to establish a

predictive framework that outlines, for a given background, in which range of parameters segmentation is likely to improve detection.

In a previous study, we introduced the effect of the target's spectral signature on segmentation efficacy [8]. Here, we extend this analysis to consider all relevant factors influencing segmentation's efficacy in detection. We propose that the observed lack of significant improvement in some cases arises from intrinsic data properties and other detector-related factors. This study provides practical guidelines to predict what possible benefit can be gained if the detector uses the segmented data.

The introductory section outlines the foundational principles. Section 2 identifies key rules that explain how each factor simultaneously influences segmentation efficacy. Section 3 presents simulations and various tests on a standard dataset. Finally, Section 4 presents practical conclusions regarding the conditions under which segmentation improves detection efficacy.

1.1. Matched-Filter Target Detection

A standard algorithm for detecting small targets is the Matched Filter (MF), chosen in this study as the representative method for evaluating segmentation efficacy due to its widespread use and reliance on covariance-based whitening. Other algorithms, such as the Adaptive Cosine Estimator (ACE), also depend on covariance properties, suggesting that the findings here will generalize to ACE and similar methods. Ongoing research is testing this hypothesis.

Assuming the background data fit a multivariate normal distribution (MVN) once the background power has been mitigated and assuming an additive target model, it is essentially defined by $t^T\Phi^{-1}(x-m)$. For each pixel vector x, it subtracts the background m, whitens the residual by the inverse covariance matrix Φ, and matches the result to the desired target signature t. Using the Generalized Likelihood Ratio Test (GLRT), m and Φ are estimated from the data itself [9]. This provides a score to each pixel, related to the probability it contains a target. The target is then detected by testing the hypothesis that x contains only "background" against the hypothesis that it contains a target. In this study, we assume an additive target model of $x \coloneqq x + pt$, where p is the target power.

Consider the hypothesis that there is no target present in the dataset. While collecting statistics of such scores "without target" being present can be conducted directly on the original dataset (due to the overwhelming number of no-target pixels), robust evaluation for the "with target" hypothesis becomes possible using an implantation procedure, as explained in [10]. This procedure involves implanting desirable targets alternately into the dataset, enabling systematic analysis of algorithm performance across a range of controlled target-to-background interactions.

The distributions of scores for the two hypotheses ("with target" and "without target") typically overlap, making it challenging to define an optimal detection threshold that balances detection rates with false alarms. The ROC curve is frequently used to optimize this threshold, as it provides a graphical representation of the trade-off. Additionally, the ROC curve serves as a metric of algorithm quality, with higher curves indicating greater cumulative detections for a given false alarm rate.

These two domains derive two different thresholds' notations used in this paper: (i) "η", indicating the *decision threshold* in the distributions' domain, versus (ii) "th", indicating the number of false alarms in the integral domain, called the *error threshold*. The relationship between these thresholds is:

$$th = \int_{\eta}^{\infty} f_N(z)dz, \tag{1}$$

where $f_N(z)$ is the scores' distribution with no targets (N) present.

1.2. Normalized SMF

Since data are often not stationary and the variance varies throughout the data, estimating one global covariance from the whole image might be wrong [1]. On the other hand, calculating it locally is problematic due to insufficient statistics [11]. A common approach to address this challenge is segmentation: dividing the image into distinct areas based on the similarity of clustered points, estimating the covariance for each area individually, and applying the detector locally using the corresponding covariance. This results in two variations of the MF algorithm: a local version and a global version [12]:

$$SMF \triangleq t^T \Phi_s^{-1}(x-m),\ GMF \triangleq t^T \Phi_G^{-1}(x-m)\ , \tag{2}$$

where SMF and GMF refer to the segmented and the global version, respectively, Φ_s and Φ_G are the segmented and the global covariance, respectively, and the other parameters are as defined above.

This work focuses on the NSMF version, where each MF is normalized by its standard deviation. Additionally, its superior performance [8,10], it has the advantage of having its t component scaling invariant, as can be shown by rewriting the expression as follows:

$$NSMF \triangleq \frac{t^T \Phi_s^{-1}(x-m)}{\sqrt{t^T \Phi_s^{-1} t}} = \tilde{t}^T \overset{\sim}{\Phi}_s^{-1}(x-m);\ \tilde{t} = \frac{t}{|t|},\ \overset{\sim}{\Phi}_s = \Phi_s \cdot \sqrt{\tilde{t}^T \Phi_s^{-1} \tilde{t}}. \tag{3}$$

We can thus examine the target direction effects separately from its power. Therefore, in the sequel, the symbol t will assume a normalized vector.

1.3. Problem Statement and Prior Work

The effectiveness of segmentation in hyperspectral target detection remains a subject of ongoing debate, with outcomes often varying based on the target and data characteristics. For instance, in this study, segmentation applied to the same dataset and algorithm yielded divergent results: one target showed significant performance improvements with a higher NSMF ROC-Curve than the global one (NGMF), while another target showed no improvement despite identical conditions (Figure 1).

These observations suggest that segmentation efficacy is influenced not only by the data structure but also by other critical algorithmic factors. Indeed, the literature reflects this ambiguity: while segmentation sometimes improves detection [1–3], in other cases, it does not [4]. This paper builds on the foundational work of Ben Yakar [12] and provides an in-depth analysis of the simultaneous effects of key factors, including the target power (p), the covariance matrix (Φ), the target spectrum (t), and the error threshold (th). A comprehensive report, including mathematical analyses and experimental details, is available in the Supplementary Materials.

Segmentation of hyperspectral datasets has been an active area of research for both classification and target detection processes [13–17], driven by the non-stationary nature of hyperspectral imagery in both spatial and spectral domains. Recent studies closely aligned with our focus have demonstrated the benefits of segmentation preprocessing for subpixel target detection. For example, Liang et al. [5,6] employed segmentation across spectral and spatial regions to detect both real and implanted targets, showing improved detection probabilities (see Figures 7 and 8 in [5]). Similarly, Stalley [7] compared various segmentation methods, including k-means, Gaussian mixture models, and subspace clustering, demonstrating that all methods outperformed non-segmented detection algorithms. These

findings underscore the relevance and potential of segmentation preprocessing, further motivating its exploration in this study.

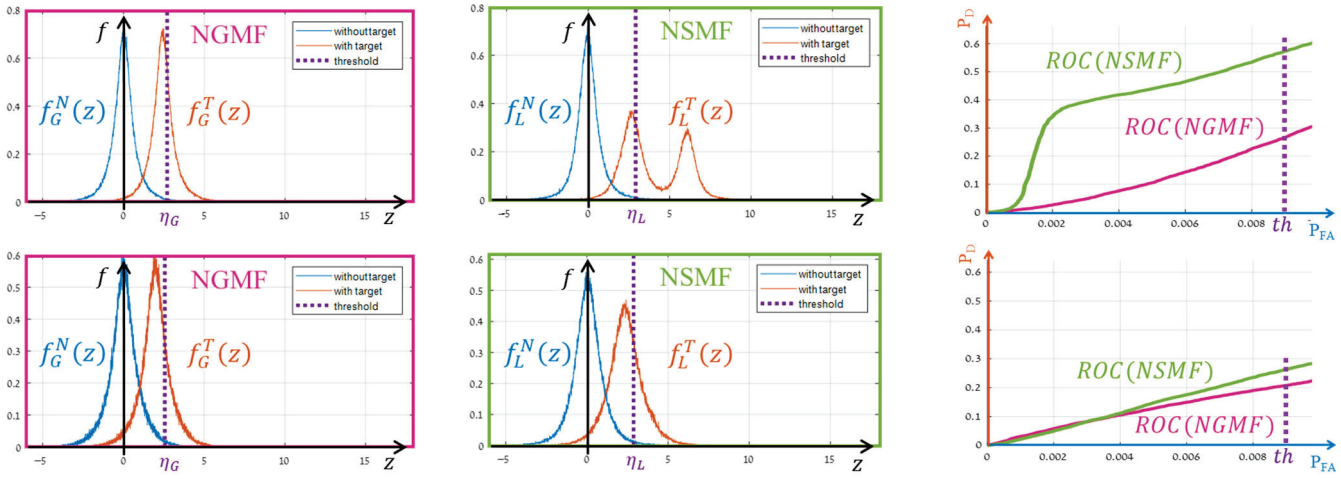


Figure 1. NSMF results on real data for two targets: the top row shows a real case of improved detection with segmentation, while the bottom row shows no improvement despite using the same data and segmentation. Column (**a**) displays global distributions, column (**b**) shows local distributions, and column (**c**) presents the ROC curves. Subscripts G /L denote global/local distributions, and superscripts N/T represent distributions without/with a target, respectively.

1.4. Evaluation Metrics

To quantify segmentation impact, we use two key metrics. The first metric is A_{th}, defined as:

$$A(th) \triangleq \frac{\int_0^{th} P_D(\tau)d\tau - 0.5\ th^2}{th - 0.5\ th^2}, \tag{4}$$

where P_D represents the probability of detection, and th denotes the error threshold. Introduced by Caefer et al. [10], this metric measures the area under the ROC curve as a function of the maximum false alarm rate. It produces a scalar value between 0 and 1, with higher values indicating better average ROC performance.

The second metric is B_{th}, defined as:

$$B(th) \triangleq \frac{A_L(th)}{A_G(th)}, \tag{5}$$

where A_G and A_L refer to A_{th} for a Matched Filter using a global or segmented estimated covariance matrix, respectively. Originally introduced by Ben-Yakar [12], this metric quantifies the improvement in detection performance attributable to segmentation. Higher values indicate greater benefits provided by segmentation.

1.5. Datacubes

This study employs a progressive reduction and deduction approach, starting from controlled synthetic scenarios and advancing towards fully real-world data. This methodology ensures robust and reliable conclusions applicable to practical hyperspectral target detection. This consists of three types of data, as depicted in Figure 2:

(a) Synthetic dataset: This dataset consists of two standard Gaussian distributions with two covariance matrices (Φ_1, Φ_2) in two-dimensional space and a target t with power p Each Gauss represents the spectral cross-section of the second moment of the residual noise ($x - m$) of the corresponding segment and has 3 degrees of freedom: aspect

ratio, scaling factor, and rotation angle. The target is represented as a 2D unit vector with two degrees of freedom—its magnitude and its rotation angle—corresponding to the target's power and spectrum, accordingly. This minimalistic setup efficiently covers the range of possible target-to-background interactions while avoiding the complexities of high-dimensional data. This dataset serves as the fundamental building block for comprehensive synthetic simulations.

(b) Simple In-house Cube (SIC): This 91-channel dataset contains stationary data with only two distinct areas. Its non-MVN distributions enable realistic simulations, bridging the gap between synthetic and real-world data. By manipulating the covariance matrices of two areas, we can simulate different types of inhomogeneity as interacted with any desirable target. This allows gradually increasing the complexity while retaining control over key parameters. The SIC serves as a baseline for exploring what can be practically deduced in more complex scenarios.

(c) RIT Cube: This hyperspectral dataset depicts a scene around Cooke City, Montana, provided by the Rochester Institute of Technology (RIT) [18]. It contains highly non-stationary data with a mix of natural and manmade materials. The data were collected using a high-resolution imaging spectrometer under controlled conditions, with details on the camera, lens, and weather provided in Snyder et al. [18]. Real-world data from the RIT Cube closes the gap entirely by addressing the final layer of complexity, including multiple segments with natural spatial variability and diverse noise characteristics. These complexities allow for testing the full applicability of the guidelines and insights derived from the simpler datasets.

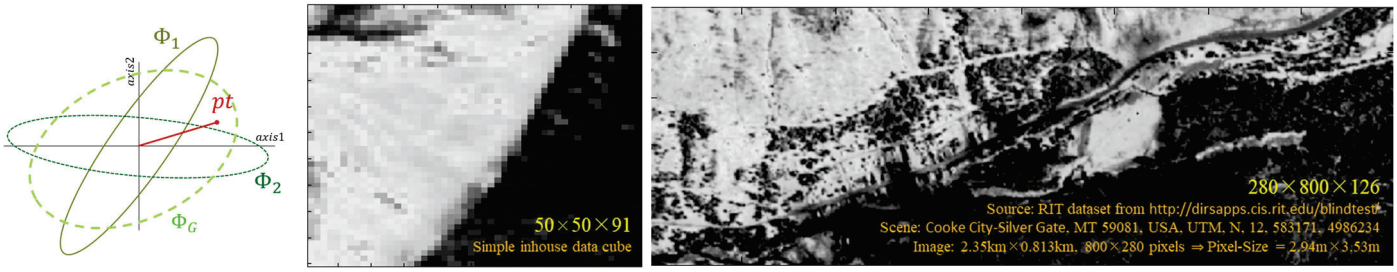


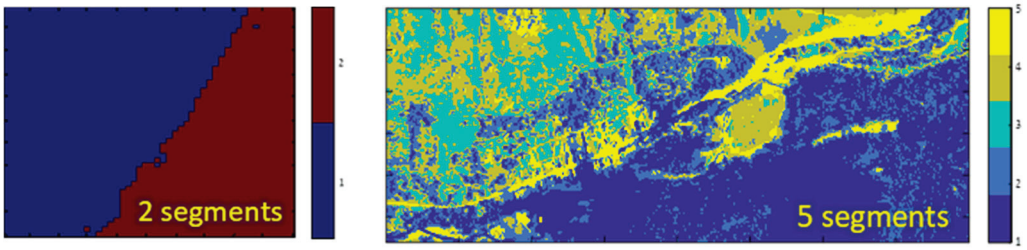
Figure 2. The three datasets used for this study: (**a**) synthetic data of Gaussian distributions, (**b**) one band from a Simple In-house Cube (SIC), and (**c**) one band from a data-cube, provided by Rochester Institute of Technology (RIT).

In addition, a second standard dataset, the Via Reggio dataset, provided by [19], was tested. This dataset features significantly different noise characteristics and environmental conditions and was used to validate the practical conclusions and guidelines presented in this paper. However, its results are omitted here due to redundancy.

A key aspect of the experimental framework in this study is the implantation of targets into datasets. This approach, described in Section 1.1, allows for systematic evaluation of algorithms under controlled conditions, enabling flexible testing of various target-to-background interactions. Unlike traditional methods relying on fixed, real-world targets, our strategy ensures that the interaction between targets and their backgrounds can be comprehensively analyzed. This methodology has been used and justified by other researchers [20].

For segmentation we utilized the K-Means algorithm [21], chosen for its simplicity and widespread use. This approach was sufficient for comprehensively studying the indirect effects of the various factors influencing segmentation efficacy. As a pixelwise classifier, K-Means does not account for spatial properties, focusing solely on spectral similarity.

Considering the trade-off regarding the number of segments [22], we divided the SIC dataset into two segments and the RIT dataset into five segments, as shown in Figure 3.



(**a**) (**b**)

Figure 3. Segmentation maps used for (**a**) SIC datacube and (**b**) RIT datacube. The labels' indexes correspond to the local covariances used in the sequel, that is, Φ_1 corresponds to segment #1, Φ_2 to segment #2, etc.

2. Factors of Influence

This section analyzes how each of the involved factors affects the segmentation success in the order of the algorithm (Equation (2)): it starts with the effect of p, the target power, added to the pixel assuming an additive model. It continues with the effect of Φ, the covariance matrix directly affected by the data inhomogeneity. It continues with the effect of t, the target direction in the spectral domain. It then ends with the effect of th, the error threshold in terms of the number of false alarms.

2.1. Influence of the Target Power

This section analyzes the influence of the target power on segmentation's efficacy.

2.1.1. Effect on the Distributions

The expectation of the Matched Filter without a target present is zero. But when a target is added, a bias occurs:

$$\mu_i = p \cdot \sqrt{t^T \Phi_i^{-1} t}, \tag{6}$$

where p and t refer to the target power and direction, respectively, and Φ_i refers to one of the global or local covariance matrices. Therefore, increasing the target power (p) increases the expectation proportionally, shifting the "with target" distributions to the right, providing more detections (P_D). Since this is true for any selected threshold, the performance (A) improves as well (see Equation (4)). Additionally, since Equation (6) holds for any kind of covariance, both the global and the local performance improve.

However, it is unclear what happens to the benefit (B), defined by the local to global performance ratio (Equation (5)), whether it remains constant or it grows in some way. Answering this requires delving into the role of p.

2.1.2. Statistical Influence of the Target Power

Figure 2a may represent a spectral cross-section of two segments' covariances, their composed covariance (Φ_G), and a target vector (t) with some power (p). As such, the illustration shows that as p increases, the Signal-to-Noise Ratio (SNR) increases relative to any covariance, whether global or local. Equation (7) shows that this increase happens proportionally regardless of the domain:

$$SNR \triangleq \frac{expected\ signal}{standard\ deviation} \underset{\uparrow (2)}{=} \frac{\mu}{\sigma} \underset{\uparrow (3)}{=} \mu \underset{\uparrow (6)}{\propto} p. \tag{7}$$

This insight leads to the intuition that there are two extremes and one intermediate state:

- When the target is weak, its SNR is bad even locally. There is, therefore, no point in segmenting since the resulting performance would remain bad in any case.
- When the target is strong, its SNR is already good globally, and segmentation is redundant.
- However, an intermediate p might exist, as in the example of Figure 2, where the SNR is still bad globally but already good locally. This case is where segmentation is worthwhile since it provides good local performance, which is, at the same time, significantly better than the global one.

Therefore, the response $B(p)$ is expected to be unimodal, low at the extrema, and maximal in the middle. Still, some characteristics of this shape are unclear, such as the location of the optimum, the maximal performance improvement, and the efficacy range of p values where segmentation is beneficial.

2.1.3. Range of Effective Target Powers

A synthetic simulation, illustrated in Figure 4, enabled characterizing the shape of $B(p)$. Increasing p gradually from a tiny value showed that although the performance improves in both domains, a gap occurring between the global (A_G) and local (A_L) performance function causes the benefit function (B), given by their ratio (Equation (5)), to become bell-shaped. Below, the origin of this gap is analyzed through the performance functions.

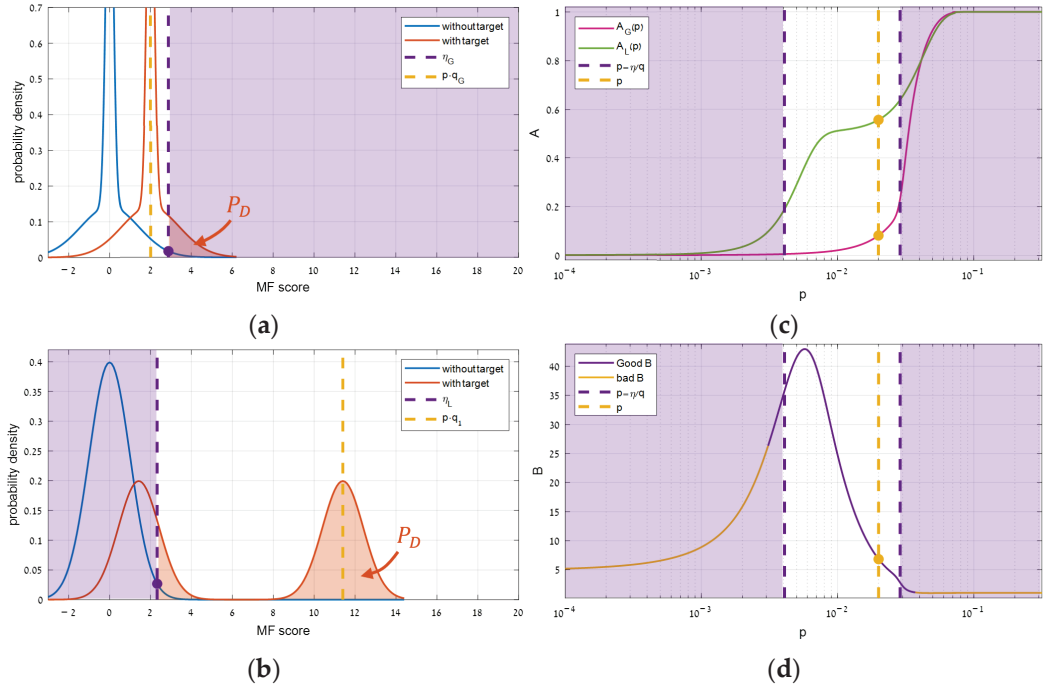


Figure 4. Range of effective target powers through different domains: (**a**) the global distributions, (**b**) the local distributions, (**c**) the respective performance, and (**d**) the resulted segmentation benefit. The purple grayed-out region represents the powers that are out of the efficacy range.

It can be seen in Figure 4c that the local A_L is the one that rises first, and therefore B initially rises (Figure 4d). This rising starts to become significant when the highest segment expectation in the local domain (Figure 4b) just crosses its decision threshold (η_L). This cross is related to the fastest increase in P_D, causing the local almost steepest ascent of A_L. Segmentation starts to be worthwhile.

As the target keeps getting stronger, B rises a bit more but reaches its maximum rapidly and begins to fall (Figure 4d). This fall happens due to its denominator A_G that is starting to grow, thanks to the global distributions in which detection is beginning to be feasible (Figure 4a).

Thereafter, the benefit B keeps decreasing until becoming so low that segmentation is no longer worthwhile. This decrease comes from the steep ascent of A_G (Figure 4c), originated in the global distribution (Figure 4a) that crosses its decision threshold (η_G).

This analysis reveals that the range of target powers in which segmentation might be effective is well defined by locating, both globally and locally, where the with-target expectation meets the decision threshold. We call these domain boundaries: p_1 after the leading local expectation and p_G after the global expectation. Substituting Equation (6) gives explicit expressions for these indicators:

$$p_1 \triangleq \frac{\eta_L}{\max_s \sqrt{t^T \Phi_s^{-1} t}}, \; p_G \triangleq \frac{\eta_G}{\sqrt{t^T \Phi_G^{-1} t}}. \tag{8}$$

These values could be viewed as anchors regarding p, since its impact on segmentation efficacy does not depend on its absolute value but only on its position relative to these values. As p is a multiplier, the degree of segmentation efficacy is indicated by the ratio of these anchors, that is,

$$\Delta p \triangleq \frac{p_G}{p_1}. \tag{9}$$

This index is called the "efficacy range's width".

2.1.4. The Optimal Target Power

It can be seen in Figure 4d above that the benefit function $B(p)$ becomes optimal shortly after p_1. Further experiments, summarized in Figure 5, show that this behavior is consistent for any type of data structure and that in the pure Gauss case, the optimal target power (p_{max}) satisfies approximately $p_{max} \approx \sqrt{2} \cdot p_1$.

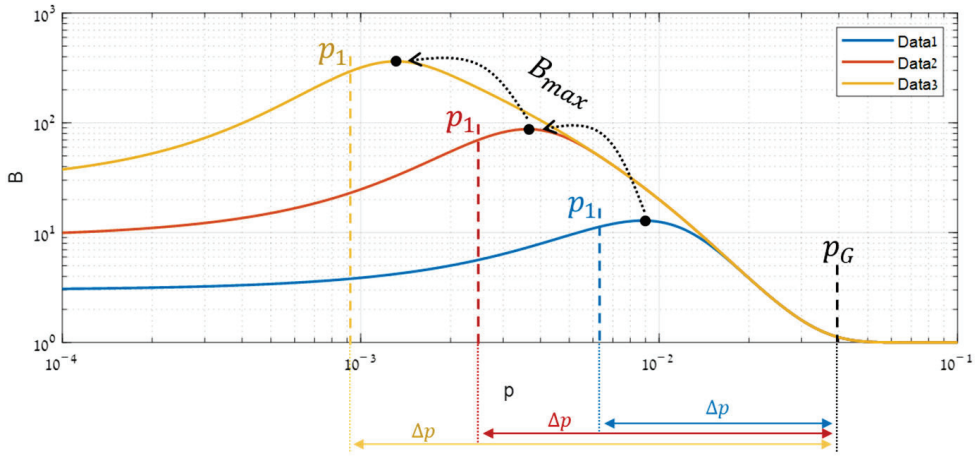


Figure 5. Evolution of $B(p)$ as a function of inhomogeneity. The global SNR is fixed, hence, p_G remains constant.

Figure 5 also shows that the maximal benefit (B_{max}) increases together with the efficacy range (Δp). Their connection can be explained mathematically. At the p_{max} point, A_L is always 0.5, so the major effect on B comes from A_G (Equation (5)). Since μ_G keeps decreasing, the numerator in Equation (4) tends to zero. Numerical approximations such as Taylor's can show that this tendency decreases proportionally to μ_G. Substituting

Equation (8) and the p_{max} approximation into Equation (6) gives $\mu_G \approx \sqrt{2} \cdot p_1 \frac{\eta_G}{p_G} \propto \frac{1}{\Delta p}$. Compiling all together gives:

$$B_{max} \propto \frac{1}{\mathrm{A_G}} \propto \frac{1}{\mu_G} \propto \Delta p. \tag{10}$$

The logic in this result is that the more the data structure challenges the detection due to its lack of homogeneity, the broader the range of targets to which segmentation can contribute. As such, relatively weaker targets become detectable, and segmentation brings a higher benefit.

2.1.5. Principles

The target power (p) affects the segmentation benefit (B) through the SNR. It has a unimodal response that has a range of efficacy bounded by well-defined p_1 and p_G anchors. The optimal target power goes with p_1 and the respective benefit is proportional to these anchors' ratio (Δp).

2.2. Influence of the Data Inhomogeneity

This section analyzes the influence of the covariance matrix (Φ), mainly affected by the degree of inhomogeneity of the data from which it is estimated.

2.2.1. A New Measure, "Kb"

The previous section concludes that the maximal segmentation benefit depends on the efficacy range (Δp), as defined in Equation (9). Substituting the anchors' explicit expressions from Equation (8) gives a two-part expression, with the error threshold affecting only the first part, where the covariances and the target direction affect mainly the second part:

$$\Delta p = \frac{\eta_G}{\eta_L} \cdot \frac{\max\limits_s \sqrt{t^T \Phi_s^{-1} t}}{\sqrt{t^T \Phi_G^{-1} t}} \Rightarrow K_b \triangleq \frac{\max\limits_s \sqrt{t^T \Phi_s^{-1} t}}{\sqrt{t^T \Phi_G^{-1} t}}. \tag{11}$$

The latter part is called K_b after "benefit factor" since this scalar factor has a proportional effect on Δp, hence also on the maximal benefit.

Therefore, when the data structure changes and the covariances layout is affected, the segmentation performance is influenced by how this change is reflected through that key factor K_b. Therefore, it is essential to understand the significance of this new intermediate factor.

2.2.2. Meaning of "Kb"

Substituting the Matched Filter expectation (Equation (6)) into K_b definition (Equation (11)) gives a ratio between the maximal local expectation and the global expectation. However, K_b might be better comprehended in the spectral domain where Equation (6)'s expression signifies "Mahalanobis Distance" along the target direction (t), as illustrated in Figure 6a:

Such a distance means "the length of the full target vector $p \cdot t$, relative to its intersection point with the covariance matrix's ellipsoid", a point denoted here as XS. Hence there exists an inverse relationship between this intersection point (XS) and the expectation (μ) above. This connection derives that the K_b expression can be rewritten as a ratio of intersection points:

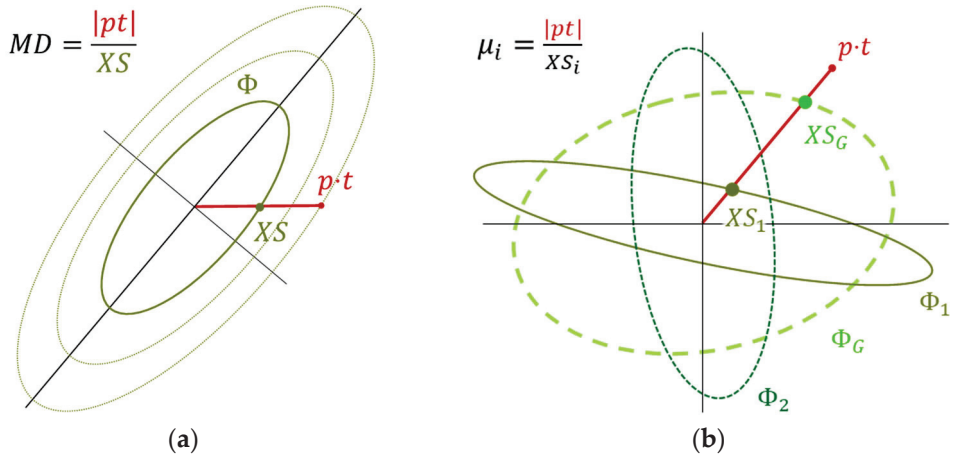


Figure 6. The spectral meaning of K_b: (**a**) expectation as a Mahalanobis Distance and (**b**) ratio of intersection points.

$$K_b = \frac{\max_s \mu_s}{\mu_G} = \frac{XS_G}{\min_s XS_s}. \tag{12}$$

This expression gives, in a layout like the one illustrated in Figure 6b: the intersection with the global Φ_G relative to the local Φ_1, or simply, XS_G/XS_1.

Two types of inhomogeneity are illustrated in Figure 7, with their corresponding K_b annotated. Type (a) is "scaling", where the segments' covariances differ just by a scalar factor, and type (b) is "angular", where the covariances differ by some planar rotation. In this illustration, despite the different inhomogeneity types, the K_b ratio obtains a completely identical value. However, this identity sustains only in the specific illustrated angle of t, where in other angles, the K_b ratio becomes different. This follows that K_b reflects the degree of the data inhomogeneity, though not in the general sense, but rather in how it acts along the specific target direction.

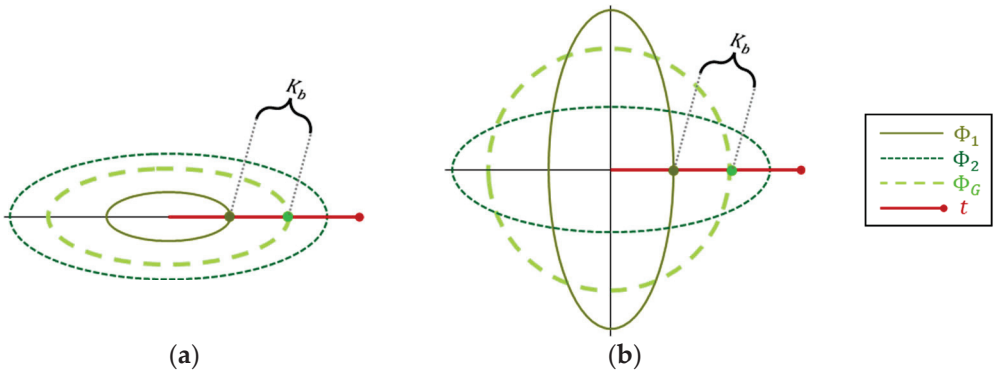


Figure 7. Comparing K_b on two kinds of inhomogeneities: (**a**) scaling and (**b**) angular.

2.2.3. Principles

A newly introduced factor called K_b was found to be a key in understanding the impact of variations in the data. That is since when the data structure varies, the covariances change by an influence that, to a specific target, is condensed into that scalar K_b, which has an immediate connection to Δp and affects the optimal benefit B_{max} proportionally.

This new factor is given by a closed-form expression, much easier to calculate than the benefit that involves complex optimization over integrations. This expression is equivalently a ratio of expectations or a ratio of intersection points, indicating the degree of "directional inhomogeneity" of the data. Moreover, this expression implies that inho-

mogeneity does not determine the segmentation impact in general but rather its highest possible impact, where the specific impact depends on the specific target of interest.

2.3. Influence of the Target Direction

This section analyzes the influence of t, the target direction in the spectral domain.

2.3.1. A Rule of Thumb: Theoretical Perspective

As discussed in Section 2.2.2, analyzing the target direction impact on the segmentation benefit is possible by tracking its effect on the intermediate factor K_b. Figure 8 below illustrates a case of angular inhomogeneity and a rotating target over some cross-section plane. The illustration shows that when t rotates, the ratio of the intersection points does not change significantly. Hence the maximal benefit B_{max} is not changing significantly as a function of the target direction.

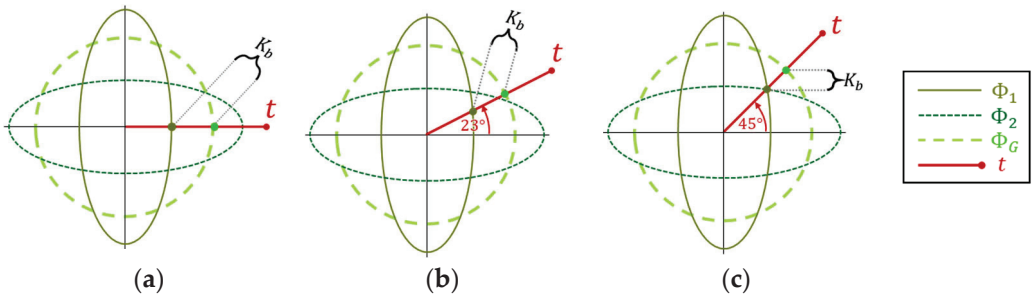


Figure 8. Angular inhomogeneity through a cross-section plane and a target at three angles: (**a**) 0°, (**b**) 23°, and (**c**) 45°.

However, this is not the only possibility. Additional examples are shown in Figure 9. Specifically, similar to case (a), where K_b is inherently rotation invariant, and case (b), which we have just analyzed, case (c) is also nearly invariant, as it represents a linear transformation of case (b); hence, proportions are preserved, and the K_b ratio remains intact. However, cases (d) and (e) are different. In case (d), the covariances also have a scaling difference on top of just rotation. This scaling causes the target to achieve a K_b ratio much smaller at 90° than at 0°. A similar phenomenon occurs in case (e), where only the major eigenvectors are rescaled.

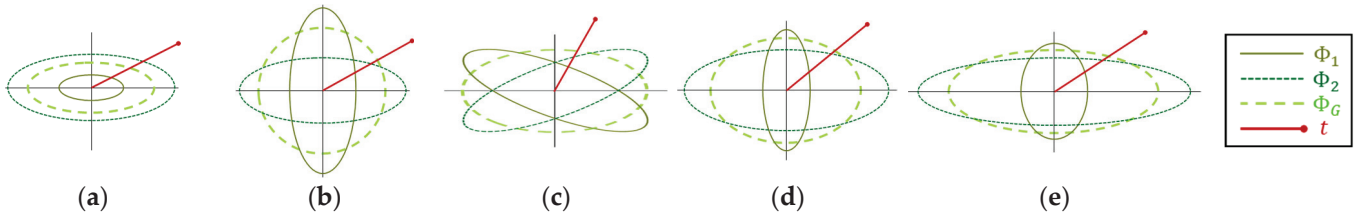


Figure 9. Target direction impact on different layouts: (**a**) scaling, (**b**) Φ_1 rotation; upon it, (**c**) appends system rotation and scaling, (**d**) appends rescaling of both ellipsoids, and (**e**) appends rescaling of the major axis of both ellipsoids.

This phenomenon reveals that it is likely to find the optimal target direction close to one of the strongest eigenvectors where the most significant gap between the global and the local intersections usually occurs.

2.3.2. Estimating the Optimal Direction

The rule of thumb valid on two segments is not necessarily true on multiple segments. Fortunately, however, an analytical method for finding such a direction has been discovered. Formally, the optimal direction should be around the maximal $K_b(t)$, that is, in the direction

where the highest ratio between XS_G and the minor XS_s occurs (Equation (12)). But since that minor value might jump from one local segment to another, it is easier to first focus on solving one single segment, Φ_1, as in Figure 10a, in which the remaining objective is to maximize XS_G/XS_1.

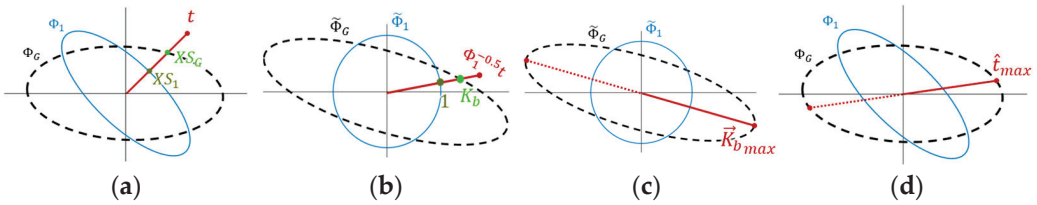


Figure 10. Analytic process of maximizing $K_b(t)$: (**a**) the objective with respect to one segment, (**b**) the effect of whitening, (**c**) the optimal K_b in the whitened domain, and (**d**) the equivalent in the original domain.

It turns out that, in this reduced case, whitening the system by Φ_1 provides an interesting result. As illustrated in Figure 10, the whitening transforms XS_1 to 1. In consequence, by Equation (12), XS_G becomes none other than K_b (Figure 10b). Since this property holds for any direction t, the whole ellipsoid of the whitened Φ_G represents K_b in all possible directions. The biggest K_b is therefore obtained where the largest radius occurs, which is simply on the major eigenvector (Figure 10c). All that remains, then, is to transform this vector back to the original domain, thus obtaining the optimal target direction in terms of K_b (Figure 10d). In the general case, this process is repeated for every segment, and the highest K_b is selected, as defined mathematically in the following equations:

$$\hat{t}_{\max} \triangleq \underset{t}{\operatorname{argmax}}\{K_b(t)\} = \Phi_{\hat{s}}^{0.5} \vec{K}^{(\hat{s})}_{bmax},\ \hat{s} = \underset{s}{\operatorname{argmax}}\left\{\left\|\vec{K}^{(s)}_{max}\right\|\right\},\ \vec{K}^{(s)}_{bmax} \triangleq \max_{\mathrm{t}} \vec{K}^{(s)}_{b} = \pm\mathrm{major}\left(\Phi_{\mathrm{s}}^{-0.5}\Phi_G\Phi_{\mathrm{s}}^{-0.5}\right). \quad (13)$$

It is worth noting that the proposed process relies solely on covariance matrices and involves highly efficient operations, making it computationally fast. Any optimal direction $\hat{t}_{\max}$ is only defined up to a sign ($\pm t$).

On real data, maximizing K_b provides only an estimation of the optimal direction. Where this is generally reasonable in predicting the maximal efficacy width (Δp), due to their direct relationship (Equation (11)), it is less accurate in predicting the maximal benefit. Not only that, the actual benefit at $\pm\hat{t}_{\max}$ may vary significantly due to asymmetries in the data structure, but the real maximal benefit (B_{max}) usually does not occur exactly there, as this estimation is based on the second moment and does not fully capture the real data distribution. Similarly, the range of effective target directions can be approximated by solving $\underset{t}{\arg} K_b(t) > \beta$ for a desirable threshold β.

2.3.3. Principles

The target direction t interacts with the covariances, affecting segmentation efficacy through the key factor K_b. The optimal direction can be estimated analytically by maximizing K_b, which gives a closed-form solution, easy-to-calculate. Likewise, the directions' efficacy range can be estimated by comparing K_b to some desirable threshold.

2.4. Influence of the Decision Threshold

This section analyzes the influence of th, the error threshold, in terms of the number of false alarms.

2.4.1. The Joint Impact of the Error Threshold

According to Equation (1), permitting fewer false alarms by a lower error threshold (th) derives a higher decision threshold (η). Since this property holds for both the global and the segmented MF, both p_1 and p_G anchors increase proportionally (Equation (8)). Hence, while the optimal power ($p_{\max}$), which goes with the first anchor, should increase (Section 2.1.4), the efficacy width (Δp), given by the ratio of these anchors (Equation (9)), is not expected to change significantly. However, the target direction (t) operates orthogonally, since its impact on the performance passes mainly through the K_b factor, which is independent of the thresholds (Equation (11)).

A synthetic simulation, shown in Figure 11, demonstrates the effect of reducing th threshold. Indeed, the optimal target power (p_{max}) shifts right along with both anchors, while the relative width (Δp) remains constant. Surprisingly, yet, along with them, the optimal benefit (B_{max}) increases, instead of remaining constant after its proportional relation to Δp (Equation (10)).

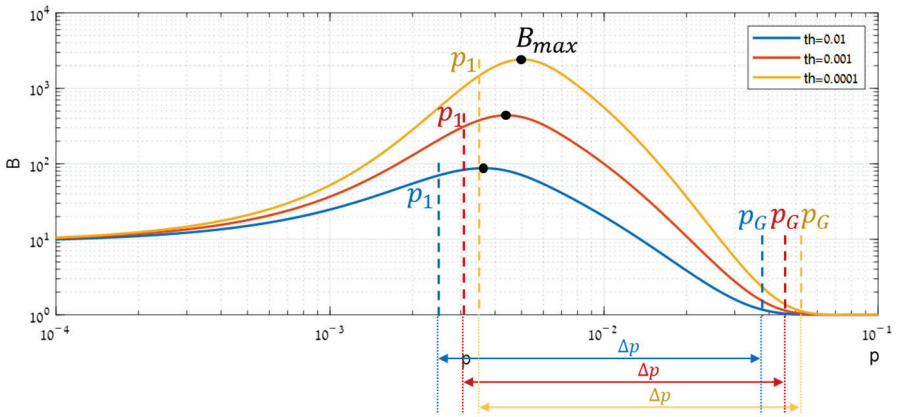


Figure 11. Evolution of $B(p)$ as a function of different thresholds.

2.4.2. Impact on the Maximal Benefit

Understanding the root of B_{max}'s rising was possible using a synthetic simulation, illustrated in Figure 12. In that simulation, th was reduced gradually from a high value, causing η to shift right in both the global and the local distributions. Simultaneously, for tracking the impact on B_{max}, the target power was adjusted as to sit on its optimal point (p_{max}). Since such an adjustment causes the highest local expectation to sit consistently just after the local threshold (Figure 12b), P_D barely changes, and the local performance (A_L) remains almost constant (Figure 12c). Contrarily, globally, while the threshold shifts similarly, the with-target distribution lags behind (Figure 12a), P_D decreases consistently, the global performance (A_G) drops monotonically (Figure 12c), and the benefit function grows endlessly (Figure 12d). Thus, unlike $B(p)$, which responds unimodally (Figure 4), th has a monotonic response.

Intuitively, this behavior indicates that as the global detection becomes more challenging due to a stricter threshold, the impact of segmentation becomes more pronounced. Nevertheless, it remains intriguing why the behavior differs so significantly between the global and local domains. After all, shifting a threshold is equivalent to inversely shifting expectations, so one would expect the change in the number of detections to change similarly in both domains.

Delving into the above theory reveals that due to Equation (1), reducing the error threshold shifts both the global and the local decision thresholds in similar proportions. Whereas, as a target power increases, each global or local expectation grows relative to the intersection point in its domain (Figure 6), whose ratio is nothing but K_b (Equation (12)).

The simulation above (Figure 12) used $K_b = 8$, which caused the global expectation to barely move, although the respective threshold kept shifting. Thus, unlike expectations, changing a target power is inequivalent to an opposite shift of the respective decision threshold. This insight means that the global domain, where the target is weaker relative to the reference noise, requires more relative efforts to improve the SNR.

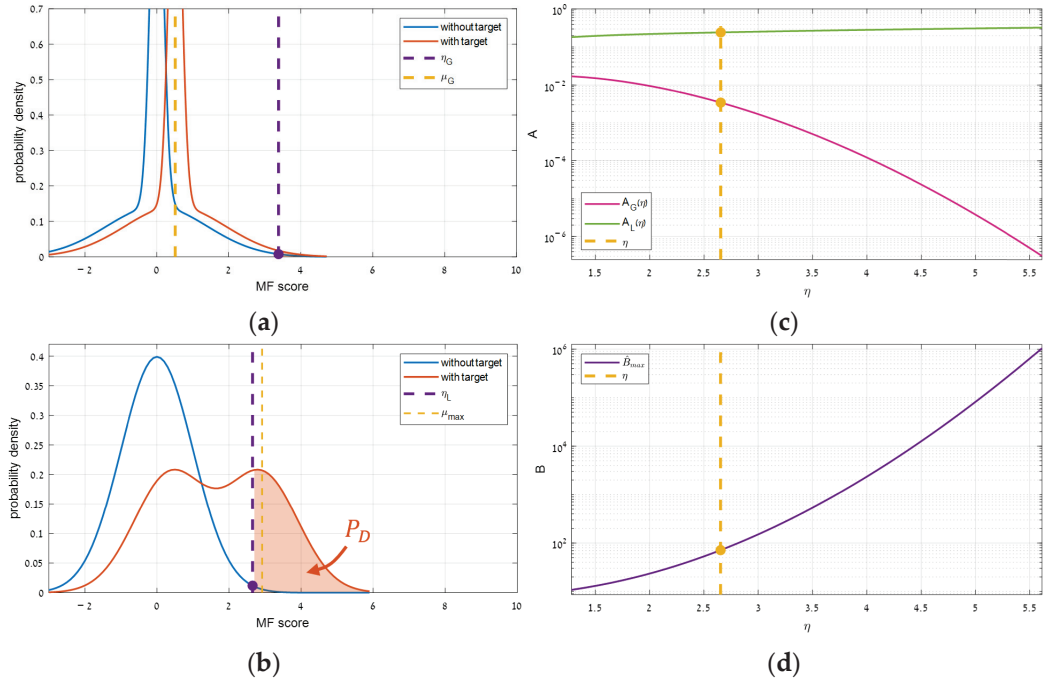


Figure 12. The influence of the error threshold in different domains: (**a**) the global distributions, (**b**) the local distributions, (**c**) the respective performance, and (**d**) the resulted segmentation benefit.

2.4.3. Impact per Domain

The definite trend of the optimal point can be extracted from mathematics. The optimal power ($p_{\max}$) goes with the first anchor (Section 2.1.4), which substituting Equation (9)'s components with Equation (11) and Equation (8) gives a proportion of η_L / K_b for any given global SNR. The corresponding benefit (B_{max}) evolves with A_G, due to the constant A_L (Figure 12c), hence it depends on an integration over P_D in the global domain (Equation (4)). Since P_D is based on the with-target distribution (f_T), which is a shifted version of the without-target distribution (f_N), its integration gives a similar response as P_{FA} which gives $0.5th^2$ [10]. This implies that A_G in Equation (4) is proportional to $th^2 / \left(th – 0.5th^2\right)$ which tends to th as th goes to zero. Thus, B_{max} tends to the inverse of th. Additionally, Equation (11) gives $\Delta p \propto K_b$ which, substituting into (10) gives $B_{max} \propto K_b$. Combining all together gives, asymptotically, the following proportions:

$$p_{max}\ , p_1 \propto \frac{\eta_L}{K_b},\ B_{max} \propto \frac{K_b}{th}. \tag{14}$$

Interestingly, the threshold effect is related to the domain of each factor: the target power factors vary with η that is in the scores' domain, where the optimal benefit varies with th which acts in the integral domain.

2.4.4. Principles

The more challenging the detection becomes due to a tighter threshold, the more the p_1 and p_G anchor points increase, as well as the position and intensity of the optimal benefit. The response to such a change is monotonic and tends to evolve proportionally, unlike the target power that responds unimodally. The threshold has a negligible effect on

the target's efficacy range and has an orthogonal effect to K_b and thus to any of the target direction factors.

2.5. Guidelines

This section analyzes how each of the NSMF factors affects the segmentation success:

- The target power (p) affects the efficacy of segmentation through the SNR. The segmentation's benefit varies unimodally with respect to two key anchors called p_1 and p_G. The optimal p is proportional to the small anchor (p_1), while the width of the efficacy range is determined by the ratio of p_G to p_1 (Equation (9)).
- The covariances (Φ_s), reflecting variations in the data structure, affect segmentation's efficacy through a scalar key factor named K_b. This factor qualifies "directional inhomogeneity" along the target direction and has a proportional influence on both the targets' efficacy range (Δt) and the maximal benefit (B_{max}).
- The target direction (t) interacts with the covariances, affecting efficacy through K_b factor. Maximizing K_b enables estimating the optimal direction using a closed-form expression that is computationally straightforward.
- Threshold (th): Tightening the threshold raises both p_1 and p_G anchors, resulting in a monotonic increase in the optimal segmentation benefit. Such a change triggers proportional relationships: the target power factors vary with the local decision threshold (η_L), while the optimal benefit varies with the error threshold (th).

These properties provide a framework for characterizing practical cases, allowing for informed decision-making regarding the range of targets and thresholds where segmentation is worthwhile in the detection process.

3. Experiments

This section presents the experimental results conducted to validate the theoretical conclusions. It begins by introducing the preparations made for these experiments. Subsequently, it delves into the influences of the data structure on the results. Finally, it explores the influence of the target signature on real datasets.

3.1. Experiments Considerations

For our experiments, we utilized the three types of data described in detail in Section 1.5: the Synthetic Dataset, the Simple In-House Cube (SIC), and real-world data, represented by the RIT Cube. Each dataset offered unique characteristics and challenges for experimentation. To ensure the reproducibility of our results, this section outlines the special preprocessing steps applied to prepare these datasets for experimentation.

Each of these three datasets posed unique preprocessing challenges: The Synthetic Dataset required adjustments to maintain analytical consistency. The SIC dataset involved targeted manipulations to introduce controlled inhomogeneities. Finally, the RIT Cube required addressing significant boundary estimation biases caused by the non-convexity of real-world segments.

The principal methods used to address these challenges are outlined below. A more detailed description is available in the comprehensive study report referenced in the Supplementary Materials.

3.1.1. Preprocessing

Several preprocessing steps were necessary for SIC data simulations. One critical step involved ensuring that data manipulations were based on a neutral starting point. First, we addressed the imbalance in segment sizes by cropping the image from 50×50 to 34×47, effectively reducing the populations imbalance from 40% to 0.2%. Next, we

addressed stationarity by balancing the segment covariances using a unique whitening transform, defined as $\Phi_G^{0.5}\Phi_s^{-0.5}$. This process whitens segment s by its own covariance and then unwhitens it by the global covariance. As a result, this manipulation effectively reduced the covariances' deviation from 5° to 0°.

The SIC data simulations were carried out over cross-sections of interest, for example, by the plane spanned by the major and minor Φ_G's eigenvectors. Such a simulation includes linear transformations for each segment across planar sections by transforming the data by VTV^T, where V is a basis, such as the eigenvectors matrix, and T is a desirable transform matrix. The latter is the identity matrix, excluding four cells determining a 2D transform over the two selected dimensions.

3.1.2. Practical Aspects

Each type of data had its challenges after its specific characteristics and its respective processing.

In some extreme cases, the synthetic simulations encountered numerical errors. To address this, a more accurate method was developed, incorporating three key principles: 1. maintaining pure analytic forms consistently, 2. employing standard numeric solutions for Gaussian distributions, and 3. applying approximation such as L'Hôpital's theorem.

In the SIC realistic simulations, the presence of long-tailed distributions impacting performance, coupled with data quantization leading to high volatility, posed challenges in deriving reliable conclusions. To address this issue, we compared each result with a corresponding synthetic simulation, enabling us to characterize distribution shape properties and meticulously isolate stochastic phenomena.

In RIT's standard dataset, background estimation errors occurring along the boundaries of segments introduced a notable bias in the per-segment average noise ($x - m$). This challenge was successfully resolved by developing a unique mirror padding technique, specifically designed for non-convex borders, as elaborated in Appendix A.

3.1.3. Consistency

Special attention was paid to isolating the primal sources of influence. While target power exclusively affects the SNR (Equation (7)), data structure and target direction simultaneously affect both homogeneity (Equation (11)) and SNR (Equation (6)). To understand their roles thoroughly, we fixed the SNR while varying the data structure and the target direction. This was obtained by consistently normalizing the vector t so that the global expectation (μ_G) remains constant.

3.2. Influence of the Data Structure

This section presents results regarding the influence of inhomogeneity due to variations in the data structure. It presents representative results for each of the three types of datacubes.

3.2.1. Synthetic Simulation

Using the synthetic dataset, we gradually rescaled two identical segments. In consequence, the inhomogeneity increased and K_b grew, respectively. Figure 13a shows the effect of K_b on the optimal target power (p_{max}), in the upper graph, and on the respective benefit (B_{max}), in the lower graph, both on the logarithmic scale. The results show that for a high enough K_b, p_{max} decreases proportionally and B_{max} increases proportionally. Additionally, as the error threshold (th) decreases, both p_{max} and B_{max} grow consistently.

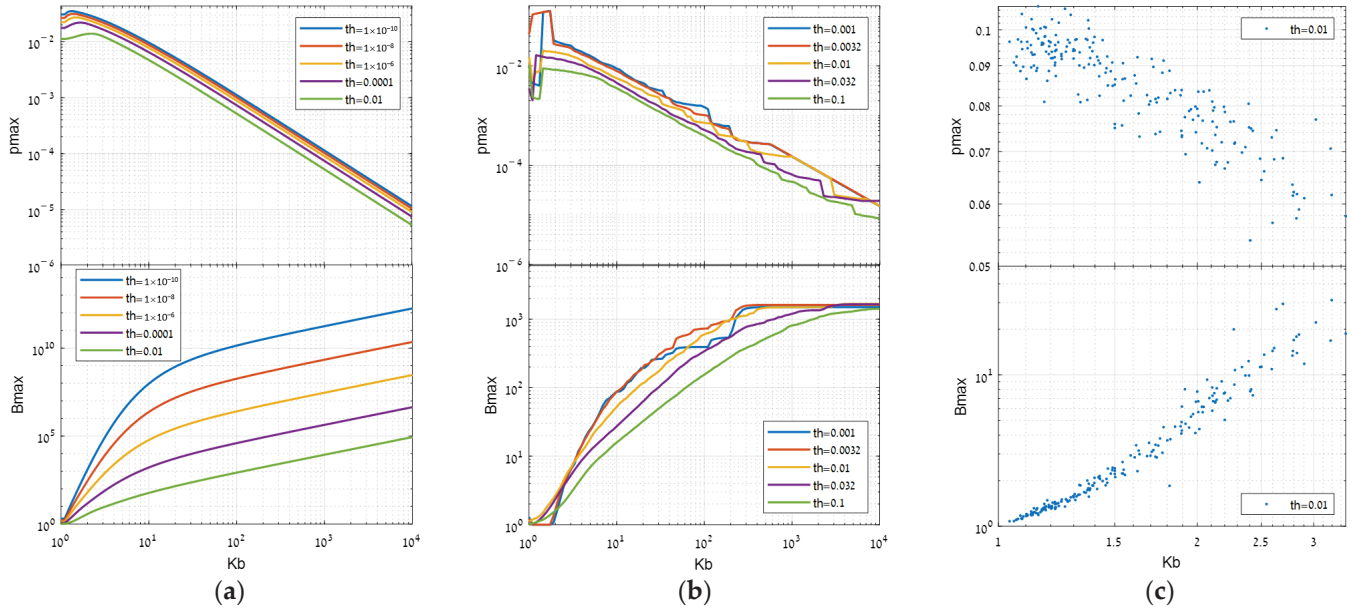


Figure 13. Influence of inhomogeneity for different thresholds on different types of datasets: (**a**) synthetic, (**b**) SIC cube, (**c**) RIT cube.

Although these results match the theoretical conclusions of Equation (14) asymptotically, it seems that at $K_b < 10$ the benefit grows faster. For example, at a threshold of 0.01, there is a quadratic growth rate of K_b^2. An alternative model that approximately satisfies the plots might be $B_{max} \approx H_{10}(K_b) \cdot K_b/10th + \overline{H}_{10}(K_b) \cdot th^{-\log K_b}$, where $H_{10}(\cdot)$ and $\overline{H}_{10}(\cdot)$ refer to a composition around 10, such as the Heaviside unit function $u(K_b - 10)$ and its complementary, respectively. However, such a model would only be applicable for data that follows normal distributions, which is not typically the case [11].

3.2.2. SIC Real Simulation

Using the SIC dataset, we repeated the simulation for real data. Starting with a purely stationary state, we gradually rescaled the two segments by some multiplicative factor so that K_b increases, regardless of the target direction. Figure 13b shows the effect on the optimal target power p_{max} and on the respective benefit.

The results closely resemble those in the synthetic case above, with the primary distinction being that B_{max} converges to a constant value. The origins of this difference can be traced back to quantization and the presence of long-tailed distributions in real data. In extreme K_b, the optimum (p_{max}) is obtained at a super weak target that shifts the global expectation to almost zero. Then, due to quantization, the number of detections becomes like the number of false alarms, the global performance stops falling, and the benefit (B_{max}) stabilizes at a fixed value.

3.2.3. RIT Standard Dataset

In the RIT standard dataset, the underlying structure is inherent. However, since K_b is related to the interaction with the target direction (Equation (11)), the data can be examined by selecting significant directions. We, therefore, chose the directions of the strongest eigenvectors from each of the five covariances of this data. For each direction, we computed K_b, determined the optimal p along with its respective benefit (B_{max}), and plotted it on scattergrams as shown in Figure 13c.

The results exhibit behavior consistent with the previous findings: as K_b rises, p_{max} drops, and B_{max} grows proportionally. The progression of B_{max} follows an approximately

quadratic rate in this specific case, as K_b is less than 10, and as we chose a threshold (th) of 0.01.

3.3. Influence of the Target Signature

This section presents results regarding the influence of the target signature on real datasets. It starts with a representative two-segment case. It then continues with a standard dataset and examines diverse types of target spectrum, striving to deduce ways to improve the performance further.

3.3.1. A Rule of Thumb: Practical Perspective

To test the target direction impact in the reduced two-segment case discussed above (Section 2.3.1), we used the SIC real dataset and a cross-section plane spanned by the major-to-minor Φ_G's eigenvectors (Section 3.1.1). Over that plane, we simulated angular inhomogeneity and rotated a target from 0° to 90° (Figure 14a). We then measured the optimal benefit obtained at each angle. The resulting upper graph in Figure 14c shows almost no rotation impact.

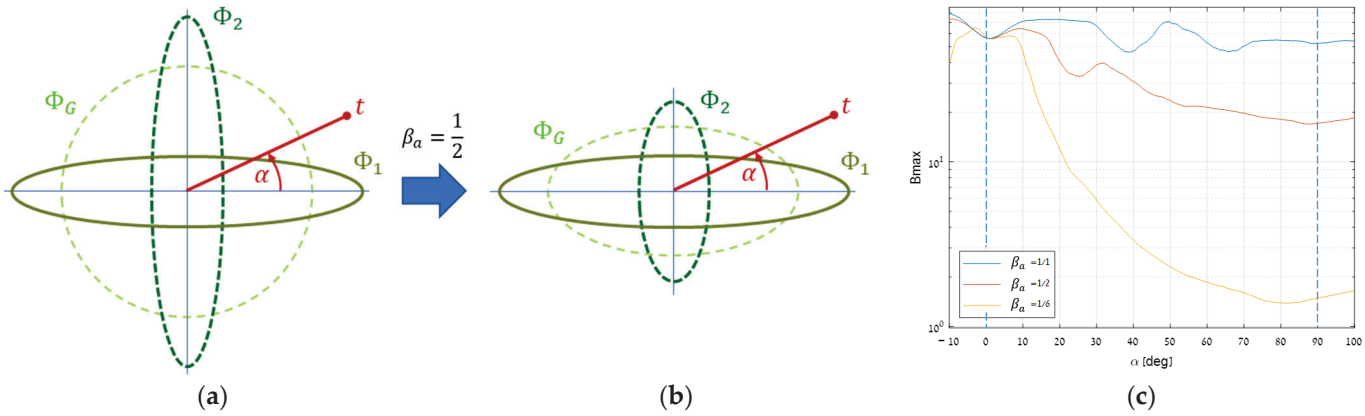


Figure 14. Influence of the target direction in the two-segment case on a representative layout: (**a**) the layout, (**b**) after reducing one major eigenvector by half, and (**c**) the resulting maximal benefit in three cases.

Thereafter, we gradually reduced the major eigenvector of Φ_2 and rechecked the rotation impact (Figure 14b). The two lower graphs in Figure 14c exhibit a decline as they approach 90°, in accordance with the decreasing factor. This decline happens due to the reduction in the ratio of K_b intersections as explained in Section 2.3.1. This reveals that since such asymmetries are common, it is likely to find the optimal target direction close to one of the most dominant eigenvectors.

3.3.2. Estimating the Global Optimum

To examine the impact of different target directions in the standard RIT dataset, we started with searching for the global optimal target (t_{max}). We first estimated this optimum by maximizing K_b using Equation (13). To get a sense of how good this estimate is, we crossed the spectral space with a plane spanned by the estimated $\hat{t}_{max}$ vector and Φ_5's major eigenvector (Figure 15a). Over this plane, we rotated a target and measured K_b and B_{max} per angle.

The results appear in Figure 15b. The lower graph shows K_b, which is indeed maximal at 0°, where $\hat{t}_{max}$ sits. The upper graph shows the actual B_{max}, whose true maximum is not much higher than its value at $\hat{t}_{max}$, but still, its position is somewhat shifted to the left. Fortunately, however, the curve between these two points looks convex, implying that the

global maximum might be achieved using standard optimization methods initialized at the estimated point.

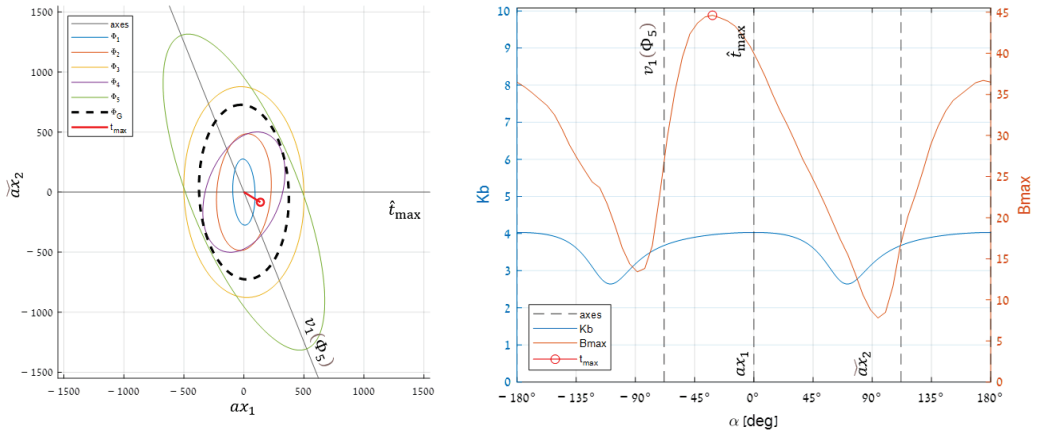


Figure 15. Segmentation performance for linear combinations of $\hat{t}_{max}$ with Φ_5's major eigenvector (v_1): (**a**) a planar cross-section for the selected pair and (**b**) segmentation performance for a range of angles.

Table 1 column (a) shows the performance values for the estimated best direction. For a 0.01 error threshold, segmentation improves performance from 0.006 to 0.252, which is 40.6 times better. By comparison, the best target found in Ben Yakar's work improved the performance by only 4.1 times (see Table 1 in [12]). Table 1 also shows that a smaller error threshold improves even more dramatically. So, our method seems to enable finding a direction with a benefit from the highest possible, which is also close enough to the absolute maximum.

Table 1. RIT cube performance obtained with the optimal target under different constraints: (a) no constraint, (b) within the positive cone, (c) belongs to the data, and (d) belongs to a given set of real targets.

Constraint	**(a)** t_{max}			**(b)** $t \geq 0$			**(c)** $t \in$ **data**			**(d)** $t \in$ **targets**		
th	**0.001**	**0.01**	**0.1**	**0.001**	**0.01**	**0.1**	**0.001**	**0.01**	**0.1**	**0.001**	**0.01**	**0.1**
K_b	4.03			3.92			2.91			1.63		
Δp	4.52	4.06	3.58	4.18	3.91	3	3.32	2.52	2.44	1.7	1.6	1.53
$p_{\max}$	0.104	0.05	0.018	0.17	0.06	0.016	0.22	0.05	0.02	34×10^{-4}	13×10^{-4}	6×10^{-4}
NGMF (A_G)	6×10^{-4}	0.006	0.039	6×10^{-4}	0.006	0.033	7×10^{-4}	0.012	0.084	0.005	0.037	0.149
NSMF (A_L)	0.273	0.252	0.241	0.257	0.218	0.201	0.224	0.154	0.202	0.135	0.123	0.208
Benefit (B)	453	40.6	6.11	451	36.4	6.1	334	13.2	2.42	26.8	3.32	1.4

3.3.3. Limiting the Optimum to the Positive Cone

Unfortunately, the just found global $\hat{t}_{max}$ cannot be a target spectrum since some of its components are negative. In Figure 16a, the area outside the positive cone is grayed out, showing that $\hat{t}_{max}$ remains outside, while Φ_4's eigenvector remains inside the cone. Figure 16b shows that the optimal direction along the rotation towards that eigenvector sits precisely on the cone border, making it relatively easy to find.

Table 1 column (b) shows that even under the positivity constraint, segmentation improves performance by 36.4 times, which is still remarkable. So, rotating the global best direction towards a target that sits within the valid area enables finding a direction that meets the required constraint but is still close enough to the absolute maximum.

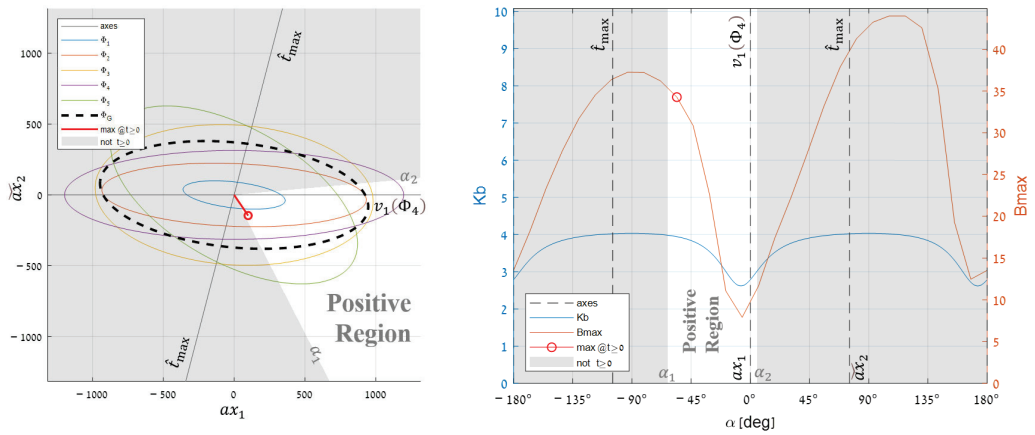


Figure 16. Segmentation optimum for constraint combinations of $\hat{t}_{max}$ with Φ_4's major eigenvector (v_1): (**a**) a planar cross-section for the selected pair and (**b**) segmentation performance for different angles.

3.3.4. Limiting the Optimum to the Image Pixels

Since forcing positivity is not enough to promise a valid signature, we additionally narrowed the constraint to the set of the image pixels so that the target would be undoubtedly valid. We searched for the pixel whose direction provides the maximal benefit among all the pixels. Figure 17a shows a map of all the pixels that give $K_b > 2$. We analyzed each of these pixels using our rules and found the pixel that offers the maximal benefit. This pixel is annotated "×" in three different views in Figure 17. Notably, this pixel belongs to segment #5, which boasts the highest benefit among the five segments, where the pixel itself is positioned on the border of that segment and visually differs from most other pixels within this segment.

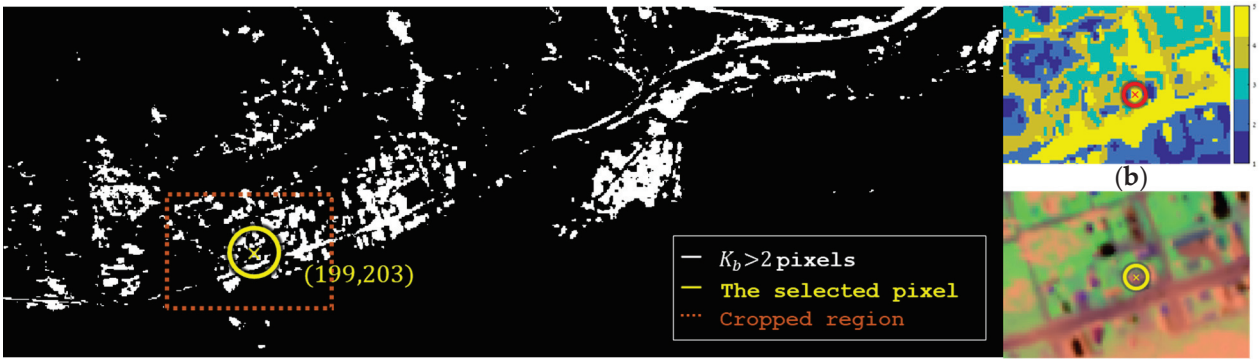


Figure 17. Several views with the best performing pixel, annotated "⊗": (**a**) a map of all $K_b > 2$ pixels, (**b**) the segmentation map cropped at (**a**)'s dashed region, and (**c**) a PCA-based visualization of the cropped datacube.

Table 1 column (c) shows that despite the extreme constraint, segmentation enhances performance by 13.2 times, which is still three times better than Ben-Yakar's best result [12]. Hence, one might get a sense of the maximal benefit of valid targets through image self-analysis of the kind presented here.

3.3.5. Efficacy with the Provided Targets

We further constrained the selected target by narrowing the possible set to only the twelve targets provided as part of the dataset. The best performer was a laboratory sampled blue cotton (F3). The result, shown in Table 1 column (d), gives that for a 0.01 error threshold,

segmentation improves by only three times. This result indicates that segmentation might be redundant for such targets.

Overall, the results in Table 1 show that although segmentation provides a high impact in some directions—while narrowing the set of valid targets, this impact drops until becoming negligible. This rapid degradation implies that the efficacy range of valid directions might be relatively tight.

3.3.6. Directions' Efficacy Range

One option for getting an impression of the efficacy range is sampling targets from the dataset itself. Figure 17a depicts such an example, showing a map of $K_b > 2$ pixels. It was found to be 97.5% accurate (measured by the relative number of true hits) equivalent to $\Delta p > 3.98$ and 95.6% accurate equivalent to $B_{max} > 3.98$, except that its computation lasts only 1.5 s instead of more than 100 h. Curiously, in this resulted map, the lit pixels occupy only 7.4% of the image. This result means that not many valid directions exist where segmentation is beneficial for this specific dataset.

One of the reasons for the limited domain of targets is the high sensitivity to the exact direction. It can be shown by comparing Φ_5's major eigenvector to the pixel with the most correlated signature, which, despite the considerable similarity (Figure 18a), tracking their cross-section reveals that B_{max} drops from 17 to 4.7 in just 2.3° (Figure 18b). This phenomenon might be explained by the above insight that K_b, from which B_{max} evolves, is represented by the radius of a multidimensional ellipsoid in the whitened domain ($\tilde{\Phi}_G$ in Figure 10c). It turns out that, like in the data [1], the first eigenvectors of this ellipsoid are relatively strong and then weaken rapidly. Therefore, when the target rotates between strong eigenvectors, K_b barely changes. Whereas, when it rotates towards most of the other weak dimensions, then a rapid degradation occurs. This behavior implies that the efficacy range of target directions is built from a manifold wide in few dimensions but narrow in most others.

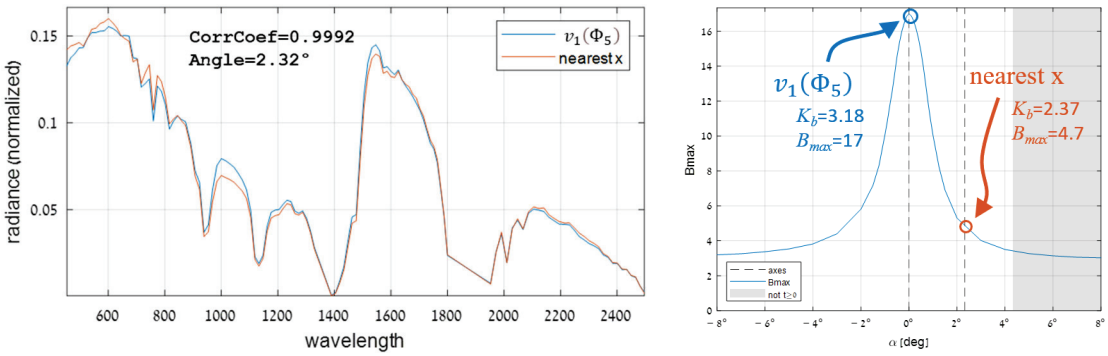


Figure 18. The performance of Φ_5's major eigenvector compared to its most similar pixel: (**a**) comparison of their two signatures and (**b**) performance analysis.

3.4. Efficacy with RIT Data

According to the results, discerning the worthiness of segmentation for the RIT dataset poses an initial challenge. On one hand, the observed benefits are pronounced in a limited range of targets; however, on the other hand, this range broadens as the required threshold diminishes, amplifying the overall segmentation's benefit. Yet, a pivotal consideration in this example may stem from the utilization of a basic "K-Means" segmentation approach, which might not inherently prioritize the creation of more homogeneous areas, contrary to our initial objective. Therefore, a transition to a more suitable technique designed to

enhance stationarity might potentially result in a substantial enhancement to the detection across a broader range of targets, even under stringent threshold conditions.

4. Conclusions

This study aimed to identify the conditions under which segmentation improves the detection of subpixel targets in hyperspectral images. While data inhomogeneity was initially hypothesized as the primary factor, our comprehensive analysis revealed that multiple algorithmic factors also significantly impact detection performance.

Our analysis of the NSMF algorithm identified the matched target as a critical factor in segmentation performance, specifically through its influence on "directional inhomogeneity". We introduced a novel factor, "K_b", representing the maximal benefit, and developed a closed-form estimator for determining the optimal target direction. These findings were distilled into practical rules describing the influence of each factor on segmentation performance.

We validated these insights through extensive simulations across a broad range of conditions. Additionally, we applied these findings to two standard datasets, evaluating segmentation performance for various targets and characterizing the conditions under which segmentation is beneficial. This enabled deducing properties about the setup and proposing ways to enhance the detection. Further investigations into additional datasets and scenarios are ongoing.

In contrast to prior studies that primarily demonstrated segmentation's potential for improving detection performance [5–7], this work focused on dissecting the underlying factors influencing segmentation efficacy. By analyzing these factors systematically, this study bridges the gap between observed benefits and the conditions that enable them.

Additionally, the theoretical insights, this work introduces practical analysis tools that can support future research and real-world applications, including:

- p-rules: Essential for characterizing segmentation's effects on detection performance.
- K_b: A key factor for efficiently predicting segmentation benefit for any target.
- $\hat{t}_{max}$: A closed-form estimator for the optimal target direction, easily computable.

These tools can be adapted for use with different models and algorithms. Unlike our prior work [8], which focused only on the influence of the matched target, this study provides an in-depth analysis of all contributing factors, supported by theoretical proofs and comprehensive experimental validations.

This study lays the groundwork for completing prior research on predicting segmentation benefits a priori. While an earlier study proposed linking data inhomogeneity to segmentation efficacy, we lacked a practical framework due to the complexity of segmentation-dependent factors. The $K_{b_{max}}$ metric introduced here simplifies this challenge by providing a single, interpretable measure of inhomogeneity for any given segmentation. This enables systematic comparisons with segmentation-independent inhomogeneity metrics, allowing evaluation of their ability to predict segmentation's potential benefit.

This study focused on the Normalized Segmented Matched Filter (NSMF) algorithm to analyze segmentation efficacy. Ongoing research is testing whether segmentation is similarly effective in other algorithms, such as the Adaptive Cosine Estimator (ACE), which also relies on covariance-based whitening. Additionally, future work will explore alternative segmentation strategies, including hierarchical clustering, to assess their impact on the conclusions drawn here. These efforts aim to validate and expand the applicability of the proposed framework across different detection algorithms and segmentation methods.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/s25010272/s1, Related code and data are available at: https://github.com/yoramfurth/spie2022_figures, accessed on 1 January 2025. Supplementary documentation of the research can be accessed at: https://drive.google.com/drive/folders/1AnwbHInJEoEpGemREbNQFz3s9-USge6c, accessed on 1 January 2025.

Author Contributions: Conceptualization, Y.F. and S.R.R.; Methodology, Y.F.; Software, Y.F.; Validation, Y.F.; Formal analysis, Y.F.; Investigation, Y.F.; Resources, S.R.R.; Data curation, Y.F.; Writing—original draft, Y.F.; Writing—review & editing, Y.F. and S.R.R.; Visualization, Y.F.; Supervision, S.R.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The synthetic data used in this study are generated programmatically using deterministic parameters, and the code for their generation is provided in the repository associated with this article. The RIT dataset is publicly available at http://dirsapps.cis.rit.edu/blindtest/, while the Via Reggio dataset is available upon request from its authors. The SIC dataset, a cropped hyperspectral image, can be provided upon request from the authors. All results are reproducible using the provided code, with no intermediate data stored.

Acknowledgments: We would like to express our gratitude to the Rochester Institute of Technology (RIT) for providing the publicly accessible hyperspectral dataset used in this study. We also thank the authors of the Via Reggio dataset for granting us access to their valuable data.

Conflicts of Interest: The authors declare that they have no competing interests or conflicts of interest related to the content of this manuscript.

Appendix A. Unbiased Background Estimation

Estimating a covariance matrix, as in Equation (2), is typically performed by averaging $(x-m)(x-m)^T$ over a set of pixels [23]. Since m itself is estimated with a spatial filter [2,10], a bias might occur, resulting in a nonzero expectation of $(x-m)$. Appending a mirror padding to the image borders overcomes such a global bias algebraically. However, this does not hold while applying it per segment.

Swapping the view from local filtering to spreading a pixel's energy to its neighbors (Figure A1a) underlies the foundation of this solution and unveils a more fitting generalization. For a square frame, the regular mirror padding compensates for any missing contribution with a respective mirrored value (the arrows in Figure A1b). In arbitrary segments, however, non-convex borders suffer from an ambiguous definition of the mirroring operation and an unbalanced redistribution of the energy within the image. In the example of an 8-neighbors averaging filter [10], illustrated in Figure A1c, a naïve mirror padding cause x_j to contribute back 11 times instead of 8 times.

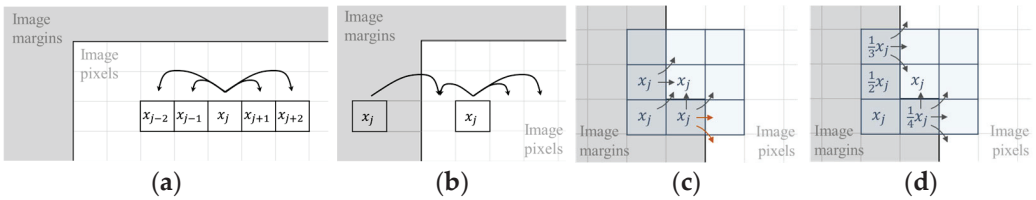


Figure A1. The concept of unbiased filtering: (**a**) the energy of a pixel x_j spreaded by a 5×1 ones filter, (**b**) the effect of mirror padding, (**c**) the origin of unbalanced contribution, and (**d**) a corrected contribution. The white area represents the set of active pixels to be padded, while the gray area represents the outer region where the padding occurs. The arrows represent the spread energy.

An alternative approach might be to mirror using an operation inverse to the following filtering so that the filtering will distribute the whole energy spread to the padded area back inwards. In the example of Figure A1d, each padded pixel contributes cumulatively exactly x_j, so that x_j is contributed precisely 8 times.

This idea can be generalized to mirroring the data using an adaptive flipped filter instead of simple direct padding. Each pixel on the margins would then get the weighted average of the pixels it sees within the image. If m is estimated by correlating x with a kernel h, then the mirror padding would be formed using the same kernel, flipped, and normalized by summing its in-segment elements. Formally, this means convolving $x \star h / (M_s \star h)$ per segment, where "/" denotes a pixel-wise divisor, "$\star$" denotes convolution, and M_s is a binary mask of segment number s.

References

1. Funk, C.; Theiler, J.; Roberts, D.A.; Borel, C.C. Clustering to Improve Matched Filter Detection of Weak Gas Plumes in Hyperspectral Thermal Imagery. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 1410–1420. [CrossRef]
2. West, J.E.; Messinger, D.W.; Schott, J.R. Comparative Evaluation of Background Characterization Techniques for Hyperspectral Unstructured Matched Filter Target Detection. *J. Appl. Remote Sens.* **2007**, *1*, 013520. [CrossRef]
3. Bajorski, P. Generalized detection fusion for hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2011**, *50*, 1199–1205. [CrossRef]
4. Pieper, M.; Manolakis, D.; Truslow, E.; Cooley, T.; Lipson, S. Performance Evaluation of Cluster-Based Hyperspectral Target Detection Algorithms. In Proceedings of the 19th IEEE International Conference on Image Processing, Orlando, FL, USA, 30 September–3 October 2012; pp. 2669–2672. [CrossRef]
5. Liang, Y.; Markopoulos, P.P.; Saber, E. Spatial–spectral segmentation of hyperspectral images for subpixel target detection. *SPIE J. Appl. Remote Sens.* **2019**, *13*, 036502. [CrossRef]
6. Liang, Y. Object Detection in High Resolution Aerial Images and Hyperspectral Remote Sensing Images. Ph.D. Thesis, Rochester Institute of Technology, New York, NY, USA, 2019.
7. Stalley, S.O. Clustered Hyperspectral Target Detection. Master's Thesis, Portland State University, Portland, OR, USA, 2020.
8. Furth, Y.; Rotman, S.R. Effective Segmentation for Point Target Detection. In Proceedings of the SPIE Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imaging XXIX, Orlando, FL, USA, 13 June 2023; pp. 227–236. [CrossRef]
9. Kelly, E.J. Performance of an Adaptive Detection Algorithm; Rejection of Unwanted Signals. *IEEE Trans. Aerosp. Electron. Syst.* **1989**, *25*, 122–133. [CrossRef]
10. Caefer, C.E.; Stefanou, M.S.; Nielsen, E.D.; Rizzuto, A.P.; Raviv, O.; Rotman, S.R. Analysis of False Alarm Distributions in the Development and Evaluation of Hyperspectral Point Target Detection Algorithms. *Opt. Eng.* **2007**, *46*, 076402. [CrossRef]
11. Manolakis, D.; Marden, D.; Shaw, G.A. Hyperspectral image processing for automatic target detection applications. *Linc. Lab. J.* **2003**, *14*, 79–116.
12. Ben-Yakar, S.; Blumberg, D.G.; Rotman, S.R. Advantages and limitations of segmentation for point target detection in hyperspectral imagery. In Proceedings of the 6th IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Lausanne, Switzerland, 24–27 June 2014; pp. 1–4. [CrossRef]
13. Dong, S.; Quan, Y.; Feng, W.; Dauphin, G.; Gao, L.; Xing, M. A pixel cluster CNN and spectral-spatial fusion algorithm for hyperspectral image classification with small-size training samples. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4101–4114. [CrossRef]
14. García, J.L.; Paoletti, M.E.; Jiménez, L.I.; Haut, J.M.; Plaza, A. Efficient semantic segmentation of hyperspectral images using adaptable rectangular convolution. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
15. Xie, F.; Wang, R.; Jin, C.; Wang, G. Hyperspectral image classification based on superpixel merging and broad learning system. *Photogramm. Rec.* **2024**, *39*, 435–456. [CrossRef]
16. Swarupa, V.V.S.; Devanathan, M. Hyperspectral Image Acquisition Methods and Processing Techniques Based on Traditional and Deep Learning Methodologies—A Study. In Proceedings of the IEEE Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 16–17 December 2022; pp. 1–13. [CrossRef]
17. Fuding, X.I.E.; Xu, L.I.; Dan, H.U.A.G.N.; Cui, J.I.N. Superpixel Merging-Based Hyperspectral Image Classification. *J. Syst. Sci. Math. Sci.* **2021**, *41*, 3268.
18. Snyder, D.; Kerekes, J.; Fairweather, I.; Crabtree, R.; Shive, J.; Hager, S. Development of a web-based application to evaluate target finding algorithms. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGRASS), Boston, MA, USA, 7–11 July 2008; Volume 2, pp. II-915–II-918. [CrossRef]
19. Acito, N.; Matteoli, S.; Rossi, A.; Diani, M.; Corsini, G. Hyperspectral airborne "Viareggio 2013 Trial" data collection for detection algorithm assessment. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 2365–2376. [CrossRef]

20. Theiler, J.; Matteoli, S.; Ziemann, A. Bayesian detection of solid subpixel targets. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Brussels, Belgium, 11–16 July 2021; pp. 3213–3216. [CrossRef]
21. Hartigan, J.A.; Wong, M.A. A K-Means Clustering Algorithm. *J. R. Stat. Soc. Ser. C-Appl. Stat.* **1979**, *28*, 100–108. [CrossRef]
22. Carlotto, M.J. A cluster-based approach for detecting man-made objects and changes in imagery. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 374–387. [CrossRef]
23. Kelly, E.J. An Adaptive Detection Algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **1986**, *22*, 115–127. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Scene Graph and Natural Language-Based Semantic Image Retrieval Using Vision Sensor Data

Jaehoon Kim and Byoung Chul Ko *

Department of Computer Engineering, Keimyung University, Daegu 42601, Republic of Korea; tyczend@gmail.com
* Correspondence: niceko@kmu.ac.kr; Tel.: +82-10-3559-4564

Abstract: Text-based image retrieval is one of the most common approaches for searching images acquired from vision sensors such as cameras. However, this method suffers from limitations in retrieval accuracy, particularly when the query contains limited information or involves previously unseen sentences. These challenges arise because keyword-based matching fails to adequately capture contextual and semantic meanings. To address these limitations, we propose a novel approach that transforms sentences and images into semantic graphs and scene graphs, enabling a quantitative comparison between them. Specifically, we utilize a graph neural network (GNN) to learn features of nodes and edges and generate graph embeddings, enabling image retrieval through natural language queries without relying on additional image metadata. We introduce a contrastive GNN-based framework that matches semantic graphs with scene graphs to retrieve semantically similar images. In addition, we incorporate a hard negative mining strategy, allowing the model to effectively learn from more challenging negative samples. The experimental results on the Visual Genome dataset show that the proposed method achieves a top nDCG@50 score of 0.745, improving retrieval performance by approximately 7.7 percentage points compared to random sampling with full graphs. This confirms that the model effectively retrieves semantically relevant images by structurally interpreting complex scenes.

Keywords: vision sensor; graph similarity learning; graph neural network; scene graph generation; semantic image retrieval; subgraph extraction

1. Introduction

Image retrieval using data acquired by vision sensors such as cameras is an active area of research in computer vision, with applications spanning search engines, medical image analysis, security and surveillance systems, and social media platforms. Traditional text-based image retrieval (TBIR) approaches utilize text annotations or tags associated with images to perform retrieval. However, these methods often fail to reflect the visual content of the images accurately and may result in the loss of important semantic information [1–3]. Pretrained large-scale vision–language models (VLMs) have recently been employed to generate textual descriptions from images for use in image retrieval. These generated captions are then used as inputs to retrieval models such as contrastive language–image pretraining (CLIP) [4] to perform image retrieval based on semantic similarity [5].

However, these methods face several challenges, including: (1) a strong dependence on captions generated by VLMs, (2) the risk of hallucinated or inaccurate captions, and (3) the lack of explicit modeling of inter-object relationships, all of which hinder accurate image retrieval.

To address the limitations of traditional TBIR approaches, content-based image retrieval (CBIR) has been proposed. However, CBIR methods also have limitations, as they rely solely on visual features and fail to capture object attributes and complex relationships among objects within scenes [6,7].

In this study, we propose a novel image retrieval system that transforms natural language queries into semantic graphs and evaluates their similarity to scene graphs constructed from an image database. While previous approaches have primarily relied on keyword matching or convolutional neural network (CNN)-based feature representations, our method leverages graph structures to more effectively capture inter-object relationships and contextual semantics, enabling a more precise similarity assessment.

As shown in Figure 1, the proposed framework retrieves semantically relevant images by generating a semantic graph from a natural language text query and matching it with a database of scene graphs extracted from images using a graph convolutional network (GCN). Scene graphs, in particular, serve as an effective tool for semantic representation, as they structurally encode complex image content into objects, attributes, and relationships. This structural abstraction enables more accurate similarity measurements compared to traditional keyword-based approaches.

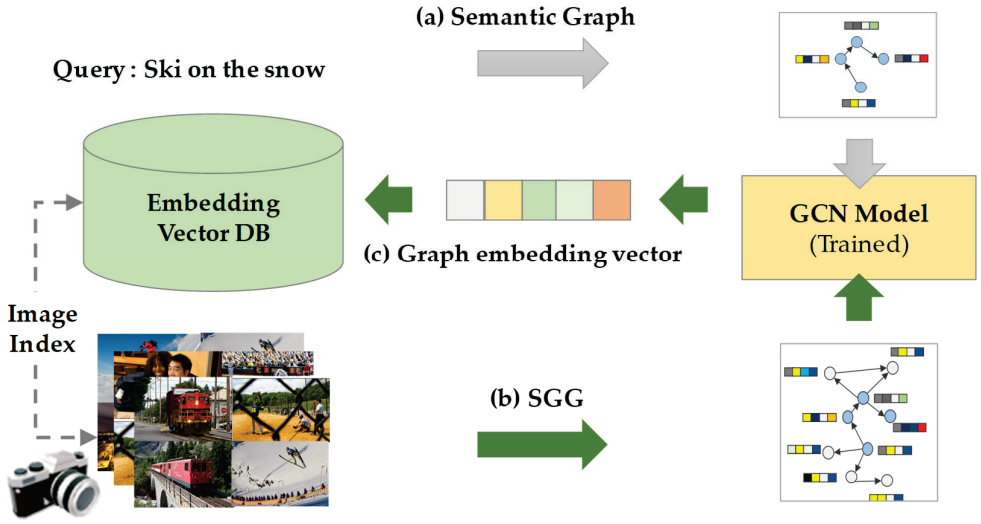


Figure 1. Overall architecture of the proposed semantic image retrieval system: (**a**) The input query sentence is first converted into a semantic graph; (**b**) scene graphs are generated from images acquired by various vision sensors through scene graph generation (SGG); (**c**) both the semantic graph and the scene graph are fed into a shared graph convolutional network (GCN), which encodes them into graph embedding vectors. These embeddings reside in a shared embedding space, allowing the system to quantitatively measure the similarity between text and image representations. The generated embedding vectors are stored in a database, enabling efficient retrieval.

For instance, a query such as “a person walking a dog” might only focus on the keywords “person” and “dog” in conventional keyword-based retrieval. In contrast, scene graph-based retrieval considers relational elements such as “walking” and “holding a leash”, leading to a more precise search outcome. Similarly, a query like “a person ordering food at a restaurant” may result in ambiguous matches when using simple keyword-based methods that emphasize only “person” and “food”. However, the use of scene graphs allows the system to capture relational semantics like “ordering food” and “interacting with a waiter”, thereby facilitating more semantically accurate retrieval.

Contribution of This Work

In this paper, we propose a graph-based image retrieval system that transforms text queries and images into semantic graphs, enabling the retrieval process to incorporate inter-object relationships. This approach allows more precise contextual understanding compared to traditional keyword-based retrieval methods. The main contributions of this paper are as follows:

- A novel similarity measurement method for heterogeneous graph structures: We propose a technique that transforms scene graphs extracted from images and semantic graphs generated from text into embedding representations that preserve both node and edge information. By leveraging a GCN, our method enables quantitative similarity computation between the two graph structures.
- Subgraph extraction from region-based scene graphs: Scene graphs generated from images often contain a large number of objects and relationships, which leads to increased computational complexity and potential interference from irrelevant information during training. To address this, we propose a region-based subgraph extraction method that selectively retains only semantically meaningful relationships around a central node, thereby reducing graph processing overhead. By employing a window-based depth-first search (DFS) strategy, the method mitigates central node bias and captures localized semantic structures. These localized subgraphs are then merged to construct a semantically focused summary of the global scene graph. Experimental results demonstrate that the dataset generated using this method contributes to improved retrieval performance.
- An effective training strategy for text-to-image retrieval: We adopt a contrastive GCN-based framework that matches semantic graphs with scene graphs to retrieve semantically similar images. To further enhance learning, we incorporate hard negative mining (HNM) [8], allowing the model to focus on more challenging negative samples. Unlike conventional negative sampling methods that select samples randomly, HNM selects hard negatives—samples the model is more likely to confuse—after an initial training phase. This approach improves the model's ability to distinguish subtle differences between similar images, leading to enhanced retrieval accuracy and better similarity evaluation performance relative to training time.

2. Related Work

2.1. Scene Graph Generation

A scene graph is a structured representation of visual information in the form of a graph, where nodes correspond to objects and their attributes, and edges represent relationships between objects. Scene graph generation (SGG) [9] refers to the process of analyzing an image to construct a scene graph, enabling the extraction of semantic data that captures complex inter-object relationships within the visual content.

2.2. Image Retrieval

Image retrieval has traditionally been categorized into two main approaches: TBIR and CBIR. TBIR relies on manually annotated textual metadata, such as tags or descriptions, to retrieve relevant images [1–4]. This method primarily depends on matching textual information rather than analyzing the visual characteristics of the image. However, it often fails to capture the actual visual semantics of the content, resulting in a potential loss of semantic information and limited retrieval performance.

CBIR retrieves images by leveraging the visual features extracted directly from the images themselves. Traditional CBIR methods rely on local features such as SIFT [10] and SURF [11] or utilize CNNs to extract visual representations for comparing query

inputs with candidate images [12]. Although CBIR captures visual similarity effectively, it often struggles to represent complex semantic content, thereby limiting retrieval accuracy. To address this limitation, recent studies have proposed the use of deep learning-based VLMs to bridge the semantic gap between images and textual queries [5,13,14]. Notably, models like CLIP [4] learn joint representations of images and text, enabling retrieval based on semantic similarity between visual content and textual descriptions. These models substantially improve retrieval performance by aligning images with their contextual meanings. However, these approaches still face critical challenges. Because VLM-based retrieval often relies on generated captions or textual representations, inaccurate captioning can degrade retrieval performance. Furthermore, hallucination effects—where models produce plausible but incorrect captions—may compromise the reliability of retrieval outcomes. In addition, the current VLMs often fail to fully capture object-level interactions and contextual relationships, making it difficult to accurately align the semantics of an image with a textual query. Collectively, these limitations hinder the ability to perform precise and trustworthy image retrieval.

2.3. Subgraph Extraction

Scene graphs are designed to represent holistic relationships within an image; however, they often include excessive information, which increases computational complexity and complicates the identification of relevant content. In particular, scene graphs generated through SGG tend to exhibit fully connected structures, linking nearly all possible object pairs. In such cases, GCN-based models face challenges in selectively learning meaningful relationships, as the dense connections dilute semantic relevance. Consequently, existing subgraph extraction methods struggle to isolate the core semantic structures from these densely connected graphs.

To address this issue, we propose a relationship-aware subgraph extraction method that leverages a DFS strategy to traverse the scene graph. By selectively excluding semantically irrelevant nodes, our method constructs localized and balanced subgraphs that retain the essential relational structure while reducing noise and redundancy. This approach enables more effective and focused representation learning within graph-based models.

3. Preliminaries

3.1. Natural Language Processing

To convert textual queries into a graph structure, we utilize the natural language processing (NLP) toolkit spaCy [15] to perform syntactic parsing of input sentences. Based on the results of dependency parsing, we construct a directed graph where each word is represented as a node, and syntactic relationships between words are modeled as edges with directionality. Each node is further initialized using global vectors for word representation (GloVe) embeddings [16], providing a numerical representation of the word's semantic meaning. This text graph construction serves as a preprocessing step to enable structural alignment with image scene graphs in subsequent stages of the framework.

3.2. Sentence-BERT (SBERT)

To quantitatively compare the semantic similarity between text and image graphs, we employ sentence-BERT (SBERT) [17] in our study. SBERT is a sentence embedding model built upon the bidirectional encoder representations from transformers architecture, specifically designed to generate fixed-length vector representations of sentences. These embeddings can be directly used for similarity computations such as cosine similarity or dot product.

In our framework, we utilize the sentence embeddings generated by SBERT to compute the semantic similarity between the semantic graph and the scene graph. This similarity score serves as a surrogate relevance signal [18], which is incorporated into the loss function during training. By integrating this measure, the GCN model is guided to learn more semantically aligned and meaningful representations.

3.3. SGG (Scene Graph Generation)

Semantic information can be effectively extracted from an image by representing object-level relationships in the form of a graph. To this end, we adopt the COOK model [19], which incorporates the concept of object co-occurrence into the graph structure. Scene graph generation enables the detection of objects, attributes, and relationships within an image, and represents them as a structured graph.

In our study, the scene graph is constructed as a directed graph, where all detected objects, their attributes, and inter-object relationships are represented as nodes. Each node is initialized with a 100-dimensional GloVe embedding, providing a dense vector representation of the corresponding word. This setup facilitates structured reasoning over the visual content and supports downstream tasks such as image–text alignment and semantic matching.

3.4. GCN (Graph Convolutional Network)

The GCN is a neural architecture designed to model relationships between nodes based on graph structures. Through convolution operations utilizing the adjacency matrix, GCNs iteratively update node representations by aggregating features from neighboring nodes. This process is grounded in spectral graph theory, enabling the network to effectively capture structurally connected semantic information. The learned node embeddings are typically aggregated using operations such as mean pooling to produce a fixed-size graph-level embedding, which can then be used to compute similarity scores via inner products or distance-based measures. The Siamese network [20], known for its effectiveness in measuring similarity between two inputs through shared weights, has also been widely adopted for comparing graph structures. Building on this concept, several graph neural network (GNN) architectures have been proposed to address the challenge of comparing graphs with different structures [21].

In this study, we propose using semantically focused subgraphs as input to the GCN, rather than processing the entire graph, to improve efficiency and relevance. By filtering out irrelevant information and concentrating on meaningful local structures, this approach enhances both the expressiveness of embeddings and the retrieval performance.

4. Semenatic Image Retrieal

We propose a novel framework that enables fine-grained semantic retrieval between textual queries and images by transforming both modalities into structured graph representations and comparing their structures using GCNs. While previous research [1–5,13,14] has predominantly focused on similarity retrieval between images, our approach extends semantic retrieval to align natural language queries with visual content. Instead of relying on global feature embeddings, as in traditional TBIR or CBIR, our method captures internal structures and relationships within queries and images through semantic graph representations. Although CLIP-based models [5,13,14] achieve strong performance through large-scale pretraining, our framework is designed to enable semantic retrieval without depending on CLIP, by focusing on direct structural matching. For evaluation purposes, however, CLIP is used as a reference point to compare retrieval performances.

Figure 2 illustrates the overall model training process, which leverages semantic scene graphs and contrastive GCN matching. As shown in Figure 2a, the retrieval query is text-based. The input sentence is processed using the natural language processing toolkit spaCy [15], which performs tokenization and extracts morphological, part-of-speech, and syntactic dependency information. For the image database, each image is automatically converted into a scene graph using a pretrained SGG model, as illustrated in Figure 2b. The SGG captures objects, their attributes, and inter-object relationships from the image, enabling the construction of structured semantic representations for image retrieval. Surrogate relevance [18], shown in Figure 2e, serves as an alternative metric to estimate semantic similarity without the need for manual human labeling. This is achieved using SBERT [17], which computes semantic similarity scores between image captions. These similarity scores are then used as alignment targets during the graph embedding learning process, guiding the model to produce semantically consistent retrieval results.

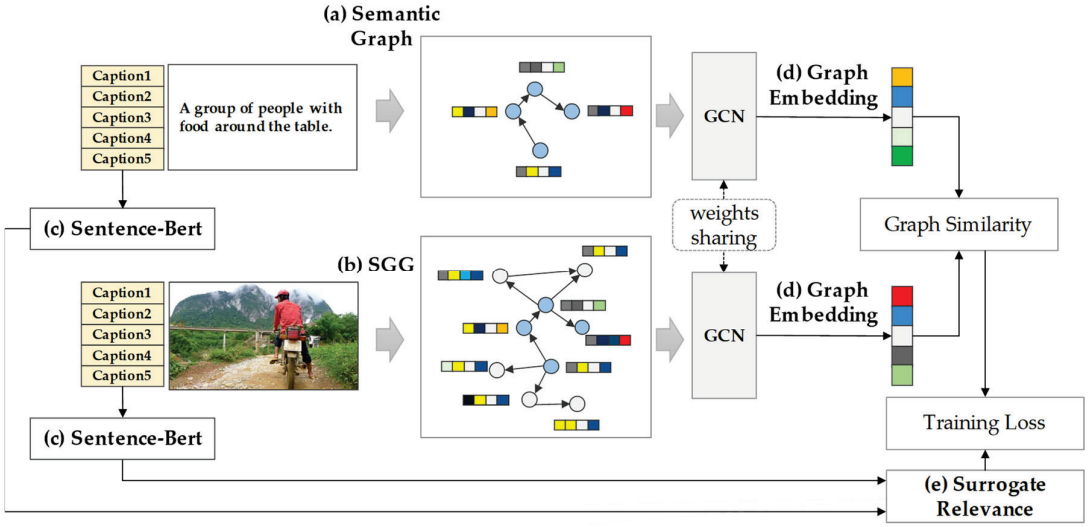


Figure 2. Training process using semantic scene graphs and contrastive GCN matching: (**a**) The search query is transformed into a semantic graph, (**b**) while the image is converted into a scene graph. (**c**) SBERT computes caption similarity without direct human labeling and quantifies the similarity between images. (**d**) The GCN model generates graph embedding and (**e**) uses the precomputed surrogate relevance for training.

When multiple natural language queries describe the same visual content, surrogate relevance scores help identify semantically similar sentence pairs, which are used to inform the graph-level contrastive learning objective. This process enhances the model's ability to learn fine-grained semantic similarity, ultimately improving retrieval performance.

4.1. Semantic Text Graph Construction

To convert a textual query into a graph structure, the input sentence is first tokenized into a sequence of words. Then, spaCy is used to analyze part-of-speech tags and dependency parsing information for each token, enabling the construction of a structured graph representation based on the syntactic relationships between words.

$$S = \left\{ w_1, w_2, \ldots, w_n \right\} \tag{1}$$

To normalize variations in word forms caused by tense, number, and inflection, lemmatization is applied. Based on the lemmatized tokens, the node set is defined as follows:

$$V = \left\{ w_i^{(\text{lemma})} \mid w_i \in S \right\} \tag{2}$$

Based on the results of dependency parsing, directional edges are established between words to represent their syntactic relationships. The resulting semantic (text) graph is then formulated as follows:

$$G = (V, E),\ E = \{(v_i \rightarrow v_j) \mid \text{dependency}(v_i, v_j)\} \tag{3}$$

Each node $v_i \in V$ is transformed into a d-dimensional vector representation $h_i \in \mathbb{R}^d$ using pretrained GloVe embeddings. The overall node feature matrix is then constructed as follows:

$$H = \left[h_1^T;\ h_2^T;\ \ldots;\ h_n^T\right] \in \mathbb{R}^{n \times d},\ h_i = \text{GloVe}(v_i) \tag{4}$$

where the matrix H serves as the input to the GCN and is utilized to learn the structural semantic representations of the text graph. Here, v_i and v_j represent nodes corresponding to lemmatized words. n denotes the number of words (nodes) in the input sentence. d denotes the dimensionality of the GloVe embeddings, which is set to 100 in this study.

4.2. Scene Graph Similarity

The structural similarity between the graph G_{query} generated from the text query, and the scene graph G_{image} constructed from the image, is defined in terms of the similarity between their semantic representations. This approach reformulates the problem of text–image semantic alignment as a similarity computation task based on graph embeddings, thereby enabling a structural comparison of heterogeneous input modalities within a unified graph representation space.

The similarity between graphs is computed via the dot product of their corresponding embedding vectors, as formulated below:

$$Sim_{\text{graph}}(G_{\text{query}}, G_{\text{image}}) = z_{\text{query}}^{\top} z_{\text{image}} \tag{5}$$

where z_{query} and z_{image} are fixed-dimensional vectors obtained by applying mean pooling to the embeddings of the query and image graphs, respectively, after processing them through a GCN. Each graph embedding vector can be defined as follows:

$$z_{\text{query}} = \text{MeanPooling}(\text{GCN}(G_{\text{query}})),\ z_{\text{image}} = \text{MeanPooling}(\text{GCN}(G_{\text{image}})) \tag{6}$$

The embeddings of image scene graphs can be precomputed and stored in advance, allowing efficient similarity comparison by computing the query graph embedding in real time when a query is given.

4.3. Pair Sampling and Training Loss

A major practical limitation in learning semantic alignment between images is that the number of image pairs required for training grows quadratically as $O(N^2)$ with the size of the dataset. This issue becomes more prominent in real-world retrieval systems, where new images are continuously added, making repeated training over the entire dataset highly inefficient in terms of scalability. This challenge is analogous to the indexing structure in information retrieval systems, highlighting the need for a selective sampling strategy that constructs semantically meaningful training pairs while maintaining or improving retrieval performance. In this study, we address this requirement by designing a training pair sampling strategy based on surrogate relevance between images and define a corresponding training loss function.

(1) Definition of Surrogate Relevance

Each image I_i and I_j is associated with n captions, denoted as $c_i^{(k)}$ and $c_j^{(l)}$, respectively. For each caption, the embeddings $h_i^{(k)}$ and $h_j^{(l)}$ are obtained using SBERT [17]. The surrogate relevance r_{ij} between the two images is then defined as the average cosine similarity between all pairs of caption embeddings, as formulated below:

$$r_{ij} = \frac{1}{n^2} \sum_{k=1}^{n} \sum_{l=1}^{n} \cos\left(h_i^{(k)}, h_j^{(l)}\right) \tag{7}$$

This serves as a proxy measure that reflects actual text-based semantic similarity and is utilized as a training label.

(2) Uniform Pair Sampling (Initial Training Strategy)

In this study, uniform pair sampling (UPS) refers to a strategy in which image pairs are classified as positive or negative based on their surrogate relevance r_{ij}, and training pairs are then constructed by randomly sampling from each group at an equal 1:1 ratio. This approach helps maintain a balanced distribution between positive and negative pairs during the early stages of training, enabling the model to learn stable and discriminative representations.

(3) Hard Negative Mining (Progressive Strategy)

Once the initial representation learning has reached a certain level of convergence, an HNM strategy is employed to facilitate more fine-grained representation learning. HNM prioritizes the selection of negative pairs with low surrogate relevance but high graph embedding similarity, thereby encouraging the model to better distinguish subtle semantic differences during training:

$$r_{ij} < \tau \text{ and } \mathrm{Sim}_{\mathrm{graph}}\left(G_i, G_j\right) > \delta \tag{8}$$

where τ denotes the upper bound of surrogate relevance, and δ represents the lower bound of graph similarity. Hard negative pairs are selected as a subset of all negative pairs, with their sampling ratio gradually increasing as training progresses. This progressive hard negative sampling strategy aligns with the principles of curriculum learning, offering a balance between training stability and effectiveness.

(4) Definition of Loss Function

The model is trained to ensure that the predicted graph embedding similarity $\hat{r}_{ij}$ approximates the ground truth surrogate relevance r_{ij}. The loss function is defined using mean squared error (MSE), as follows:

$$\mathcal{L} = \frac{1}{\mid B \mid} \sum_{(i,j) \in B} \left(\hat{r}_{ij} - r_{ij}\right)^2 \tag{9}$$

where B denotes the set of training pairs within a batch. This training strategy enables efficient representation learning without the need for repeated training on the entire dataset. As a result, it facilitates the construction of an optimized framework that reduces training time while maintaining retrieval performance.

5. Proposed Subgraph Extraction

5.1. Motivation for Subgraph-Based Representation

Scene graphs generated from images, which include both objects and their relations, provide rich visual semantics. However, utilizing the entire graph during training is com-

putationally expensive and may introduce semantically irrelevant information, potentially hindering the model's generalization performance. In particular, the number of detected objects in an image can vary significantly, and there is often an imbalance in the spatial distribution and sizes of these objects. Such data disparities are reflected in the structure of the scene graph, introducing noise during training.

In this paper, we propose a subgraph extraction method called region-based semantic subgraph extraction (R-SSE). This method involves selecting region-based central nodes and performing a constrained DFS to extract minimal subgraphs, which are then merged. The approach is inspired by the scene graph decomposition method proposed by Zhong et al. [22], which highlights that scene graphs can vary significantly in structure depending on object count and relation density and that excessive connectivity may hinder learning.

R-SSE mitigates this issue by selecting semantically meaningful nodes within local regions of the image and constructing local minimal subgraphs that include only the most relevant relations. This strategy enhances both training efficiency and generalization performance by focusing learning on essential visual semantics.

The overall processing pipeline of the proposed R-SSE is illustrated in Figure 3. It visually demonstrates the process of selecting central nodes from the input image, constructing local subgraphs via a directed DFS, and merging them to form the final semantically focused subgraph.

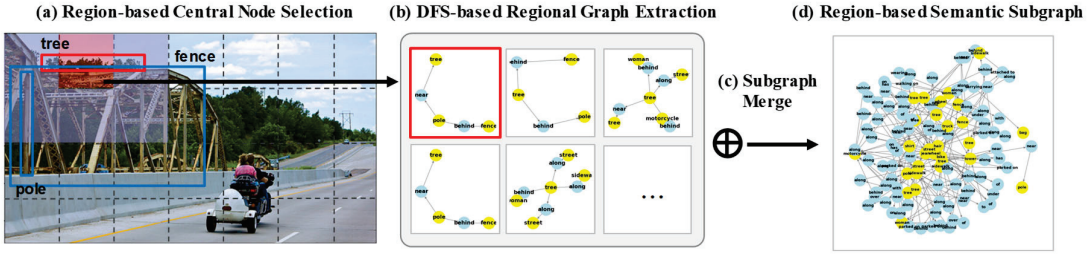


Figure 3. Proposed subgraph extraction method (R-SSE): (**a**) The input image is divided using a sliding window, and a central node is selected within each window; (**b**) directed depth-first search (DFS) is performed to extract local subgraphs; (**c**) overlapping nodes are merged to integrate the subgraphs; (**d**) Finally, region-based semantic subgraphs are constructed.

5.2. R-SSE (Region-Based Semantic Subgraph Extraction)

(1) Region-based Central Node Selection

To perform a regional partitioning of the nodes in a scene graph $G = (V, E)$, a sliding window technique is applied. A window region W of fixed size is moved across the image in a grid-like manner, and the set of nodes within each window is denoted as $V(W)$. Within each window, the central node c is selected as the object with the largest spatial extent, measured by its area.

This selection process is formally defined as:

$$c = \arg\max_{v \in V(W)} \text{Area}(v) \tag{10}$$

where $\text{Area}(v)$ denotes the bounding box area representing the spatial extent of object v. Selecting the object with the largest area aims to prioritize visually prominent objects, which are likely to carry higher semantic significance within the scene. In addition, a set of neighborhood windows $\mathcal{N}(W)$ is defined around the central window W. Subsequent exploration is restricted to the nodes located within $\mathcal{N}(W)$, excluding those in the central window itself. This design reduces the bias toward the central node and enables the

selection of semantically relevant external nodes, thereby enhancing the representational quality of the extracted subgraph.

(2) DFS-Based Regional Graph Extraction

A local subgraph $G_c = (V_c, E_c)$ is constructed with the central node c as its starting point. The exploration is performed on the directed scene graph G using a constrained DFS strategy. The search constraints are defined as follows:

- Maximum depth: $d = 2$;
- Maximum number of nodes: $m = 5$ (including the central node);
- Search node restriction: $V_{\text{search}} = \{u \in V(\mathcal{N}(W))\}$.

This exploration method incorporates both semantic relationships and spatial constraints, rather than relying solely on location-based expansion. As a result, it enables the extraction of local yet semantically meaningful subgraphs.

The overall procedure is detailed in Algorithm 1 of this paper, which includes the selection of region-based central nodes via a sliding window, exploration of surrounding window nodes, DFS-based expansion, and condition-based termination. The maximum number of nodes m is set to 5, including the central node, in order to maintain the compactness of the local subgraphs and to focus on capturing the most semantically relevant objects within each region. The sliding window size is empirically set to 75 pixels to balance local semantic concentration and spatial coverage. The maximum exploration depth d is set to 2 to maintain semantic locality and avoid overly deep traversals.

(3) Subgraph Merge

In the previously defined sliding window-based subgraph generation process, a local minimal subgraph is extracted from each region, centered around a central node and composed of semantically connected objects. These minimal subgraphs independently represent localized relationships across different spatial regions of the image and enable the model to learn distinct semantic units that may otherwise be entangled within a global scene graph.

This section describes the procedure for merging multiple local subgraphs—generated during the execution of Algorithm 1—into a single, unified subgraph. The core idea of the merging process is to normalize and consolidate duplicate object nodes that may appear across different local subgraphs. Specifically, nodes referring to the same object in different subgraphs are merged into a single unified node. The edges between these merged nodes preserve the original structural relations from their respective subgraphs, thereby forming a new, integrated semantic subgraph.

- V_merge: the normalized set of nodes obtained by removing duplicates across all merged nodes;
- E_merge: the set of edges preserving the original relationships among the merged nodes.

$$G_{\text{merge}} = \left(V_{\text{merge}}, E_{\text{merge}}\right) \tag{11}$$

Through this merging process, a focused representation composed only of semantically relevant objects can be extracted from the overall scene graph. This enables the construction of subgraphs that are locally dense in information while eliminating unnecessary or excessively connected components from the complex global graph structure. By reducing the overcomplexity of the full scene graph and providing a semantically condensed representation, this structure introduces an effective inductive bias for GNN-based learning.

Algorithm 1. Region-based Semantic Subgraph Extraction with Subgraph Merge (R-SSE)

Input: Scene graph $G = (V, E)$, sliding windows $\mathcal{W}$, max node count m, max depth d
Output: Merged subgraph $G_{\text{merge}} = (V_{\text{merge}}, E_{\text{merge}})$
$S \leftarrow$ // Set of all regional subgraphs
for each center window $W \in \mathcal{W}$ do
$c \leftarrow$ node in $V(W)$ with the largest blob area
$V_c \leftarrow \{c\}$
Stack $\leftarrow [(c, 0)]$
while Stack is not empty and $|V_c| < m$ *do*
$(v, \text{depth}) \leftarrow$ Stack:pop()
for each $(v \rightarrow u) \in E$ do
if $u \in V(\mathcal{N}(W))$ and $u \notin V_c$ and depth $+ 1 \leq d$ and
$|V_c| < m$ then
$V_c \leftarrow V_c \cup \{u\}$
Stack:push$(u, \text{depth} + 1)$
$E_c \leftarrow \{(u \rightarrow v) \in E \mid u, v \in V_c\}$
$G_c \leftarrow (V_c, E_c)$
$S \leftarrow S \cup \{G_c\}$
$V_{\text{merge}} \leftarrow$ union of all nodes in S, duplicates removed
$E_{\text{merge}} \leftarrow$ all edges among V_{merge} from E
Return $G_{\text{merge}} = (V_{\text{merge}}, E_{\text{merge}})$

6. Dataset and Experimental Results

6.1. Dataset and Sampling Strategy

In this study, we constructed the experimental dataset based on the publicly available aaai2021-scene-graph-image-retrieval dataset [23], which was originally built by selecting semantically similar images and captions from the MS-COCO [22] dataset. Semantic similarity in the original dataset was determined through a combination of human annotations and SBERT-based semantic similarity scores between captions. A total of 3216 images were included in the dataset.

To enable graph-based structural semantic retrieval research, we further extended the dataset by incorporating scene graph annotations from the Visual Genome [24] dataset for each corresponding image. Each training sample is organized as a pair consisting of a scene graph and a caption graph derived from the same image. For example, if image A has the caption "A cat sits on a mat", the sentence is converted into a graph structure and paired with the corresponding scene graph extracted from the image for training purposes.

The total number of images used in the experiment is 3216, and with five captions per image across all samples to maintain consistency and fairness in training, as provided by the MS-COCO dataset. This yields a total of $3216 \times 5 \times 3216$ training pairs. To ensure efficient training, only 25% of these pairs were used during model training. For evaluation, the first caption associated with each image (3216 captions in total) was used as query data to assess retrieval performance.

Semantic similarity between text and image modalities was learned based on these training pairs. Because the quality of pair construction significantly impacts final performance in contrastive learning frameworks, additional experiments were conducted to analyze performance differences according to the pair sampling strategies. The training pairs were constructed using three different sampling methods.

The dataset for comparative experiments was constructed by utilizing caption information from MS-COCO and scene graph annotations from Visual Genome. Because the proposed method generates semantic graphs from captions and extracts scene graphs from images, it can be applied to any dataset that provides both images and caption information. Therefore, the framework is designed to be applicable to other datasets that satisfy these conditions.

(1) Random Sampling:

Image pairs are randomly selected without considering similarity and are then used to construct positive and negative pairs. While this approach is straightforward, it lacks control over training difficulty and may result in a high proportion of easy negative samples, potentially reducing training efficiency.

(2) Uniform Pair Sampling (UPS):

Image pairs are classified as positive or negative based on their surrogate relevance scores, and training pairs are then constructed by randomly sampling from each category at a 1:1 ratio. This method is employed during the early stages of training to ensure stability in representation learning while maintaining data diversity.

(3) Hard Negative Mining (HNM):

Negative pairs are selected such that they exhibit low surrogate relevance but high graph structural similarity. By including these hard negative samples—those that are challenging for the model to distinguish—this strategy enhances the precision of representation learning and contributes to improved generalization performance. HNM is not applied during the initial stages of training but is introduced at a fixed ratio after a certain number of training epochs.

6.2. Training Details

The Adam optimizer is employed with an initial learning rate of 0.0001 and a weight decay of 5×10^{-4}. The batch size is set to 384, and the model is trained for a total of 20 epochs.

6.3. Evaluation

In this study, normalized discounted cumulative gain (nDCG) is employed to quantitatively evaluate the performance of text-to-image retrieval. nDCG is a representative ranking-based evaluation metric that assigns higher weights to items appearing at the top of the retrieved list, effectively capturing the balance between relevance and ranking position. The metric is defined as follows:

Discounted Cumulative Gain (DCG):

$$DCG_k = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)} \tag{12}$$

Normalized NCG (nDCG):

$$nDCG_k = \frac{DCG_k}{IDCG_k} \tag{13}$$

where rel_i denotes the surrogate relevance score of the image retrieved at rank i, and $IDCG_k$ represents the ideal DCG value for the given query. In this experiment, nDCG@k for k = 10, 20, 30, 40, 50 is used as the primary evaluation metric. nDCG effectively reflects the alignment between the ranking of retrieved images and their surrogate relevance scores, making it a practical and reliable measure of retrieval system performance.

(1) Baseline Methods

The performance of the proposed model was compared against the following three baseline methods:

- Keyword Match: A simple method that computes similarity based on the number of matching words between the text query and image captions. This approach does not account for structural or semantic relationships between words and therefore yielded the lowest performance.
- Contrastive Language–Image Pretraining (CLIP): Similar to the method proposed by Karthik et al. [5], this baseline utilizes a pretrained model based on ResNet and Vision Transformer (ViT) to directly learn semantic similarity between text and images. Among the text-based methods, CLIP achieved the highest performance in this experiment.
- Ground Truth Scene Graph (GT/SG): This approach uses manually annotated scene graphs provided by Visual Genome without modification. By leveraging accurate object and relation information, this method achieved the best performance among the graph-based approaches.

(2) Scene Graphs

Table 1 presents a quantitative comparison of text-to-image retrieval performance across different graph representation methods and GNN architectures. Experimental results show that the GCN (Node) approach consistently outperforms others across all evaluation points (nDCG@10 to @50), achieving a score of 0.738 at nDCG@50. This represents an improvement of approximately 7.2 percentage points over graph attention networks (GATs) [25] (Node) and 0.4 percentage points over the GCN (Edge) under the same graph representation scheme.

Table 1. Semantic image retrieval performance comparison across various GNN architectures and graph representation types using random sampling. "Caption" refers to ground truth textual annotations from the MS-COCO dataset, used as input for text-based models such as CLIP and Keyword Match. "GT/SG" denotes ground truth scene graphs from Visual Genome, and "I/SG" indicates automatically generated scene graphs via SGG.

Methods	Data	nDCG@ 10	nDCG@ 20	nDCG@ 30	nDCG@ 40	nDCG@ 50
Keyword Match	Caption	0.205	0.213	0.218	0.223	0.229
CLIP	Caption	0.845	0.846	0.845	0.844	0.844
GAT (Node)	GT/SG	0.628	0.644	0.653	0.660	0.666
GAT (Edge)	GT/SG	0.632	0.648	0.657	0.664	0.669
GCN (Node)	GT/SG	0.681	0.705	0.718	0.729	0.738
GCN (Edge)	GT/SG	0.676	0.700	0.715	0.725	0.734
GCN (Node)	I/SG	0.625	0.641	0.649	0.658	0.668
GCN (Edge)	I/SG	0.624	0.638	0.649	0.658	0.662

The observed performance gap can be attributed to the structural differences between GATs and GCNs. While GATs learn the importance of neighboring nodes through an attention mechanism, GCNs aggregate information from neighboring nodes using a normalized, fixed weighting scheme. In this experiment, the normalized aggregation of GCNs proved more robust in complex or noisy graph environments than the flexibledk weighting of GATs. This suggests that in scene graphs, where relational density is often imbalanced or certain nodes are overly connected, attention weights in GATs may become skewed, leading to biased representation learning.

Furthermore, performance differences were observed across graph representation strategies. The edge-based approach (GCN Edge) represents relations solely as edges, which limits the extent to which relational information is incorporated into node embeddings. In

contrast, the node-based approach (GCN Node) explicitly models relationships as separate nodes, enabling the GCN to learn embeddings for both objects and relations. This led to better performance; for instance, in the GT-based GCN setting, the node-based method achieved 0.738 at nDCG@50, while the edge-based method reached 0.734, with the node-based model consistently outperforming across all evaluated ranks (k = 10 to 50).

On the other hand, performance dropped when using automatically generated scene graphs (I/SG) compared to manually annotated ground truth scene graphs (GT/SG). For example, the GCN (Node) model achieved 0.738 on GT/SG, whereas it scored 0.668 on I/SG—a decrease of approximately 7.0 percentage points. This performance drop—from 0.738 (GT/SG) to 0.668 (I/SG) in GCN (Node) models—reflects the limitations of automatically generated scene graphs, which tend to include over-detected objects and redundant relationships. These artifacts lead to overly dense graph structures that reduce semantic diversity and introduce noise, thereby impairing the generalization ability of the GCN. To address this, we propose a combination of improved pair sampling and subgraph-based representation learning (Table 2) in Section 6.3, which demonstrably mitigates these issues and recovers performance close to that of GT/SG.

(3) Sampling Pair

The sampling strategy for training data plays a critical role in the performance of contrastive learning-based models. Through prior experiments, the limitations of random sampling were identified, and it was observed that the importance of negative pairs increases as training progresses. Accordingly, various sampling strategies were compared, and the results are summarized in Table 2. First, UPS involves separating all samples into positive and negative categories based on their surrogate relevance scores and then sampling an equal number of pairs (1:1 ratio) from each category. This method is particularly effective in the early stages of training, as it ensures representation stability and data diversity. Based on the full graph setting, UPS achieved an nDCG@50 of 0.714, showing an improvement of approximately 4.6 percentage points over random sampling (0.668).

Table 2. Comparison of semantic image retrieval performance across subgraph extraction methods and sampling strategies. 'Full' uses the entire scene graph, while 'Top(k)' selects the Top-k relations based on predicted similarity. R-SSE is the proposed method focusing on localized semantic subgraphs. The best performance is achieved by combining R-SSE with UPS + HNM, highlighting the effectiveness of both structured subgraph selection and hard negative sampling.

Subgraph Extraction	Sampling Pair	nDCG@ 10	nDCG@ 20	nDCG@ 30	nDCG@ 40	nDCG@ 50
Full	Random	0.625	0.641	0.649	0.658	0.668
Full	UPS	0.680	0.695	0.704	0.710	0.714
Top (k = 20)	UPS	0.696	0.711	0.721	0.727	0.732
Top (k = 40)	UPS	0.697	0.716	0.726	0.733	0.739
Top (k = 80)	UPS	0.691	0.710	0.721	0.729	0.735
R-SSE	UPS	0.697	0.715	0.725	0.733	0.740
Full	UPS + HNM	0.679	0.697	0.707	0.713	0.717
Top (k = 40)	UPS + HNM	0.699	0.718	0.728	0.735	0.740
R-SSE	**UPS + HNM**	**0.704**	**0.722**	**0.732**	**0.740**	**0.745**

In addition, the combination strategy UPS + HNM augments UPS with HNM, which progressively increases training difficulty by selecting negative pairs that are semantically dissimilar but have similar surrogate relevance scores. Under the full graph setting, UPS + HNM slightly outperformed UPS, achieving an nDCG@50 of 0.717, representing a 0.3 percentage point improvement. More notably, the performance gap was more pronounced in the subgraph-based setting. In particular, when UPS + HNM was applied in conjunction with the R-SSE subgraph extraction method, the model achieved the highest performance across all experimental conditions, recording an nDCG@50 of 0.745.

The Top-k approach constructs subgraphs by selecting the Top-k relationships based on the similarity ranking of node pairs predicted by the SGG module. While this method demonstrated improved performance compared to the full graph, a performance drop was observed as the number of selected nodes increased. For example, Top-k (k = 40) achieved a high nDCG@50 score of 0.739, but performance declined to 0.735 at Top-k (k = 80). This suggests that excessive increases in the number of nodes and edges may raise graph complexity, leading to potential overfitting or degradation in generalization during training.

In contrast, the proposed R-SSE method consistently outperformed the Top-k approach across all evaluation points (nDCG@10 to @50). R-SSE offers structural advantages by effectively capturing region-based semantic representations while maintaining a controlled number of nodes. On the other hand, the Top-k method demonstrates optimal performance only at specific values of k (e.g., k = 40), indicating limitations in its general applicability.

It is also worth noting that the evaluation benchmark used in this study—MS-COCO—provides natural language captions that typically describe the image as a whole. As a result, the advantages of R-SSE, which focuses on localized semantic representation, may not be fully captured in this setting. Nevertheless, R-SSE possesses structural properties that allow it to effectively capture semantic information from local regions within an image. Furthermore, it offers flexibility in adjusting semantic granularity through configurations such as the exploration scope and central node selection. These characteristics are expected to yield greater benefits in future application scenarios such as region-level scene queries or fine-grained scene descriptions.

(4) Subgraph Structure Analysis

To analyze the structural complexity introduced by different subgraph generation methods, we compared the number of nodes and edges in subgraphs generated from the same image set (a subset of Visual Genome, n = 5) using the 'Full', 'Top-k' (with k = 20, 40, 80), and the proposed R-SSE methods. Figure 4 visualizes the structural sizes of the subgraphs produced by each method, presenting the distribution of average node and edge counts.

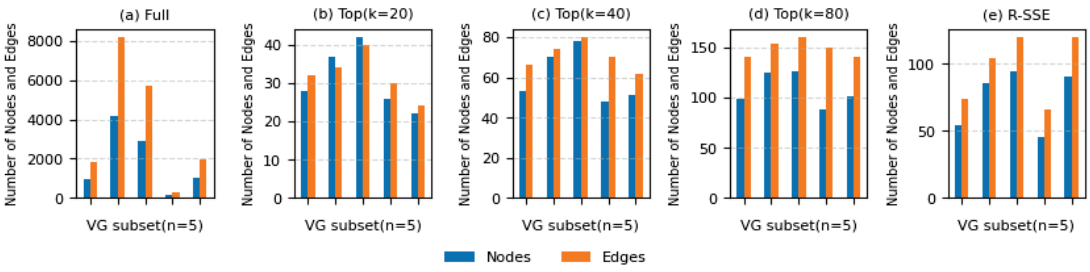


Figure 4. Comparison of node and edge counts for different subgraph extraction methods over five Visual Genome subset scenes: (**a**) Full, (**b**) Top (k = 20), (**c**) Top (k = 40), (**d**) Top (k = 80), (**e**) R-SSE.

The 'Full' method includes all possible relations between detected objects, resulting in an excessively dense structure with significantly increased graph size and computational overhead. In contrast, (*b–d*) the 'Top-k' methods limit graph size by selecting only the Top-k ranked relations; however, they exhibit large variations in the number of nodes and edges across images, indicating a lack of structural consistency. Notably, as k increases, the number of nodes tends to saturate, while the number of edges continues to grow proportionally. This suggests the inclusion of an excessive number of redundant or unnecessary connections.

The proposed R-SSE method constructs a more balanced graph by selecting region-based central nodes and semantically related objects within a constrained scope, effectively suppressing excessive connectivity while preserving only the most essential structure.

Experimental results show that R-SSE consistently maintains stable levels of both node and edge counts, striking a balance between graph compactness and representational richness. These structural characteristics indicate that R-SSE provides an efficient subgraph structure that supports both computational efficiency and expressive power during model training.

While R-SSE introduces sliding windows and DFS traversal, these operations are confined to local subgraph segments with significantly reduced node counts (83.0 nodes on average compared to 1140.9 in full scene graphs). Moreover, sliding windows are applied only to Top-k object regions, avoiding exhaustive search. Although these steps are performed during offline preprocessing, they contribute to generating compact and semantically focused subgraphs. As shown in Table 3, this structure results in efficient training performance, with R-SSE requiring 83 ms per iteration—an approximately 16% increase compared to Top (k = 40)—while making it effective for controlled subgraph construction depending on task objectives.

Table 3. Summary of the average preprocessing and training times for each subgraph extraction method.

Step	Full	Top (k = 20)	Top (k = 40)	Top (k = 80)	R-SSE
Preprocessing (Subgraph)	4 ms	0.1 ms	0.24 ms	0.5 ms	8 ms
Training (Iteration)	917 ms	54 ms	71 ms	125 ms	83 ms

6.4. Qualitative Results

To qualitatively evaluate the semantic alignment performance of the proposed model, we conducted a visual analysis comparing the top retrieved results for two natural language queries. Figure 5 presents the retrieval outcomes using different methods: (a) SBERT (GT), (b) CLIP, (c) GAT + Full, (d) GCN + Full, (e) GCN + Top-40, and (f) GCN + R-SSE



Figure 5. Comparison of retrieval results and methods: high similarity results (Query 1) vs. low similarity results (Query 2). (**a**) SBERT (GT), (**b**) CLIP, (**c**) GAT + Full, (**d**) GCN + Full, (**e**) GCN + Top-40, and (**f**) GCN + R-SSE (proposed).

For Query 1 (“A desk with a laptop, monitor, keyboard, mouse and speakers.”), most models successfully retrieved semantically relevant desk images. However, for Query 2 (“A man asleep against his baggage on top of bench.”), CLIP retrieved relatively well-matched images, while the GCN + SGG model returned less relevant results. While overall performance remained limited, the proposed R-SSE (f) was the only graph-based method to retrieve images containing key objects—man, bench, and baggage—indicating its relative advantage in handling complex semantic queries. This discrepancy can be attributed to two main factors.

First, CLIP, having been pretrained on large-scale image–text pairs, demonstrates strong generalization across diverse expressions. In contrast, the performance of the GCN + SGG model is heavily dependent on the quality of the graph structures generated via SGG and the training pairs.

Second, the current SGG system constructs scene graphs based only on a predefined set of object and relation categories, limiting its ability to fully capture the semantic richness of the input query. This can result in misalignment between the semantic structure of the query and that of the scene graph. Furthermore, as contrastive learning relies on diverse and informative training samples, insufficient variation in representation may lead to degraded generalization performance.

To overcome these limitations, future work should consider incorporating visual features into the scene graphs generated by SGG through a multimodal graph representation approach. By integrating CNN- or ViT-based visual information into node embeddings, the expressive power of graph-based semantic alignment models can be significantly enhanced, thereby improving the model’s ability to learn semantic similarity between text and images.

7. Conclusions

In this paper, we proposed a method to enhance text-based image retrieval by learning semantic structural similarity between scene graphs and natural language queries. Existing image–text alignment methods often rely on global representations of entire images or, when using graph structures, fail to fully capture their underlying semantics. We designed a GCN-based model to address these limitations by learning the structural similarity between scene graphs generated from images and semantic graphs converted from text. We further introduced an R-SSE using a sliding window approach, as well as an HNM-based sampling strategy.

Through quantitative experiments, we demonstrated that both the graph representation method and the sampling strategy significantly influence model performance. The proposed method achieved a top nDCG@50 score of 0.745, improving retrieval performance by approximately 7.7 percentage points compared to random sampling with full graphs. In addition, qualitative analysis revealed structural limitations of the current approach and outlined directions for future improvement. In particular, we identified the limited expressiveness of automatically generated scene graphs and the lack of diversity in training samples as key factors contributing to performance degradation.

While our model demonstrates competitive performance across many natural language queries, we observe that it performs sub-optimally on complex or abstract queries such as “A man glides along a frozen lake”, as illustrated in Figure 5. This is primarily due to the limited scale and diversity of our training dataset (3216 images), which contrasts with the large-scale web-scraped pretraining data used in CLIP. Nevertheless, our model achieves strong performance on a wide range of grounded and compositionally simpler queries, highlighting its effectiveness in data-constrained settings.

We acknowledge that reliance on predefined relationship categories may limit the capacity to capture free-form or abstract semantics. To mitigate this, we integrate relation

augmentation using CLIP-based similarity and employ soft textual–visual alignment during matching. Future work will explore structure-free or latent graph models to enable finer semantic matching.

Author Contributions: J.K. was responsible for the design, writing, and overall investigation. B.C.K. was responsible for the supervision and editing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2022-NR075648).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Karpathy, A.; Li, F.-F. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3128–3137.
2. Gordo, A.; Almazán, J.; Revaud, J.; Larlus, D. Deep Image Retrieval: Learning Global Representations for Image Search. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016, Proceedings, Part VI*; Springer: Cham, Switzerland, 2016; pp. 241–257.
3. Xuan, H.; Stylianou, A.; Liu, X.; Pless, R. Hard Negative Examples Are Hard, but Useful. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part XIV 16*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 126–142.
4. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models from Natural Language Supervision. In Proceedings of the International Conference on Machine Learning, ICML 2021, Virtual, 18–24 July 2021; pp. 8748–8763.
5. Karthik, S.; Roth, K.; Mancini, M.; Akata, Z. Vision-by-Language for Training-Free Compositional Image Retrieval. *arXiv* **2023**, arXiv:2310.09291.
6. Datta, R.; Li, J.; Wang, J.Z. Content-Based Image Retrieval: Approaches and Trends of the New Age. In Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval, Singapore, 10–11 November 2005; pp. 253–262.
7. Baldrati, A.; Bertini, M.; Uricchio, T.; Del Bimbo, A. Conditioned and Composed Image Retrieval Combining and Partially Fine-Tuning CLIP-Based Features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, New Orleans, LA, USA, 19–20 June 2022; pp. 4959–4968.
8. Robinson, J.; Chuang, C.-Y.; Sra, S.; Jegelka, S. Contrastive Learning with Hard Negative Samples. *arXiv* **2020**, arXiv:2010.04592.
9. Xu, D.; Zhu, Y.; Choy, C.B.; Li, F.-F. Scene Graph Generation by Iterative Message Passing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5410–5419.
10. Ali, N.; Bajwa, K.B.; Sablatnig, R.; Chatzichristofis, S.A.; Iqbal, Z.; Rashid, M.; Habib, H.A. A Novel Image Retrieval Based on Visual Words Integration of SIFT and SURF. *PLoS ONE* **2016**, *11*, e0157428. [CrossRef] [PubMed]
11. Wasson, V. An Efficient Content Based Image Retrieval Based on Speeded Up Robust Features (SURF) with Optimization Technique. In Proceedings of the 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 19–20 May 2017; pp. 730–735.
12. Gautam, G.; Khanna, A. Content Based Image Retrieval System Using CNN based Deep Learning Models. *Procedia Comput. Sci.* **2024**, *235*, 3131–3141. [CrossRef]
13. Li, L.H.; Yatskar, M.; Yin, D.; Hsieh, C.-J.; Chang, K.-W. VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv* **2019**, arXiv:1908.03557.
14. Gupta, S.; Malik, J. Visual Semantic Role Labeling. *arXiv* **2015**, arXiv:1505.04474.
15. spaCy. Available online: https://spacy.io/ (accessed on 1 March 2025).
16. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
17. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3982–3992.

18. Yoon, S.; Kang, W.Y.; Jeon, S.; Lee, S.; Han, C.; Park, J.; Kim, E.-S. Image-to-Image Retrieval by Learning Similarity between Scene Graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 10718–10726.
19. Kim, H.; Kim, S.; Ahn, D.; Lee, J.T.; Ko, B.C. Scene Graph Generation Strategy with Co-occurrence Knowledge and Learnable Term Frequency. *arXiv* **2024**, arXiv:2405.12648.
20. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese Neural Networks for One-Shot Image Recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 6–11 July 2015; pp. 1–8.
21. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24. [CrossRef] [PubMed]
22. Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D.A.; et al. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vis.* **2017**, *123*, 32–73. [CrossRef]
23. Kim, J.; Ko, B.C. Semantic Image Retrieval Using Scene Graphs and Natural Language Queries. In Proceedings of the 2025 International Conference on Electronics, Information, and Communication (ICEIC), Osaka, Japan, 19–22 January 2025; pp. 1–5.
24. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision (ECCV 2014), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
25. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–12.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Enhancing UAV Object Detection in Low-Light Conditions with ELS-YOLO: A Lightweight Model Based on Improved YOLOv11

Tianhang Weng and Xiaopeng Niu *

School of Computer Science and Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China; 2330702051@st.btbu.edu.cn

* Correspondence: nxp_btbu@btbu.edu.cn

Abstract

Drone-view object detection models operating under low-light conditions face several challenges, such as object scale variations, high image noise, and limited computational resources. Existing models often struggle to balance accuracy and lightweight architecture. This paper introduces ELS-YOLO, a lightweight object detection model tailored for low-light environments, built upon the YOLOv11s framework. ELS-YOLO features a re-parameterized backbone (ER-HGNetV2) with integrated Re-parameterized Convolution and Efficient Channel Attention mechanisms, a Lightweight Feature Selection Pyramid Network (LFSPN) for multi-scale object detection, and a Shared Convolution Separate Batch Normalization Head (SCSHead) to reduce computational complexity. Layer-Adaptive Magnitude-Based Pruning (LAMP) is employed to compress the model size. Experiments on the ExDark and DroneVehicle datasets demonstrate that ELS-YOLO achieves high detection accuracy with a compact model. Here, we show that ELS-YOLO attains a mAP@0.5 of 74.3% and 68.7% on the ExDark and DroneVehicle datasets, respectively, while maintaining real-time inference capability.

Keywords: low-light conditions; YOLOv11; lightweight; model pruning

1. Introduction

Drone-view object detection (DVOD) aims to locate and classify objects in images or videos captured by unmanned aerial vehicles (UAVs) [1]. With the rapid advancements in computer vision and UAV technologies, DVOD has become prevalent in diverse applications, including security surveillance, intelligent transportation systems, and environmental monitoring, achieving significant results. For instance, Wu et al. [2] proposed CCR-Net, a multimodal feature fusion network that improves operational efficiency in disaster response and emergency relief missions. Huang et al. [3] developed UFPMP-Det, which accurately identifies crop diseases and pests from UAV imagery. Zhan et al. [4] introduced ARGNet, effectively detecting forest fire smoke. Hu et al. [5] proposed CM-YOLO, which enhances the detection performance of aircraft and ships under cloud and mist conditions. Peng et al. [6] designed the MMDLTH framework to achieve robust detection of small infrared targets against heavy cloud clutter. By integrating deep learning algorithms, UAV systems can monitor critical events such as traffic violations and accidents in real time, providing essential support for informed decision making.

Nighttime security patrols and search-and-rescue missions require real-time and precise monitoring over large regions. However, traditional manual inspections are inefficient and prone to missed detections. Conventional surveillance equipment struggles with

capturing detailed imagery under low-light conditions and lacks automated data analysis capabilities. UAV systems equipped with object detection algorithms offer rapid deployment, mobility, and automated recognition, enabling efficient wide-area monitoring in a short time [7]. Therefore, developing robust and efficient object detection techniques for low-light conditions has become a critical research focus [8].

Currently, most object detection methods for low-light conditions primarily rely on image preprocessing techniques, such as brightness adjustment and noise suppression, to enhance image quality and thereby improve detection performance [9]. For example, Guo et al. [10] proposed an illumination map estimation method that initializes low-light images based on maximum RGB channel values. Hu et al. [11] mitigated color distortion in low-light images through saturation adjustment. Jeon et al. [12] combined atmospheric scattering models with pixel-adaptive gamma correction for image enhancement. However, these enhancement-based methods exhibit inherent limitations. First, the enhancement process can introduce artifacts that obscure essential image details. Second, reliance on fixed prior knowledge restricts adaptability to dynamically changing lighting conditions and limits the model's ability to learn deeper, high-level semantic features. Moreover, the computational overhead associated with image enhancement is significant for resource-constrained edge devices, severely limiting their real-time performance and practical deployment.

The YOLO (You Only Look Once) [13] series models represent single-stage object detection frameworks capable of performing localization and classification simultaneously in a single forward pass. These models, characterized by simple architectures, effectively balance detection accuracy and real-time performance, making them suitable for resource-constrained edge devices. Lightweight variants such as YOLOv8-nano, YOLOv9-tiny [14], and YOLOv10-nano [15] demonstrate robust performance on natural image datasets like Pascal VOC [16] and MS COCO [17], but they are not optimized specifically for low-light or UAV-captured imagery. Consequently, their performance significantly deteriorates under complex backgrounds and weak object features.

To address these issues, we propose a lightweight object detection model tailored specifically for low-light conditions called ELS-YOLO. This model builds upon the YOLOv11s framework and aims to balance detection accuracy with architectural efficiency.

The main contributions of this work are summarized as follows:

1. We design the re-parameterized backbone ER-HGNetV2, which integrates Re-parameterized Convolution (RepConv) [18] and Efficient Channel Attention (ECA) [19] mechanisms to effectively capture high-quality features, suppress noise, and enhance feature representation in low-light environments.
2. We develop LFSPN, which enables efficient cross-scale feature fusion and improves both model generalization and detection capability across diverse object scales.
3. We propose SCSHead, a lightweight detection head that leverages shared convolutions with separate batch normalization layers to minimize computational complexity and enhance inference efficiency. Furthermore, we incorporate Layer-Adaptive Magnitude-Based Pruning (LAMP) [20] to precisely prune redundant parameters, thereby reducing computational costs without compromising detection performance.
4. Extensive experiments conducted on the ExDark and DroneVehicle datasets demonstrate that ELS-YOLO achieves an optimal balance between detection accuracy and inference speed, validating its practical deployment potential.

2. Background

2.1. DVOD: Drone-View Object Detection

As an emerging research direction in remote sensing, DVOD faces unique challenges compared with conventional ground-view detection. Images captured by drones often contain numerous targets with significant scale variations, complicating detection accuracy and robustness. Existing DVOD approaches can be broadly classified into three categories: super-resolution-based, context-based, and representation fusion-based methods.

Super-resolution-based methods [21,22] enhance small object detectability by reconstructing low-resolution regions into high-resolution representations, typically through a three-stage pipeline: candidate region proposal, super-resolution reconstruction, and detection. Although these methods significantly improve object perception, their multi-stage structures often introduce redundant computation and complicate the training process, limiting their practicality and end-to-end optimization capability.

Context-based methods [23,24] leverage local and global contextual information to build spatial relationships and semantic dependencies, enhancing semantic representation and scene understanding. However, drone imagery often presents complex backgrounds and ambiguous semantic boundaries, hindering effective context modeling and reducing overall detection accuracy.

Representation fusion-based methods [25,26] integrate fine-grained spatial details from shallow features with high-level semantic features from deeper layers, primarily using architectures such as the Feature Pyramid Network (FPN) and its variants. Nonetheless, under low-light conditions, the representational gap between scales is pronounced, and direct fusion may introduce noise, degrading the discriminative ability of the model.

In addition, object detection from the UAV perspective faces multiple challenges arising from environmental factors. First, high-speed flight often results in image blur and motion trails, which significantly increase the difficulty of real-time detection. Second, weather variations such as fog, rain, and wind can degrade image quality and affect flight stability, thereby impairing object perception. Third, UAVs are susceptible to both intentional and unintentional electromagnetic interference, which may cause motor stalling, sensor drift, or even communication link failures [27,28]. All these factors can adversely impact the real-time performance and accuracy of object detection systems.

2.2. LLOD: Low-Light Object Detection

Existing research on object detection in low-light environments mainly focuses on improving image quality through low-light image enhancement (LLIE) and enhancing detection performance through architectural optimization.

LLIE techniques aim to restore critical information in dark regions by enhancing image brightness, contrast, and overall visual quality. Early LLIE methods relied on pixel intensity mapping and local statistical modeling. Techniques such as exposure correction [29] adjust global brightness distributions to enhance visibility, whereas histogram equalization [30] redistributes pixel intensity histograms to increase contrast. The Retinex theory [31] offers a physically interpretable enhancement framework by decomposing an image into illumination and reflectance components, thereby modeling contributions from lighting and surface texture. In recent years, deep neural networks have achieved significant advancements in LLIE tasks. LLNet [32] was the first deep autoencoder-based network designed for simultaneous low-light enhancement and denoising. Wei et al. [33] combined the Retinex theory with convolutional neural networks, incorporating Gaussian filtering and logarithmic transformation to perform adaptive brightness correction. Guo et al. proposed Zero-DCE [34,35], a method that achieves fast, reference-free image enhancement by learning pixel-wise luminance adjustment curves. Xu et al. [36] developed an

SNR-aware network that adaptively enhances images through global attention mechanisms and local structure modeling.

Architectural optimization aims to improve feature extraction and object recognition under low-light conditions by refining network structures and incorporating attention mechanisms. Long et al. [37] proposed a multi-level illumination learning framework, SCINet, which enhances feature extraction under complex backgrounds. Qiu et al. [38] introduced Efficient Attention Pyramid Transformer (EAPT), which integrates deformable attention and a global encoder–decoder structure to improve multi-scale feature modeling. Hu et al. [39] proposed an occlusion-aware attention module, MPCM, to alleviate detection difficulties caused by occlusion. Peng et al. [40] enhanced detection performance in low-light scenarios by optimizing attention mechanisms and the loss function. Wu et al. [41] developed the progressive enhancement network AENet, combining Yeo–Johnson transformation with a Transformer architecture to improve dynamic feature representation. Wang et al. [42] and Li et al. [43] mitigated the challenges posed by complex environments by employing geometry-aware learning and advanced signal reconstruction techniques.

3. Baseline Algorithm

YOLOv11 [44] is the latest version of the YOLO series, and provides five model variants—n, s, m, l, and x—to support deployment across a spectrum of platforms, ranging from edge devices to high-performance servers. As illustrated in Figure 1, YOLOv11 adopts a classic three-stage architecture consisting of a backbone, neck, and head, which are responsible for feature extraction, feature fusion, and object detection, respectively.

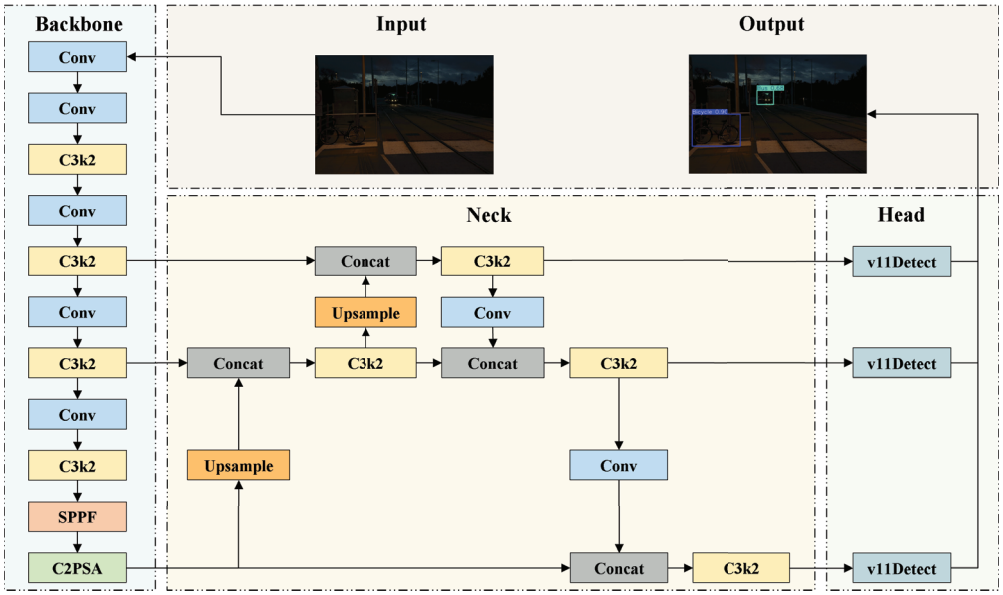


Figure 1. Overall architecture of the YOLOv11 model.

As a significant advancement in real-time object detection, YOLOv11 inherits the high efficiency and end-to-end detection capabilities of previous YOLO models, while incorporating multiple architectural innovations and optimizations. An overview of YOLOv11 primary modules is presented in Figure 2. The C3k2 module serves as the core structural unit of YOLOv11 and adjusts the kernel size by modifying the C3k parameter to meet the feature extraction requirements of different scenarios. The C2PSA module uses spatial attention to guide the model to focus on key regions and improve the accuracy of small or occluded objects. In the detection head, YOLOv11 employs depthwise separable convolutions (DWConv) to replace standard convolutions, thereby further reducing the parameter count and accelerating inference speed.

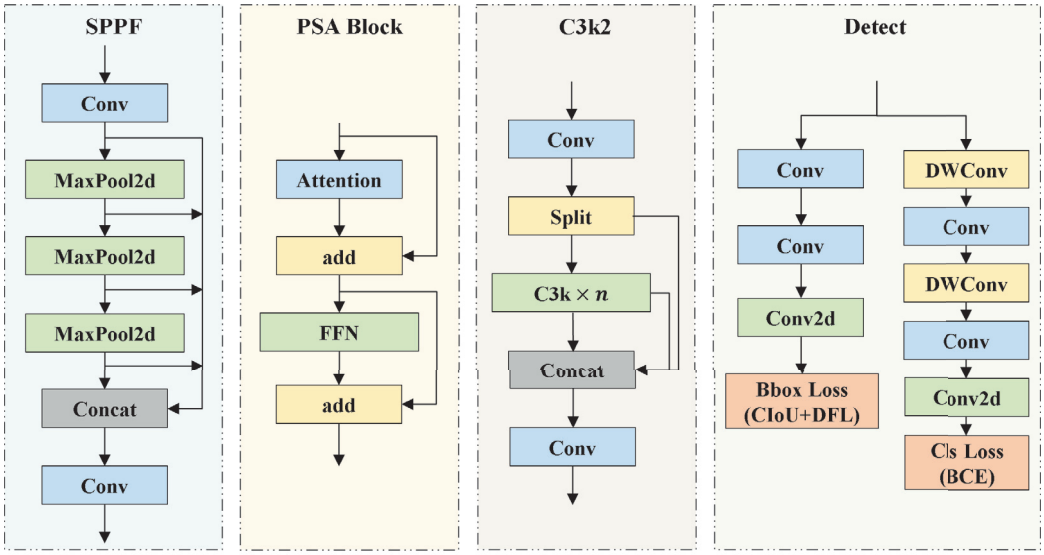


Figure 2. Detailed architectural design of core modules in YOLOv11.

We selected YOLOv11s as the baseline due to its superior performance under identical experimental conditions. As demonstrated in Tables 1 and 2, YOLOv11s achieved an optimal balance between detection accuracy, parameter count, and computational complexity on the ExDark dataset relative to other YOLO variants. Moreover, compared with the "n" variants, the "s" versions provided higher detection precision with only a slight increase in computational overhead, making them particularly suitable for lightweight UAV deployments.

Table 1. Comparison with the YOLOv11 series on the ExDark dataset.

Models	mAP@0.5/%	mAP/%	Params/M	GFLOPs/G	FPS
YOLO11n	67.6	42.2	**2.6**	**6.3**	**282**
YOLO11s	71.4	45.7	9.4	21.3	251
YOLO11m	73.2	47.7	20.0	67.7	218
YOLO11l	<u>74.6</u>	<u>48.9</u>	25.2	86.6	183
YOLO11x	**75.7**	**49.7**	56.8	194.5	169
ELS-YOLO	74.3	48.5	<u>4.6</u>	<u>15.0</u>	<u>274</u>

The best value for each metric is shown in bold, and the second-best value is underlined.

Table 2. Comparison with other models on the ExDark dataset.

Models	P/%	R/%	mAP@0.5/%	mAP/%	Params/M	GFLOPs/G
YOLOv8n	70.2	59.6	65.7	41.1	3.0	<u>8.1</u>
YOLOv8s	73.9	<u>62.7</u>	<u>70.4</u>	44.3	11.1	28.5
YOLOv9t	74.0	56.7	65.2	40.8	**2.0**	**7.6**
YOLOv9s	74.1	62.1	69.8	<u>44.8</u>	7.2	26.8
YOLOv10n	71.8	58.1	65.0	40.5	<u>2.7</u>	8.2
YOLOv10s	<u>77.2</u>	60.2	69.0	43.8	8.1	24.5
Faster R-CNN	67.4	52.6	58.9	35.2	41.2	208
RetinaNet	66.3	50.7	57.6	33.9	36.5	210
DETR	71.9	57.3	63.8	39.7	40.8	86.2
RT-DETR-r50	75.4	61.5	67.1	42.2	41.9	125.7
RT-DETR-L	73.1	58.1	64.6	39.9	32.0	103.5
ELS-YOLO	**79.2**	**65.8**	**74.3**	**48.5**	4.6	15.0

The best value for each metric is shown in bold, and the second-best value is underlined.

4. Methodology

4.1. ER-HGNetV2: Re-Parameterized Backbone

The backbone network of YOLOv11 primarily consists of alternating stacks of standard convolutional layers and C3k2 modules. Although this design demonstrates strong feature extraction capabilities, its increasing depth and channel width lead to parameter redundancy and substantial computational overhead. Moreover, this structure struggles to effectively capture complex global semantic features in low-light UAV imagery. To address this issue, inspired by the HGNetV2 framework [45], we design a re-parameterized backbone, ER-HGNetV2, which is illustrated in Figure 3.

ER-HGNetV2 begins with a Stem module that consists of standard convolution and max-pooling operations, performing initial spatial downsampling and extracting fundamental feature representations. ERBlock forms the core of the network and applies a multi-scale feature extraction strategy based on stacked convolution layers to refine features and enhance representational accuracy. The LDS layer integrates Depthwise Convolution (DWConv), batch normalization, and SiLU activation, enabling independent channel-wise convolution that enhances feature expressiveness while maintaining minimal computational overhead.

We construct ERBlock using the RepConv and ECA mechanism. As illustrated in Figure 4, RepConv adopts a multi-branch training structure that incorporates 3×3 convolutions, 1×1 convolutions, and identity mappings to capture diverse feature patterns. During inference, RepConv applies re-parameterization to fuse convolutional layers and batch normalization into a single standard convolution, thereby achieving a balance between representational capacity and inference efficiency. As shown in Figure 3, the ECA mechanism leverages a lightweight 1D convolution to adaptively compute channel-wise attention weights. This compact design effectively emphasizes critical feature channels with minimal computational cost, thereby improving the model's ability to distinguish targets under low-light conditions.

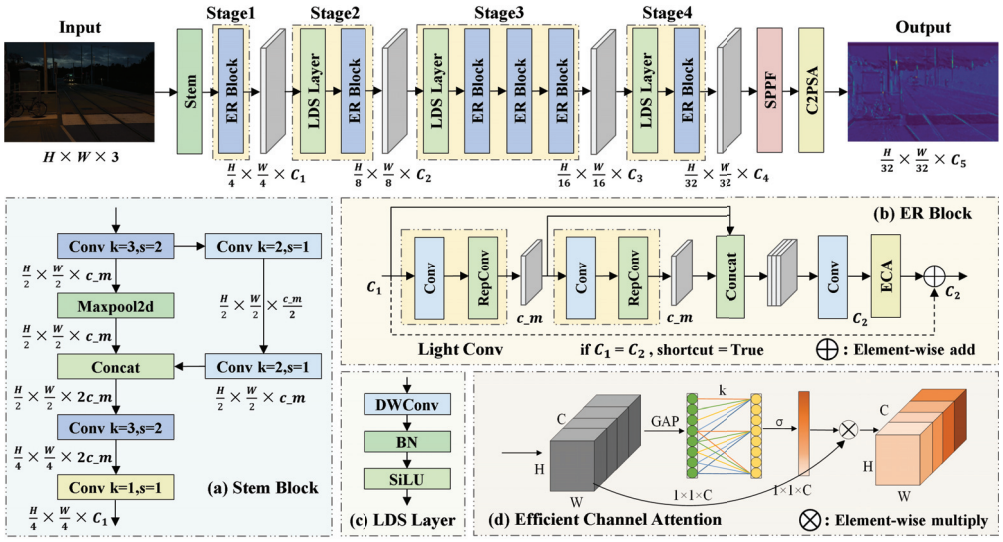


Figure 3. Network architecture of the proposed ER-HGNetV2 backbone.

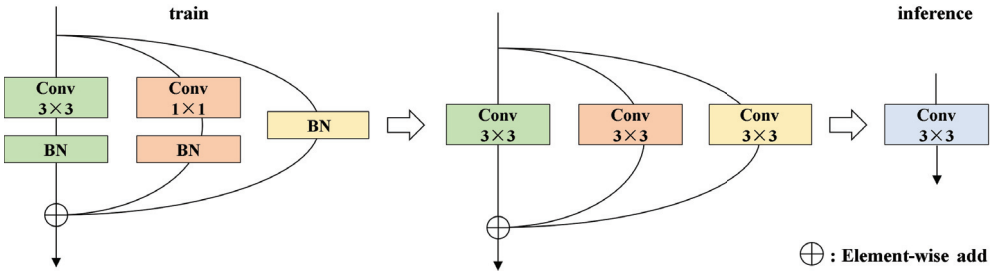


Figure 4. Re-parameterization process of the RepConv module.

4.2. LFSPN: Lightweight Feature Selection Pyramid Network

The Path Aggregation Feature Pyramid Network (PAFPN) used in YOLOv11 suffers from unselective aggregation and cross-level semantic inconsistency during feature fusion, making it difficult to obtain discriminative multi-scale representations. To address this issue, we design the LFSPN, which selectively strengthens semantically relevant features and suppresses redundant background information to significantly improve model robustness under challenging lighting conditions. As shown in Figure 5, LFSPN consists of two stages: an attention-weighted stage and a dynamic cross-scale feature fusion stage.

The attention-weighted stage performs initial feature selection on the multi-scale feature maps extracted by the backbone and enhances spatial position awareness via the Coordinate Attention (CA) [46] mechanism. As shown in Figure 6, the CA mechanism first applies global average pooling along the horizontal and vertical directions to capture spatial dependencies. It then concatenates the pooled features along the channel dimension and applies a 1×1 convolution to model cross-directional spatial relationships and generate intermediate attention weights. Finally, pixel-wise weighting is applied to the original feature map using the generated attention weights to emphasize informative regions and suppress redundant features, improving the quality of representations for the subsequent fusion stage.

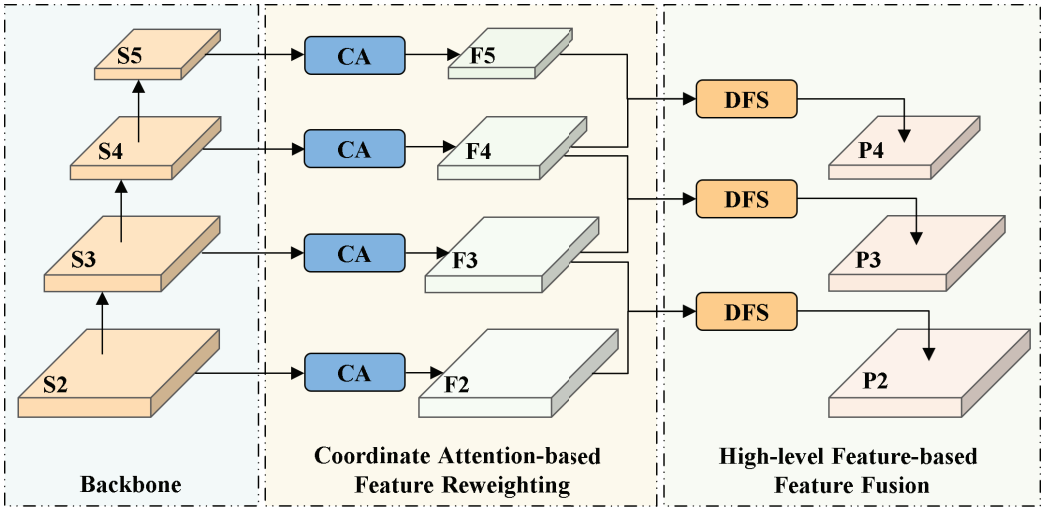


Figure 5. Architecture of the proposed Lightweight Feature Selection Pyramid Network.

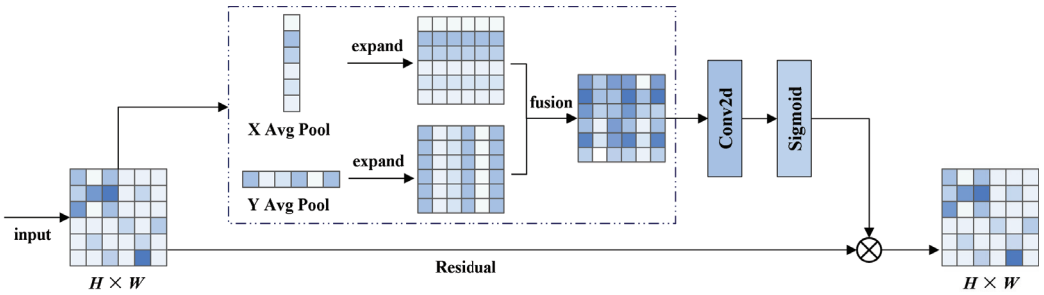


Figure 6. Illustration of the CA mechanism.

To further improve the effectiveness of feature fusion, we design a Dynamic Feature Selection (DFS) module, as shown in Figure 7. Taking the high-level feature map F_4 and the low-level feature map F_3 as an example, we first apply a 1×1 convolution to F_4 for nonlinear dimensionality reduction to lower computational cost. Then we upsample F_4 using transposed convolution to match the spatial resolution of F_3. The CA mechanism is applied to the upsampled feature map F_4 to generate attention weights for dynamic feature selection. These weights are element-wise multiplied with F_3 to selectively enhance spatially informative regions in the low-level feature map. The refined F_3 is then fused with F_4 through element-wise addition to produce a feature representation that preserves both fine-grained spatial details and high-level semantic information. The fused features are passed through a C3k2 module to further enhance feature representations, resulting in the final output feature map, as

$$f_{\text{tmp}} = \text{Trans}\Big(\text{Conv}_{1\times1}(f_{\text{high}})\Big) \tag{1}$$

$$f_{\text{out}} = \text{C3k2}\big(f_{\text{low}} \cdot \text{CA}(f_{\text{tmp}}) + f_{\text{tmp}}\big) \tag{2}$$

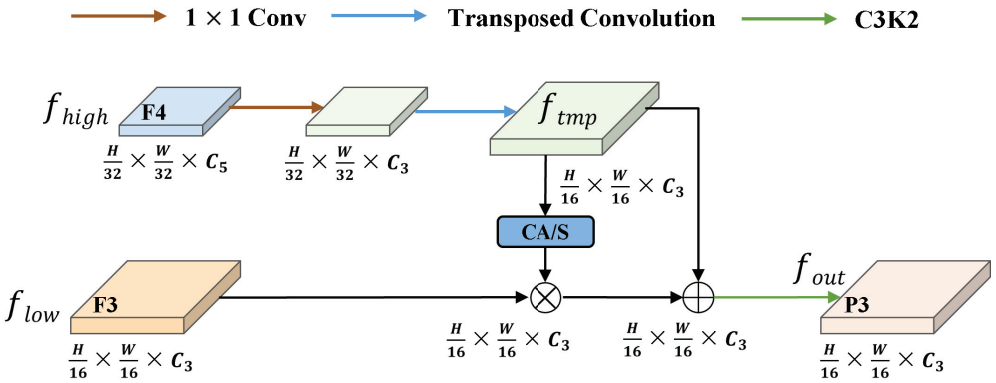


Figure 7. Framework of the Dynamic Feature Selection module.

4.3. SCSHead: Shared Convolution and Separate Batch Normalization Head

On UAV and other edge devices, detection head must balance high accuracy and low computational cost. However, conventional designs typically use separate convolutional branches for each prediction task, which leads to parameter redundancy. To address this issue, we propose the SCSHead, as shown in Figure 8.

SCSHead adopts a shared-weight convolutional architecture. Feature maps from different scales are first processed by individual 3×3 convolutional layers to align their channel dimensions, thereby ensuring consistent cross-scale feature representation. The aligned feature maps are then passed through a shared-weight convolutional module consisting of two successive convolutional stages. First, a 3×3 convolutional layer is

applied to extract cross-scale features and enhance representational capacity. Subsequently, a 1×1 convolutional layer performs channel-wise compression and reorganization, thereby reducing the number of parameters and computational cost. Finally, the processed features are fed into two parallel convolutional layers, Conv_Reg for predicting bounding box coordinates and Conv_Cls for classification scores.

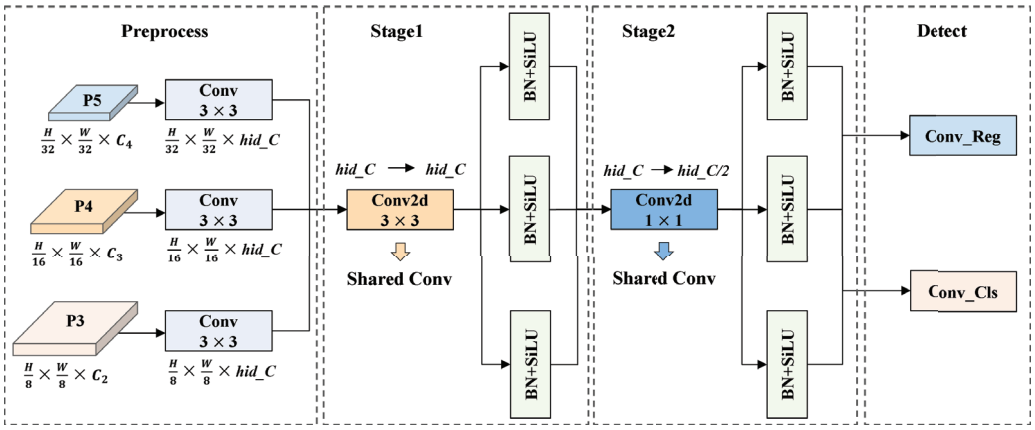


Figure 8. Architecture of the proposed SCSHead.

To accommodate statistical discrepancies across feature maps of different scales, we introduce a scale-specific normalization strategy. Specifically, each input branch applies an independent BN layer. This design allows each scale to adaptively normalize its feature statistics, thereby preserving and emphasizing the discriminative properties of individual feature maps.

4.4. Network Structure of ELS-YOLO

The overall architecture of the proposed ELS-YOLO is shown in Figure 9. The input image is first processed by the ER-HGNetV2 backbone to extract hierarchical features. These features are passed through LFSPN for dynamic cross-scale fusion to improve multi-scale representation. The fused features are then sent to SCSHead to produce the final detection results.

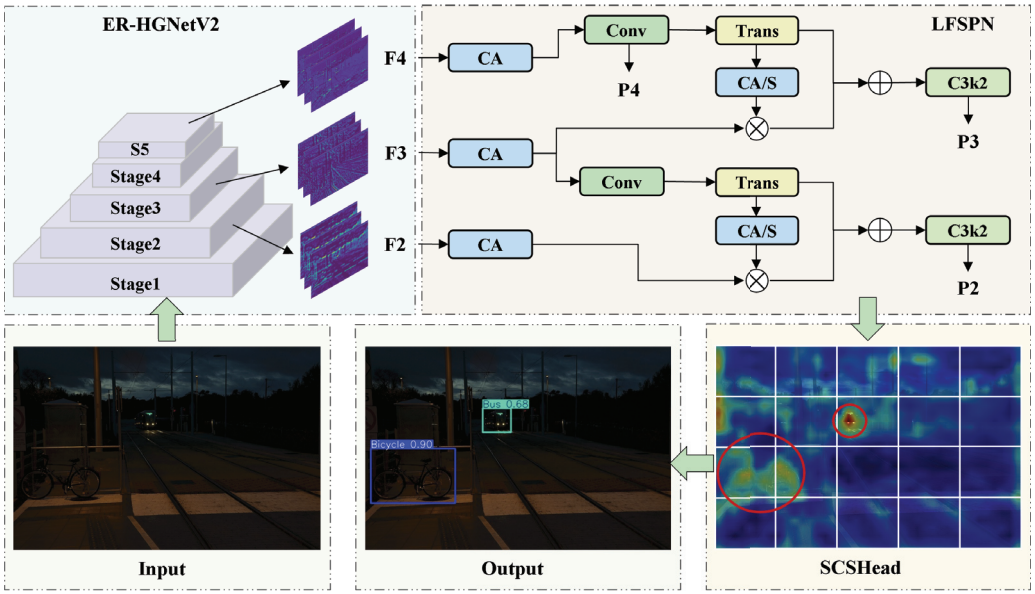


Figure 9. Overall architecture of the proposed ELS-YOLO model.

4.5. Channel Pruning

Deploying object detection systems on UAV platforms necessitates real-time processing of aerial imagery to accurately identify and localize traffic signs, vehicles, pedestrians, and other critical targets. However, UAVs are typically constrained by limited computational resources, memory, and battery capacity. Therefore, detection models must be carefully optimized to reduce both computational complexity and memory consumption. Model pruning has emerged as an effective structural optimization technique that significantly compresses model size while preserving near-original performance. To further improve the real-time inference capability of the proposed model on edge devices, we apply the LAMP method to eliminate redundant parameters and reduce model complexity.

Traditional magnitude-based pruning methods apply a global threshold to prune all layers of the network uniformly. However, such approaches fail to account for the varying importance of parameters across layers, often resulting in excessive pruning of critical layers and subsequent degradation in model accuracy. To address this issue, the LAMP method introduces a layer-wise adaptive importance scoring mechanism. By reweighting the importance of weights within each layer, LAMP determines an appropriate pruning ratio for each layer. As shown in Figure 10, for a given layer's weight tensor W, its values are first flattened into a one-dimensional vector and sorted in ascending order based on magnitude. Based on this sorted vector, LAMP assigns each weight u a corresponding score that reflects its relative importance within the layer, as

$$\text{score}(u, W) = \frac{(W[u])^2}{\sum_{v \geq u} (W[v])^2} \tag{3}$$

where the numerator represents the squared value of the current weight, and the denominator denotes the sum of squares of all weights whose magnitudes are greater than or equal to that of the current weight.

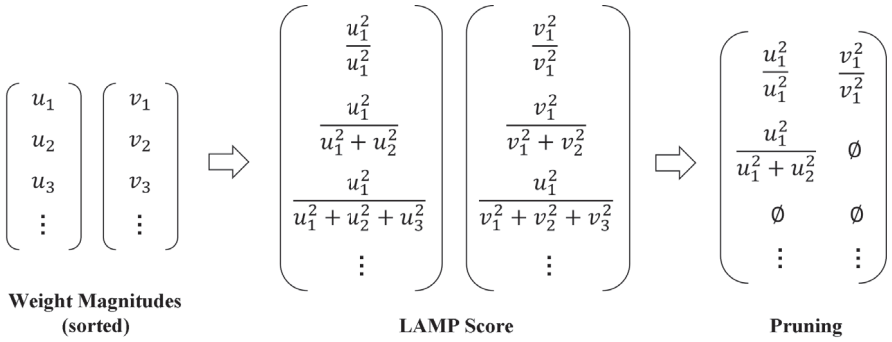


Figure 10. The LAMP pruning process.

The LAMP score reflects the relative contribution of each weight among the remaining connections. During pruning, the model removes connections with lower scores on a per-layer basis until the desired global sparsity level is reached. This strategy accounts for the varying contributions of different layers to the overall network, thereby enabling substantial model compression while preserving detection accuracy to the greatest extent possible.

5. Experimental Results

5.1. Dataset

To evaluate the effectiveness and generalizability of the proposed ELS-YOLO model for object detection in low-light conditions, we conducted extensive experiments on two

publicly available datasets: ExDark [47] and DroneVehicle [26]. We randomly split the dataset into training, validation, and test sets in an 8:1:1 ratio.

5.1.1. ExDark

The ExDark dataset is designed for object detection under low-light conditions. It contains 7363 real-world images captured in various challenging lighting environments and provides annotations for 12 object categories. Figure 11 shows typical image examples and a statistical overview of the annotations.

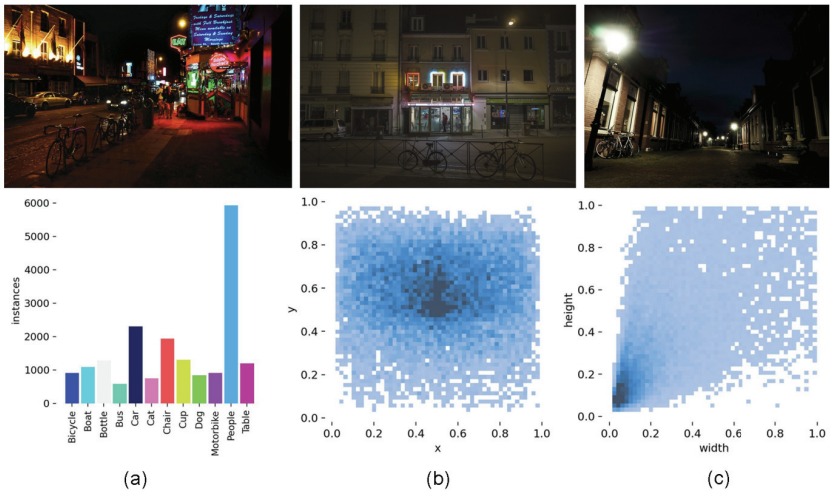


Figure 11. Illustration and statistical analysis of the ExDark dataset. (**a**) Class distribution of annotated objects, (**b**) distribution of normalized bounding box sizes, and (**c**) spatial distribution of bounding box center points across the image plane.

5.1.2. DroneVehicle

The DroneVehicle dataset contains aerial images captured by UAVs across various urban scenes such as city roads, residential areas, parking lots, and highways. The images are categorized into four lighting conditions: Day, Hazy, Night, and Dark Night. Figure 12 shows typical examples under each condition. Since this study focuses on object detection in low-light environments, we retain only the images labeled as Night and Dark Night.

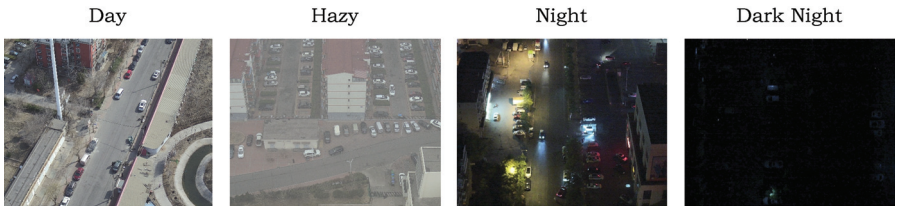


Figure 12. Images under varying illumination conditions.

5.2. Experimental Environment

The experiment is conducted using Python 3.10.16 and PyTorch 2.3.1 on an NVIDIA GeForce RTX 4090 GPU. We use SGD as the optimizer with an initial learning rate of 0.01, momentum set to 0.937, and weight decay of 0.0005. Each model is trained for 200 epochs with a batch size of 16 and no pretrained weights. All comparison experiments use the same settings.

5.3. Evaluation Indicators

The performance of the proposed model is evaluated using commonly adopted object detection metrics, including precision, recall, mean average precision (mAP), frames per second (FPS), and GFLOPs. The definitions of these metrics are presented as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$AP = \int_0^1 \text{Precision}(\text{Recall})\, d(\text{Recall}) \tag{6}$$

$$\text{mAP} = \frac{1}{K} \sum_{i=1}^{K} AP_i \tag{7}$$

where TP denotes the number of correctly detected positive samples, FP denotes the number of falsely detected positive samples, FN denotes the number of ground truth objects that were missed by the detector, K denotes the total number of object categories in the dataset, and AP_i represents the average precision for the i-th category.

5.4. Experimental Analysis on the ExDark Dataset

5.4.1. ER-HGNetV2 Experiment

To validate the effectiveness of the proposed ER-HGNetV2 backbone for low-light object detection, we conducted comparative experiments using YOLOv11s as the baseline and by replacing its backbone with ER-HGNetV2 and other existing lightweight alternatives. Table 3 presents the experimental results on the ExDark dataset.

Table 3. Comparative experiments with different backbone networks.

Backbone	mAP@0.5/%	mAP/%	Params/M	GFLOPs/G	FPS
baseline	<u>71.4</u>	<u>45.7</u>	9.4	21.3	251
EfficientViT [48]	68.5	43.1	7.98	19.0	214
RepViT [49]	69.3	43.9	10.14	23.5	201
HGNetV2	69.7	44.6	<u>7.61</u>	18.9	220
MobileNetV4 [50]	66.3	41.9	9.53	27.8	**267**
StarNet [51]	65.8	40.1	8.63	**17.6**	174
ER-HGNetV2	**72.6**	**46.5**	**7.6**	<u>18.3</u>	<u>255</u>

The best value for each metric is shown in bold, and the second-best value is underlined.

The proposed ER-HGNetV2 backbone demonstrates clear advantages across all evaluation metrics, achieving the highest scores in both mAP@0.5 and mAP. Compared with the original HGNetV2, ER-HGNetV2 yields better detection performance while incurring lower computational complexity.

5.4.2. Comparison with YOLOv11

To evaluate the performance advantages of ELS-YOLO over the YOLOv11 series models, we conducted a quantitative analysis on the ExDark dataset. The experimental results are presented in Table 1.

In terms of detection performance, ELS-YOLO achieves a mAP@0.5 of 74.3% and mAP of 48.5%. Compared with the baseline model YOLOv11s, ELS-YOLO improves mAP@0.5 by 2.9% and mAP by 2.8%, demonstrating stronger robustness in low-light conditions. Regarding model complexity, ELS-YOLO contains only 48.9% of the parameters and 70.4% of the computational cost of YOLOv11s, demonstrating excellent lightweight characteristics. Figure 13 illustrates the training performance curves of four key metrics for both ELS-YOLO and YOLOv11s. The proposed ELS-YOLO outperforms YOLOv11s across all four metrics throughout the training process.

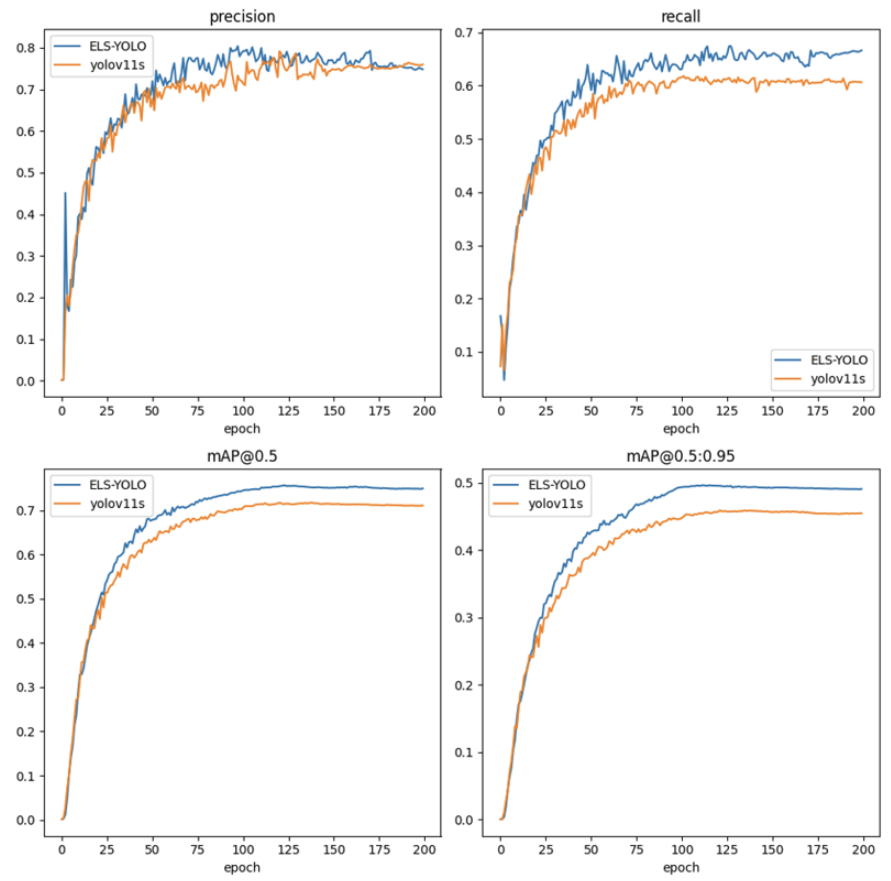


Figure 13. Training curves of ELS-YOLO and YOLOv11s.

5.4.3. LAMP Experiment

We conducted a detailed analysis of the ELS-YOLO model's performance under different pruning ratios. The pruning ratio is defined as the ratio of computational cost before pruning to that after pruning. For example, a pruning ratio of 1.33 indicates a 25% reduction in computation, a ratio of 2 indicates a 50% reduction, and a ratio of 4 indicates a 75% reduction.

As shown in Table 4, when the pruning ratio is set to 1.33, the number of parameters is reduced from 4.6 M to 2.4 M, corresponding to a compression rate of 47.8%. In terms of detection performance, mAP@0.5 remains at 74.3%, consistent with the unpruned model, while mAP decreases by only 0.1%. When the pruning ratio is increased to 2.0, the model achieves an optimal balance between compression and performance. Specifically, the parameter count further decreases to 1.3M, and mAP@0.5 is maintained at 74.2%. However, when the pruning ratio increases to 4.0, although the number of parameters continues to decline, detection performance drops sharply, with mAP@0.5 decreasing to 62.4% and mAP falling to 37.5%.

Table 4. Model performance under different pruning rates.

Models	mAP@0.5/%	mAP/%	Params/M	GFLOPs/G	FPS
ELS-YOLO	**74.3**	**48.5**	4.6	15.0	274
ELS-YOLO (ratio = 1.33)	**74.3**	<u>48.4</u>	2.4	11.2	283
ELS-YOLO (ratio = 2.0)	<u>74.2</u>	48.1	<u>1.3</u>	<u>7.4</u>	<u>298</u>
ELS-YOLO (ratio = 4.0)	62.4	37.5	**0.5**	**3.7**	**359**

The best value for each metric is shown in bold, and the second-best value is underlined.

Based on the above experimental results, the pruning ratio is set to 2.0, which provides the best trade-off between model compactness and detection accuracy.

5.4.4. Ablation Experiments

To comprehensively assess the contribution of each key component in the proposed ELS-YOLO, we conducted a series of ablation experiments. The corresponding experimental results are summarized in Table 5.

Table 5. Ablation experiments performed with the proposed ELS-YOLO.

Models	mAP@0.5/%	mAP/%	Params/M	GFLOPs/G	FPS
baseline	71.4	45.7	9.4	21.3	251
+A	72.6	46.5	7.6	18.3	255
+B	72.2	46.2	9.03	20.4	253
+C	72.7	45.9	6.64	18.7	262
+A+B	<u>73.8</u>	<u>47.9</u>	7.3	17.6	268
+A+C	73.5	47.6	<u>4.84</u>	<u>15.5</u>	<u>271</u>
+A+B+C	**74.3**	**48.5**	**4.6**	**15.0**	**274**

A denotes ER-HGNetV2, B denotes SCSHead, and C denotes LFSPN. The best value for each metric is shown in bold, and the second-best value is underlined.

First, we replace the original backbone with the proposed ER-HGNetV2 to enhance the model's ability to represent and generalize target features under low-light conditions. As shown in Table 5, this substitution resulted in a 1.2 increase in the mAP@0.5 metric, while the computational complexity was reduced by 3 GFLOPs. These results indicate that ER-HGNetV2 effectively eliminates redundant computation and improves the network's capacity to extract complex features in dark environments.

Next, we apply the proposed SCSHead, which further increases mAP@0.5 to 73.8%. This result indicates that SCSHead, through its shared convolutional structure and scale-adaptive normalization, enhances the network's robustness and discriminative capacity for multi-scale object detection in low-light conditions.

Finally, we introduce LFSPN to enhance multi-scale feature fusion. The addition of LFSPN improved mAP@0.5 and mAP by 0.5% and 0.6%, respectively. Meanwhile, the model's parameter and computational complexity were significantly reduced. These results confirm that LFSPN effectively integrates multi-scale information while suppressing redundant features, thereby improving both the network's generalization capability and computational efficiency.

5.4.5. Comparison Experiments with Other Baseline Methods

To further evaluate the effectiveness and performance advantages of the proposed ELS-YOLO, we conduct comparison experiments under the same settings with several mainstream detection models.

As presented in Table 2, ELS-YOLO demonstrates clear advantages across all key performance metrics, outperforming lightweight YOLO variants and DETR-based models. In terms of model complexity, ELS-YOLO maintains a compact architecture with only 4.6 M parameters and 15.0 GFLOPs, which is significantly lower than that of larger models such as Faster R-CNN and DETR. These results further support the feasibility of deploying ELS-YOLO for real-time inference on resource-constrained edge devices.

5.4.6. Visualization Analysis

To intuitively evaluate the detection performance of the proposed ELS-YOLO under low-light conditions, four representative scenes from the ExDark dataset were selected.

These scenes encompass various challenges, including small object detection, multi-scale targets, severe occlusion, and complex lighting environments.

From the visualization results in Figure 14, it is evident that ELS-YOLO demonstrates superior object detection and localization capabilities compared with YOLOv11s. Specifically, in the first row, under extremely poor illumination, ELS-YOLO accurately detects partially occluded targets in dim lighting conditions. In the second row, which depicts a distant and weakly illuminated scene, ELS-YOLO effectively detects small-scale pedestrians and vehicles, whereas YOLOv11s exhibits clear omissions. These results suggest that the ER-HGNetV2 backbone and the LFSPN structure enhance ELS-YOLO's ability to capture fine-grained features and small object information in low-light environments. In the third row, representing a multi-scale detection scenario, YOLOv11s shows noticeable false detections. In contrast, ELS-YOLO successfully identifies multiple object instances with higher precision and better localization, highlighting the benefits of the proposed SCSHead in cross-scale feature sharing and adaptive normalization. Additionally, in the fourth row, an urban night scene, ELS-YOLO provides clear and accurate detections across objects of various scales. In comparison, YOLOv11s performs poorly on distant targets, leading to both false negatives and false positives.

We use Gradient-weighted Class Activation Mapping (Grad-CAM) [52] to visualize the focus regions of both ELS-YOLO and the baseline YOLOv11s during prediction. As shown in Figure 15, ELS-YOLO exhibits more accurate attention to critical object regions compared with YOLOv11s. In the first row, under a dimly lit scene, YOLOv11s displays weak activation for distant pedestrians and bicycles. In contrast, ELS-YOLO significantly enhances attention to these small objects. In the second row, which depicts a complex nighttime environment, YOLOv11s either overlooks or diffusely attends to vehicles and distant pedestrians. However, ELS-YOLO produces more concentrated attention centered on the key regions of the targets. The third row further confirms the robustness and superiority of ELS-YOLO. Under partial occlusion, YOLOv11s generates vague and dispersed attention, whereas ELS-YOLO accurately highlights the partially occluded small objects.



Figure 14. Detection results of different models on the ExDark dataset. From left to right: ground truth,YOLOv11s, and ELS-YOLO.

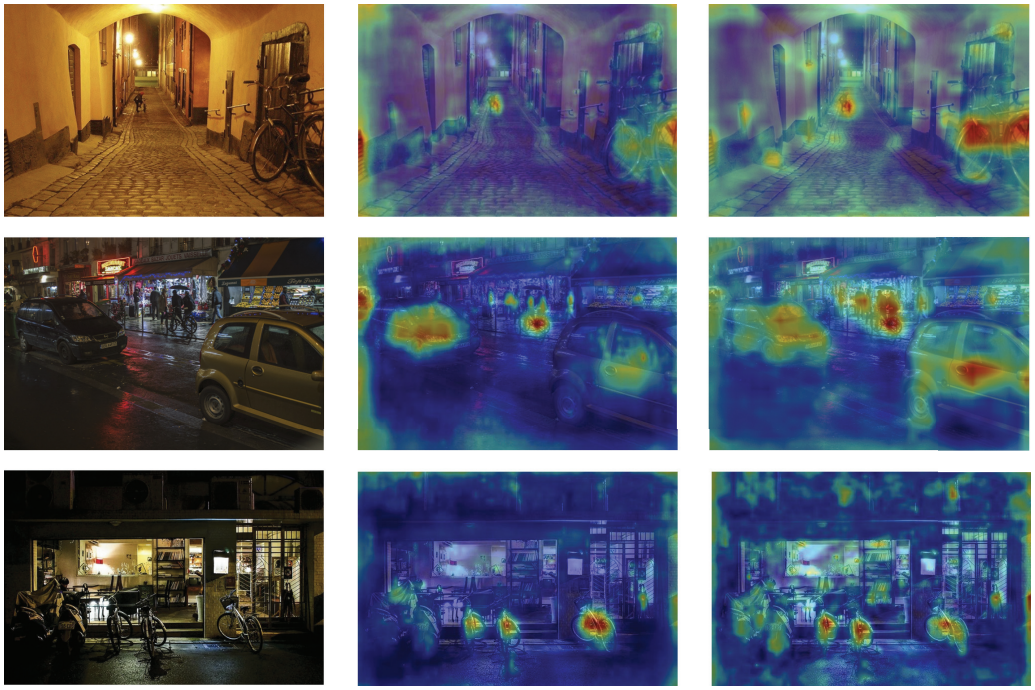


Figure 15. Heatmap results. From left to right: original images, YOLOv11s, and ELS-YOLO.

5.5. Experimental Analysis on the DroneVehicle Dataset

To further assess the performance of the proposed ELS-YOLO model in DVOD tasks, we conducted experiments on the DroneVehicle dataset. This dataset comprises aerial images captured by UAVs in various urban scenarios, including city roads, residential areas, parking lots, and highways. Representative inference results are presented in Figure 16, and the corresponding performance comparison is summarized in Table 6.

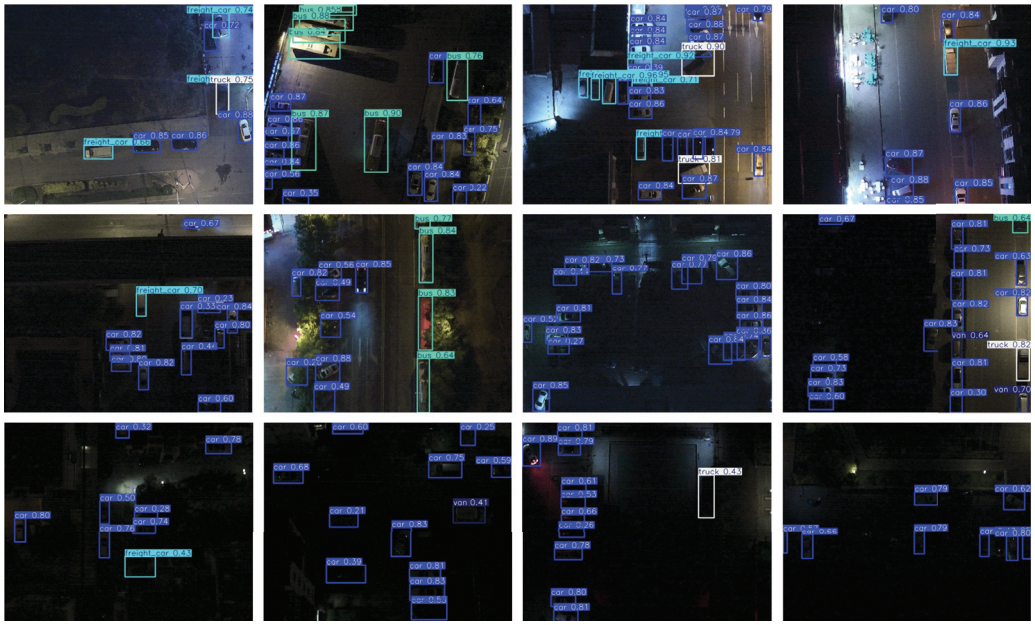


Figure 16. Detection results of ELS-YOLO on the DroneVehicle dataset.

Table 6. Comparative experiments of different object detection algorithms on the DroneVehicle dataset.

Models	P/%	R/%	mAP@0.5/%	mAP/%
YOLO11n	58.4	58.9	61.7	38.6
YOLO11s	**68.2**	63.7	67.2	42.9
RT-DETR-r50	65.7	63.2	66.7	41.2
RT-DETR-L	67.9	66.4	68.1	43.3
ELS-YOLO	<u>68.3</u>	**67.5**	**68.7**	**44.5**
ELS-YOLO (ratio = 2.0)	68.2	<u>67.3</u>	<u>68.5</u>	<u>44.2</u>

The best value for each metric is shown in bold, and the second-best value is underlined.

As shown in Table 6, ELS-YOLO achieves a precision of 68.3%, recall of 67.5%, mAP@0.5 of 68.7%, and mAP of 44.5%. Compared with the baseline model YOLOv11s, ELS-YOLO improves mAP@0.5 by 1.5% and mAP by 1.6%, demonstrating superior target localization capabilities.

6. Discussion

We conducted extensive experiments on the representative ExDark and DroneVehicle datasets to evaluate the object detection performance of ELS-YOLO under complex low-light environments. These datasets include diverse low-light scenarios and drone perspectives characterized by substantial scale variations and complex backgrounds. The experimental results indicate that ELS-YOLO achieves a mAP@0.5 of 74.3% and 68.7% on ExDark and DroneVehicle, respectively, significantly surpassing YOLOv11s and other mainstream detection models. This performance enhancement is primarily attributed to structural innovations, resulting in superior capabilities in feature extraction and fusion strategies. Compared with other lightweight models, ELS-YOLO exhibits advantages in parameter scale, computational complexity, and inference efficiency, highlighting its potential for deployment in edge computing environments.

Although ELS-YOLO demonstrates strong overall performance, there remains room for improvement under certain conditions. For instance, its detection accuracy still needs enhancement in scenarios involving extreme low-light conditions or severe occlusion. In addition, although this study has significantly reduced the model's parameter size, further improving inference speed without sacrificing accuracy remains an important direction for future research.

7. Conclusions

In this paper, we propose the lightweight object detection model ELS-YOLO to address the challenges of limited detection performance in complex low-light conditions and constrained computational resources on edge devices. We design the re-parameterized backbone ER-HGNetV2 to enhance the ability to extract and represent critical features under low-light environments. To overcome the limitations of conventional fusion strategies, we introduce LFSPN, which enables efficient multi-scale feature integration. We also develop the lightweight detection head SCSHead to reduce computational cost and parameter count. Furthermore, we apply the LAMP pruning strategy to compress model size without sacrificing accuracy. Extensive experiments on the ExDark and DroneVehicle datasets demonstrate that ELS-YOLO achieves superior detection accuracy and real-time inference efficiency compared with existing lightweight models.

In future work, we will further explore optimization strategies for the model under extreme conditions, such as incorporating advanced attention mechanisms or dynamic feature fusion methods, to enhance the model's adaptability in complex environments.

Additionally, we will investigate integrating techniques like knowledge distillation and quantization training to further improve the real-time inference capability of the model, thereby facilitating the deployment and application of object detection technologies in a broader range of practical scenarios.

Author Contributions: Conceptualization, T.W. and X.N.; methodology, T.W.; software, T.W.; validation, T.W.; formal analysis, T.W.; investigation, T.W.; resources, T.W.; data curation, T.W.; writing—original draft preparation, T.W.; writing—review and editing, X.N.; visualization, T.W.; supervision, X.N.; project administration, X.N.; funding acquisition, X.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62472010.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and Tracking Meet Drones Challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7380–7399. [CrossRef] [PubMed]
2. Wu, X.; Hong, D.; Chanussot, J. Convolutional Neural Networks for Multimodal Remote Sensing Data Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5517010. [CrossRef]
3. Huang, Y.; Chen, J.; Huang, D. UFPMP-Det: Toward Accurate and Efficient Object Detection on Drone Imagery. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 22 February–1 March 2022.
4. Zhan, J.; Hu, Y.; Zhou, G.; Wang, Y.; Cai, W.; Li, L. A high-precision forest fire smoke detection approach based on ARGNet. *Comput. Electron. Agric.* **2022**, *196*, 106874. [CrossRef]
5. Hu, J.; Wei, Y.; Chen, W.; Zhi, X.; Zhang, W. CM-YOLO: Typical Object Detection Method in Remote Sensing Cloud and Mist Scene Images. *Remote Sens.* **2025**, *17*, 125. [CrossRef]
6. Peng, L.; Lu, Z.; Lei, T.; Jiang, P. Dual-Structure Elements Morphological Filtering and Local Z-Score Normalization for Infrared Small Target Detection against Heavy Clouds. *Remote Sens.* **2024**, *16*, 2343. [CrossRef]
7. Zhou, L.; Dong, Y.; Ma, B.; Yin, Z.; Lu, F. Object detection in low-light conditions based on DBS-YOLOv8. *Clust. Comput.* **2025**, *28*, 55. [CrossRef]
8. Kaur, R.; Singh, S. A comprehensive review of object detection with deep learning. *Digit. Signal Process.* **2023**, *132*, 103812. [CrossRef]
9. Liu, X.; Wu, Z.; Li, A.; Vasluianu, F.A.; Zhang, Y.; Gu, S.; Zhang, L.; Zhu, C.; Timofte, R.; Jin, Z.; et al. NTIRE 2024 Challenge on Low Light Image Enhancement: Methods and Results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 17–18 June 2024; pp. 6571–6594. [CrossRef]
10. Guo, X.; Li, Y.; Ling, H. LIME: Low-Light Image Enhancement via Illumination Map Estimation. *IEEE Trans. Image Process.* **2017**, *26*, 982–993. [CrossRef] [PubMed]
11. Hu, C.; Yi, W.; Hu, K.; Guo, Y.; Jing, X.; Liu, P. FHSI and QRCPE-Based Low-Light Enhancement With Application to Night Traffic Monitoring Images. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 6978–6993. [CrossRef]
12. Jeon, J.J.; Park, J.Y.; Eom, I.K. Low-light image enhancement using gamma correction prior in mixed color spaces. *Pattern Recognit.* **2024**, *146*, 110001. [CrossRef]
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
14. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In Proceedings of the European Conference on Computer Vision (ECCV), Milan, Italy, 29 September–4 October 2024; pp. 1–21. [CrossRef]
15. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011.

16. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
17. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Lecture Notes in Computer Science, Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014; Volume 8693. [CrossRef]
18. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets Great Again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13728–13737. [CrossRef]
19. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539. [CrossRef]
20. Lee, J.; Park, S.; Mo, S.; Ahn, S.; Shin, J. Layer-Adaptive Sparsity for the Magnitude-Based Pruning. In Proceedings of the 9th International Conference on Learning Representations (ICLR), Virtually, 3–7 May 2021.
21. Deng, S.; Li, S.; Xie, K.; Song, W.; Liao, X.; Hao, A.; Qin, H. A Global-Local Self-Adaptive Network for Drone-View Object Detection. *IEEE Trans. Image Process.* **2021**, *30*, 1556–1569. [CrossRef] [PubMed]
22. Bai, Y.; Zhang, Y.; Ding, M.; Ghanem, B. SOD-MTGAN: Small Object Detection via Multi-Task Generative Adversarial Network. In *Lecture Notes in Computer Science, Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Cham, Switzerland, 2018; Volume 11217. [CrossRef]
23. Xi, Y.; Zheng, J.; He, X.; Jia, W.; Li, H.; Xie, Y.; Feng, M.; Li, X. Beyond context: Exploring semantic similarity for small object detection in crowded scenes. *Pattern Recognit. Lett.* **2020**, *137*, 53–60. [CrossRef]
24. Li, G.; Liu, Z.; Zeng, D.; Lin, W.; Ling, H. Adjacent context coordination network for salient object detection in optical remote sensing images. *IEEE Trans. Cybern.* **2023**, *53*, 526–538. [CrossRef] [PubMed]
25. Zhao, W.; Kang, Y.; Chen, H.; Zhao, Z.; Zhao, Z.; Zhai, Y. Adaptively attentional feature fusion oriented to multiscale object detection in remote sensing images. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 5008111. [CrossRef]
26. Sun, Y.; Cao, B.; Zhu, P.; Hu, Q. Drone-based RGB-Infrared cross-modality vehicle detection via uncertainty-aware learning. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 6700–6713. [CrossRef]
27. Jie, H.; Zhao, Z.; Zeng, Y.; Chang, Y.; Fan, F.; Wang, C.; See, K.Y. A review of intentional electromagnetic interference in power electronics: Conducted and radiated susceptibility. *IET Power Electron.* **2024**, *17*, 1487–1506. [CrossRef]
28. Jie, H.; Zhao, Z.; Li, H.; Gan, T.H.; See, K.Y. A Systematic Three-Stage Safety Enhancement Approach for Motor Drive and Gimbal Systems in Unmanned Aerial Vehicles. *IEEE Trans. Power Electron.* **2025**, *40*, 9329–9342. [CrossRef]
29. Farid, H. Blind inverse gamma correction. *IEEE Trans. Image Process.* **2001**, *10*, 1428–1433. [CrossRef] [PubMed]
30. Zuiderveld, K. Contrast limited adaptive histogram equalization. In *Graphics Gems IV*; Academic Press Professional, Inc.: Williston, VT, USA, 1994; pp. 474–485.
31. Li, M.; Liu, J.; Yang, W.; Sun, X.; Guo, Z. Structure-revealing low-light image enhancement via robust Retinex model. *IEEE Trans. Image Process.* **2018**, *27*, 2828–2841. [CrossRef] [PubMed]
32. Lore, K.G.; Akintayo, A.; Sarkar, S. LLNet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognit.* **2017**, *61*, 650–662. [CrossRef]
33. Li, X.; Wang, W.; Feng, X.; Li, M. Deep parametric Retinex decomposition model for low-light image enhancement. *Comput. Vis. Image Underst.* **2024**, *241*, 103948. [CrossRef]
34. Guo, C.; Li, C.; Guo, J.; Loy, C.C.; Hou, J.; Kwong, S.; Cong, R. Zero-reference deep curve estimation for low-light image enhancement. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1777–1786. [CrossRef]
35. Li, C.; Guo, C.; Loy, C.C. Learning to enhance low-light image via zero-reference deep curve estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 4225–4238. [CrossRef] [PubMed]
36. Xu, X.; Wang, R.; Fu, C.-W.; Jia, J. SNR-aware low-light image enhancement. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 17693–17703. [CrossRef]
37. Ma, L.; Ma, T.; Liu, R.; Fan, X.; Luo, Z. Toward fast, flexible, and robust low-light image enhancement. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5627–5636. [CrossRef]
38. Qiu, H.; Li, H.; Wu, Q.; Meng, F.; Xu, L.; Ngan, K.N.; Shi, H. Hierarchical context features embedding for object detection. *IEEE Trans. Multimed.* **2020**, *22*, 3039–3050. [CrossRef]
39. Hu, J.; Cui, Z. YOLO-Owl: An occlusion aware detector for low illuminance environment. In Proceedings of the 2023 3rd International Conference on Neural Networks, Information and Communication Engineering (NNICE), Guangzhou, China, 24–26 February 2023; pp. 167–170. [CrossRef]

40. Zhang, Y.; Wu, C.; Zhang, T.; Liu, Y.; Zheng, Y. Self-attention guidance and multiscale feature fusion-based UAV image object detection. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 6004305. [CrossRef]
41. Wu, R.; Huang, W.; Xu, X. AE-YOLO: Asymptotic enhancement for low-light object detection. In Proceedings of the 2024 17th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 26–28 October 2024; pp. 1–6. [CrossRef]
42. Wang, H.; Zhang, G.; Cao, H.; Hu, K.; Wang, Q.; Deng, Y.; Gao, J.; Tang, Y. Geometry-Aware 3D Point Cloud Learning for Precise Cutting-Point Detection in Unstructured Field Environments. *J. Field Robot.* 2025, *Early View*. [CrossRef]
43. Li, X.; Hu, Y.; Jie, Y.; Zhao, C.; Zhang, Z. Dual-Frequency Lidar for Compressed Sensing 3D Imaging Based on All-Phase Fourier Transform. *J. Opt. Photon. Res.* **2023**, *1*, 74–81. [CrossRef]
44. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv*, **2024**, arXiv:2410.17725. [CrossRef]
45. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974. [CrossRef]
46. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13708–13717. [CrossRef]
47. Loh, Y.P.; Chan, C.S. Getting to Know Low-light Images with The Exclusively Dark Dataset. *Comput. Vis. Image Underst.* **2019**, *178*, 30–42. [CrossRef]
48. Cai, H.; Li, J.; Hu, M.; Gan, C.; Han, S. EfficientViT: Lightweight Multi-Scale Attention for High-Resolution Dense Prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 17256–17267. [CrossRef]
49. Wang, A.; Chen, H.; Lin, Z.; Han, J.; Ding, G. Rep ViT: Revisiting Mobile CNN From ViT Perspective. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 15909–15920. [CrossRef]
50. Qin, D.; Leichner, C.; Delakis, M.; Fornoni, M.; Luo, S.; Yang, F.; Wang, W.; Banbury, C.; Ye, C.; Howard, A.; et al. MobileNetV4: Universal Models for the Mobile Ecosystem. In *Lecture Notes in Computer Science, Proceedings of the Computer Vision—ECCV 2024, Milan, Italy, 29 September–4 October 2024*; Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G., Eds.; Springer: Cham, Switzerland, 2025; Volume 15098. [CrossRef]
51. Dai, X.; Bai, Y.; Wang, Y.; Fu, Y. Rewrite the Stars. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 5694–5703. [CrossRef]
52. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 618–626. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Enhanced SOLOv2: An Effective Instance Segmentation Algorithm for Densely Overlapping Silkworms

Jianying Yuan, Hao Li, Chen Cheng, Zugui Liu, Sidong Wu and Dequan Guo *

School of Automation, Chengdu University of Information Technology, Chengdu 610225, China; yuanjy@cuit.edu.cn (J.Y.); lh0612123@163.com (H.L.); chengc0728@163.com (C.C.); lzgcuit2@163.com (Z.L.); wsd@cuit.edu.cn (S.W.)
* Correspondence: guodq@cuit.edu.cn

Abstract

Silkworm instance segmentation is crucial for individual silkworm behavior analysis and health monitoring in intelligent sericulture, as the segmentation accuracy directly influences the reliability of subsequent biological parameter estimation. In real farming environments, silkworms often exhibit high density and severe mutual occlusion, posing significant challenges for traditional instance segmentation algorithms. To address these issues, this paper proposes an enhanced SOLOv2 algorithm. Specifically, (1) in the backbone network, Linear Deformable Convolution (LDC) is incorporated to strengthen the geometric feature modeling of curved silkworms. A Haar Wavelet Downsampling (HWD) module is designed to better preserve details for partial visible targets, and an Edge-Augmented Multi-attention Fusion Network (EAMF-Net) is constructed to improve boundary discrimination in overlapping regions. (2) In the mask branch, Dynamic Upsampling (Dysample), Adaptive Spatial Feature Fusion (ASFF), and Simple Attention Module (SimAM) are integrated to refine the quality of segmentation masks. Experiments conducted on a self-built high-density silkworm dataset demonstrate that the proposed method achieves an Average Precision (AP) of 85.1%, with significant improvements over the baseline model in small- (APs: +10.2%), medium- (APm: +4.0%), and large-target (APl: +2.0%) segmentation accuracy. This effectively advances precision in dense silkworm segmentation scenarios.

Keywords: intelligent sericulture; silkworm instance; segmentation enhanced; SOLOv2 EAMF-Net

1. Introduction

The sericulture industry, as a vital component of the agricultural economic sector, plays an indispensable role in promoting rural economic development, increasing farmers' income, and driving progress in the textile industry in major silk-producing countries such as China and India. With the transition of modern agriculture towards intelligent and precision-oriented practices, the research and application of smart sericulture technologies have emerged as the core driving force for enhancing industrial efficiency. In this context, computer vision technology has gradually become a critical solution for silkworm growth monitoring, behavioral analysis, and health assessment due to its non-contact nature and operational efficiency. Dense silkworm instance segmentation, serving as the foundation for individual silkworm vitality monitoring, directly determines the accuracy of subsequent biological parameter estimations for each silkworm. Consequently, it constitutes a critical technical foundation for constructing intelligent sericulture systems.

Currently, in the field of intelligent sericulture, silkworm segmentation primarily involves two approaches: traditional image segmentation techniques and deep learning-based methods. For traditional approaches, threshold-based segmentation and clustering-based segmentation are commonly utilized. Threshold-based segmentation separates silkworms as foreground from the background by analyzing feature differences and calculating optimal thresholds. Yang et al. [1] used Otsu and maximum entropy to determine the segmentation threshold for silkworms and mulberry branches in an automated feeding system for the fifth-instar Bombyx mori. Hu et al. [2] proposed a segmentation method for Bombyx mori microsporidia that combines fuzzy contrast enhancement and color feature extraction. This approach first enhances the image, then segments the microsporidia using Otsu on the color channel of the microsporidia. Clustering-based algorithms partition images by grouping pixels with similar features into distinct categories. Luo et al. [3] proposed an improved K-means algorithm to address challenges in analyzing silkworm phenotypic traits, optimizing threshold computation to mitigate background interference and accurately measure silkworm areas. Although traditional methods demonstrate simplicity and computational efficiency, they are highly sensitive to illumination variations and fail to resolve overlapping individuals in dense populations.

In recent years, with advancements in deep learning, numerous researchers have applied deep learning-based image segmentation methodologies to silkworm segmentation. H et al. [4] conducted a comparative study employing three deep learning models: U-Net, Mask R-CNN, and FCN for automated silkworm segmentation. The comparative analysis demonstrated that the U-Net model achieved the highest Intersection over Union (IoU) score of 0.89 among the three approaches. Wang et al. [5] developed an automated system for Bombyx mori counting and instar stage identification by leveraging the YOLOv5 model with real-time image acquisition and instance segmentation algorithms, significantly enhancing the automation level of sericulture practices. In the related advancement, He et al. [6] introduced an improved Mask R-CNN model incorporating pixel reweighting and bounding box refinement strategies. Experimental evaluations on samples with complex background interference demonstrated that their approach achieved a silkworm recognition accuracy of 96.83%. Currently, research on silkworm instance segmentation is still in its early stages. In the realm of instance segmentation for insects similar to silkworms, they proposed an enhanced instance segmentation model combining Pyramid Vision Transformer (PVT) with Mask R-CNN, targeting the challenge of detecting camouflaged pests (e.g., cotton bollworms) [7]. This approach significantly improved pest recognition performance in complex agricultural environments. For the coffee fruit fly, Tassis et al. [8] integrated instance segmentation with semantic segmentation, achieving mean Intersection over Union (mIoU) scores of 94.25% and 93.54% for UNet and PSPNet networks, respectively. For the tomato leafminer, Uygun et al. [9] employed the YOLOv8l-Seg model for training, demonstrating that the proposed model and methodology can effectively detect damage caused by the pest. Chen et al. [10] proposed a new approach, local maximum of boundary intensity (LMBI), which generates segmentation candidates from a region growing process to achieve quantitative analysis of nematodes in soil. To identify precious tree pests in real complex natural environments, Guo et al. [11] proposed an improved instance segmentation method that replaces the backbone network of Mask R-CNN with Swin Transformer. The experimental findings demonstrated that the suggested method successfully segmented pests in a variety of situations, including shaded, overlapped, and foliage- and branch-obscured pests.

Although instance segmentation techniques have witnessed remarkable advancements in general object domains (e.g., pedestrians, vehicles), their deployment in the context of silkworm segmentation, especially under high-density breeding scenarios, has remained

largely underexplored. Unlike pedestrians and vehicles, silkworm instance segmentation encounters distinct challenges stemming from the extremely high density, flexible morphological variations, and severe mutual occlusion among silkworms. To address these challenges, a dedicated silkworm dataset was first constructed, and a systematic evaluation of state-of-the-art instance segmentation methods was conducted, including YOLOv5n-seg, YOLOv5s-seg, YOLACT, SOLOv1, SOLOv2, and Mask R-CNN. Among these, SOLOv2 was identified as the most promising candidate due to its superior segmentation accuracy and was thus selected as the baseline model for further optimization. Subsequently, architectural refinements were introduced to enhance SOLOv2's capability in handling densely overlapping silkworms. First, to overcome the limitations of the original ResNet backbone in feature extraction for dense silkworm clusters in SOLOv2, a Linear Deformable Convolution (LDConv) was incorporated, enabling better adaptation to morphological diversity. Second, conventional stride convolution was replaced with a Haar Wavelet Downsampling (HWD) module to improve feature extraction for partially visible silkworms. Third, an Edge-Augmented Multi-attention Fusion Network (EAMF-Net) was integrated between network layers to address geometric continuity issues in occluded regions. For the issue of low-quality mask generation in dense silkworm segmentation, the mask feature branch of SOLOv2 was enhanced through the sequential integration of Dynamic Upsampling (DySample), Adaptive Spatial Feature Fusion (ASFF), and Simple Attention Module (SimAM), thereby producing higher-quality mask features.

The main contributions of this paper are as follows:

- Dataset Construction: A dedicated dataset for silkworm segmentation was built, comprising 1264 images captured under high-density breeding conditions to encompass diverse scenarios. This dataset provides essential data support for silkworm instance segmentation research.
- Baseline Model Identification: A systematic comparison of mainstream instance segmentation methods (including YOLOv5n-seg, YOLOv5s-seg, YOLACT, SOLOv1, SOLOv2, and Mask R-CNN) was conducted for silkworm segmentation. Based on comprehensive evaluation metrics, SOLOv2 was identified as the optimal baseline model.
- Enhanced SOLOv2 Architecture: An improved SOLOv2 architecture with dual-branch modifications was proposed. In the ResNet backbone network, three modules (LDConv, HWD, and EAMF-Net) were introduced to adapt to silkworm morphological variations, enhance feature extraction for partially visible targets, and resolve geometric continuity in occluded regions. In the mask feature branch, three components (DySample, ASFF, and SimAM) were sequentially integrated for improving mask generation quality.
- Comprehensive Experimental Validation: Extensive experiments were conducted on the self-constructed dataset, with both qualitative and quantitative results validating the effectiveness and accuracy of the proposed algorithm for silkworm segmentation. Notably, the method demonstrates superior segmentation precision for partially visible silkworms and achieves higher boundary adherence when segmenting overlapping silkworm regions.

2. Materials and Methods

2.1. Overall Workflow

To achieve accurate instance segmentation of silkworms in high-density overlapping scenarios, this study first constructed a silkworm image dataset that is consistent with real breeding environments. A professional image acquisition system was employed to capture silkworm images, and a standardized dataset was formed by integrating data augmentation and detailed annotation. Subsequently, systematic performance testing and comparative

analysis were conducted on current mainstream instance segmentation methods using this self-built dataset, leading to the selection of SOLOv2—the most suitable method for silkworm segmentation tasks—as the baseline model. Furthermore, based on the problems exposed by the baseline model in experiments, such as the missed segmentation of partially visible silkworms and inaccurate boundaries in overlapping regions, targeted improvement strategies were designed from two dimensions: feature extraction of the backbone network and quality optimization of the mask branch. Modules including Linear Deformable Convolution (LDConv), Haar Wavelet Downsampling (HWD), Edge-Augmented Multi-attention Fusion Network (EAMF-Net), Dynamic Upsampling (DySample), Adaptive Spatial Feature Fusion (ASFF), and Simple Attention Module (SimAM) were introduced to construct the enhanced SOLOv2 algorithm. Finally, through comparative experiments and visual analysis, the effectiveness of the improved model was fully verified, and its performance advantages in silkworm instance segmentation tasks were thoroughly analyzed.

2.2. Dataset Construction

The dataset construction process is illustrated in Figure 1. First, the silkworm images were collected and augmented. Subsequently, instance-level mask annotations were performed. Following this, the dataset was partitioned into training, testing, and validation subsets. Finally, the dataset was converted into the standard COCO dataset format compatible with MMDetection. The detailed construction process will be elaborated on in the subsequent sections.

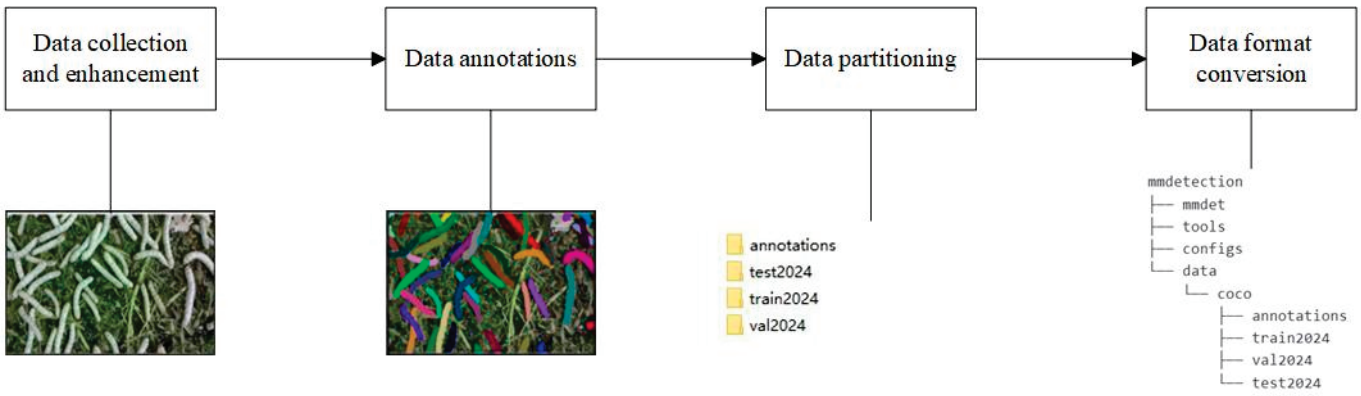


Figure 1. Silkworm data construction.

The data collection was conducted in the silkworm rearing rooms of silkworm farmers in Lezhi County, Sichuan Province, China, as illustrated in Figure 2. A specialized image acquisition system was constructed to obtain real silkworm data, consisting of an industrial color camera (IDS uEye SE UI-3130SE. Produced by Guangzhou Hongke Electronics Technology Co., Ltd., Guangzhou, China) with 1280 × 720 pixel resolution, two linear bar light sources, an adjustable mounting apparatus, and a data processing workstation. The camera was positioned approximately 0.2 m above the specimen tray, enabling consistent capture of the silkworms at densities of about 70 individuals per frame. To obtain silkworm data under different activity states, the data acquisition process was divided into two distinct phases: pre-feeding and post-feeding periods. Given the characteristically slow locomotion of silkworms, the imaging system was configured to capture frames at 5 s intervals to mitigate redundant data accumulation caused by excessive frame rates. The entire data collection spanned a total duration of 2 h.



(**a**) (**b**) (**c**)

Figure 2. Data acquisition: (**a**) Hardware; (**b**) Silkworm image; (**c**) Annotated image.

To enhance the model's robustness and generalization capabilities, multiple data augmentation strategies were applied to the collected silkworm images, including brightness adjustment, noise addition, and random rotation. After obtaining the augmented silkworm images, segmentation annotation of the silkworms was performed to create the silkworm dataset. This study employed the interactive semi-automatic image annotation tool (ISAT) for silkworm annotation. Following annotation, the dataset was partitioned into training, testing, and validation sets based on quantity and image redundancy in an 8:1:1 ratio. The final constructed silkworm dataset comprises 1264 dense silkworm images, with 1012 training images, 126 test images, and 126 validation images. To ensure the representativeness of each subset, the entire dataset was uniformly shuffled before partitioning, and the split was performed randomly to avoid sample bias caused by non-random grouping.

2.3. Baseline Model Selection for Silkworm Instance Segmentation

Since the inception of instance segmentation by Hariharan et al. [12] in 2014, numerous algorithms have emerged. To identify the optimal algorithm for silkworm segmentation, this study systematically evaluated the mainstream single-stage (e.g., YOLACT, SOLO, SOLOv2, YOLOv5n-seg, YOLOv5s-seg) and two-stage (e.g., Mask R-CNN) instance segmentation architectures. Quantitative experiments and qualitative analyses demonstrated that SOLOv2 achieved the highest segmentation accuracy (see [13]) for experimental details). Consequently, SOLOv2 was adopted as the baseline framework.

Despite its superior accuracy, SOLOv2 exhibits limitations in real-world silkworm breeding scenarios characterized by high-density distributions, severe inter-silkworm overlapping, and frequent occlusion by mulberry leaves. Under these conditions, SOLOv2 suffers from (1) missed segmentation of partial visual silkworms, (2) over-segmentation of individual silkworms, and (3) fragmented segmentation of overlapping instances with poor boundary adherence. These deficiencies undermine the reliability of key metrics in intelligent farming systems, such as silkworm area and perimeter estimation. To address these challenges, an enhanced SOLOv2 architecture was designed.

2.4. Enhanced SOLOv2 Model

This section first outlines the original SOLOv2 architecture, followed by a detailed description of our proposed enhancements.

2.4.1. SOLOv2 Model Overview

As illustrated in Figure 3, the SOLOv2 architecture comprises three main components: a ResNet backbone network, a Feature Pyramid Network (FPN), and an instance segmentation network [14]. The ResNet backbone is primarily responsible for extracting image

features. The FPN enhances the model's capability to detect objects at different scales. The instance segmentation network, serving as the core component of SOLOv2, performs segmentation of individual object instances in images through three parallel processing branches: the category prediction branch (Category Branch), the convolutional kernel prediction branch (Kernel Branch), and the mask feature branch (Mask Feature Branch). SOLOv2 adopts a simple yet effective single-stage instance segmentation framework that achieves high segmentation accuracy. By simultaneously predicting semantic categories and mask categories, it enables both distinction between different objects and identification of similar objects across various regions in an image. Our improvements focus on optimizing the ResNet backbone and enhancing the mask feature branch (highlighted in red in Figure 3).

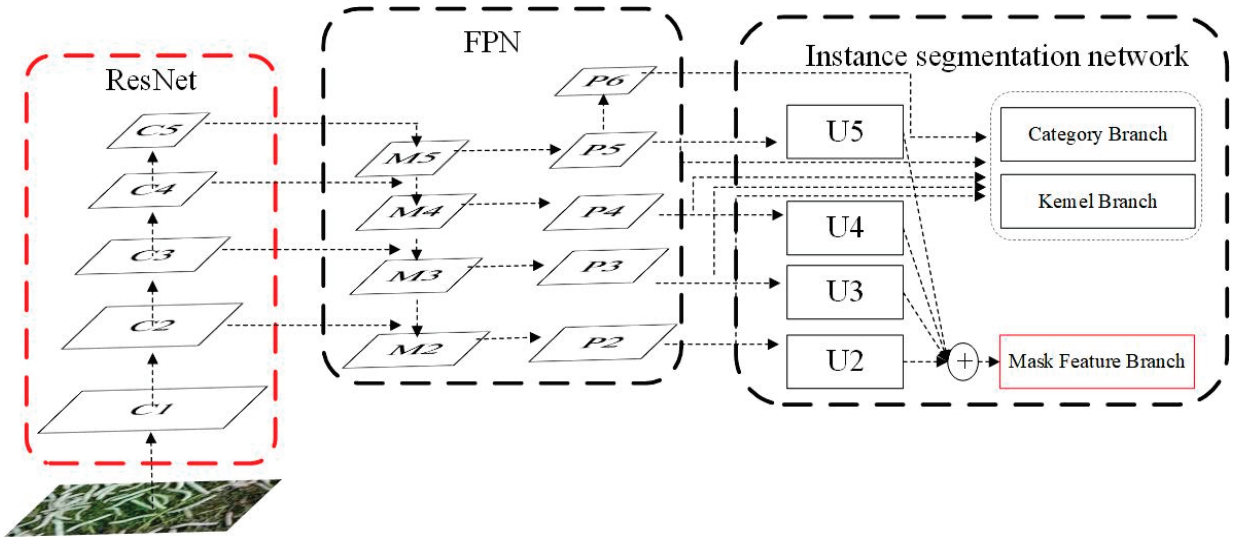


Figure 3. The architecture of Solov2.

2.4.2. Feature-Enhanced Backbone Network

The SOLOv2 takes ResNet-50 as the backbone. On this basis, this study introduces LDConv, HWD, and EAMF-Net to enhance the feature extraction ability.

1. Linear Deformable Convolution (LDConv)

The ResNet-50 uses traditional convolution to extract target features; however, the traditional convolution suffers from limited receptive fields and rigid geometric sampling structures, making it ineffective for silkworm segmentation, where targets exhibit significant posture changes and geometric discontinuities due to occlusion.

In response to this, this paper introduces an LDConv module in the ResNet-50 to replace the original traditional convolution module. LDConv was proposed by Zhang et al. [15]; it can obtain a variety of sampling shapes. Specifically, LDConv generates the original coordinate according to the number of parameters of the convolution kernel firstly, and then learns the offset through the convolution operation and adds the offset to the initial sampling coordinates to generate a new modified coordinate. Finally, the feature of the corresponding position is extracted by interpolation and resampling, and the convolution operation is applied to output the result. Compared with standard convolution and deformable convolution, it changes the growth trend of the number of parameters into linear growth and reduces the computational overhead.

2. Haar Wavelet-Based Downsampling (HWD)

In ResNet-50, downsampling is implemented through strided convolutions. As the convolution stride increases, the step size of the convolutional kernel on the input data

becomes larger, leading to two potential issues. Firstly, degraded local feature capture: The increased stride reduces the coverage density of the convolutional kernel over local regions, making it difficult for the network to capture fine-grained features. Secondly, feature sparsity: The feature density in the output feature maps decreases, leading to sparser feature distributions, which reduces the model's sensitivity to details. These issues directly impair the model's ability to identify small target features, ultimately degrading performance in classification and segmentation tasks.

In silkworm segmentation, significant size variations exist among silkworms, and partial visual silkworms frequently appear at image boundaries due to the field-of-view (FOV) truncation. Traditional downsampling methods progressively lose detailed information about these small targets, as the resolution of feature maps decreases across layers. To address this problem, an HWD module was introduced to replace the strided convolutions in ResNet-50. The HWD module, proposed by Xu et al. [16], utilizes the Haar wavelet transform to decompose input feature maps into low-frequency and high-frequency components. The low-frequency components retain the main structural and global semantic information, whereas the high-frequency components represent detailed features, such as edges and textures. By fusing multi-frequency information, this module preserves both the primary morphological characteristics and fine details of silkworm targets during the downsampling process. Furthermore, this module can directly replace traditional pooling or strided convolution layers, thereby enhancing the model's adaptability to multi-scale features with minimal computational overhead.

3. Edge-Augmented Multi-attention Fusion Network (EAMF-Net)

To address the challenges of geometric discontinuity and weak edge segmentation caused by overlapping and occluded silkworms, an EAMF-Net was proposed. The EAMF-Net innovatively integrates four components: a Channel Attention Module (CAM), a Spatial Attention Module (SAM), an Edge Enhancement Module (EEM), and a Multi-attention Fusion Module (MAFM). As illustrated in Figure 4, the CAM and SAM are connected sequentially, while the EEM is incorporated in parallel through a bypass connection. Finally, the feature maps are summed and fed into the MAFM to generate the refined output.

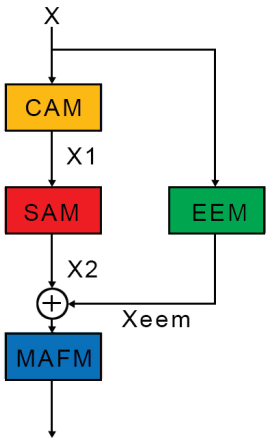


Figure 4. Structural Diagram of the EAMF-Net Module.

Within this framework, the CAM models cross-channel feature dependencies to enhance selective attention across channel dimensions, enabling better association and utilization of critical information. The SAM performs spatial sensitivity analysis to strengthen feature discriminability in spatial dimensions, improving localization and recognition of geometrically discontinuous silkworms. The EEM amplifies edge contours and texture features, facilitating precise boundary detection of overlapping specimens. The MAFM

synthesizes multi-attention features through learnable fusion strategies. The detailed architectures of these modules are elaborated in the following text.

CAM: As illustrated in Figure 5a, the input feature map $X^{B\times C\times H\times W}$ is processed through parallel Adaptive Average Pooling (AAP) and Adaptive Max Pooling (AMP) operations to generate two 1D channel-wise feature weight vectors: $K_{CM}^{B\times C\times 1\times 1}$ and $K_{CA}^{B\times C\times 1\times 1}$. These vectors are then summed to produce $K_{C}^{B\times C\times 1\times 1}$. A 3 × 3 1D convolutional layer is applied to fuse the weight information, followed by a Sigmoid activation function to generate the final channel attention weights. The detailed mathematical formulation is provided in Equation (1)

$$X_1 = \mathrm{CAM}(X) = \sigma\left(\mathrm{C1D}_{3\times 3}\left(P_{\max}(X) + P_{\mathrm{avg}}(X)\right)\right) \circledast X \tag{1}$$

where CAM() denotes the channel attention function; $P_{\max}()$ represents the max pooling operation; $P_{\mathrm{avg}}()$ indicates the average pooling operation; $\mathrm{C1D}_{3\times 3}()$ corresponds to a 1D convolution, where the subscript specifies the kernel size; $\sigma()$ signifies the Sigmoid activation function; and $\circledast$ denotes the Hadamard product operation; X_1 is the output of CAM.

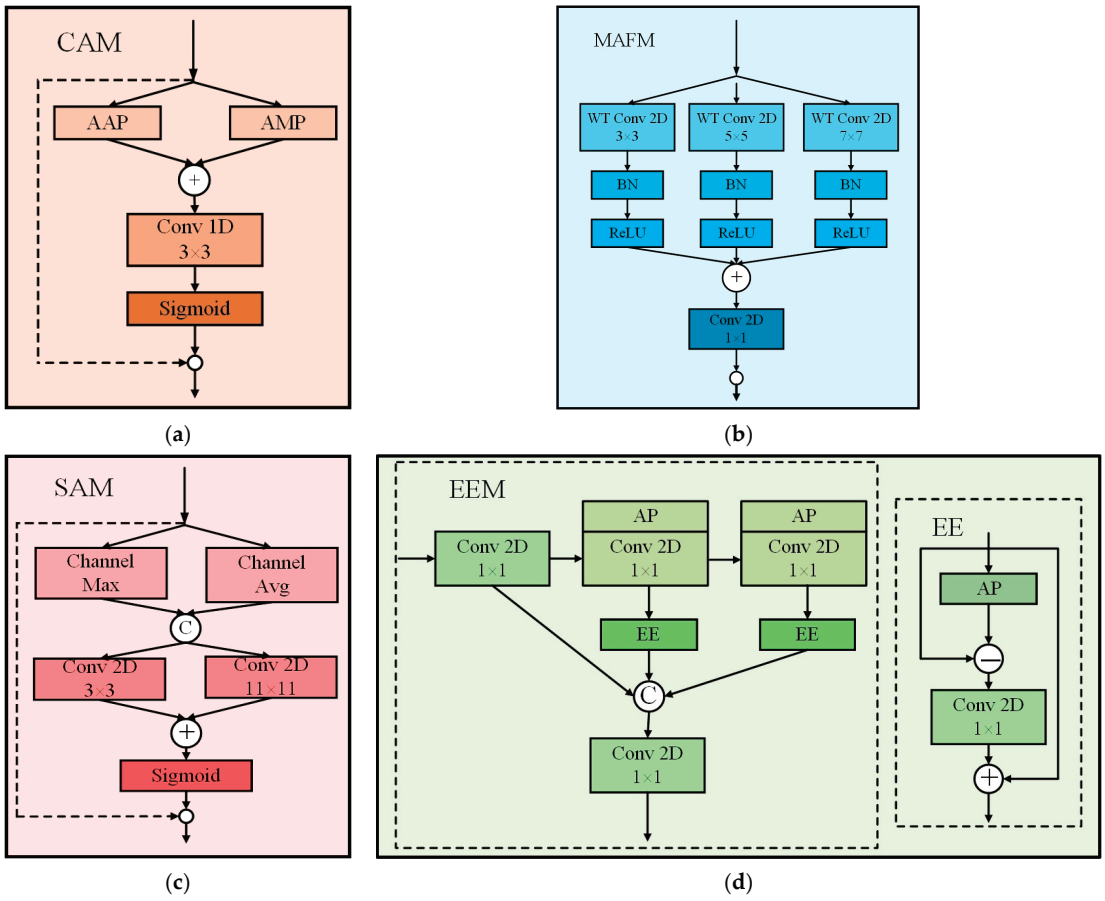


Figure 5. Block diagrams of the structures of each module: (**a**) CAM; (**b**) MAFM; (**c**) SAM; (**d**) EEM.

SAM: As shown in Figure 5c, the feature map $X_1^{B\times C\times H\times W}$ output by the CAM is fed into the SAM. Within SAM, the channel-wise maximum $K_{SM}^{B\times 1\times H\times W}$ and channel-wise average $K_{SA}^{B\times 1\times H\times W}$ are extracted from $X_1^{B\times C\times H\times W}$. These features are concatenated along the channel dimension to form $K_{S}^{B\times 2\times H\times W}$. Parallel 3 × 3 and 11 × 11 convolutional layers are applied to $K_{S}^{B\times 2\times H\times W}$, and their outputs are summed and normalized via a Sigmoid function to produce $X_2^{B\times C\times H\times W}$. The detailed formulation is given in Equation (2)

$$X_2 = \mathrm{SAM}(X) = \sigma\left(C_{3\times3}(\mathrm{Ct}[\mathrm{Ch}_{\max}(X_1),\mathrm{Chavg}(X_1)]) + C_{11\times11}(\mathrm{Ct}[\mathrm{Ch}_{\max}(X_1),\mathrm{Ch}_{\mathrm{avg}}(X_1)])\right) \circledast X_1 \quad (2)$$

where $C_{3\times3}(\cdot)$ denotes standard 2D convolution with a 3 × 3 kernel; $\mathrm{Ch}_{\mathrm{avg}}(\cdot)$ is channel-wise average pooling; $\mathrm{Ch}_{\max}(\cdot)$ represents channel-wise max pooling; and Ct[·] indicates channel-wise concatenation.

EEM: As depicted in Figure 5d, the core structure of this module is the Edge Enhancement (EE) operation [17]. The EE module improves the segmentation accuracy of overlapping or weak edges in silkworm images by subtracting the feature average pooling results from the original feature map. Let $X_e^{B\times C_e\times H\times W}$ denote the input feature map of the EE module, its output X_{ee} is formulated in Equation (3)

$$X_{ee} = \mathrm{EE}(X_e) = X_e + C_{1\times1}\left(X_e - P_{\mathrm{avg}}(X_e)\right). \quad (3)$$

MAFM: As illustrated in Figure 5b, the summed features of $X_2^{B\times C\times H\times W}$ and $X_{ee}^{B\times C\times H\times W}$ are used as the input to the MAFM, with the computational process defined in Equations (4) and (5). Within the MAFM, three parallel wavelet-based convolutional layers with varying kernel sizes are employed to fuse multi-scale features extracted from preceding modules

$$X_3 = X_{ee} + X_2 \quad (4)$$

$$X_4 = C_{1\times1}\left(\mathrm{Ct}\left[\mathrm{WTC}_{3\times3}(X_3), \mathrm{WTC}_{5\times5}(X_3), \mathrm{WTC}_{7\times7}(X_3)\right]\right). \quad (5)$$

The proposed EAMF-Net is incorporated into the SOLOv2 backbone network after Stages 1 to 4, with its detailed architectural configuration illustrated in Figure 6. Where H and W represent the height and width of the image, respectively. During training, the image size is randomly selected from [(720, 1280), (640, 1280), (560, 1280), (480, 1280)], while a unified size of (560, 1280) is used during testing.

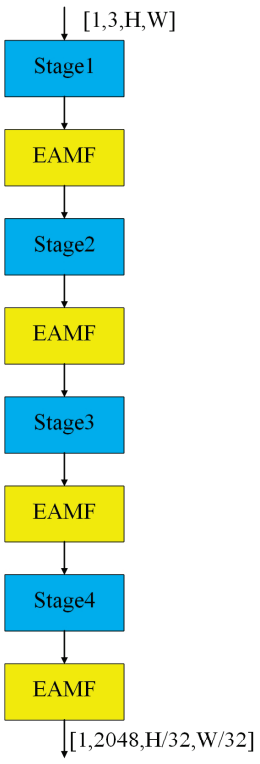


Figure 6. Architecture diagram of the backbone network with the integrated EAMF-Net.

The EAMF-Net is integrated into the backbone network, with the specific connectivity mechanism between residual blocks of adjacent stages illustrated in Figure 7.

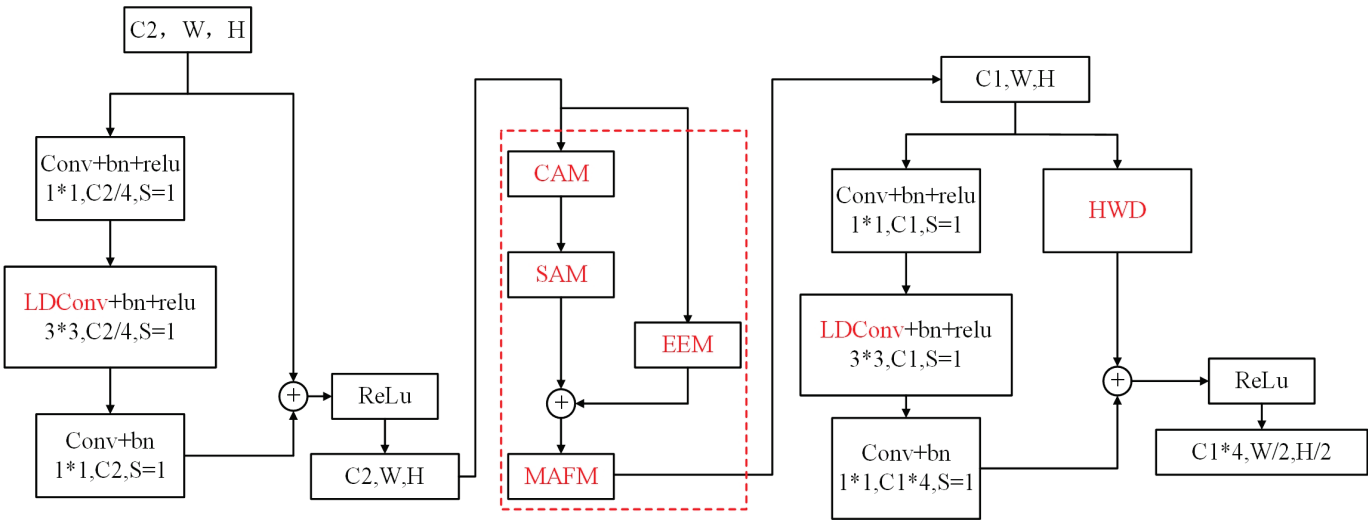


Figure 7. Schematic diagram of the connection between the EAMF-Net module and the residual blocks in the preceding and succeeding stages.

2.4.3. Enhanced Mask Feature Branch Network

The core functionality of the SOLOv2 mask feature branch lies in constructing foundational features for instance segmentation through multi-scale feature fusion. As depicted in Figure 7, this module processes four-level feature maps from the Feature Pyramid Network (FPN). Initially, hierarchical features are unified to 1/4 of the input image resolution via convolutional operations and bilinear upsampling to achieve spatial dimension standardization. Subsequently, multi-scale information is integrated through progressive summation of feature maps, yielding a composite feature tensor for mask prediction. To enhance silkworm segmentation quality, this study augments the original SOLOv2 mask feature branch by incorporating three novel components: (1) the DySample network that replaces fixed bilinear interpolation with learnable upsampling for adaptive spatial detail recovery; (2) the ASFF network that optimizes cross-scale feature integration using channel-wise and spatial attention mechanisms; and (3) the SimAM, which strengthens feature discriminability without computational overhead. This refined architecture preserves the primary morphological integrity of silkworm targets while effectively capturing fine-grained edge and texture details, thereby improving segmentation accuracy for partially visible specimens and multi-scale objects in complex imaging scenarios.

1. Dynamic Upsampling Module (Dysample)

In Figure 7, when the summation of the four-layer feature maps (P2 to P5) is performed, upsampling is required for feature maps of varying resolutions. In the original SOLOv2 model, bilinear interpolation is employed for this operation. However, this method fails to account for the influence of grayscale gradient variations among adjacent pixels on interpolation results, which often leads to the loss of edge and texture information after upsampling. Consequently, the clarity and fine details of target objects are inadequately preserved, resulting in edge blurring. To address this limitation, the DySample upsampling module, proposed by Liu et al. at ICCV 2023 [18], is introduced. Its core principle is defined as point sampling, where content-aware sampling points are generated by learning coordinate offsets from input feature maps, enabling adaptive resampling of feature maps.

2. Adaptively Spatial Feature Fusion (ASFF)

In the original SOLOv2, the summed P2-P5 features are spatially normalized and directly aggregated to form a composite feature tensor, which is then convolved to generate the mask feature. This approach neglects semantic discrepancies across hierarchical feature levels during fusion, which can induce cross-target feature confusion and geometric distortions in instance mask shapes. To mitigate this, an ASFF module [19] is integrated. By leveraging adaptive pooling, secondary weight learning for semantic information across different hierarchical levels is facilitated, thereby enhancing the fusion of local features and improving segmentation accuracy under occlusion scenarios.

3. Simple Attention Module (SimAM)

The SimAM, a lightweight and parameter-free attention module, is designed to emphasize feature saliency by suppressing activations of neighboring neurons [20]. This module is demonstrated to significantly boost recognition accuracy in dense scenes for object detection and enhance boundary sensitivity in image segmentation models. Unlike conventional channel or spatial attention mechanisms, 3D attention weights for each neuron are inferred by computing feature similarity through SimAM, thereby amplifying critical features while suppressing irrelevant ones without introducing additional parameters to the baseline network. In this work, the SimAM module is appended to the final layer of the segmentation head, following the output of the ASFF module.

The improved mask feature branch is illustrated in Figure 8, where components highlighted in red are defined as the novel modules introduced in this study. Where H and W represent the height and width of the image, respectively. During training, the image size is randomly selected from [(720, 1280), (640, 1280), (560, 1280), (480, 1280)], while a unified size of (560, 1280] is used during testing.

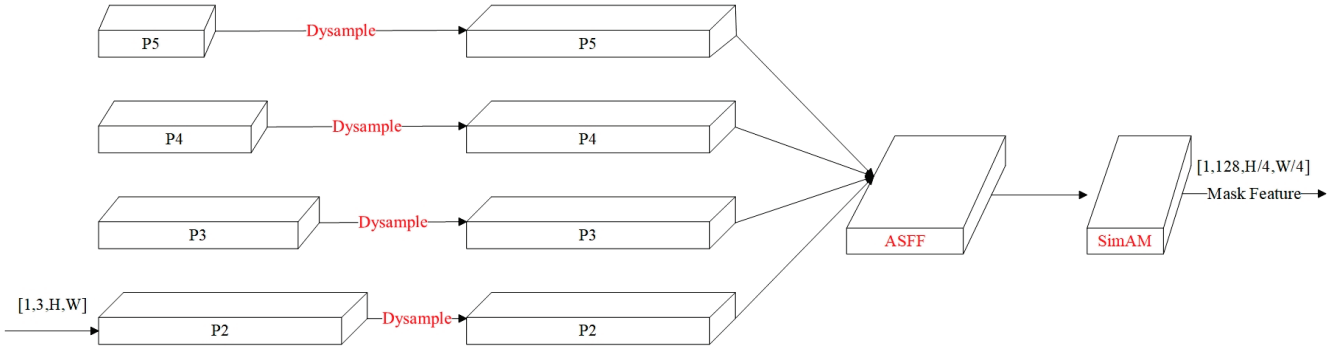


Figure 8. Enhanced mask feature branch network.

3. Experimental Results and Discussion

3.1. Experimental Setup

The experimental hardware configuration was configured as follows: an Intel Core 13900KF processor, an NVIDIA GeForce RTX 3090 GPU, and a Windows 10 operating system. The software environment was implemented using the PyTorch 1.10.0 deep learning framework, CUDA 11.3 for computational acceleration, Python 3.8.10, and the MMdetection 3.0.0 framework. During the model training phase, hyperparameters were set as follows: the initial learning rate was fixed at 0.00125 to balance training stability and convergence acceleration; a momentum factor of 0.9 was applied to optimize convergence and suppress oscillations; and a weight decay coefficient of 0.0001 was adopted to prevent overfitting. The model was trained for 36 epochs with a batch size of 1 using the stochastic gradient descent (SGD) optimizer. Input images were resized to 1280 $\times$ 720 pixels to balance computational efficiency and detail preservation. Evaluation results were recorded, and weight files were saved once per epoch.

3.2. Results Analysis

To validate the performance of the proposed algorithm in silkworm segmentation, comparative experiments were designed as follows: (1) comparison between the proposed algorithm and the original SOLOv2; (2) performance comparison with mainstream instance segmentation algorithms.

3.2.1. Silkworm Segmentation Metrics

For silkworm segmentation in this paper, the IoU is first calculated, defined in Equation (6), where Mpred represents the predicted mask and Mgt represents the ground-truth mask

$$\text{IoU} = \frac{M_{pred} \cap M_{gt}}{M_{pred} \cup M_{gt}}. \tag{6}$$

In the MMdetection framework, by setting an IoU threshold, the precision and recall of segmentation can be calculated, as defined in Equations (7) and (8). Here, TP (true positive) indicates that positive samples are correctly identified; TN (true negative) means that negative samples are accurately identified; FP (false positive) represents that negative samples are incorrectly classified as positive; and FN (false negative) denotes that positive samples are missed or misclassified

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{8}$$

In this paper, the mean Average Precision (mAP) is used as the core performance metric. It is obtained by calculating the Average Precision (AP) for each detection category and then taking the arithmetic mean of these AP values. Its mathematical definition is shown in Equation (9)

$$\text{AP} = \int_0^1 P(R)dR \tag{9}$$

where P represents precision and R represents recall. Let N denote the total number of categories, then, the mAP can be expressed as Equation (10)

$$\text{mAP} = \frac{\sum_{i=1}^{N} AP_i}{N}. \tag{10}$$

In this paper, since the silkworm is the only object that needs to be segmented (N = 1), the AP and mAP are completely identical. Given that the dataset created in this paper follows the standard format of the COCO dataset, when training and testing instance segmentation models on the COCO dataset, common evaluation metrics include Segm_AP, Segm_AP$_{50}$, Segm_AP$_{75}$, Segm_AP$_{s}$, Segm_AP$_{m}$, and Segm_AP$_{l}$. Among them, Segm_AP represents the average precision within the IoU threshold range from 0.5 to 0.95. Segm_AP$_{50}$ and Segm_AP$_{75}$ represent the segmentation precision at IoU = 0.5 and IoU = 0.75, respectively. Segm_AP$_{s}$, Segm_AP$_{m}$, and Segm_AP$_{l}$ represent the segmentation precision for small objects (area less than 32^2 px), medium objects (area between 32^2 and 96^2 px), and large objects (area greater than 96^2 px), respectively. This paper selects Segm_AP, Segm_AP$_{s}$, Segm_AP$_{m}$, and Segm_AP$_{l}$ to measure the segmentation performance of the proposed algorithm.

3.2.2. Ablation Study on Improved SOLOv2

To investigate the impact of different modules on the segmentation performance of the SOLOv2 network, ablation experiments were conducted by incrementally adding proposed

modules to the baseline SOLOv2. In the table, "√" indicates the inclusion of a specific module, while "×" denotes its exclusion. The model size and instance segmentation metrics (AP, AP_s, AP_m, AP_l) are compared in Table 1.

Table 1. Ablation analysis of key components in dense silkworm segmentation.

Module	SOLOv2						AP	AP_s	AP_m	AP_l	Size (Mb)
Baseline	×	×	×	×	×	×	0.809	0.626	0.818	0.856	178
LDConv	√	×	×	×	×	×	0.817	0.638	0.823	0.867	160
HWD	√	√	×	×	×	×	0.828	0.669	0.837	0.872	208
Dysample	√	√	√	×	×	×	0.833	0.682	0.848	0.871	209
ASFF	√	√	√	√	×	×	0.840	0.687	0.847	0.876	209
SimAM	√	√	√	√	√	×	0.843	0.707	0.851	0.869	209
EAMF-Net	√	√	√	√	√	√	0.851	0.728	0.858	0.876	306

As shown in Table 1, the AP value was progressively improved by sequentially integrating LDConv, HWD, DySample, adaptive spatial fusion (ASFF), SimAM, and multi-attention fusion (EAMF-Net) into SOLOv2. The AP value was enhanced by 0.8%, 1.1%, 0.5%, 0.7%, 0.3%, and 0.8%, respectively, relative to the previous configuration. Overall, the AP value was increased by 4.2% in the final improved model. The segmentation accuracy for small-, medium-, and large-scale silkworm targets was significantly improved by 10.2%, 4.0%, and 2.0%, respectively. Notably, the HWD, EAMF-Net, and SimAM modules contributed to 3.1%, 2.1%, and 2.0% improvements in small-target segmentation accuracy. The model size was increased from 178 MB to 306 MB. The largest parameter expansion (97 MB) was observed when the EAMF-Net module was integrated into the backbone network. Conversely, replacing traditional convolutions with LDConv reduced the parameter size by 18 MB while improving segmentation performance.

3.2.3. Comparative Experiments with Mainstream Instance Segmentation Methods

To comprehensively evaluate the performance of the improved algorithm, comparative experiments were conducted between the enhanced SOLOv2 model and several high-performance instance segmentation algorithms, followed by detailed analysis. As instance segmentation represents a critical phase in image processing research, with its primary research period spanning from 2017 to 2021 before being largely superseded by panoptic segmentation tasks, this study selected two-stage instance segmentation algorithms—Mask Scoring R-CNN [21] and Cascade Mask R-CNN [22] and the single-stage algorithm CondInst [23]—as benchmarks. The former two models, as derivatives of the Mask R-CNN architecture, have long been recognized for their high accuracy, while CondInst also exhibits robust performance. The comparative results are presented in Figure 9.

As shown in Table 2, the proposed enhanced SOLOv2 achieves the highest overall average precision (AP: 0.851) among evaluated instance segmentation models, outperforming both two-stage methods (Mask Scoring R-CNN: 0.794; Cascade Mask R-CNN: 0.721) and single-stage counterparts (original SOLOv2: 0.809; CondInst: 0.814). Notably, it demonstrates substantial improvements in small-target segmentation (AP_s: 0.728 vs. original SOLOv2's 0.626), while maintaining leading performance on medium (AP_m: 0.858) and large objects (AP_l: 0.876). Despite a model size increase to 306 MB (72% larger than the original SOLOv2), it remains more compact than two-stage architectures (464–564 MB) and achieves real-time inference at 105.4 ms, balancing accuracy and efficiency for multi-scale segmentation tasks.



(**a**) (**b**)

Figure 9. The segmentation results of two methods: (**a**) result of SOLOv2; (**b**) result of our method.

Table 2. Performance comparison of different models.

Category	Model	AP	AP_S	AP_m	AP_l	Size (Mb)	Time (ms)
Two-stage	Mask Scoring R-CNN	0.794	0.753	0.786	0.834	464	120.8
	Cascade Mask R-CNN	0.721	0.734	0.704	0.791	564	140.2
Single-stage	SOLOv2	0.809	0.626	0.818	0.856	178	88.3
	CondInst	0.814	0.661	0.818	0.871	236	100.5
Ours		0.851	0.728	0.858	0.876	306	105.4

3.2.4. Visual Evaluation

Figure 9 presents a comparative case of the segmentation results between the SOLOv2 algorithm and the proposed method for silkworm instance segmentation. In Figure 9a, the blue rectangular boxes highlight various failure cases of the original SOLOv2 algorithm, including missed segmentation in overlapping silkworm regions (Region 1), missed segmentation of partially visible silkworms (Region 2), over-segmentation (Region 3), and inaccurate boundary segmentation where the predicted edges deviate from the true boundaries (Region 4). In Figure 9b, the method proposed in this paper addresses all the aforementioned issues. Figure 10 provides zoomed-in comparisons for these four cases.

In Figure 10a, comparing the left (SOLOv2) and right (proposed method) results reveals that the original SOLOv2 algorithm fails to segment overlapping silkworms, whereas our approach achieves accurate segmentation. Figure 10b shows a silkworm (bottom-left) with only minimal head visibility. While SOLOv2 fails to segment it, our method successfully detects and segments the silkworm. In Figure 10c, two overlapping silkworms are displayed, with their segmentation masks represented by different colors. The SOLOv2 result incorrectly assigns multiple colors to each silkworm (indicating over-segmentation), while our method generates a single, consistent mask for each silkworm. Finally, Figure 10d demonstrates that SOLOv2 produces a significant boundary misalignment at overlapping regions (left), whereas our method achieves highly precise boundary adherence (right).

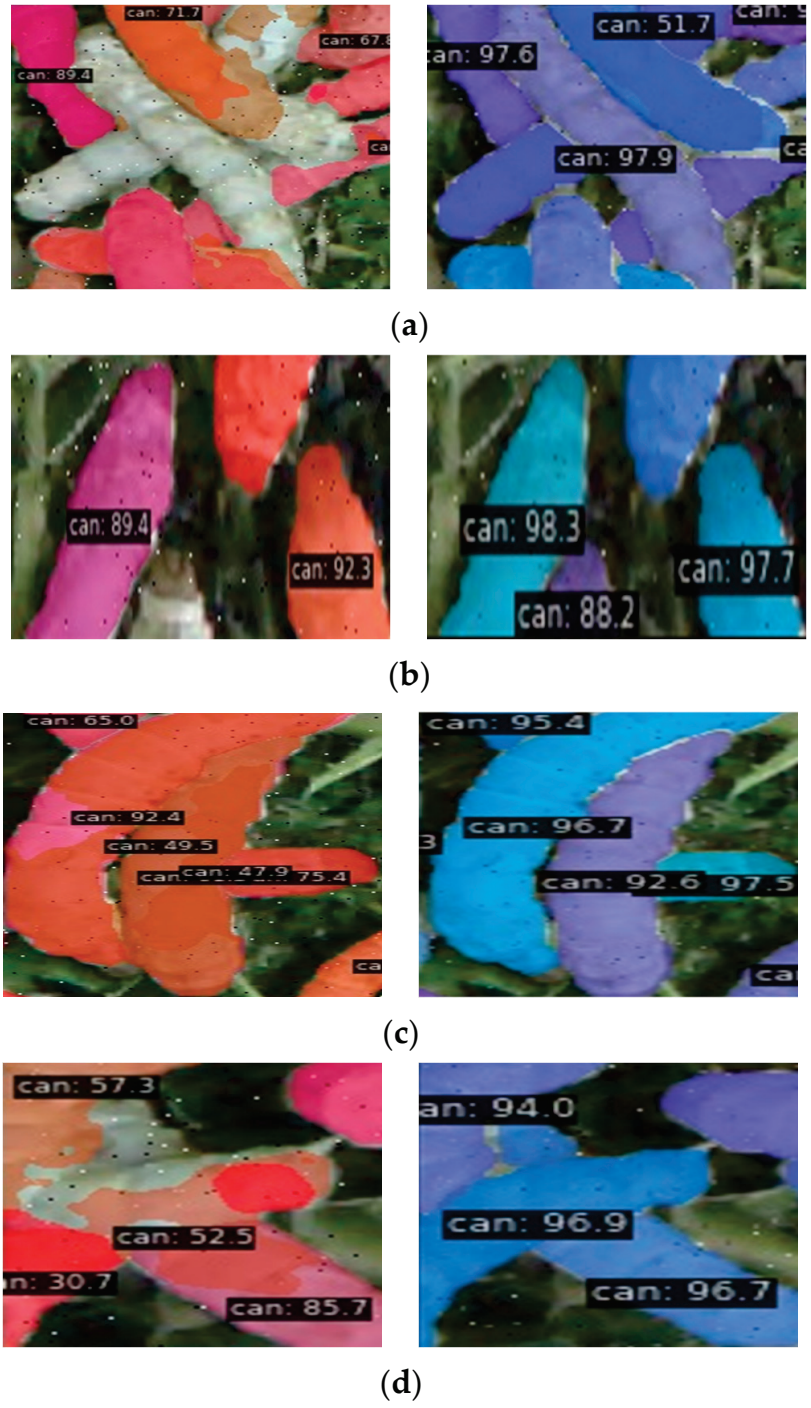


Figure 10. Zoom in on areas 1 to 4 in Figure 9: (**a**) zoom in on area 1 (left: SOLOv2, right: our method); (**b**) zoom in on area 2 (left: SOLOv2, right: our method); (**c**) zoom in on area 3 (left: SOLOv2, right: our method); (**d**) zoom in on area 4 (left: SOLOv2, right: our method).

Figure 11 presents additional segmentation examples. Due to space limitations, only the full-image segmentation results are displayed here, without zoomed-in comparative analysis. However, it is evident that the segmentation inaccuracies present in Figure 11a,c are effectively resolved in Figure 11b,d.

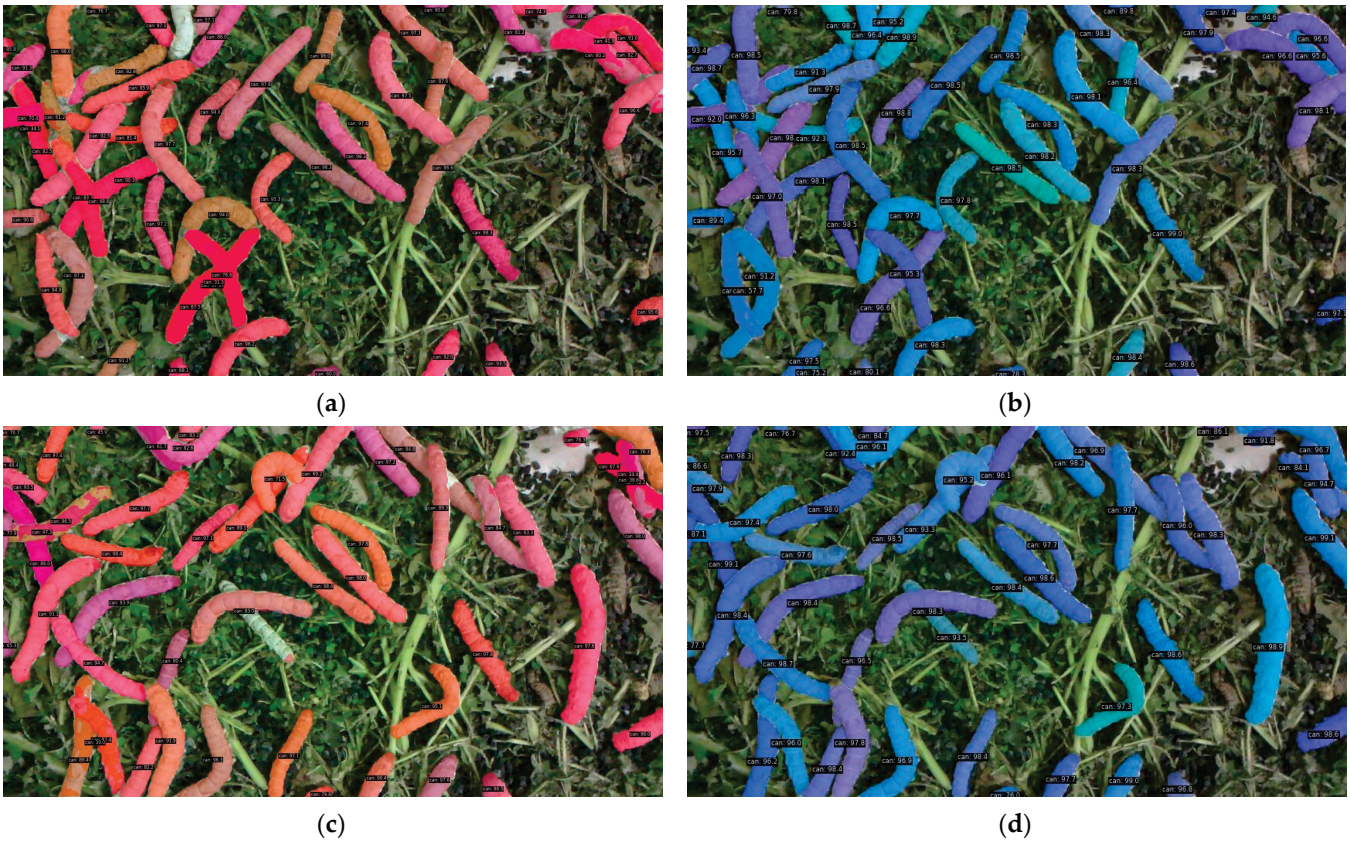


Figure 11. Segmentation examples: (**a**) result of SOLOv2; (**b**) result of our method; (**c**) result of SOLOv2; (**d**) result of our method.

3.2.5. Segmentation Confidence

The confidence of all instances in the images from the experimental results was statistically analyzed. As shown in Table 3, after model refinement, our proposed method demonstrates significant improvements in confidence distribution compared to the original model. Specifically for SOLOv2, only 9.61% of instances achieved high confidence (score > 0.9), whereas our approach substantially increased this proportion to 88.8%. Conversely, the percentage of low-confidence instances (score < 0.7) decreased dramatically from 19.57% in the baseline to merely 1.50% in our enhanced method.

Table 3. Segmentation results with different confidence levels (high/medium/low).

Confidence Levels	SOLOv2 (%)	Ours (%)
Highly confident (>0.9)	9.61	88.80
Confident (0.7–0.9)	78.83	9.70
Low confidence (<0.7)	19.57	1.50

3.3. Feasibility Analysis for Deployment on Low-Resource Devices

The memory capacity of low-resource devices is usually limited (e.g., the Jetson Nano has 4 GB of memory, and embedded terminals typically have 1–2 GB of memory), so the model's parameter scale and runtime memory usage are core constraints. As shown in Table 2, the model size of enhanced SOLOv2 is 306 MB. Although this is a 72% increase compared to the original SOLOv2 (178 MB), it is significantly smaller than that of two-stage models (Mask Scoring R-CNN: 464 MB; Cascade Mask R-CNN: 564 MB) and far below the memory upper limit of low-resource devices (easily loadable on devices with 4 GB of memory).

In terms of runtime memory usage, the memory consumption of the model during the inference phase mainly comes from feature map storage. Based on the tensor dimension design in Sections 2.4.2 and 2.4.3, the maximum dimension of the feature map output by the backbone network is B × 512 × 80 × 45 (approximately 1.8 MB when B = 1), and the maximum dimension of the feature map in the mask branch is B × 256 × 320 × 180 (approximately 18.4 MB when B = 1). The total feature map memory usage for single-sample inference is less than 50 MB, which can meet the requirements even on embedded devices with 1 GB of memory.

In addition, the model supports dynamic batch inference (inference latency of 105.4 ms when B = 1, Table 2). The memory pressure can be further reduced by controlling the batch size, enabling adaptation to low-memory devices. Therefore, the model proposed in this paper can be deployed on low-resource devices.

4. Conclusions

Instance segmentation of silkworms is the foundation for achieving intelligent sericulture; however, there is currently limited research in this area. This paper explores instance segmentation of silkworms. Firstly, a dataset specifically for silkworm segmentation was self-constructed, and current mainstream instance segmentation algorithms (Mask R-CNN, YOLACT, YOLO v5s, YOLO v5n, SOLOv1, and SOLOv2) were tested on this dataset. The SOLOv2 algorithm, which achieved the highest segmentation accuracy, was selected as the baseline model for silkworm segmentation. Subsequently, to address the issues of missed segmentation of incomplete silkworms and mis-segmentation of overlapping silkworms that still persisted when using SOLOv2 for silkworm segmentation tasks, an enhanced SOLOv2 model tailored for silkworm segmentation was designed. Specifically, to tackle the problem of the ResNet backbone network's insufficient feature extraction capability for dense silkworms, LDConv was introduced into the ResNet backbone network to better adapt to the variable morphologies of silkworms. Traditional strided convolution was replaced with an HWD to enhance the feature extraction capability for small, incomplete silkworm targets. An edge-enhanced multi-attention fusion network was incorporated between stages to better parse the geometric continuity of occluded silkworms. Additionally, to address the low-quality mask generation for dense silkworm segmentation in the mask feature branch network, DySample, ASFF, and the SimAM parameter-free attention module were sequentially introduced to generate higher-quality mask features.

The experimental results demonstrate that the proposed algorithm in this paper improves upon the original SOLOv2 by 4.2%, 10.2%, 4.0%, and 2.0% in terms of the AP, APs, APm, and APl metrics, respectively. Visualization of the segmentation results reveals that the proposed algorithm enhances the segmentation performance for small, incomplete silkworm targets and overlapping silkworms. Meanwhile, to fully demonstrate the superiority of the improved algorithm, comparative experiments were also conducted with Mask Scoring R-CNN, Cascade Mask R-CNN, and CondInst. The improved model proposed in this paper also achieved certain improvements in the AP, AP_s, and AP_l metrics. In addition to silkworm segmentation, the method proposed in this paper can also provide a reference for the instance segmentation of other agricultural pests such as wheat aphids and orchard red spiders.

Author Contributions: Conceptualization, Project administration, Resources, Writing—original draft, J.Y.; Data curation, Software, Validation, Visualization, Writing—original draft; H.L.; Formal analysis, Investigation, Writing—original draft; C.C.; Methodology, Software; Z.L.; Funding acquisition, Writing—review and editing; D.G.; Writing—review and editing; S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Sichuan Province Science and Technology Support Projects, China (2023YFN0077, 2024YFFK0442, 2023YFN0009, 2022YFN0020, 2024YFNH0018, 2025YFHZ0031), Xizang Autonomous Region Science and Technology Support Project, China (CGZH2024000151), and Chengdu Science and Technology Project (2024-YF08 00027-GX, 2024-YF05-02414-SN), Key Laboratory of Agricultural Equipment Technology for Hilly and Mountainous Areas, Ministry of Agriculture and Rural Affairs, China (2024QSNZ04).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yang, S.; Zuo, X.; Zhao, Z. Research on bombyx mori recognition in rearing seats based on image processing technology. *Acta Sericol. Sin.* **2022**, *48*, 5.
2. Hu, X.; Qian, L.; Zhang, D. Research on segmentation technology of *Nosema bombycis* images. *Mod. Electron. Tech.* **2018**, 41–44.
3. Luo, Y.; Wen, C.; Wen, J.; Xu, Q.; Liang, X.; Yang, J.; Lan, Z. Data density-based RGB color component feature extraction and area measurement method for silkworm. *J. Asia Pac. Entomol.* **2023**, *26*, 102153. [CrossRef]
4. Gowramma, B.H.; Poornima, B.; Ranjana, B.J. Enhancing Silkworm Breeding Through Advanced Image Segmentation: A Comparative Study of U-Net, Mask R-CNN, And FCN. In Proceedings of the 2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC), Davangere, India, 24–25 October 2024.
5. Wang, Y.; Jiang, X.; Liu, Y. Design and implementation of silkworm detection system based on deep learning. *Electron. Compon. Inf. Technol.* **2024**, *8*, 67–70.
6. He, R.; Zheng, K.; Wei, Q. Research on silkworm recognition and counting in industrialized rearing based on improved Mask R-CNN model. *Smart Agric.* **2022**, *4*, 163–173.
7. Meng, K.; Xu, K.; Cattani, P.; Mei, S. Camouflaged cotton bollworm instance segmentation based on PVT and Mask R-CNN. *Comput. Electron. Agric.* **2024**, *226*, 109450. [CrossRef]
8. Tassis, L.M.; de Souza, J.E.T.; Krohling, R.A. A deep learning approach combining instance and semantic segmentation to identify diseases and pests of coffee leaves from in-field images. *Comput. Electron. Agric.* **2021**, *186*, 106191. [CrossRef]
9. Uygun, T.; Ozguven, M.M. Determination of tomato leafminer: *Tuta absoluta* (Meyrick) (Lepidoptera: Gelechiidae) damage on tomato using deep learning instance segmentation method. *Eur. Food Res. Technol.* **2024**, *250*, 1837–1852. [CrossRef]
10. Chen, L.; Strauch, M.; Daub, M.; Jansen, M.; Luigs, H.-G.; Merhof, D. Instance Segmentation of Nematode Cysts in Microscopic Images of Soil Samples. In Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 23–27 July 2019.
11. Guo, Y.; Gao, J.; Wang, X.; Jia, H.; Wang, Y.; Zeng, Y.; Tian, X.; Mu, X.; Chen, Y.; OuYang, X. Precious Tree Pest Identification with Improved Instance Segmentation Model in Real Complex Natural Environments. *Forests* **2022**, *13*, 2048. [CrossRef]
12. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Simultaneous detection and segmentation. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part VII 13; Springer International Publishing: Berlin/Heidelberg, Germany, 2014; pp. 297–312.
13. Yuan, J.; Cheng, C.; Liu, Z.; Li, H. A segmentation method for high-density overlapping silkworms based on the SOLOv2 framework. *J. Phys. Conf. Ser.* **2025**. *submitted*.
14. Wang, X.; Zhang, R.; Kong, T.; Li, L.; Shen, C. SOLOv2: Dynamic and Fast Instance Segmentation. In Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada, 6–12 December 2020.
15. Zhang, X.; Song, Y.; Song, T.; Yang, D.; Ye, Y.; Zhou, J.; Zhang, L. LDConv: Linear deformable convolution for improving convolutional neural networks. *Image Vis. Comput.* **2024**, *149*, 105190. [CrossRef]
16. Xu, G.; Liao, W.; Zhang, X.; Li, C.; He, X.; Wu, X. Haar wavelet downsampling: A simple but effective downsampling module for semantic segmentation. *Pattern Recognit.* **2023**, *143*, 109819. [CrossRef]
17. Gao, S.; Zhang, P.; Yan, T.; Lu, H. Multi-scale and detail-enhanced segment anything model for salient object detection. In Proceedings of the 32nd ACM International Conference on Multimedia, Melbourne, Australia, 28 October 2024.
18. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to upsample by learning to sample. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023.
19. Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv* **2019**, arXiv:1911.09516. [CrossRef]

20. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 18–24 July 2021.
21. Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; Wang, X. Mask scoring r-cnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
22. Cai, Z.; Vasconcelos, M. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1483–1498. [CrossRef] [PubMed]
23. Tian, Z.; Shen, C.; Chen, H. Conditional convolutions for instance segmentation. In *Computer Vision–ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Proceedings, Part I; Springer International Publishing: Glasgow, UK, 2020; Volume 16, pp. 282–298.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Keyword-Conditioned Image Segmentation via the Cross-Attentive Alignment of Language and Vision Sensor Data

Hye Rim Kim and Byoung Chul Ko *

Department of Computer Engineering, Keimyung University, Daegu 42601, Republic of Korea; 1114728@stu.kmu.ac.kr

* Correspondence: niceko@kmu.ac.kr; Tel.: +82-10-3559-4564

Abstract

Advancements in multimodal large language models have opened up new possibilities for reasoning-based image segmentation by jointly processing visual and linguistic information. However, existing approaches often suffer from a semantic discrepancy between language interpretation and visual segmentation as a result of the lack of a structural connection between query understanding and segmentation execution. To address this issue, we propose a keyword-conditioned image segmentation model (KeySeg) as a novel architecture that explicitly encodes and integrates inferred query conditions into the segmentation process. KeySeg embeds the core concepts extracted from multimodal inputs into a dedicated [KEY] token, which is then fused with a [SEG] token through a cross-attention-based fusion module. This design enables the model to reflect query conditions explicitly and precisely in the segmentation criteria. Additionally, we introduce a keyword alignment loss that guides the [KEY] token to align closely with the semantic core of the input query, thereby enhancing the accuracy of condition interpretation. By separating the roles of condition reasoning and segmentation instruction, and making their interactions explicit, KeySeg achieves both expressive capacity and interpretative stability, even under complex language conditions.

Keywords: vision–language model; keyword condition; keyword-conditioned image segmentation; reasoning segmentation; multimodal learning; vision sensors

1. Introduction

Recent advancements in large language models (LLMs) have enabled high-level reasoning tasks based on their powerful capabilities for natural language understanding and generation. Consequently, LLMs are being increasingly applied across a wide range of domains, including multimodal learning, natural language question answering, code generation, and visual recognition. In particular, vision–language models (VLMs), which jointly process linguistic and visual information, have emerged as a unified framework that bridges these two modalities. These models can perform complex tasks, including generating visual outputs such as object locations or segmentation masks, in response to textual queries.

Such VLM technologies have been extended to text-guided image segmentation, where a system identifies and segments specific objects in an image based on natural language descriptions provided by users. In this context, the image data are typically acquired using vision sensors such as RGB cameras, which serve as the fundamental source for object-level perception and segmentation tasks. This task requires the integration of

various advanced techniques, including semantic alignment between language and vision, multimodal conditional reasoning, and object-level representation learning. Therefore, it is widely regarded as a representative benchmark for evaluating the true reasoning capabilities of VLMs.

Recently, several studies [1–5] have attempted query-based segmentation using multimodal large language models (MLLMs). These approaches typically enhance segmentation performance by incorporating special tokens such as the [SEG] token, visual attention mechanisms, and pixel-level language conditioning strategies. These VLM technologies have been actively applied to text-based image segmentation, where the system identifies specific objects within an image and generates corresponding segmentation masks based on natural language descriptions provided by users. This task inherently involves multiple challenging components, including semantic alignment between language and vision, multimodal conditional reasoning, and object-level representation learning, making it a representative benchmark for evaluating the true understanding capabilities of VLMs.

However, such models commonly suffer from interpretative discrepancies between linguistic responses and visual segmentation outputs. For example, even when a model generates a plausible textual response to a natural language query, it may fail to segment the corresponding object in the image accurately, either by omitting it entirely or by producing imprecise masks. This issue stems from a structural limitation, where the core concepts of the query are not explicitly reflected in the segmentation conditions. Consequently, there is often a lack of consistency between the language understanding and visual execution.

Figure 1 illustrates this problem. For the query '*Which object in this room is used for sleeping?*', LISA [6] generates a reasonable textual response of '*The object in this room used for sleeping is a bed*' but fails to produce a segmentation mask for the target object. In contrast, the proposed keyword-conditioned image segmentation model (KeySeg) generates both a coherent textual response and an accurate segmentation mask for the bed, demonstrating more effective task alignment between language and vision.

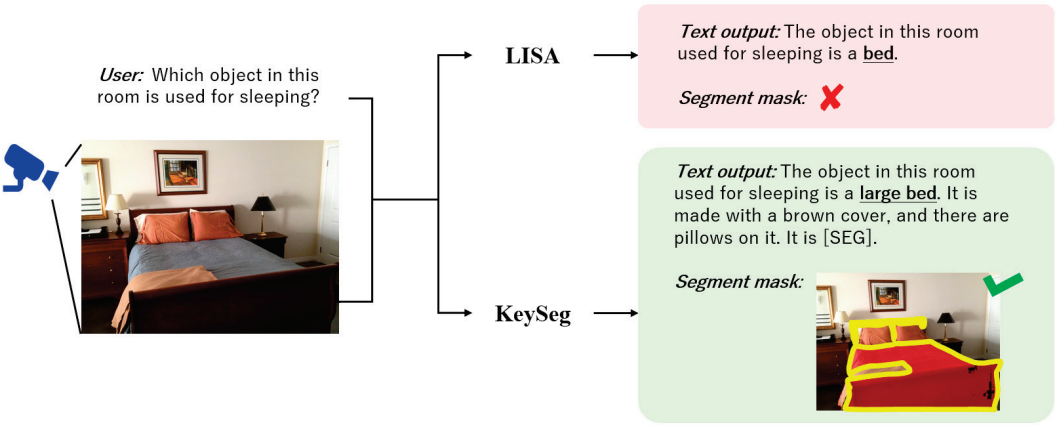


Figure 1. An example of cross-modality task interpretation discrepancies using image data acquired from vision sensors. While LISA generates an appropriate language response to a natural language query, it fails to segment the corresponding object visually. In contrast, the proposed KeySeg model produces both a consistent textual response and an accurate segmentation mask for the same query.

Existing segmentation models [7–9] typically incorporate natural language conditions either as part of the input context or through attention mechanisms that indirectly influence segmentation. However, these methods do not explicitly define the structural links between the linguistic conditions and the segmentation process. To address this limitation, we propose a framework that jointly interprets image and natural language queries, extracts keywords via multimodal reasoning, and utilizes them as explicit segmen-

tation conditions, thereby enabling more precise and semantically aligned segmentation outputs. Existing vision–language segmentation models usually rely on a single special token that simultaneously conveys multiple roles. For example, a single token represents the conceptual meaning of the query or acts as the execution signal for segmentation. This implicit design often results in inconsistencies between the linguistic interpretation and the visual segmentation. In contrast, the proposed framework assigns these roles to two distinct tokens. The [KEY] token encodes the conceptual target inferred from multimodal reasoning over both the query and the image, while the [SEG] token is specialized solely for guiding the mask generation process. By disentangling these roles, KeySeg provides a clearer and more explicit alignment between condition interpretation and segmentation execution, addressing the limitations of previous approaches.

Specifically, we designed KeySeg to encode the inferred keyword information into a [KEY] token, which interacts with the [SEG] token responsible for generating the segmentation mask through a cross-attention-based fusion module. This module generates a semantically rich representation of the segmentation condition via [KEY] token embedding and precisely integrates it into the representation of the [SEG] token, thereby improving segmentation accuracy.

To ensure that the [KEY] token semantically captures the core concept inferred through multimodal reasoning, we introduce a keyword alignment loss that is integrated into the overall training objective. This loss encourages the [KEY] token to represent the intended conditional meaning derived from both the query and the visual context and enables the model to learn meaningful condition representations that directly influence segmentation decisions.

By explicitly leveraging conditional information and enforcing semantic alignment between the query and visual output, this architecture allows for more consistent and accurate segmentation, particularly in complex scenes or under linguistically challenging queries. The effectiveness of KeySeg was quantitatively validated through extensive experiments across multiple datasets. The main contributions of this work can be summarized as follows.

- We propose a keyword-conditioned segmentation framework to resolve discrepancies in modality interpretation. The core concepts of natural language queries are inferred and embedded into a [KEY] token that is explicitly fused with the [SEG] token responsible for mask generation. This explicit integration bridges the gap between linguistic understanding and visual segmentation, thereby mitigating the discrepancies between the two modalities.
- We designed a cross-attention-based fusion module with keyword alignment loss. The fusion module facilitates semantic alignment between the [KEY] and [SEG] tokens, whereas keyword alignment loss encourages the [KEY] token to learn the intended semantics of the core concept inferred through multimodal reasoning from the query and the image. This approach improves the precision and effectiveness of condition-based learning.
- We construct a Reason-Seg-Key dataset and empirically validate the proposed model. Building on the existing ReasonSeg dataset, we augmented it with keyword annotations for each query to enable keyword-guided training. Experimental results on multiple datasets demonstrate that our model achieves superior accuracy and consistency compared with previous methods, validating the effectiveness of the proposed architecture.

2. Related Works

2.1. Multimodal Large Language Models

MLLMs typically perform text generation and reasoning based on visual context by integrating aligned image–language representations into LLMs. For example, BLIP-2 [10] employs a lightweight query transformer on top of a pre-trained vision encoder to project visual features into the LLM input space to delegate natural language generation to the LLM. Flamingo [11] introduces a perceiver re-sampler to summarize visual features, which are then incorporated into LLMs via cross-attention, enabling effective few-shot reasoning.

Models such as LLaVA [12] and MiniGPT-4 [13] align CLIP-derived visual features with LLM embeddings through linear projection and apply instruction tuning to specialize in tasks such as visual question answering and multimodal reasoning. Kosmos-1 [14] jointly trains images and text within a single transformer architecture to support end-to-end grounding, captioning, and OCR. Recently, GPT-4V [15] and Google Gemini [16] extended this framework to support generalized multimodal reasoning across diverse input and output modalities.

Although these MLLMs demonstrate impressive performance in language generation and high-level reasoning tasks, they remain limited to fine-grained pixel-level prediction tasks such as segmentation. A key limitation arises when natural language conditions are not consistently reflected in visual outputs, primarily because of the lack of explicit structural integration of conditional information into the segmentation process.

2.2. Referring Expression Segmentation

Referring Expression Segmentation is a task that segments a specific object in an image based on natural language expressions. It has been widely applied in various domains such as human–computer interaction, robotic vision, and augmented reality [17–21]. Early studies primarily focused on single-target settings, where each expression referred to exactly one object. Representative models such as MAttNet [22] and RRN [23] have implemented vision–language fusion architectures to segment the object referred to by explicitly referring to expressions. To address the complexity of real-world conditions, a generalized RES (GRES) model has been proposed as an extended setting in which a single expression may refer to multiple objects or none at all [24]. The gRefCOCO dataset was introduced to cover single-, multiple-, and no-target scenarios. GRES emphasizes the need for models that can jointly explain region-to-region and region-to-language alignments. For example, CGFormer [25] integrates a query-token-based mask classification framework with a cross-modal reasoning module to learn fine-grained correspondences between expressions and visual regions. Furthermore, it employs contrastive learning to reinforce object-level alignment and improve generalization to various forms of reference expressions.

Another major challenge in RES research is the limited availability of annotated training data. To address this issue, RESMatch [26], which is a semi-supervised learning framework, combines strong visual augmentation, text-based data augmentation, and pseudo-label refinement strategies, achieving performance comparable to that of fully supervised methods, even under weak supervision. Additionally, recent efforts have explored RES in zero-shot, weakly supervised settings. For example, global–local CLIP [27] leverages a pre-trained CLIP model to infer segmentation conditions by combining the global context with local phrase-level semantics, demonstrating strong generalization even with limited annotated data.

Despite these advancements, most existing RES approaches incorporate linguistic conditions indirectly through input embedding or attention mechanisms. Consequently, the segmentation criteria are not structurally encoded within the model, leading to frequent semantic misalignments between the language input and visual output.

2.3. Reasoning Segmentation

Reasoning Segmentation extends beyond simple object identification by requiring the interpretation of complex linguistic cues such as multiple conditions, relational descriptions, and conceptual classifications embedded in natural language queries. LISA [6] defines this task as 'vision segmentation conditioned on high-level reasoning', emphasizing that, unlike directive-based segmentation, it requires the model to understand and reason over the semantic structure and contextual relationships of the query prior to segmentation. In this context, models such as LISA and LLM-Seg [28] have attempted to leverage the language reasoning capabilities of LLMs to perform semantically grounded segmentation, even when queries lack explicit referring expressions. Reasoning-focused benchmarks such as ReasonSeg, MUSE [29] and LLM-Seg40K [28] have been introduced to evaluate segmentation under complex linguistic conditions and relational reasoning tasks.

However, a major limitation of existing approaches is their inability to handle segmentation conditions in a structurally explicit manner. Most models simply append query information to an input sequence or rely on attention-based mechanisms for implicit conditioning. Consequently, segmentation outputs often fail to align with the intended query semantics, leading to frequent interpretation discrepancies between language and visual responses.

To address this limitation, we propose a KeySeg architecture that structurally encodes key concepts inferred from multimodal reasoning. Specifically, we extract the core concepts from the query as a [KEY] token and explicitly align them with a [SEG] token, which is responsible for segmentation, through a cross-attention mechanism. By disentangling query interpretation and condition injection at the token level, our approach enables more accurate and consistent alignment between complex language inputs and visual segmentation tasks.

3. Method

3.1. Special Token Mechanism in MLLMs

MLLMs can utilize special tokens within their input sequences to achieve semantic distinction, assign roles, and inject specific conditions into inputs. In this study, as part of our special token design, we introduce the [SEG] and [KEY] tokens to convey the visual segmentation conditions to the model explicitly as linguistic expressions.

The [SEG] token carries an imperative meaning akin to '*generate a mask based on this point/region*' and functions as a semantic query for the segmentation target object within the MLLMs. The model generates an embedding corresponding to this token as its output, which is subsequently used as a condition for segmentation mask generation by a dedicated mask decoder.

The [KEY] token encapsulates conceptual information about the segmentation target, as determined by the model's comprehensive interpretation of both the image and natural language query. This mechanism differs from directly extracting keywords from the query. Instead, the model assesses the most suitable target based on the visual context and linguistic conditions, expressing this judgment as an embedding. The [KEY] token provides a dedicated representation of the core concept required to answer a query. Rather than relying solely on a literal keyword or a fully implicit embedding, it captures this concept through multimodal reasoning over both the query and the image. This design preserves the clarity of explicit conditioning while retaining the flexibility of implicit representations. Consequently, the segmentation process is guided by conditions that are both precise and semantically aligned. This [KEY] embedding is then used to refine and modulate the segmentation conditions through its semantic interaction with the [SEG] token.

The [SEG] and [KEY] tokens perform complementary functions and play a pivotal role in structuring segmentation conditions. Specifically, [SEG] operates as an execution criterion that informs the mask generation process, whereas [KEY] functions as a conceptual representation that defines the semantic criteria for segmentation. This explicit design of two functionally distinct tokens enables the model to construct a stepwise process for interpreting complex input conditions and inferring the corresponding segmentation target, thereby facilitating adaptable and context-aware segmentation performance across diverse scenarios.

This architecture promotes robust semantic alignment between visual and linguistic information, providing a strong foundation for the model to interpret intricate and composite conditions. Visual information is transformed into patch-wise embeddings using a CLIP-based image encoder [30], and textual inputs are tokenised and processed into language embeddings. The representations of each modality, along with the [SEG] and [KEY] tokens, are integrated into a single sequence and fed into the MLLMs. Based on this unified representation, the model identifies the target segmentation object according to natural language conditions and subsequently generates a precise mask for the corresponding regions.

$$\mathcal{F}\left(\mathrm{x}_{\mathrm{img}}, \mathrm{x}_{\mathrm{txt}}\right) = \{[\mathrm{SEG}], [\mathrm{KEY}], \ldots, [\mathrm{EOS}]\}, \tag{1}$$

In this context, the [EOS] token indicates the end of a text sequence and delineates the overall input boundaries. This particular sequence composition is designed to go beyond mere information concatenation as it enables the model to learn the intricate relationships between various representations and execute segmentation judgments based on complex query conditions. Specifically, the reasoning segmentation task addressed in this study requires sophisticated condition interpretation and inferential capabilities. Therefore, a sequence design that structurally reflects the underlying meaning of a query is highly effective at enhancing both the model's comprehension of the conditions and its segmentation accuracy. This special token-based input design provides structural flexibility within the MLLM for condition interpretation and segmentation execution and offers two key functional advantages.

- Semantic Role Separation: The [SEG] and [KEY] tokens perform distinct and independent functions, representing the execution point for mask generation and the semantic criterion, respectively. This functional separation allows the model to differentiate and learn role-based representations clearly during training.
- Explicit Conditioning: The [KEY] token explicitly encodes conceptual information derived from the internal inference of the model. This mechanism ensures that the implicit linguistic meaning embedded within the segmentation conditions is effectively reflected and directly integrated into the input structure.

Unlike conventional approaches, this special token mechanism explicitly specifies and controls the conditional information at the token level, thereby simultaneously enhancing the interpretability of segmentation conditions and promoting the stability of representation learning.

3.2. Model Baseline

KeySeg is constructed based on pre-trained MLLMs and designed to interpret visual information and natural language queries holistically to perform precise object segmentation under complex conditions. We adopt LLaVA (Meta, Menlo Park, CA, USA) as the base multimodal large language model, which combines a CLIP vision encoder with an LLaMA-based language model. Our proposed [SEG] and [KEY] tokens and the fusion

module are designed on top of this backbone. The overall model architecture is illustrated in Figure 2.

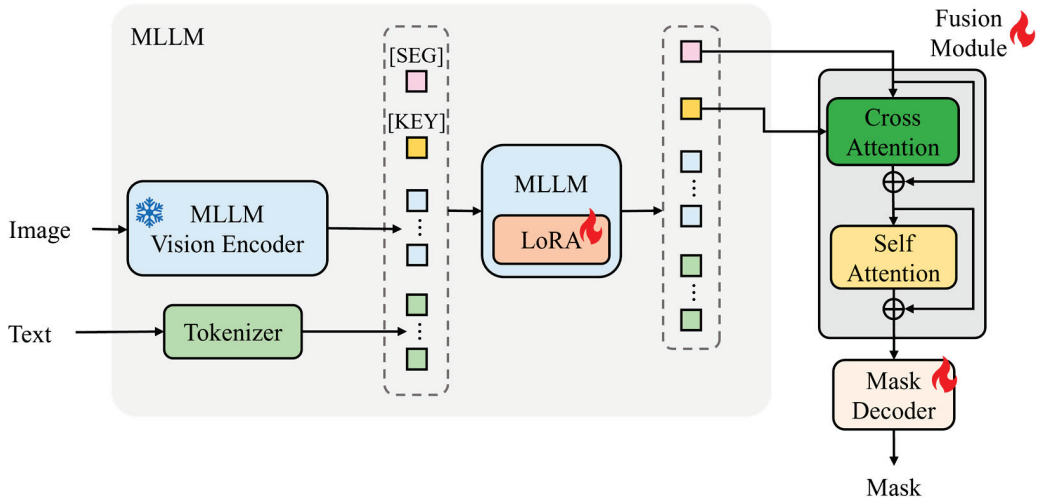


Figure 2. Overall architecture of the proposed KeySeg model. The input image and query are processed through a vision encoder and tokenizer, and then passed to the MLLMs with [SEG] and [KEY] tokens indicating the segmentation condition. The [SEG] and [KEY] tokens are newly introduced and optimized during fine-tuning. At inference, the mask decoder uses the refined [SEG] embedding enriched via cross-attention with the [KEY] token. Unlike LISA, which implicitly combined query interpretation and segmentation in a single token, KeySeg separates these roles into two tokens explicitly connected through the fusion module.

The input consists of an RGB image and a natural language query, which are transformed into visual patch representations using a CLIP-based vision encoder and language token embeddings using a tokenizer. These representations, along with the special tokens [SEG] and [KEY], form a unified input sequence that is fed into the MLLMs. This configuration encourages semantic alignment between the natural language condition and visual context, enabling the model to infer an object that corresponds to the query condition effectively.

The KeySeg MLLMs were constructed using pre-trained LLMs. To learn the semantics of the newly introduced special tokens effectively, we apply low-rank adaptation (LoRA) (Microsoft Research, Redmond, WA, USA) [31]. Because conventional pre-trained language models are not trained with special tokens such as [SEG] and [KEY], an additional process is necessary to teach them the functional roles of these tokens effectively, such as indicating segmentation criteria or expressing condition-centric representations. However, fine-tuning an entire large-scale model is computationally expensive and inefficient. Therefore, we adopted the LoRA method, which selectively trains only low-rank parameters while keeping most of the model's weights frozen. This approach allows the model to recognize special tokens as new semantic units and acquire the representational capacity to utilize them appropriately in the multimodal condition interpretation process.

From the MLLM output sequence, the embedding corresponding to the [SEG] token is passed to a fusion module. This module refines the [SEG] representation to align precisely with the segmentation conditions through semantic interaction with the [KEY] token embedding. The corrected embedding is subsequently used as a core condition representation for the final segmentation judgment. This representation is then forwarded to a segment anything model (SAM)-based mask decoder [32] to generate a binary mask for the target object on a high-resolution visual feature map.

KeySeg is designed to separate query interpretation and condition injection structurally, and to define the semantic alignment process explicitly. This approach effectively mitigates the common issue of condition interpretation mismatch observed in existing LLM-based segmentation models, enabling the generation of high-precision segmentation results, even under complex natural language conditions.

3.3. Fusion Module

Among the MLLM output embeddings, the [SEG] token is semantically augmented through interaction with the keyword information provided by the [KEY] token before being transmitted to the mask decoder. To this end, a cross-attention-based fusion module was designed. This module enhances the interpretability of condition-based object segmentation by ensuring that the segmentation target concept inferred from the natural language query and represented by the [KEY] token is reflected more precisely and effectively in the [SEG] token.

The fusion module takes an MLLM output sequence as an input and performs a cross-attention (CrossAttn) operation. In this operation, the [SEG] token embedding serves as a query, whereas the [KEY] token embedding acts as both a key and a value. Through this process, the [SEG] token directly incorporates semantic conditional information from the [KEY] token, thereby refining its representation to become more condition-centric for the segmentation target. Each embedding is projected into a shared latent space to facilitate the attention operation, ensuring the seamless integration of keyword information into the [SEG] representation.

Following the cross-attention operation, the resulting output is added back to the original [SEG] embedding through a residual connection. This structural design enables the injection of precise conditional information while simultaneously preserving the original visual-context representation, ensuring both learning stability and the consistency of the learned representations. Furthermore, this approach allows the model to reflect new semantic information cumulatively without incurring information loss, thereby facilitating efficient training even in deep network architectures.

The refined [SEG] embedding subsequently undergoes a self-attention (SelfAttn) operation to learn the interactions with both visual and linguistic representations within the surrounding sequence. This self-attention mechanism dynamically learns contextual correlations, thereby enabling the [SEG] embedding to align effectively with the global visual context. Subsequently, this representation is further transformed into a final expression using a multilayer perceptron (MLP), which enhances the representational power of the model through nonlinear transformations in a high-dimensional space.

The entire process can be expressed mathematically as follows:

$$\hat{\mathrm{S}} = \mathrm{MLP}(\mathrm{SelfAttn}(\mathrm{CrossAttn}(\mathrm{S}, \mathrm{K}) + \mathrm{S})), \tag{2}$$

where S is the initial embedding of the [SEG] token, K is the [KEY] token embedding, and $\hat{\mathrm{S}}$ is the final refined [SEG] embedding. The refined embedding $\hat{S}$ is then reordered to match the sequence length before being passed to the mask decoder ($\mathcal{D}$), which serves as the input for generating the final object segmentation mask.

$$\mathcal{M} = \mathcal{D}(\hat{S}), \tag{3}$$

This structure enables the precise reflection of keyword-centric segmentation conditions, even in complex scenes or situations involving multiple objects, thereby contributing to the generation of more consistent and accurate segmentation masks by the model. With explicit condition injection facilitated by cross-attention, robust information preservation

achieved through residual connections, and effective context alignment ensured by self-attention, all of which are designed in an integrated manner, this module plays a pivotal role in enhancing not only segmentation accuracy but also the reliability of condition interpretation.

3.4. Keyword-Conditioned Learning

In the proposed model, the [KEY] token is not merely used as a conditional input but is further extended to serve as an explicit learning signal. To this end, we define an auxiliary keyword prediction loss that is integrated into the overall training objective. This learning strategy encourages the model to reflect the semantics of the designated keyword for each training sample more precisely.

The keyword loss is computed as follows. First, we extract the embedding vector K corresponding to the [KEY] token from the output of the MLLMs. This embedding is passed through a linear projection and then compared with a vocabulary embedding table $E \in \mathbb{R}^{V \times d}$ by computing the dot product to obtain the logits. Here V denotes the vocabulary size and d is the hidden dimension of the model. A temperature parameter τ is applied to modulate the output distribution and a cross-entropy loss is computed using the ground-truth keyword token y as the target. The complete formulation of keyword loss is defined as follows:

$$\mathcal{L}_{\text{keyword}} = \text{CrossEntropy}\left(\frac{\mathrm{W_k \cdot E^\top}}{\tau}, \mathrm{y}\right), \tag{4}$$

Here, W_k denotes the result of applying a linear transformation to the [KEY] token embedding K and E refers to the vocabulary embedding table. In addition to the keyword loss described above, KeySeg adopts a multi-objective loss framework to optimize various aspects of the model jointly, including language generation, segmentation accuracy, and condition expressiveness. Specifically, we include the following components:

- Language generation loss $\mathcal{L}_{CE}$, which preserves the language modeling capabilities of the MLLM;
- Binary cross-entropy loss $\mathcal{L}_{BCE}$, which maximises the alignment between the predicted mask and ground-truth mask; and
- Dice loss $\mathcal{L}_{DICE}$, which further enhances segmentation consistency.

The total loss function for KeySeg is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{DICE}} + \mathcal{L}_{\text{keyword}}, \tag{5}$$

This multicomponent loss design balances different learning objectives while enabling the model to generate high-quality segmentation results that accurately reflect natural language conditions. The keyword loss term $\mathcal{L}_{\text{keyword}}$ plays a crucial role in guiding the semantic representation of the [KEY] token. Rather than relying on a keyword as a simple input embedding, this loss encourages the model to infer and internalise the segmentation criterion through its reasoning process.

Consequently, KeySeg goes beyond basic condition-based segmentation by enabling the model to infer and represent segmentation criteria actively based on a broader linguistic context, ultimately supporting a more flexible and context-aware visual understanding.

4. Experiment

In this study, we evaluated the performance of our model for object segmentation conditioned on natural language queries using four benchmark datasets: refCOCO [33], refCOCO+ [33], refCOCOg [34], and ReasonSeg [6]. These datasets require models to

generate segmentation masks based on the correspondence between visual content and natural language expressions. Each dataset varies in linguistic complexity and expression style, allowing a comprehensive assessment of a model's reasoning capabilities across different semantic challenges.

4.1. Datasets

4.1.1. The refCOCO Series

The refCOCO series is a large-scale vision language segmentation benchmark based on the MS COCO image dataset [35]. It focuses on expression segmentation tasks in which a natural language description is used to identify and segment a specific object in an image. The refCOCO dataset consists of 19,994 images, more than 50,000 annotated objects, and 142,209 reference expressions. These expressions tend to be short and direct, making the dataset well suited for evaluating basic referential grounding. For evaluation, refCOCO provides two separate testing splits to analyse performance differences based on target type and to control for visual and linguistic biases. Test A includes only expressions that refer to human subjects, whereas Test B includes expressions that refer to non-human objects. This split allows for a more detailed evaluation of a model's ability to handle queries targeting people versus background- or object-focused queries, which may involve different levels of segmentation difficulty and visual attention requirements. refCOCO+ shares a similar structure with refCOCO but introduces an additional challenge by removing spatial terms from the referring expressions. This design prevents models from relying on explicit spatial cues, thereby requiring a deeper semantic understanding. The dataset contains 19,992 images, 49,856 objects, and 141,564 reference expressions. In contrast, refCOCOg consists of longer and more complex sentences, with a significant portion of queries involving ambiguity or relational descriptions. This dataset includes 25,799 images, 49,822 objects, and 95,010 reference expressions. Many queries require reasoning over the relationships between objects or an understanding of high-level contexts, making them particularly suitable for evaluating advanced language comprehension in segmentation tasks. Although these three datasets share the same image base as MS COCO, they differ significantly in terms of linguistic complexity and query construction. This allows for a multifaceted evaluation of a model's ability to understand natural language and perform conditioned segmentation across various levels of semantic difficulty.

4.1.2. ReasonSeg Dataset

ReasonSeg is a reasoning-based object segmentation dataset introduced in LISA and designed to go beyond explicit reference expression datasets. Unlike conventional datasets, where the target object is directly mentioned in the query, ReasonSeg presents queries that rely on indirect cues such as the object's attributes, roles, usage, or contextual surroundings. The model must interpret both the image and query holistically to infer an appropriate segmentation target without an explicit reference. The dataset contains 1218 image–query–mask samples split into 239 training, 200 validation, and 779 testing examples. This setup provides an evaluation environment that assesses a model's capability for high-level vision–language reasoning, rather than simple directive-based understanding.

While the refCOCO series is used to assess performance in explicit, directive-based segmentation, ReasonSeg focuses on evaluating a model's reasoning-driven segmentation capabilities in more abstract and context-dependent language conditions. We conducted both quantitative and qualitative analyses across these datasets to evaluate the proposed model's ability to perform accurate and consistent segmentation under varying levels of linguistic complexity and reasoning demands.

4.2. Model Training Using Reason-Seg-Key Dataset

The ReasonSeg dataset provides natural language queries and the corresponding object segmentation masks for images that often contain complex scenes. However, one limitation of this dataset is the absence of explicit annotations for key concepts or salient objects within each query, which requires the model to infer visual attention implicitly without guidance. To address this limitation, we constructed an extended version of the dataset called Reason-Seg-Key, which augments ReasonSeg with keyword-level supervision. The extended dataset retains the same number of samples as ReasonSeg but also includes automatically extracted keywords for each sample. These keywords are generated as single words or short noun phrases representing the core concept or referent of a query and its associated visual targets.

Keyword extraction is performed using the pre-trained FLAN-T5 [36] model with an instruction-style prompt such as '*Extract the most important object from the following text as a single word or short noun phrase*'. The resulting keywords are appended to the original dataset as a new field labeled '*keyword*'. To ensure the reliability of the automatically generated keywords, we combined simple post-processing with manual verification. Keywords that were incomplete or redundant were removed, and synonyms or multi-word expressions were standardized into a single representative form. Additionally, a subset of the extracted keywords was manually inspected and corrected as needed to ensure semantic consistency. By using this combination of normalization rules and manual checks, the dataset maintained consistent and reliable supervision signals across all samples.

In this study, we utilize the Reason-Seg-Key dataset for model training, treating the extracted keyword as the target for keyword-conditioned learning. Specifically, each keyword served as a supervised signal for training the [KEY] token, allowing the model to learn semantically grounded keyword-centric segmentation behaviors more effectively. By combining normalization rules and manual checks, the dataset maintained consistent and reliable supervision signals across all samples.

4.3. Experimental Setup

To evaluate the performance of the proposed keyword-conditioned segmentation model, experiments were conducted using three popular benchmarks for query-based object segmentation: refCOCO, refCOCO+, and refCOCOg. These datasets serve as standard benchmarks for expression segmentation, where the goal is to segment a specific object in an image based on a natural language query. Each dataset differs in segmentation difficulty, depending on the linguistic complexity and length of the referring expressions.

The refCOCO+ dataset presents a particularly challenging setup because of the presence of multiple similar objects in visually cluttered backgrounds, which makes it difficult to perform accurate segmentation based on text–image alignment alone. In contrast, refCOCOg contains relatively longer and more complex expressions, with an average of 8.4 words per query, increasing the overall reasoning difficulty. Therefore, refCOCOg is considered more challenging than refCOCO and refCOCO+, and strong performance on this dataset is indicative of the superior segmentation capabilities of a model under complex linguistic conditions.

Model performance was evaluated using two metrics. The centre-based intersection over union (cIoU) and generalised intersection over union (gIoU) assess the alignment between the predicted segmentation mask and ground-truth mask in terms of centrality and overall spatial consistency, respectively. For a fair comparison, all baseline models were evaluated using the same experimental pipeline. The results of LISA were reproduced using the official checkpoint and source code provided by its authors. The evaluations employed the same dataset splits and metrics used for KeySeg. The reproduced values may

differ slightly from those reported in the LISA paper due to differences in preprocessing, metric implementation, and experimental environments.

4.4. Qualitative Evaluation

Table 1 presents the experimental results for the ReasonSeg dataset, where gIoU and cIoU were used as performance metrics to evaluate the proposed model's inference-based object segmentation capabilities. ReasonSeg contains highly challenging queries that demand multistep reasoning and attribute inference without explicit instructions, making it suitable for assessing visual–linguistic integrated reasoning abilities beyond simple language understanding.

Table 1. Evaluation on the ReasonSeg Dataset.

Method	ReasonSeg	
	gIoU	cIoU
OVSeg [37]	26.1	20.8
GRES [25]	21.3	22.0
X-Decoder [38]	21.7	16.3
SEEM [39]	24.3	18.7
Grounded-SAM [40]	21.3	16.4
LISA [6]	36.8	34.1
KeySeg	**42.6**	**47.9**

The experimental results reveal that KeySeg achieved an accuracy of 42.6% for the gIoU and 47.9% for the cIoU, outperforming all compared methods. Compared with LISA, our strongest competitor, KeySeg demonstrated significant performance improvements of 5.8% (gIoU) and 13.8% (cIoU), as well as exhibited a substantial performance gap when compared with other conventional segmentation models. These findings suggest that the proposed keyword-centric condition generation and semantic refinement approach offers a more robust representational structure for reasoning segmentation compared with simple mask condition injection methods or prompt-selection-based approaches.

The results empirically demonstrate that the proposed KeySeg architecture maintains consistent segmentation performance across various levels of linguistic complexity. Furthermore, our approach proves that keyword-based segmentation condition interpretation and representation refinement operate effectively, even on tasks requiring high-level reasoning. Notably, the incorporation of keyword loss learning enables the model to represent the central concepts inherent in the conditions accurately, providing strong alignment performance when converting MLLM outputs into segmentation criteria. Although this evaluation is based on a single dataset, ReasonSeg contains highly complex and reasoning-oriented queries, and the clear improvements of KeySeg under these conditions highlight the structural robustness of the proposed approach in handling reasoning-based segmentation tasks.

Table 2 presents a quantitative comparison of the proposed KeySeg model with existing methods on three standard benchmarks: refCOCO, refCOCO+, and refCOCOg. On the refCOCO dataset, KeySeg achieved cIoU values of 71.8% on Test A and 63.6% on Test B, outperforming all baseline models on both splits. This result demonstrates that the proposed keyword-conditioned architecture effectively facilitates precise object identification, even in scenarios involving relatively simple directive expressions. For refCOCO+, KeySeg achieved 63.1% accuracy on Test A and 46.4% accuracy on Test B. On the Test A split, KeySeg surpassed the previous best-performing VLT model (61.0%) by 2.1%. Although KeySeg's performance on Test B is slightly lower than that of VLT, KeySeg still outperforms LISA and MCN by 4.6% and 3.7%, respectively. Although KeySeg performs slightly worse

than VLT on the Test B split, this outcome reflects the characteristics of the data. Test B primarily consists of object-centric queries, where success relies more on robust object category recognition than on reasoning over semantic attributes or relations. VLT benefits from large-scale pretraining that enhances general object recognition and thus gains an advantage in this setting. In contrast, KeySeg demonstrates superior performance on Test A and refCOCOg, where queries demand higher-level reasoning beyond simple object identification. On the more challenging refCOCOg dataset, KeySeg achieved a cIoU of 62.4%, outperforming both LISA and VLT, and achieved the highest performance among all compared models.

Table 2. Evaluation on rerfCOCO series datasets (cIoU).

Method	refCOCO		refCOCO+		refCOCOg
	Test A	Test B	Test A	Test B	Test
MCN [41]	64.2	59.7	55.0	42.7	49.4
VLT [42]	70.5	65.2	61.0	**50.1**	57.7
LISA [6]	68.7	59.8	56.9	41.8	57.0
KeySeg	**71.8**	**63.6**	**63.1**	46.4	**62.4**

Both refCOCO+ and refCOCOg pose significant challenges because of the removal of spatial cues and presence of longer and more complex queries, respectively. These settings require a higher level of linguistic understanding and reasoning beyond simple directive segmentation. The improved performance under these conditions highlights the effectiveness of the structural design of KeySeg, particularly in terms of capturing and utilizing semantic conditions for accurate object segmentation. In particular, the cross-attention-based fusion module between the [KEY] and [SEG] tokens enables the model to integrate the core concepts from a query into a semantically meaningful segmentation instruction. Additionally, the incorporation of keyword loss provides an explicit training signal that refines the model's ability to represent and focus on the correct target object with greater precision.

4.5. Ablation Study

4.5.1. Effect of KEY Token Insertion

Table 3 presents the results of an ablation study conducted on the refCOCOg dataset to evaluate the impact of the [KEY] tokens on segmentation performance quantitatively. For this experiment, we adopted a LISA-based architecture as the baseline, where only the [SEG] token was inserted into the multimodal input. Since LISA itself is implemented on top of LLaVA, keeping the same backbone ensures a fair comparison. Therefore, all improvements shown in Table 3 can be attributed to our proposed [KEY] token and the fusion mechanism, rather than differences in the underlying language model.

Table 3. Ablation study on [SEG], [KEY], and the Fusion Module on RefCOCOg. CA and SA denote cross-attention and self-attention, respectively.

[SEG]	[KEY]	Fusion Module		refCOCOg	
		CA	SA	gIoU	cIoU
✓				57.69	57.07
✓			✓	59.89	58.78
✓	✓	✓	✓	**63.85**	**62.40**

As an intermediate variant, we first experimented with a configuration that included only the [SEG] token but enhanced its contextual understanding by integrating it with a

self-attention-based fusion module. This setting achieved 59.89% gIoU and 58.78% cIoU, representing improvements of 2.2% and 1.71%, respectively, over the baseline. These results suggest that enriching the [SEG] representation through self-attention-based fusion partially contributes to improved context-aware condition modelling.

The best performance was achieved when both [SEG] and [KEY] tokens were incorporated, and a fusion module combining cross-attention and self-attention was used to enable bidirectional interaction. This configuration achieved 63.85% gIoU and 62.40% cIoU, representing improvements of 6.16% and 5.33%, respectively, over the baseline. In this setup, key semantic concepts were explicitly extracted from the query and encoded into the [KEY] token. This information was then fused into the [SEG] token through the designed attention mechanisms, enabling the model to interpret the intended segmentation criteria more precisely and align them with the generated mask.

This structural enhancement is particularly effective under the high linguistic complexity of refCOCOg, where queries are often long and semantically rich. The explicit decomposition of query semantics and integration of these cues through cross-attention allow the model to produce more consistent and accurate segmentation results by aligning core meanings with visual outputs.

These ablation results empirically demonstrate that keyword-guided conditioning and cross-attentive fusion modules are key architectural components for structurally improving referring expression segmentation. Although extending the ablation to multiple datasets would provide stronger validation, we focused on refCOCOg because it best represents the reasoning-oriented scenarios targeted in this study. We recognize the importance of broader multi-dataset analysis and regard it as a key direction for future work.

4.5.2. Fusion Module Configuration Analysis

Table 4 presents a quantitative analysis of the effects of the internal components of the fusion module on the conditional alignment and segmentation performances on the refCOCOg dataset. This ablation study compared the three configurations based on the inclusion of cross-attention, self-attention, and residual connections. The first configuration applies a simple cross-attention mechanism in which condition-centric information from the [KEY] token is directly injected into the [SEG] token without any internal interaction or feedback. This unidirectional information flow limits the ability of the model to interpret complex or lengthy queries because the [SEG] token cannot holistically incorporate the full structure of the input query. Consequently, this setting achieved 56.51% gIoU and 55.83% cIoU, suggesting that under such conditions, the segmentation output suffers from insufficient semantic alignment.

Table 4. Ablation study on attention mechanisms and residual connection design in the fusion module.

CA	SA	Residual Connection	refCOCOg	
			gIoU	cIoU
✓			56.51	55.83
✓	✓		55.41	55.57
✓	✓	✓	**63.85**	**62.40**

The second configuration enhanced the previous setup by introducing self-attention into the [SEG] token, allowing the [SEG] representation to interact with the entire input sequence, helping it better contextualise the condition signal and suppress interference from irrelevant background objects or non-target elements. This configuration aims to improve the selectivity and consistency of condition interpretation, particularly in complex scenes with multiple objects or where spatial and attribute-level reasoning is required. However,

it exhibited a slightly lower performance (55.41% gIoU and 55.57% cIoU) than the cross-attention-only setting. This result suggests that although self-attention can enrich contextual understanding, it may also dilute the model's focus on the core semantic signal conveyed by the [KEY] token. The final configuration combines cross-attention and self-attention, and introduces a residual connection that aggregates the fused condition representation back into the original [SEG] embedding. This design aims to preserve the semantic precision of the condition representation while gradually reinforcing the alignment throughout training. This configuration achieved the best performance, with 63.85% gIoU and 62.40% cIoU, demonstrating robust and consistent segmentation results, even under ambiguous or linguistically complex queries.

These results indicate that simply adding individual attention mechanisms is insufficient. The interaction between attention types and the overall flow of information within the fusion module plays a crucial role in enhancing multimodal condition understanding. Notably, the final fusion design proved to be both semantically reliable and structurally effective for keyword-based alignment and segmentation instructions, providing empirical validation of its robustness under complex natural language conditions.

4.6. Quantitative Evaluation

Figure 3 visually compares the segmentation results of LISA and our KeySeg model on the refCOCOg dataset. refCOCOg is characterised by the frequent presence of multiple similar objects, complex backgrounds, and lengthy queries, demanding both precise language–visual alignment and accurate object distinction. In this comparison, we analysed the performance of the proposed model by focusing on three major difficulty scenarios: multi-object coexistence, distinction between similar objects, and background information interference.

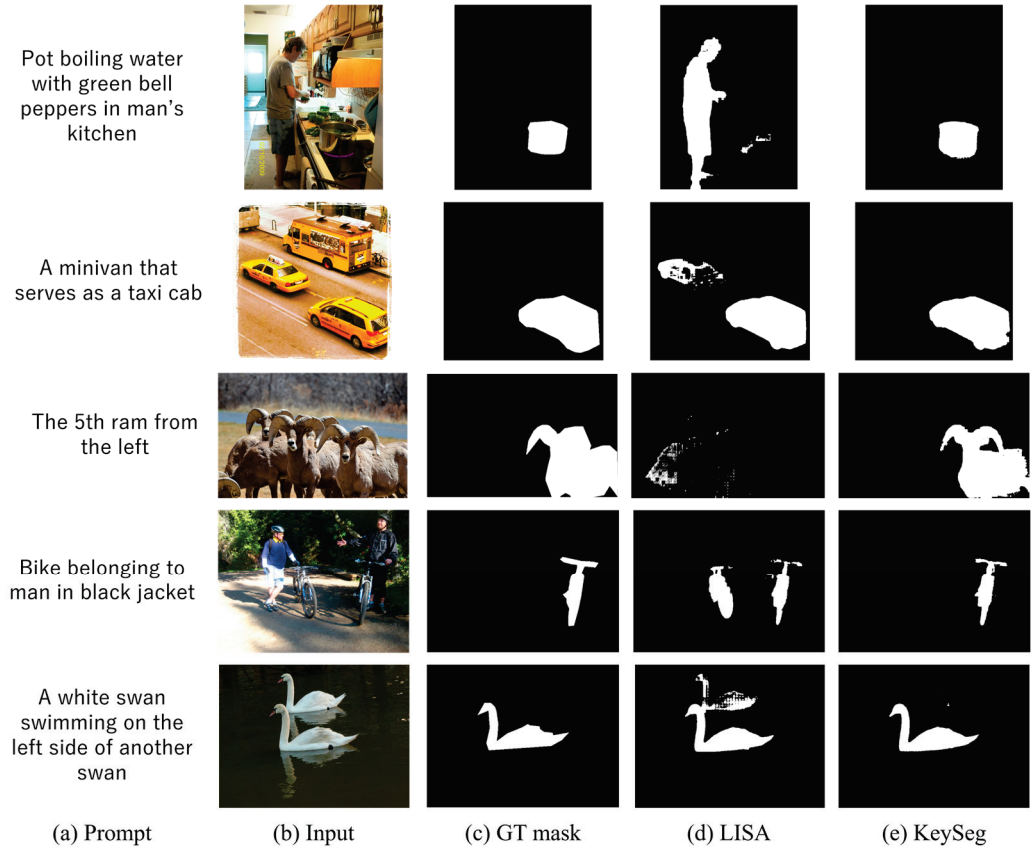


Figure 3. Comparison of object segmentation results on the refCOCOg testset.

First, in instances where multiple similar objects are present within the frame (e.g., vehicles in the second row, mountain goats in the third row, and cyclists in the fourth row), standard segmentation models often suffer from selection errors or the erroneous inclusion of multiple objects, despite the query specifically indicating a single target object, as a result of the high visual similarity among objects. Indeed, LISA was observed to include multiple objects or generate blurred masks that failed to distinguish the designated target accurately. In contrast, by leveraging its complementary special token structure, KeySeg successfully segmented only a specified single object by precisely reflecting the meaning of the query. This outcome demonstrates that the keyword fusion module effectively incorporates semantic differences between objects into its representation.

Furthermore, situations in which complex background elements overlap with the main object or similar shapes in the background interfere with object identification (e.g., the kitchen scene in the first row and the swan and water reflection in the last row), highlighting the critical role of a model's representational power in terms of segmentation boundary consistency and accuracy. KeySeg effectively suppresses visual interference factors and extracts clear contours of the designated object. Conversely, LISA exhibits a lack of consistency, including either the background around the object or incorrect segmentation of the reflected shapes. This result indicates that a keyword-based refinement structure contributes to condition-centric representation reconstruction beyond simple visual features.

These qualitative comparison results demonstrate that the proposed model is capable of performing semantic-based representation alignment, even under complex and ambiguous visual–linguistic conditions. It can accurately infer and segment only conditioned objects, even in scenarios where multiple or similar objects are present. In particular, an integrated design combining a cross-attention-based fusion structure with keyword-aware learning is crucial for effectively isolating semantic differences between objects and simultaneously ensuring the stability and precision of segmentation performance.

5. Conclusions

In this paper, we proposed KeySeg, a keyword-conditioned image segmentation model designed to mitigate the performance discrepancies often observed in MLLM-based object segmentation. The proposed KeySeg model was engineered to incorporate semantic-centric information into the segmentation process by extracting keyword information from natural language queries using [KEY] tokens and fusing them with [SEG] token embeddings through a dedicated fusion module. Furthermore, we introduced keyword loss as an explicit learning signal for the [KEY] token, enabling the model to learn conditional information clearly. Experimental results demonstrated that the proposed model outperforms existing baseline models, thereby validating the effectiveness of our keyword fusion module and conditional learning structure in terms of enhancing object segmentation accuracy.

Future work will focus on strengthening both the robustness and interpretability of the proposed framework. Beyond refCOCOg, evaluation on multiple datasets with diverse linguistic complexities would provide stronger evidence of generalizability. In parallel, a more fine-grained analysis of the loss design in Equation (5) could reveal how different objectives contribute to stabilizing [KEY] token learning and overall segmentation quality. Finally, richer qualitative comparisons will be explored to illustrate how KeySeg handles ambiguous queries, failure cases, and subtle differences across models, thereby offering deeper insights into its reasoning process.

Author Contributions: H.R.K. was responsible for the design, data curation and writing the draft of the manuscript. B.C.K. was responsible for the data curation, supervision and writing and editing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education of the Republic of Korea (No. RS-2025-25399600).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yang, Y.; Jiang, P.T.; Wang, J.; Zhang, H.; Zhao, K.; Chen, J.; Li, B. Empowering segmentation ability to multi-modal large language models. *arXiv* **2024**, arXiv:2403.14141. [CrossRef]
2. Jang, D.; Cho, Y.; Lee, S.; Kim, T.; Kim, D.S. MMR: A large scale benchmark dataset for multi-target and multi-granularity reasoning segmentation. In Proceedings of the International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025.
3. Xia, Z.; Han, D.; Han, Y.; Pan, X.; Song, S.; Huang, G. GSVA: Generalized segmentation via multimodal large language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 3858–3869.
4. Ou, R.; Hu, Y.; Zhang, F.; Chen, J.; Liu, Y. GeoPix: Multi-modal large language model for pixel-level image understanding in remote sensing. In Proceedings of the International Conference on Machine Learning (ICML), Montréal, Canada, 21–27 July 2025; pp. 1–14.
5. Wei, C.; Zhong, Y.; Tan, H.; Zeng, Y.; Liu, Y.; Zhao, Z.; Yang, Y. InstructSeg: Unified instructed visual segmentation. *arXiv* **2024**, arXiv:2412.14006. [CrossRef]
6. Lai, X.; Tian, Z.; Chen, Y.; Li, Y.; Yuan, Y.; Liu, S.; Jia, J. LISA: Reasoning segmentation via large language model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 9579–9589.
7. Chen, Y.C.; Li, W.H.; Sun, C.; Wang, Y.C.F.; Chen, C.S. SAM4MLLM: Enhance multi-modal large language model for referring expression segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Milan, Italy, 29 September–4 October 2024; Springer Nature Switzerland: Cham, Switzerland, 2024; pp. 323–340.
8. Qian, R.; Yin, X.; Dou, D. Reasoning to attend: Try to understand how <SEG> token works. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Atlanta, GA, USA, 16–26 June 2025; pp. 24722–24731.
9. Lu, Y.; Cao, J.; Wu, Y.; Li, B.; Tang, L.; Ji, Y.; Wu, C.; Wu, J.; Zhu, W. RSVP: Reasoning segmentation via visual prompting and multi-modal chain of thought. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), Kuala Lumpur, Malaysia, 20–25 July 2025; pp. 1–12.
10. Li, J.; Li, D.; Savarese, S.; Hoi, S. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Proceedings of the International Conference on Machine Learning (ICML), Honolulu, HI, USA, 23–29 July 2023; pp. 19730–19742.
11. Alayrac, J.B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*; Neural Information Processing Systems Foundation: San Diego, CA, USA, 2022; Volume 35, pp. 23716–23736.
12. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*; Neural Information Processing Systems Foundation: San Diego, CA, USA, 2023; Volume 36, pp. 34892–34916.
13. Zhu, D.; Chen, J.; Shen, X.; Li, X.; Elhoseiny, M. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023; pp. 1–12.
14. Huang, S.; Dong, L.; Wang, W.; Hao, Y.; Singhal, S.; Ma, S.; Lv, T.; Cui, L.; Mohammed, O.K.; Patra, B.; et al. Language is not all you need: Aligning perception with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*; Neural Information Processing Systems Foundation: San Diego, CA, USA, 2023; Volume 36, pp. 72096–72109.
15. OpenAI. GPT-4 technical report. *arXiv* **2023**, arXiv:2303.08774. [CrossRef]
16. Google DeepMind. Gemini: A family of highly capable multimodal models. *arXiv* **2023**, arXiv:2312.11805. [CrossRef]
17. Wu, Y.; Zhang, Z.; Xie, C.; Zhu, F.; Zhao, R. Advancing referring expression segmentation beyond single image. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 2628–2638.
18. Qian, H.H.; Lu, Y.; Ren, K.; Wang, G.; Khargonkar, N.; Xiang, Y.; Hang, K. RISeg: Robot interactive object segmentation via body frame invariant features. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 13954–13960.

19. Lu, K.; Ma, C.; Hori, C.; Romeres, D. KitchenVLA: Iterative vision language corrections for robotic execution of human tasks. In Proceedings of the 1st Workshop on Safely Leveraging Vision-Language Foundation Models in Robotics: Challenges and Opportunities, Kyoto, Japan, 26 May 2025; pp. 1–8.
20. Samson, M.; Muraccioli, B.; Kanehiro, F. Scalable, training-free visual language robotics: A modular multi-model framework for consumer-grade GPUs. In Proceedings of the 2025 IEEE/SICE International Symposium on System Integration (SII), Harbin, China, 14–17 January 2025; pp. 193–198.
21. Tsagkas, N.; Mac Aodha, O.; Lu, C.X. VL Fields: Towards language grounded neural implicit spatial representations. In Proceedings of the ICRA Workshop on Language and Robot Learning, London, UK, 29 May 2023; pp. 1–6.
22. Yu, L.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; Bansal, M.; Berg, T.L. MAttNet: Modular attention network for referring expression comprehension. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1307–1315.
23. Li, R.; Li, K.; Kuo, Y.C.; Shu, M.; Qi, X.; Shen, X.; Jia, J. Referring image segmentation via recurrent refinement networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5745–5753.
24. Liu, C.; Ding, H.; Jiang, X. GRES: Generalized referring expression segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada, 18–22 June 2023; pp. 23592–23601.
25. Tang, J.; Zheng, G.; Shi, C.; Yang, S. Contrastive grouping with transformer for referring image segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada, 18–22 June 2023; pp. 23571–23580.
26. Zang, Y.; Fu, C.; Cao, R.; Zhu, D.; Zhang, M.; Hu, W.; Zhu, L.; Chen, T. RESMatch: Referring expression segmentation in a semi-supervised manner. *Inf. Sci.* **2025**, *694*, 121709.
27. Yu, S.; Seo, P.H.; Son, J. Zero-shot referring image segmentation with global-local context features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada, 18–22 June 2023; pp. 19456–19465.
28. Wang, J.; Ke, L. LLM-Seg: Bridging image segmentation and large language model reasoning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2024, Seattle, WA, USA, 17–21 June 2024; pp. 1765–1774.
29. Wang, Z.; Yang, X.; Liu, Y.; Feng, S.; Wang, D.; Zhang, Y. Muse: A multimodal conversational recommendation dataset with scenario grounded user profiles. *arXiv* **2024**, arXiv:2412.18416. [CrossRef]
30. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Satry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning (ICML), Vienna, Austria, 18–24 July 2021; pp. 8748–8763.
31. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-rank adaptation of large language models. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022; pp. 1–16.
32. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rollins, N.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y. Segment anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 4015–4026.
33. Kazemzadeh, S.; Ordonez, V.; Matten, M.; Berg, T. Referitgame: Referring to objects in photographs of natural scenes. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 787–798.
34. Yu, L.; Poirson, P.; Yang, S.; Berg, A.C.; Berg, T.L. Modeling context in referring expressions. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; Springer Nature Switzerland: Cham, Switzerland, 2016; pp. 69–85.
35. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; Springer Nature Switzerland: Cham, Switzerland, 2014; pp. 740–755.
36. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tai, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.* **2024**, *25*, 1–53.
37. Liang, F.; Wu, B.; Dai, X.; Li, K.; Zhao, Y.; Zhang, H.; Zhang, P.; Vajda, P.; Marculescu, D. Open-vocabulary semantic segmentation with mask-adapted CLIP. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada, 18–22 June 2023; pp. 7061–7070.
38. Zou, X.; Dou, Z.Y.; Yang, J.; Gan, Z.; Li, L.; Li, C.; Dai, X.; Behl, H.; Wang, J.; Yuan, L.; et al. Generalized decoding for pixel, image, and language. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada, 18–22 June 2023; pp. 15116–15127.
39. Zou, X.; Yang, J.; Zhang, H.; Li, F.; Li, L.; Gao, J.; Lee, Y.J. Segment everything everywhere all at once. In *Advances in Neural Information Processing Systems (NeurIPS)*; Neural Information Processing Systems Foundation: San Diego, CA, USA, 2023; Volume 36, pp. 19769–19782.

40. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Li, C.; Yang, J.; Su, H.; Zhu, J.; et al. Grounding DINO: Marrying DINO with grounded pre-training for open set object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; Springer Nature Switzerland: Cham, Switzerland, 2023; pp. 38–55.
41. Luo, G.; Zhou, Y.; Sun, X.; Cao, L.; Wu, C.; Deng, C.; Ji, R. Multi-task collaborative network for joint referring expression comprehension and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 14–19 June 2020; pp. 10034–10043.
42. Ding, H.; Liu, C.; Wang, S.; Jiang, X. Vision-language transformer and query generation for referring segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 16321–16330.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Lightweight 3D Multi-Object Tracking via Collaborative Camera and LiDAR Sensors

Dong Feng [1,*,†], Hengyuan Liu [2,†] and Zhiyu Liu [3]

[1] School of Computer Science, Peking University, Beijing 100871, China
[2] School of Data Science, Qingdao University of Science and Technology, Qingdao 266061, China; liuhengyuan@mails.qust.edu.cn
[3] School of Information Science and Engineering, Ocean University of China, Qingdao 266100, China; lzy1280@ouc.edu.cn
* Correspondence: fengdong@pku.edu.cn
† These authors contributed equally to this work.

Abstract

With the widespread adoption of camera and LiDAR sensors, 3D multi-object tracking (MOT) technology has been extensively applied across numerous fields such as robotics, autonomous driving, and surveillance. However, existing 3D MOT methods still face significant challenges in addressing issues such as false detections, ghost trajectories, incorrect associations, and identity switches. To address these challenges, we propose a lightweight 3D multi-object tracking framework via collaborative camera and LiDAR sensors. Firstly, we design a confidence inverse normalization guided ghost trajectories suppression module (CIGTS). This module suppresses false detections and ghost trajectories at their source using inverse normalization and a virtual trajectory survival frame strategy. Secondly, an adaptive matching space-driven lightweight association module (AMSLA) is proposed. By discarding global association strategies, this module improves association efficiency and accuracy using low-cost decision factors. Finally, a multi-factor collaborative perception-based intelligent trajectory management module (MFCTM) is constructed. This module enables accurate retention or deletion decisions for unmatched trajectories, thereby reducing computational overhead and the risk of identity mismatches. Extensive experiments on the KITTI dataset show that the proposed method outperforms state-of-the-art methods across multiple performance metrics, achieving Higher Order Tracking Accuracy (HOTA) scores of 80.13% and 53.24% for the Car and Pedestrian categories, respectively.

Keywords: deep learning; computer vision; multi-object tracking; autonomous driving; ghost trajectory; data association; trajectory management

1. Introduction

Multi-object tracking is a technique used to associate dynamic objects across successive frames, and it plays a crucial role in computer vision and intelligent perception. It is essential for tasks like autonomous driving, surveillance, and robot navigation. Three-dimensional multi-object tracking requires the continuous and stable localization and association of multiple objects in three-dimensional space. The color images captured by camera sensors contain rich color and texture information, but their quality is significantly affected by lighting conditions and they inherently lack depth information. In contrast, the point cloud data acquired by 3D LiDAR sensors provides direct depth measurements and captures the geometric structure of objects, effectively overcoming the

limitations of color cameras. This has prompted many researchers to combine cameras and 3D LiDAR for multi-object tracking. However, with the rapid advancement of 3D object detection technology and the growing complexity of application scenarios, existing 3D multi-object tracking methods have revealed several limitations, failing to meet the rigorous requirements for tracking accuracy, efficiency, and robustness in practical applications.

In the 3D multi-object tracking task, existing research has generally overlooked the effective handling of false detections and ghost trajectories, which thereby affects overall tracking performance. Although some works [1,2] attempt optimization through detection result preprocessing or trajectory management strategies, they still have limitations and fail to fundamentally resolve the impact of ghost trajectories on tracking performance, as shown in Figure 1a. Furthermore, data association, a critical component of multi-object tracking, is typically performed using global matching strategies in existing methods [3,4], which calculate the matching cost between all detection results and predicted trajectories within the scene. This approach not only leads to a high redundancy in the association cost matrix and significant computational overhead, but the computational efficiency issue becomes worse, especially when multiple decision factors are introduced to improve association accuracy. Some studies [2,5] attempt to enhance matching robustness through multi-stage association strategies, but this increases both the risk of incorrect matches and the computational burden, as shown in Figure 1b. Finally, in trajectory management, existing methods [6,7] struggle to accurately determine whether an unmatched trajectory corresponds to a temporarily occluded object or one that has exited the scene. Most methods adopt a uniform strategy of keeping trajectories for a fixed number of frames, which increases both the risk of incorrect matches and computational costs. Moreover, the handling of objects with long-term occlusion is inadequate, as illustrated in Figure 1c.

To address the aforementioned issues, we propose a lightweight 3D multi-object tracking method via collaborative camera and LiDAR sensors. Our contributions are primarily as follows:

- We propose a confidence inverse normalization guided ghost trajectories suppression method. This method restores the original confidence distribution by reversing the normalization of detection result confidence. This enables effective discrimination between true objects and false detections, thereby mitigating the performance degradation caused by false detections induced by complex illumination and dynamic backgrounds. We also introduce virtual trajectories and an activation threshold strategy. Based on the confidence of virtual trajectories, we dynamically adjust their confirmation cycle, significantly reducing ghost trajectories caused by false detections while reducing the risk of trajectory loss.
- We design an adaptive matching space-driven lightweight association strategy. This strategy does not rely on computationally expensive global matching methods. Instead, it dynamically determines the matching range based on detection confidence. The detection results only build an association cost matrix with predicted trajectories within the specified range, thereby reducing computational cost and improving association accuracy. We also use Euclidean distance, relative distance, and motion direction angles to build the association cost matrix. This approach reduces computational costs while improving association accuracy, thereby meeting the real-time tracking requirements of autonomous driving and robot navigation.
- We construct a multi-factor collaborative perception-based intelligent trajectory management method. This method overcomes the limitation of retaining all unmatched trajectories. It uses factors such as trajectory position, object motion trends, and the observation distance from the data acquisition vehicle to collaboratively judge whether an unmatched trajectory corresponds to a temporarily occluded object or one that has

left the scene. By extending the retention period for occluded objects and terminating trajectories of objects that have exited the scene, we effectively reduce redundant computation and mitigate identity switches caused by occlusions.



Figure 1. Problems in the 3D multi-object tracking task. (**a**) Consecutive false detections are incorrectly identified as real objects and assigned an ID (548). The corresponding trajectories disappear after a single frame, forming typical ghost trajectories; (**b**) The ID (2472) of a vehicle is incorrectly assigned to a neighboring vehicle during prolonged occlusion due to the baseline's association strategy. When the original vehicle reappears, the system misidentifies it as a new object and assigns a new ID (2512); (**c**) A van passing by causes a prolonged occlusion of the vehicle with ID 12782. After the occlusion, the system fails to recognize the reappearing vehicle, mistakenly classifying it as a new object and assigning a new ID (12846).

2. Related Works

The three main paradigms commonly used in 3D multi-object tracking are: tracking-by-detection [8,9], joint detection and tracking [10,11], and attention-based tracking [12,13]. Among these, tracking-by-detection has become the mainstream framework due to its flexibility and scalability. Its core process includes object detection, state prediction, data association, and trajectory management. Although existing tracking methods have achieved significant improvements in accuracy, they still face significant challenges when deployed in complex real-world traffic scenarios, especially in dealing with ghost trajectories, reducing data association costs, and optimizing trajectory decisions. This paper reviews the current work in three aspects: ghost trajectories, data association, and trajectory management.

2.1. Ghost Trajectories

Ghost trajectories are abnormal trajectories caused by false detections, typically appearing in one frame and then disappearing. They often lead to issues such as incorrect

matching and identity switching. To address false detection problems, Kim et al. [14] proposed EagerMOT, which avoids false detections by projecting 3D detection results onto 2D and calculating their intersection and ratio with 2D detection results. However, this method heavily depends on the performance of the detector and tends to fail when there are discrepancies between the perception ranges of the camera and LiDAR. Pang et al. [1] proposed SimpleTrack, which uses Non-Maximum Suppression (NMS) before data association to eliminate overlapping low-quality detection results while retaining only one low-quality detection. However, this method cannot eliminate non-overlapping low-quality detections. To further enhance the robustness of NMS, Wang et al. [15] introduced YONTD-MOT, which incorporates historical trajectory confidence to guide NMS for lower-confidence prediction boxes before data association, reducing the likelihood of false detections. However, this method may mistakenly remove the true prediction box of occluded low-confidence objects. Li et al. [16] proposed Fast-Poly, which uses confidence and NMS as two independent processes to filter detection results, with accelerated computation in the NMS process. However, this method directly uses normalized confidence to filter low-quality detections, which can compress confidence score intervals, making it difficult to distinguish objects and increasing the risk of missed detections.

2.2. Data Association

Data association is critical for matching objects across frames and directly impacts tracking accuracy and robustness. Papais et al. [17] proposed SWTrack, which evaluates all trajectories within a sliding time window to find the most strongly correlated trajectory for association, thereby improving data association accuracy. However, this method requires iterating through all detection results and predicted trajectories in the entire scene, leading to high redundancy. To improve the discriminative power of the association matrix, Guo et al. [18] introduced a new association matrix model, which uses appearance features, object dimensions (length, width, height), and Mahalanobis distance to calculate appearance, geometry, and distance matrices, which together form the association matrix. Additionally, Zhang et al. [19] designed a more precise association matrix by enhancing the appearance matrix with additional correlated features and combining 3D GIoU (Generalized Intersection over Union) and 3D CIoU (Complete Intersection over Union) to improve the geometry matrix. However, while these methods, which incorporate multiple complex decision factors, provide more accurate information for data association and tracking, they also increase the redundancy and computational cost of the association matrix. To improve the robustness of data association, Pieroni et al. [20] proposed a three-level data association method, where each level processes detection results from different modal detectors. Zhu et al. [5] designed a four-stage data association method, where each stage determines which detection results to associate based on reliability, prioritizing the most reliable detections. While these multi-level data association methods enhance the robustness of the association strategy, they not only increase the risk of incorrect matches but also raise computational costs, impacting the real-time performance of the method.

2.3. Trajectory Management

Trajectory management involves post-processing the results of data association and is crucial for achieving stable tracking. Nagy et al. [21] proposed DFR-FastMOT, which addresses the long-term occlusion problem by utilizing long-term memory in the memory management module. While increasing the retention frames effectively handles long-term occlusion, it also significantly increases computational overhead. To prevent premature deletion of trajectories, which could lead to identity switching, Jin et al. [22] proposed EAFFMOT, which only deletes trajectories when there is no detection update for multiple

consecutive frames. Although this method reduces identity switching to some extent, it unnecessarily retains trajectories of objects that have exited the scene, reducing tracking efficiency. To address the complexity and error-proneness of manually handling unmatched trajectories and detection methods, Sadjadpour et al. [23] proposed ShaSTA-Fuse, which extends the affinity matrix by adding four types of anchor points, thus transforming lifecycle management into an extended matching problem of the affinity matrix. While this method enhances the robustness of trajectory management, it cannot distinguish whether an object is temporarily occluded or has left the scene, leading to suboptimal performance. Doll et al. [24] proposed S.T.A.R.-Track, which introduces learnable embedding vectors to implicitly encode the trajectory lifecycle, enhancing trajectory continuity and suppressing identity switching caused by repeated initialization after brief occlusion. However, this method lacks robustness in the case of long-term occlusions or objects in complex edge regions, as its uniform retention strategy often results in the incorrect deletion of trajectories, thus triggering identity switching.

3. Methodology

The performance of 3D multi-object tracking in complex scenarios is often limited by several fundamental and interrelated issues. First, the confidence scores from deep learning-based detectors, after normalization, result in compressed distribution intervals, making it difficult to effectively distinguish between true objects and high-confidence false detections. These persistent false detections lead to disruptive ghost trajectories, which severely compromise tracking stability. Second, mainstream methods rely on global data association strategies, where computational costs increase quadratically with the number of objects. Moreover, introducing multiple complex decision factors further exacerbates computational redundancy, making it challenging to meet real-time application requirements. Finally, traditional trajectory management uses a "one-size-fits-all" fixed frame retention strategy, which cannot distinguish whether an object is temporarily occluded or has permanently left the scene. This leads to unnecessary computations for objects that have left the scene and identity switching caused by the premature deletion of temporarily occluded objects.

This study proposes a lightweight 3D multi-object tracking method via collaborative camera and LiDAR sensors. The main contributions include the confidence inverse normalization guided ghost trajectories suppression (CIGTS) module, the adaptive matching space-driven lightweight association (AMSLA) module, and the multi-factor collaborative perception-based intelligent trajectory management (MFCTM) module. The overall process is shown in Figure 2. The image and point cloud data are processed by a detector to generate corresponding detection results. Then, the confidence of the detection results undergoes inverse normalization and is filtered based on a set threshold for low-confidence detections. During the data association phase, an adaptive matching space-driven lightweight association strategy is applied. The filtered detection results are matched with predicted trajectories, yielding three outputs: matched detection–trajectory pairs, unmatched detection results, and unmatched predicted trajectories. Specifically, the matched detection–trajectory pairs undergo state updates using the Kalman filter. Unmatched detection results are used to initialize virtual trajectories, which are managed according to a survival frame count strategy based on their confidence levels. The unmatched trajectories are fed into an intelligent trajectory management method with multi-factor collaborative perception for further assessment, which determines the current state of each object corresponding to an unmatched trajectory and applies targeted processing accordingly.

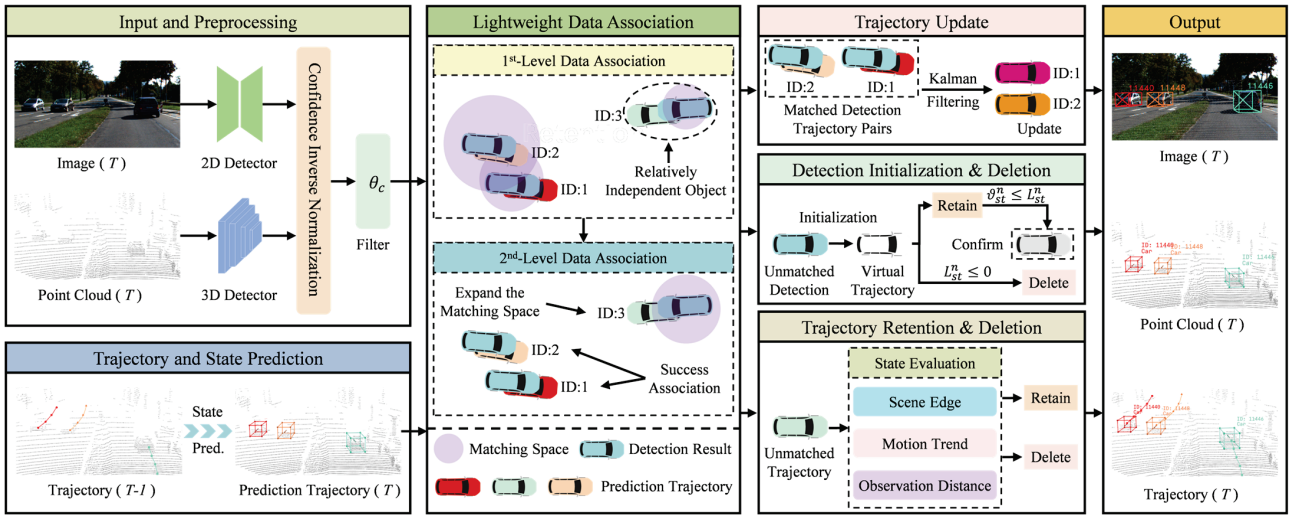


Figure 2. Overall framework of the proposed method.

3.1. Confidence Inverse Normalization Guided Ghost Trajectories Suppression (CIGTS)

Existing object detectors often generate a large number of detection results to improve recall rates, and these redundant detections can easily lead to false detections and ghost trajectories. However, most recent methods overlook the preprocessing of detection results and the suppression of ghost trajectories, resulting in suboptimal performance. To address this, as shown in Figure 3, we propose a confidence inverse normalization guided ghost trajectories suppression (CIGTS) module. This module first applies inverse normalization to the object confidence scores, restoring their original distribution to improve the distinguishability of low-quality detections. It then filters out low-quality detection results based on a preset low-confidence threshold, reducing false detections. Furthermore, leveraging the inherent difficulty for false detections to form continuous trajectories, a virtual trajectory management mechanism based on survival frame count is introduced. Specifically, for newly detected objects, a virtual trajectory with an associated survival frame count is initialized. When the survival frame count exceeds a preset activation threshold, the trajectory is converted into a formal trajectory, with different activation thresholds set based on the object's confidence.

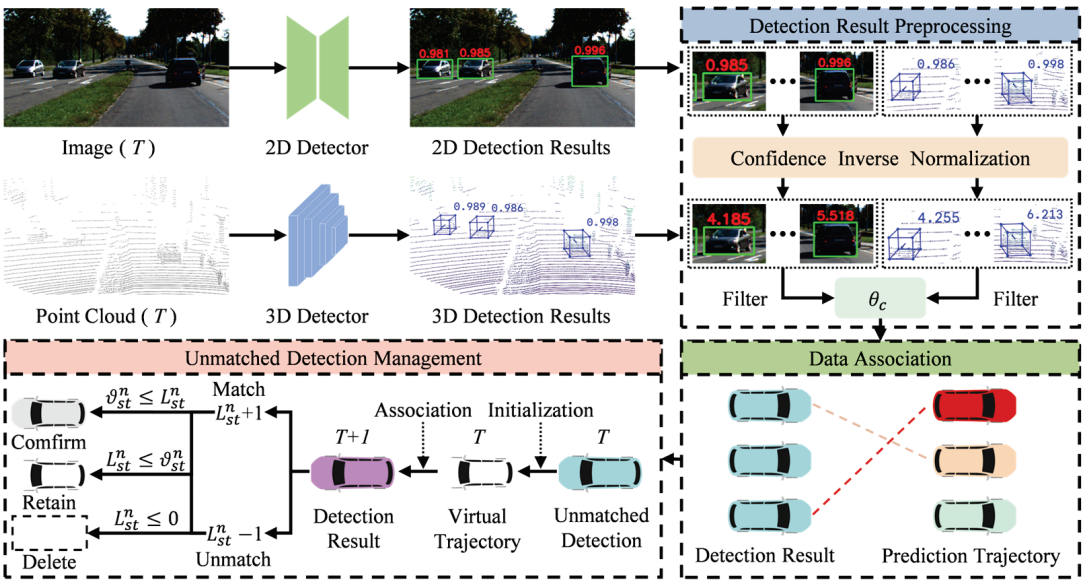


Figure 3. Confidence inverse-normalization-guided ghost trajectories suppression module.

3.1.1. Detection Result Preprocessing

Image data and point cloud data are processed separately by 2D and 3D detectors to obtain the corresponding detection result sets D_{st}^{2d} and D_{st}^{3d}, where s denotes the current scene index and t denotes the current frame index. Since each detection result set is composed of a series of detection results, they can be further expressed as $\left\{D_{st}^{i}\right\}_{i=1}^{N_s^t}$ and $\left\{D_{st}^{j}\right\}_{j=1}^{N_s^t}$, where N_s^t denotes the number of detection results in frame t of scene s. Here, $D_{st}^{i} \in \mathbb{R}^{M\times 1}$ represents the i-th 2D detection result, and $D_{st}^{j} \in \mathbb{R}^{N\times 1}$ represents the j-th 3D detection result, with M and N denoting the dimensionality of the detection results. The specific forms are given as: $D_{st}^{j} = \left[t, type, x_{lst}^{j}, y_{lst}^{j}, x_{rst}^{j}, y_{rst}^{j}, s_{st}^{j}, h, w, l, x, y, z, r, a\right]$ and $D_{st}^{i} = \left[t, x_{lst}^{i}, y_{lst}^{i}, x_{rst}^{i}, y_{rst}^{i}, s_{st}^{i}\right]$. t is the current frame index. $(x_{lst}^{i}, y_{lst}^{i})$, $(x_{lst}^{j}, y_{lst}^{j})$ and $(x_{rst}^{i}, y_{rst}^{i})$, $(x_{rst}^{j}, y_{rst}^{j})$ are coordinates of the top-left and bottom-right corners of the 2D bounding box, respectively. s_{st}^{i} and s_{st}^{j} are normalized confidence scores of 2D/3D detections. $type$ denotes object category. (h, w, l) are height, width, and length of the 3D bounding box. (x, y, z) denote center coordinates of the 3D bounding box. r denotes motion orientation angle of the object. a is observation angle of the object.

When outputting detection results, existing detectors typically apply the sigmoid function to normalize confidence scores into the range [0, 1]. However, the intervals in the original confidence score space are compressed by the nonlinear sigmoid transformation. Extreme values are pushed into the saturation regions near 0 or 1, shrinking their relative differences, while intermediate values become more concentrated. This increases the sensitivity of detection results to different threshold settings and makes it more difficult to determine appropriate confidence thresholds. Moreover, this is the main reason why directly using normalized confidence scores to filter out low-quality detections often leads to missed detections. Therefore, in order to more accurately preserve high-quality detection results while eliminating redundant or erroneous low-quality detections, we first apply inverse normalization to restore the original confidence values (taking the 3D detection results as an example), as shown in Equation (1):

$$\hat{s}_{st}^{j} = \sigma^{-1}(s_{st}^{j}) = log(\frac{s_{st}^{j}}{1 - s_{st}^{j}}) \tag{1}$$

where s_{st}^{j} denotes the normalized confidence score of the 3D detection result, $\sigma^{-1}(\cdot)$ represents the inverse sigmoid operation, and $\hat{s}_{st}^{j}$ is the restored original confidence score. By recovering the confidence scores, detections can be filtered more effectively. Specifically, we remove low-confidence detections by applying a preset threshold θ_c, as defined in Equation (2):

$$\hat{D}_{st}^{3d} = \left\{ D_{st}^{j} \in D_{st}^{3d} \middle| \hat{s}_{st}^{j} \geq \theta_c \right\} \tag{2}$$

where D_{st}^{3d} represents the original set of 3D detection results, and $\hat{D}_{st}^{3d}$ denotes the filtered candidate set after applying the confidence threshold.

3.1.2. Virtual Trajectories

Although inverse normalization confidence filtering effectively reduces interference from low-quality detections, some false detections with relatively high confidence scores may still remain. These objects typically exhibit intermittent characteristics. if directly initialized as formal trajectories, they are prone to producing ghost trajectories. To address this issue, we introduce a virtual trajectory mechanism.

Virtual trajectories serve as candidate trajectories for formal ones. By validating the persistence and reliability of a newly detected object over a certain period, only those

virtual trajectories that are continuously associated with detections are ultimately confirmed as formal trajectories. The representation of a virtual trajectory is defined as: $\ddot{T}_{st}^{n} = \left[t, id, type, x_{lst}^{n}, y_{lst}^{n}, x_{rst}^{n}, y_{rst}^{n}, h, w, l, x, y, z, r, \hat{s}_{st}^{n}, L_{st}^{n}\right]$. Similar to 3D detection results, the representation includes the inverse normalization confidence score $\hat{s}_{st}^{n}$, while two additional state variables are introduced: *id* as trajectory identifier and L_{st}^{n} as survival frame count of the n-th virtual trajectory in frame t of scene s. Each virtual trajectory $\ddot{T}_{st}^{n}$ is initialized with a survival frame count of $L_{st}^{n} = 1$. For each subsequent frame, if the virtual trajectory $\ddot{T}_{st}^{n}$ is successfully matched with a detection, its survival frame count is incremented. When L_{st}^{n} exceeds a predefined activation threshold, the virtual trajectory is promoted to a formal trajectory. Conversely, if a virtual trajectory $\ddot{T}_{st}^{n}$ fails to match with any detection, its survival frame count is decremented; once $L_{st}^{n} = 0$, the virtual trajectory is deleted.

In real-world scenarios, low-confidence objects are more likely to be false detections compared to high confidence objects. Therefore, it is unreasonable to assign a uniform activation threshold ϑ to all virtual trajectories. To this end, we propose a confidence-aware stratified activation strategy, which suppresses ghost trajectories while reducing trajectory loss. Specifically, as shown in Equation (3), virtual trajectories are divided according to a confidence stratified threshold θ_l. For virtual trajectories with confidence scores greater than or equal to θ_l, the activation threshold is set to δ; otherwise, it is set to ξ:

$$\vartheta_{st}^{n} = \begin{cases} \delta, & \hat{s}_{st}^{n} \geq \theta_l \\ \xi, & \hat{s}_{st}^{n} < \theta_l \end{cases} \quad where\ \delta > \xi \tag{3}$$

where ϑ_{st}^{n} denotes the activation threshold of a virtual trajectory. This differentiated activation strategy can effectively suppress ghost trajectories while ensuring the completeness of true object trajectories as much as possible.

3.2. Adaptive Matching Space-Driven Lightweight Association (AMSLA)

Data association is the core task in multi-object tracking. Existing methods enhance the accuracy of data association through global association, the introduction of multiple complex decision factors, and the design of multi-stage association strategies, but they tend to have poor real-time performance due to their high computational costs. To address this, as shown in Figure 4, we propose an adaptive matching space-driven lightweight association (AMSLA) module. In this module, detection results are only associated with predicted trajectories that fall within the same matching space, thus avoiding the high computational costs of global association. Additionally, this module does not rely on unstable factors such as the aspect ratio of bounding boxes or appearance information with complex feature engineering for constructing the association matrix. Instead, it utilizes the intrinsic data characteristics of the object, using the motion direction angle and relative distance (the absolute difference in distance between the object and the data acquisition vehicle) as decision factors. This allows for reducing computational costs while maintaining accuracy.

The data association task involves matching detection results and predicted trajectories, followed by dividing the matched results into corresponding matching spaces based on the object's trajectory ID. The specific form of the predicted trajectory is as follows: $\hat{T}_{st}^{n} = \left[t, id, type, a, x_{lst}^{n}, y_{lst}^{n}, x_{rst}^{n}, y_{rst}^{n}, h, w, l, x, y, z, r, \hat{s}_{st}^{n}\right]$. a represents the observation angle. The state representation of a virtual trajectory is consistent with that of the detection results. After obtaining the detection results and predicted trajectories, a matching space is first determined for each detection result, whose matching radius is denoted as R, and then the association matching is performed with the predicted trajectory in that space. Since

high-confidence objects have relatively stable motion states, while low-confidence objects may exhibit issues such as drift, the matching space for high-confidence detection results is set smaller, while the matching space for low-confidence detection results is set larger, as shown in Equation (4):

$$R_{st}^{n} = R_{min} + \left(\frac{s_{st}^{max} - \hat{s}_{st}^{j}}{s_{st}^{max} - s_{st}^{min}}\right) \cdot (R_{max} - R_{min}) \tag{4}$$

where R_{st}^{n} and $\hat{s}_{st}^{j}$ represent the corresponding matching radius and confidence of detection results. R_{min} and R_{max} represent the minimum and maximum matching radii, based on the statistical results of the object's relative movement. We set R_{min} and R_{max} to 3 and 3.5, respectively. s_{st}^{max} and s_{st}^{min} represent the maximum and minimum confidence scores in the scene s with the t-th frame.

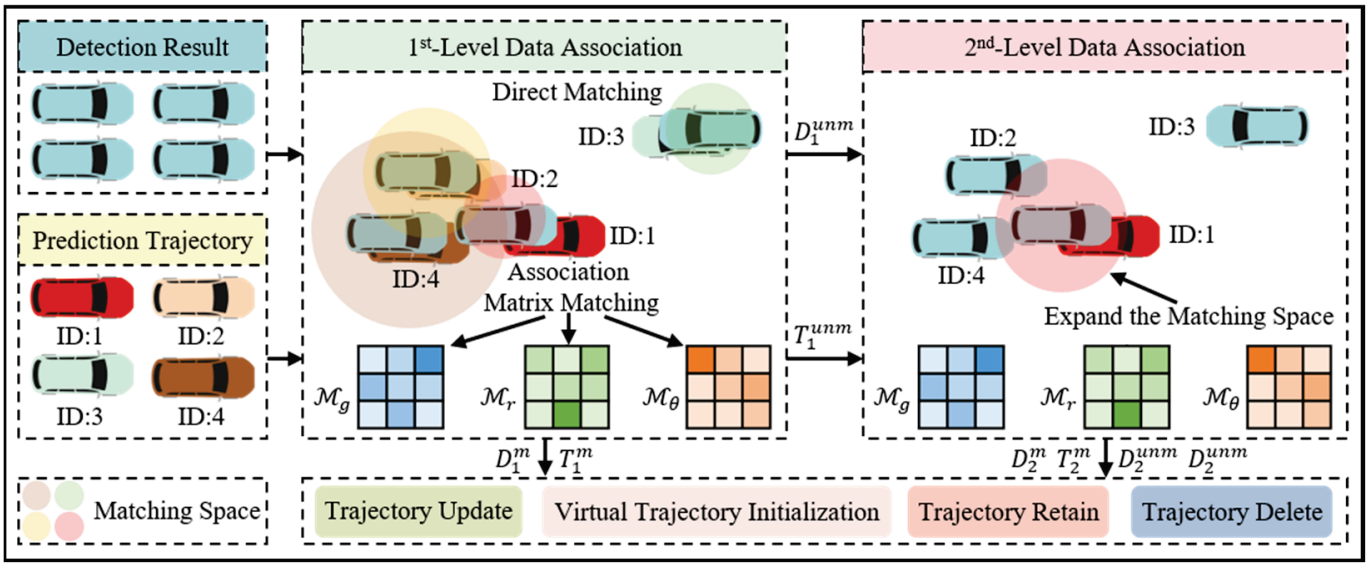


Figure 4. Adaptive matching space-driven lightweight data association module.

After determining the matching space, to establish a connection between the detection results and the predicted trajectories in the surrounding environment, we use the Euclidean distance to construct a global distance matrix. The center coordinates of the object are denoted as (x, y, z), as shown in Equation (5):

$$\mathcal{M}_g(m, n) = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + (z_m - z_n)^2} \tag{5}$$

where $\mathcal{M}_g(m, n)$ represents the global distance matrix, (x_m, y_m, z_m) and (x_n, y_n, z_n) represent the center point coordinates of two different detection results (or predicted trajectories) in the current frame. Then, the detection results and predicted trajectories in each matching space can be filtered according to the matching radius and the global distance matrix.

During data association, if a matching space contains only one detection and one trajectory, this indicates that the object is well-separated from all other objects in the scene. A direct match can be applied without further calculation. Conversely, if the matching space contains multiple detections or predicted trajectories, it is then necessary to construct an association cost matrix for all detections and trajectories within that space.

We enhance the accuracy of data association by improving the discriminative power of the association matrix. Specifically, we use the previously constructed global distance matrix as the association matrix. Furthermore, we observe that the relative distance between

an object and the data acquisition vehicle can effectively distinguish different objects, even when occlusion or deformation occurs. Therefore, we also incorporate this computationally efficient relative distance to construct the association matrix, where the distance between the object and the data acquisition vehicle is denoted as z, as shown in Equation (6):

$$\mathcal{M}_r(m,n) = |z_m - z_n| \tag{6}$$

where $\mathcal{M}_r(m,n)$ represents the relative distance matrix, z_m and z_n represent the distances from two distinct detection results (or predicted trajectories) in the current frame to the data acquisition vehicle, respectively. However, in real-world scenarios, roads are often multi-lane and support bidirectional traffic. Relying solely on relative distance fails to distinguish between objects that are close in proximity but moving in different directions. Therefore, to further enhance the discriminative power of the association matrix, we introduce the motion direction angle r from the object (or trajectory) state information to construct the association matrix, as shown in Equation (7):

$$\mathcal{M}_\theta(m,n) = min(|r_m - r_n|,\ 2\pi - |r_m - r_n|) \tag{7}$$

where $\mathcal{M}_\theta(m,n)$ represents the motion direction matrix, and r_m and r_n represent the motion direction angles of the two different detection results (or predicted trajectories) in the current frame. As mentioned earlier, when the detection results or predicted trajectories in the matching space are not unique, we will consider the overall Euclidean distance, relative distance, and motion direction angle for the association. The weight coefficients corresponding to the global distance matrix, relative distance matrix, and motion direction matrix are denoted as α, β, and γ, respectively, as shown in Equation (8):

$$\mathcal{M}(m,n) = \alpha \cdot \mathcal{M}_g(m,n) + \beta \cdot \mathcal{M}_r(m,n) + \gamma \cdot \mathcal{M}_\theta(m,n) \tag{8}$$

where $\mathcal{M}(m,n)$ denotes the weighted comprehensive association cost matrix. The weight factors satisfy $0 < \alpha, \beta, \gamma < 1$ and $\alpha + \beta + \gamma = 1$. Based on experimental data, the weight factors α, β, γ are set as 0.4, 0.3 and 0.3, respectively.

Using the comprehensive association cost matrix, data association is performed within the matching space. Successfully matched detection–trajectory pairs are removed from all matching spaces, with their corresponding matching spaces being deleted accordingly. Upon completing the data association process, four distinct outcomes are identified: matched detections D_1^m, matched trajectories T_1^m, unmatched detections D_1^{unm}, and unmatched trajectories T_1^{unm}. Considering that real-world scenarios may involve objects with significant inter-frame displacement, we proceed to conduct a second stage of data association specifically for the unmatched detections and trajectories. Specifically, we expand the matching space for unmatched detections to μ times its original size, while also incorporating 2D detection results to address unmatched trajectories caused by omissions from the 3D detector. After the secondary data association, we similarly obtain four outcomes: matched detections D_2^m, matched trajectories T_2^m, unmatched detections D_2^{unm}, and unmatched trajectories T_2^{unm}. Finally, the matched detections (D_1^m,D_2^m) and matched trajectories (T_1^m,T_2^m) from both the first and second data association stages, along with the unmatched detections D_2^{unm} and unmatched trajectories T_2^{unm} from the second association stage, are fed into the trajectory management module for operations such as updating, initializing, and terminating trajectories.

3.3. Multi-Factor Collaborative Perception-Based Intelligent Trajectory Management (MFCTM)

Trajectory management aims to process the multiple outcomes of data association, preventing identity switches and trajectory fragmentation. However, existing methods fail to distinguish between occluded objects and objects that have left the scene. They uniformly retain unmatched trajectories for a fixed number of frames, which not only increases computational overhead but also leads to issues such as erroneous matches. To address this, as shown in Figure 5, we propose a multi-factor collaborative perception-based intelligent trajectory management (MFCTM) module. This module first uses the 2D bounding box coordinates of unmatched trajectories to determine whether the corresponding object is at the boundary of the scene. Then, it combines the motion trend analysis of two consecutive frames' bounding boxes to identify the direction of the object's movement, further distinguishing whether the object at the scene's boundary is entering or leaving the scene. Finally, for objects far from the scene's boundary, it uses the observation distance between the data acquisition vehicle and the object (the depth information z from the 3D bounding box center coordinates) to determine if the object has moved beyond the perception range and has left the scene. Through this multi-factor collaborative perception mechanism, the trajectory of objects that have left the scene can be promptly terminated to reduce computational load, while reliably retaining the trajectories of occluded objects to wait for their reappearance. This approach effectively reduces the number of identity switches and improves overall tracking efficiency.

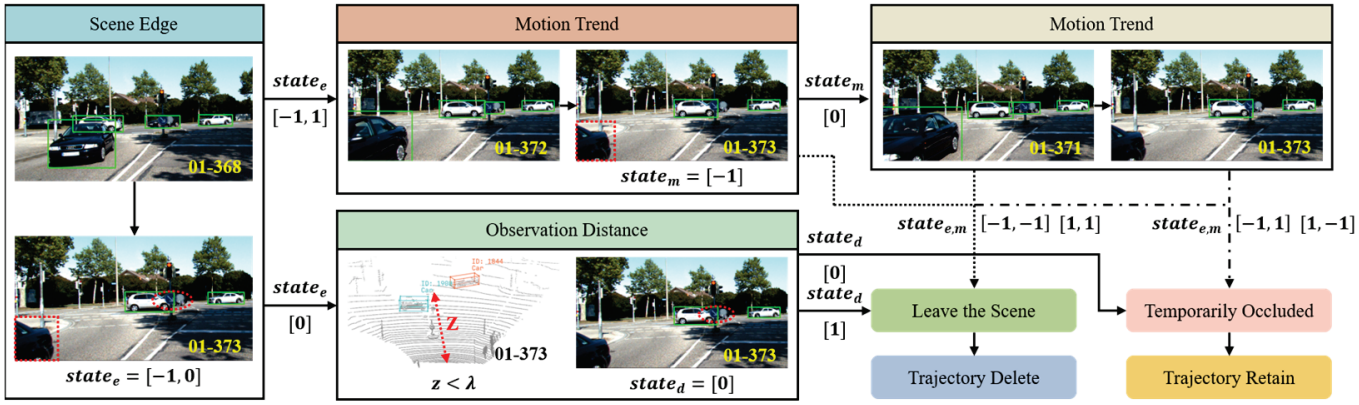


Figure 5. Multi-factor collaborative perception-based intelligent trajectory management module.

3.3.1. Trajectory Management

The adaptive matching space-driven lightweight association module outputs six sets of results to the trajectory management module. For the matched detection–trajectory pairs D_1^m_T_1^m and D_2^m_T_2^m, Kalman filtering is applied to update the corresponding trajectories T_1^m and T_2^m using the detection results D_1^m and D_2^m, respectively. To account for false detections in the detection results, a virtual trajectory is initialized for the unmatched detection result D_2^{unm}. It is converted into a formal trajectory only when its survival frame count exceeds the activation threshold. For the unmatched trajectories T_2^{unm}, the object state determination strategy is employed to determine whether the corresponding object is temporarily occluded or has left the scene. Since trajectories represent the historical output of detection results, the object's status in the current frame can be determined based on the trajectories. The decision process is as follows: (1) determine whether the trajectory is located at the scene boundary; (2) determine the motion trend of the trajectory; (3) determine whether the trajectory exceeds the perception range. If the object corresponding to the unmatched trajectories T_2^{unm} has left the scene, it is immediately deleted to reduce the computational overhead of retaining trajectories. Otherwise, the object is considered occluded, and a larger survival frame count φ is set for it. When the number of consecutive

unmatched frames exceeds φ, the corresponding trajectory is deleted. This enhances the ability to handle long-term occlusions and reduces identity switches.

3.3.2. Object State Determination

Scene Boundary: We observe that all objects consistently exit the scene from either the left or right side. Therefore, whether an unmatched trajectory corresponds to an object that has left the scene can be determined by assessing its position relative to the scene boundary. The 2D bounding box coordinates within a trajectory contain spatial information about the object's position in the scene. When the x-coordinate of the top-left corner x_{lst} of the 2D bounding box corresponding to an unmatched trajectory is approximately 0, or the x-coordinate of the bottom-right corner x_{rst} is approximately w_{max} (the maximum scene width), it indicates that the object is located at the scene boundary or has partially exited the scene. In such cases, the most likely reason for the trajectory being unmatched is that the corresponding object has already left the scene. We introduce a small value ϵ (3 pixels) to avoid noise interference. The specific decision process is shown in Equation (9):

$$state_e = \begin{cases} -1, & if\ x_{lst} - 0 \leq \epsilon \\ 1, & if\ |x_{rst} - w_{max}| \leq \epsilon \\ 0, & else \end{cases} \tag{9}$$

where $state_e$ represents the boundary state of the trajectory. -1 indicates that the trajectory is located at the left boundary, 1 indicates the right boundary, and 0 signifies the trajectory is far from any boundary. For unmatched trajectories at the scene boundary, they are not immediately deleted; instead, their motion trend is further analyzed. Similarly, trajectories far from the boundary are not immediately retained; rather, their observation distance is further evaluated.

Motion Trend: In real-world scenarios, there are cases where an object is quickly occluded after entering the scene. If the trajectory is judged solely based on whether it is located at the scene boundary, the corresponding trajectory of the object will be deleted, and when the occlusion ends and the object reappears in the scene, this will lead to an identity switch. Therefore, we introduce motion trend to further determine whether the object is entering or leaving the scene. Specifically, when the unmatched trajectory is located at the left boundary, its top-left corner's horizontal coordinate is fixed at 0, so we determine its motion trend based on the change in the horizontal coordinate of the bottom-right corner. Similarly, for unmatched trajectories located at the right boundary, their motion trend is determined based on the change in the horizontal coordinate of the top-left corner. We introduce a small value δ (2 pixels) to filter out interference caused by minor bounding box jitter, as shown in Equations (10) and (11):

$$\Delta x = \begin{cases} x_{rst} - x_{rst-1}, & if\ state_e = -1 \\ x_{lst} - x_{lst-1}, & if\ state_e = 1 \end{cases} \tag{10}$$

$$state_m = \begin{cases} -1, & if\ \Delta x < -\delta \\ 1, & if\ \Delta x > \delta \\ 0, & else \end{cases} \tag{11}$$

where x_{rst}, x_{lst} and x_{rst-1}, x_{lst-1} denote the x-coordinates of the bottom-right and top-left corners of the unmatched trajectory at frame t and frame $t-1$, respectively; Δx represents the offset of the trajectory between two consecutive frames; $state_m$ indicates the motion state of the trajectory: -1 denotes a leftward offset, 1 denotes a rightward offset, and 0 signifies that the motion trend is undefined and requires further judgment based on historical data, as shown in Equation (12):

$$\Delta x = \begin{cases} x_{rst} - x_{rst-2}, & if\ state_e = -1 \\ x_{lst} - x_{lst-2}, & if\ state_e = 1 \end{cases} \tag{12}$$

where x_{rst-2} and x_{lst-2} denote the x-coordinates of the bottom-right and top-left corners of the unmatched trajectory at frame $t-2$, respectively. When the combined state $[state_e, state_m]$ is $[-1, -1]$ or $[1, 1]$, it indicates that the object corresponding to the unmatched trajectory is in the process of leaving the scene, and the respective unmatched trajectory is immediately deleted. Conversely, a combined state of $[-1, 1]$ or $[1, -1]$ indicates that the object is entering the scene, and the respective unmatched trajectory is retained to ensure it can be reassociated when the object reappears.

Observation Distance: In real-world scenarios, there exists a special category of objects that, although far from the scene boundary, can no longer be perceived or tracked due to the limited perception range of the detector and the excessive distance between the object and the data acquisition vehicle. Since the distance between such objects and the acquisition vehicle continues to increase in subsequent frames, they should also be considered to have left the scene, and their corresponding trajectories should be deleted. Specifically, by comparing the observation distance z of an unmatched trajectory with the perception threshold λ, it is possible to determine whether the corresponding object has moved beyond the perceivable range due to excessive distance in the current frame, as shown in Equation (13):

$$state_d = \begin{cases} 1, & if\ z > \lambda \\ 0, & else \end{cases} \tag{13}$$

where $state_d$ denotes the distance state of a trajectory: 1 indicates that the object corresponding to the unmatched trajectory has exceeded the perception range, is considered to have left the scene, and the unmatched trajectory is deleted; 0 indicates that the object lies within the perception range, is considered temporarily occluded, and the unmatched trajectory is retained.

In summary, the proposed object state determination strategy employs a collaborative decision-making mechanism that integrates scene boundary awareness, motion trend modeling, and observation distance analysis. This multi-faceted approach accurately distinguishes between the exiting and occluded states of unmatched trajectories. Consequently, it significantly reduces computational overhead caused by redundant trajectory retention while effectively preventing identity switches resulting from premature deletion of the occluded trajectories, thereby achieving intelligent and efficient trajectory management in complex scenarios.

4. Experiments

4.1. Experiment Setting

4.1.1. Datasets

To evaluate the performance of the proposed 3D multi-object tracking algorithm, we conduct systematic experimental validations on the KITTI [25] tracking benchmark, which is widely adopted in the field of autonomous driving. Recognized as one of the most authoritative evaluation platforms for 3D multi-object tracking tasks, the KITTI dataset is distinguished by its high data quality, realistic scenarios, and fine-grained annotations. For the multi-object tracking task, the KITTI benchmark provides 21 fully annotated training sequences and 29 unannotated test sequences. All sequences are captured from real-road environments, covering diverse typical driving scenarios such as urban streets, rural roads, and highways. These sequences incorporate multi-dimensional challenges including variations in object density, occlusion levels, and illumination conditions, thereby enabling

a comprehensive assessment of the algorithm's robustness and generalization capability under complex traffic conditions.

4.1.2. Evaluation Metrics

We adopt a recognized and up-to-date comprehensive metric system to evaluate the tracking algorithm's accuracy and robustness from multiple dimensions. Specifically, we employ Higher Order Tracking Accuracy (HOTA) [26] as the primary evaluation metric, as it provides a more balanced and accurate characterization of performance in complex scenarios. Furthermore, we introduce a series of metrics from the CLEAR [27] evaluation protocol for supplementary validation of the method's effectiveness. These include: Multi-Object Tracking Accuracy (MOTA), Multi-Object Tracking Precision (MOTP), False Positives (FP), False Negatives (FN), Identity Switches (IDSW), Association Accuracy (AssA), and Localization Accuracy (LocA).

4.1.3. Implementation Details

The model is implemented based on the PyTorch 1.9.1 deep learning framework utilizing Tesla V100 SXM2 GPUs from NVIDIA Corporation (Santa Clara, CA, USA). The Python and CUDA versions are 3.8.16 and 11.1, respectively. The input data consists of three parts: images, point clouds, and trajectories from the previous timestamp. The image and point cloud data are in one-to-one correspondence, comprising 21 training sequences and 29 test sequences. Specifically, the training set contains 8008 frames, while the test set contains 11,095 frames. The trajectory data are updated using a Kalman filter. The cost matrix weight coefficients α, β, and γ are set to 0.4, 0.3, and 0.3, respectively. The matching radii, R_{min} and R_{max}, are set to 3 and 3.5, respectively. The optimal values for the confidence filtering threshold θ_c, confidence stratified threshold θ_l, activation thresholds δ and ξ, matching space expansion factor μ, trajectory retention frame count φ, and perception threshold λ were determined through ablation studies. The Adaptive Moment Estimation (Adam) optimizer was employed, with the initial learning rate, weight decay coefficient, and momentum set to 0.01, 0.01, and 0.9, respectively. For the 3D and 2D detectors, PointRCNN [28] and RRC [29] were used, respectively, aligning with the baseline method to ensure a fair comparison.

4.2. Quantitative Results

We conduct a systematic evaluation of the widely-used KITTI tracking benchmark to comprehensively validate the effectiveness of the proposed lightweight tracking method via collaborative camera and LiDAR sensors for the 3D multi-object tracking task. Given that existing studies are typically optimized for specific object categories (e.g., cars or pedestrians), we accordingly performed detailed comparisons between our method and current state-of-the-art approaches separately for both the Car and Pedestrian categories. The results demonstrate that our method outperforms existing techniques across multiple key metrics. The detailed evaluation results are presented in Tables 1 and 2. For clarity in presenting the performance ranking, the top first, second, and third methods for each metric are highlighted in red, blue, and green, respectively. An upward arrow (↑) denotes higher values are better, while a downward arrow (↓) signifies lower values are preferable.

As evidenced by the comprehensive comparison in Tables 1 and 2, the proposed lightweight 3D multi-object tracking method via collaborative camera and LiDAR sensors achieves highly competitive results on the KITTI test set. It demonstrates significant advantages across multiple key metrics, particularly in HOTA, AssA, LocA, MOTP, and IDSW. This superior performance is primarily attributed to the synergistic effect of the three proposed core modules. Firstly, the Confidence Inverse-Normalization-Guided Ghost Trajectories Suppression (CIGTS) module significantly enhances the discriminative capabil-

ity for objects by recovering the original confidence distribution, effectively suppressing false positives and reducing missed detections. Consequently, this leads to a reduction in FP and FN metrics and an improvement in LocA. Compared to the baseline model, our method reduces the FP and FN metrics by 21.3% and 20.1%, respectively. Secondly, the Adaptive Matching Space-Driven Lightweight Association (AMSLA) module discards the redundant global matching strategy. It adaptively defines the matching range based on object confidence and constructs a highly discriminative association cost matrix by incorporating low-complexity motion and geometric features. This approach significantly improves association accuracy while substantially reducing computational costs, leading to notable improvements in the AssA and IDSW metrics. Finally, the Multi-Factor Collaborative Perception-based Intelligent Trajectory Management (MFCTM) module accurately distinguishes between occluded objects and exiting objects through multi-dimensional state assessment of unmatched trajectories. This strategy not only avoids premature deletion of valid trajectories but also promptly terminates redundant ones, thereby preventing incorrect associations. Consequently, it further reduces IDSW while simultaneously improving the MOTP metric.

Table 1. Evaluation result of "Car" on KITTI test set.

Methods	HOTA↑	AssA↑	LocA↑	MOTA↑	MOTP↑	FP↓	FN↓	IDSW↓
DeepfusionMOT [2]	75.46	80.05	86.70	84.63	85.02	4601	601	84
BcMODT [19]	71.00	69.14	86.93	85.48	85.31	3353	1260	381
Seg2Track-SAM2 [30]	60.42	67.95	78.93	61.60	75.93	7888	5124	193
JHIT [31]	79.21	82.29	86.91	89.80	85.37	2273	1058	177
NC2 [32]	71.85	74.81	87.30	78.52	85.84	2676	4554	159
EAFFMOT [22]	72.28	73.08	86.73	84.77	85.08	3946	1185	107
SpbTracker [33]	72.66	71.43	87.48	86.51	86.07	3508	875	257
UG3DMOT [34]	78.60	82.28	87.84	87.98	86.56	2993	1111	30
MMF-JDT [35]	79.52	84.01	87.65	88.06	86.24	2317	1751	37
CollabMOT [36]	80.02	81.86	87.14	91.70	85.77	2063	583	207
S3MOT [37]	76.86	77.41	87.87	86.93	86.60	1899	2053	543
C-TwiX [38]	77.58	78.84	86.95	89.68	85.50	2814	355	381
SG-LKF [39]	79.59	82.53	87.09	90.55	85.66	2741	348	160
Ours	80.13	84.20	88.55	87.93	87.45	3620	480	52

Table 2. Evaluation result of "Pedestrian" on KITTI test set.

Methods	HOTA↑	AssA↑	LocA↑	MOTA↑	MOTP↑	FP↓	FN↓	IDSW↓
Seg2Track-SAM2 [30]	44.40	50.51	78.93	37.81	69.31	8044	6058	296
StrongFusionMOT [7]	43.42	48.83	70.53	39.04	63.89	8727	5069	316
JHIT [31]	54.07	56.88	78.38	64.95	64.59	5903	1927	284
MSA-MOT [5]	44.73	49.34	71.21	47.86	64.35	7761	4101	209
C-TWiX [38]	52.44	54.35	79.15	64.95	75.28	6632	1160	381
RAM [40]	52.71	52.19	77.70	68.40	73.61	5394	1660	262
MO-YOLO [41]	51.46	58.39	77.86	56.84	73.74	8068	1759	164
EAFFMOT [22]	40.20	45.63	71.25	42.01	64.57	10,793	2431	201
APPTracker+ [42]	42.73	41.15	78.30	55.45	74.22	7936	2076	302
SpbTracker [33]	43.25	44.79	71.87	53.55	65.28	8591	1961	200
NC2 [32]	44.30	46.75	72.08	44.18	65.68	6176	6415	159
FNC2 [43]	46.55	46.68	72.07	56.05	65.68	6160	3679	333
MMTrack [44]	49.28	55.33	79.26	56.19	75.34	9081	886	175
Ours	53.24	58.53	78.49	62.34	74.47	7242	1336	141

The pedestrian category is characterized by smaller object size, weaker appearance features, frequent occlusion, and non-rigid motion, presenting significant challenges in

distinguishing false positives from true objects, and exiting objects from temporarily occluded ones. As shown in Table 2, for the pedestrian tracking task, the proposed method achieves first place in HOTA, AssA, and IDSW metrics, and secures second place in LocA, MOTA, and MOTP metrics. These results convincingly demonstrate the effectiveness of our method in tracking non-rigid, small-scale, and easily occluded objects.

4.3. Ablation Experiments

To systematically evaluate the effectiveness of the proposed method, we conduct component-wise ablation studies on the KITTI training set for the three core modules: the CIGTS module, the AMSLA module, and the MFCTM module. These studies aim to quantify the contribution of each module to the overall tracking performance. The experiments employ the official KITTI benchmark evaluation metrics, including HOTA, MOTA, MOTP, FP, FN, and IDSW. Among these, HOTA, MOTA, and MOTP are positive metrics where higher values indicate better performance, whereas FP, FN, and IDSW are negative metrics for which lower values signify superior tracking results. The detailed ablation study results are presented in Table 3.

Table 3. Ablation study on the proposed modules.

Methods	HOTA↑	MOTA↑	MOTP↑	FP↓	FN↓	IDSW↓	FPS↑
Baseline	77.449	87.275	86.602	1028	2722	83	110
Baseline + CIGTS	78.603	88.106	87.364	910	2642	69	117
Baseline + CIGTS + AMSLA	80.125	88.97	88.155	826	2317	50	128
Baseline + CIGTS + AMSLA + MFCTM	81.498	89.663	89.110	724	2228	36	135

As evidenced in Table 3, incorporating the CIGTS module into the baseline model yields marked improvements across all evaluation metrics. This enhancement is primarily attributed to the module's ability to perform inverse normalization on detection confidence, which sharpens the distinction between true objects and false positives. Concurrently, the introduction of virtual trajectories and a hierarchical activation mechanism effectively suppresses ghost trajectories caused by false positives while mitigating the risk of losing true objects due to threshold sensitivity. Replacing the original association strategy in the baseline model with the AMSLA module further elevates multiple tracking performance metrics. This improvement stems from the module's abandonment of the computationally redundant global matching strategy in favor of adaptively defining the matching space based on detection confidence. Furthermore, it constructs a highly discriminative association cost matrix by integrating low-complexity features including Euclidean distance, relative distance, and motion direction angle, thereby significantly enhancing association accuracy while substantially reducing computational overhead. Finally, the integration of all three proposed modules into the baseline model yields the optimal overall performance. This confirms that the MFCTM module also contributes significantly to the overall tracking performance. By comprehensively leveraging multi-source information from unmatched trajectories—including 2D bounding box positions, motion trends, and observation distances—this module intelligently discriminates between temporarily occluded objects and those that have exited the scene. This capability enables precise decisions regarding trajectory retention versus termination, thereby enhancing the algorithm's tracking robustness and efficiency in complex scenarios.

According to the results in Table 3, the contribution of each module to the computational efficiency can be clearly observed. First, the introduction of the CIGTS module yields an initial improvement in Frames Per Second (FPS). This is primarily attributed to the module's ability to filter out a large number of low-quality detections at the source,

directly reducing the computational burden for subsequent data association. Furthermore, it effectively suppresses the generation of ghost trajectories, thereby avoiding unnecessary computational overhead during the association and trajectory management stages caused by these false trajectories. Second, the incorporation of the AMSLA module leads to a significant enhancement in FPS. This is mainly because it abandons the computationally redundant global association strategy, performing associations only between detections and trajectories within the same matching space, which substantially reduces the computational complexity of the association process. Additionally, this module avoids using computationally expensive similarity measures, such as appearance information, when constructing the association cost matrix, further optimizing computational efficiency. Finally, the integration of the MFCTM module achieves the highest FPS performance. This stems from its ability to promptly terminate trajectories of objects that have exited the scene. This not only reduces the memory and computational overhead associated with maintaining a large number of active trajectories but also avoids meaningless association attempts with these trajectories in subsequent frames, thereby freeing up computational resources.

During the preprocessing of detection results, the confidence filtering threshold θ_c plays a critical role in distinguishing false positives from true objects. To determine the optimal value for θ_c, we conduct an ablation study, the results of which are summarized in Table 4.

Table 4. Ablation study on the confidence filtering threshold.

θ_c	HOTA↑	MOTA↑	MOTP↑	FP↓	FN↓	IDSW↓
1	78.295	87.532	87.173	1047	2693	81
1.2	78.466	88.018	87.289	932	2668	72
1.4	78.603	88.106	87.364	910	2642	69
1.6	78.429	88.035	87.274	931	2669	72
1.8	78.237	87.486	87.036	1015	2716	85

As shown in Table 4, the model achieves optimal performance across all evaluation metrics when the confidence filtering threshold θ_c is set to 1.4. This is primarily because an excessively large threshold would incorrectly delete true objects, leading to issues such as missed detections and association failures. Conversely, an overly small threshold would retain a portion of false positives, potentially causing ghost trajectories and identity switches. Furthermore, the value of the confidence stratified threshold θ_l determines the activation period distribution of virtual trajectories. To enable the model to achieve its optimal state, we also conduct an ablation study on θ_l, with the results shown in Table 5.

Table 5. Ablation study on the confidence stratified threshold.

θ_l	HOTA↑	MOTA↑	MOTP↑	FP↓	FN↓	IDSW↓
2.5	78.233	87.919	87.108	939	2677	76
3	78.425	87.998	87.215	924	2654	74
3.5	78.603	88.106	87.364	910	2642	69
4	78.435	88.002	87.234	931	2666	73
4.5	78.229	87.794	87.134	983	2694	77

As indicated in Table 5, the model achieves optimal performance when the confidence stratified threshold θ_l is set to 3.5. To further investigate the impact of the activation thresholds on tracking performance, we conduct separate ablation studies for the activation thresholds δ and ξ. The results are presented in Tables 6 and 7.

Table 6. Ablation study on the activation threshold δ.

δ	HOTA↑	MOTA↑	MOTP↑	FP↓	FN↓	IDSW↓
1	78.207	87.736	87.160	937	2670	75
2	78.603	88.106	87.364	910	2642	69
3	78.383	87.973	87.191	931	2666	74

Table 7. Ablation study on the activation threshold ξ.

ξ	HOTA↑	MOTA↑	MOTP↑	FP↓	FN↓	IDSW↓
2	78.475	88.031	87.293	928	2673	72
3	78.603	88.106	87.364	910	2642	69
4	78.393	87.985	87.242	935	2687	74

As shown in Tables 6 and 7, the tracking performance achieves the optimum when the activation thresholds δ and ξ are set to 2 and 3, respectively. This is primarily because a smaller activation threshold increases the likelihood of false detections forming ghost trajectories, thereby leading to incorrect matches and identity switches. Conversely, a larger activation threshold increases the trajectory confirmation time, which subsequently results in association failures and trajectory fragmentation.

In the data association stage, we sequentially incorporate the global distance matrix $\mathcal{M}_g$, relative distance matrix $\mathcal{M}_r$, and motion direction matrix $\mathcal{M}_\theta$ into the proposed association module to evaluate the impact of each association matrix on tracking performance. The results are summarized in Table 8.

Table 8. Ablation study on the association matrix.

$\mathcal{M}_g$	$\mathcal{M}_r$	$\mathcal{M}_\theta$	HOTA↑	MOTA↑	MOTP↑	FP↓	FN↓	IDSW↓
√			79.589	88.496	87.800	884	2457	72
√	√		79.803	88.629	87.970	862	2403	66
√	√	√	80.125	88.970	88.155	826	2317	50

As shown in Table 8, incorporating the relative distance matrix $\mathcal{M}_r$ into the association cost matrix improves all evaluation metrics. A more substantial performance gain is observed when the motion direction matrix $\mathcal{M}_\theta$ is further added. This is primarily attributed to the fact that object motion direction can enhance the discriminative power of the relative distance criterion for distinguishing different objects, thereby improving the accuracy of data association and the robustness of tracking. Furthermore, the matching space expansion factor μ in the data association module affects the performance of the secondary data association. To explore the optimal performance of our model, we conduct an ablation study on this parameter, with the results presented in Table 9.

Table 9. Ablation study on matching space expansion factors.

μ	HOTA↑	MOTA↑	MOTP↑	FP↓	FN↓	IDSW↓
1.1	79.822	88.795	87.963	861	2355	57
1.3	79.994	88.887	88.076	845	2342	54
1.5	80.125	88.970	88.155	826	2317	50
1.7	80.031	88.912	88.097	837	2330	52
1.9	79.916	88.853	88.022	852	2342	55

As shown in Table 9, the model achieves optimal performance when the matching space expansion factor μ is set to 1.5. This is primarily because an appropriate expansion

factor can effectively handle large inter-frame displacements while simultaneously avoiding the introduction of incorrect matches.

After the object state is determined by the multi-factor collaborative perception-based intelligent trajectory management module, unmatched trajectories corresponding to objects that have left the scene are immediately purged, while those corresponding to temporarily occluded objects are retained as much as possible to address the challenge of long-term occlusion. However, as the number of retention frames increases, the computational cost of the model rises accordingly. Furthermore, the reliability of unmatched trajectories gradually diminishes, leading to an increased probability of incorrect associations. Therefore, we conduct an ablation study on the retention frame count φ for unmatched trajectories to determine its optimal value. The results are shown in Figure 6.

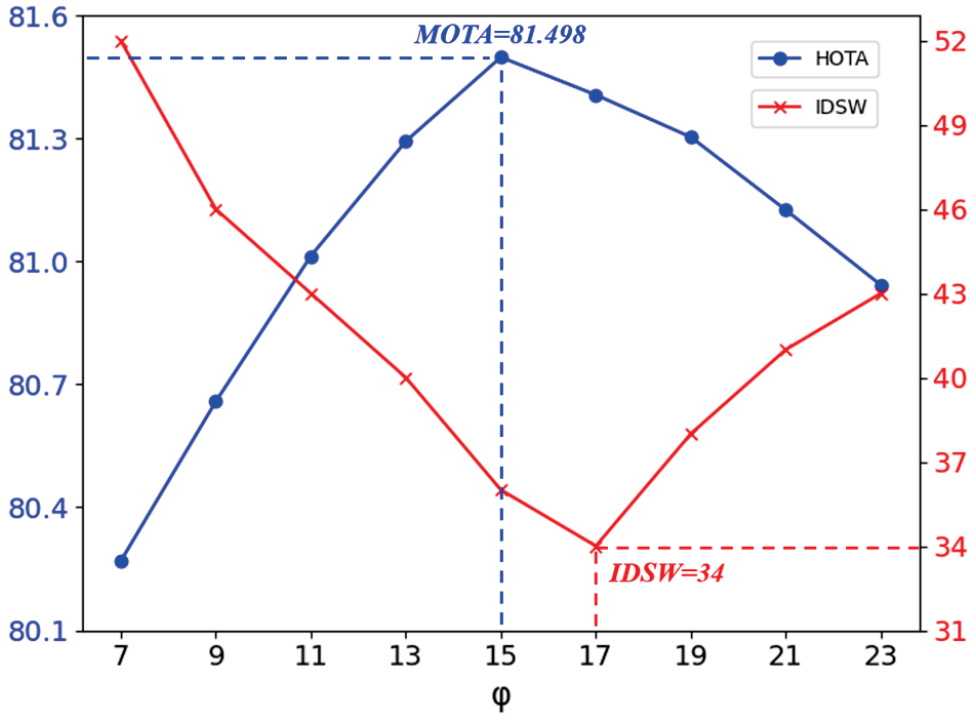


Figure 6. Results of the ablation study on trajectory retention frame count φ.

As illustrated in Figure 6, the evaluation metric HOTA achieves its optimum when the retention frame count φ for unmatched trajectories is set to 15, whereas the IDSW metric reaches its minimum at $\varphi = 17$. To balance model efficiency and performance, we ultimately set the retention frame count φ to 15. Subsequently, the value of the perception threshold λ in the object state determination strategy also influences tracking performance. Therefore, we conduct an ablation study on the perception threshold λ, with the results presented in Table 10.

Table 10. Ablation study on perception threshold.

μ	HOTA↑	MOTA↑	MOTP↑	FP↓	FN↓	IDSW↓
70	81.200	89.497	88.920	752	2268	44
75	81.375	89.560	89.045	736	2248	39
80	81.498	89.663	89.110	724	2228	36
85	81.405	89.605	89.052	735	2241	38
90	81.274	89.526	88.978	750	2259	42

As shown in Table 10, the model fails to achieve optimal performance with either excessively large or small values of the perception threshold λ. This occurs primarily

because an overly small λ tends to misclassify temporarily occluded objects as having left the scene, leading to the premature deletion of their unmatched trajectories. When these objects reappear, identity switches are likely to occur. Conversely, an excessively large λ tends to retain unmatched trajectories for objects that have actually left the scene, misclassifying them as temporarily occluded. This increases computational costs and elevates the risk of incorrect associations.

Most parameters in the proposed method maintain stable tracking performance when varied within a reasonable range, indicating low sensitivity to precise parameter values. Moreover, their dependencies are simple and tied to inherent physical constraints, making them insensitive to scene changes and highly generalizable across different environments.

4.4. Visualization Analysis

To intuitively validate the effectiveness of the proposed method in addressing core challenges such as false positives, ghost trajectories, long-term occlusions, and identity switches, we conduct a systematic visual comparison of the tracking results between our method and the baseline model on the KITTI dataset. By selecting representative complex scenarios, the tracking outcomes are presented from both 2D image (left) and 3D point cloud (right) perspectives. The specific visualization results are shown in Figures 7–9.

As shown in the left subfigures of Figure 7, the baseline model produces ghost trajectories in both frames 72 and 73 of the fifth scene. This occurs because it neglects the preprocessing of false positives, resulting in consecutive false detections that manifest as ghost trajectories. In contrast, the visualization results on the right demonstrate that the proposed method is completely free from ghost trajectories. This clearly indicates that our proposed CIGTS module effectively filters out false positives. Furthermore, the introduction of virtual trajectories optimizes the trajectory initialization strategy, collectively leading to effective suppression of ghost trajectories.

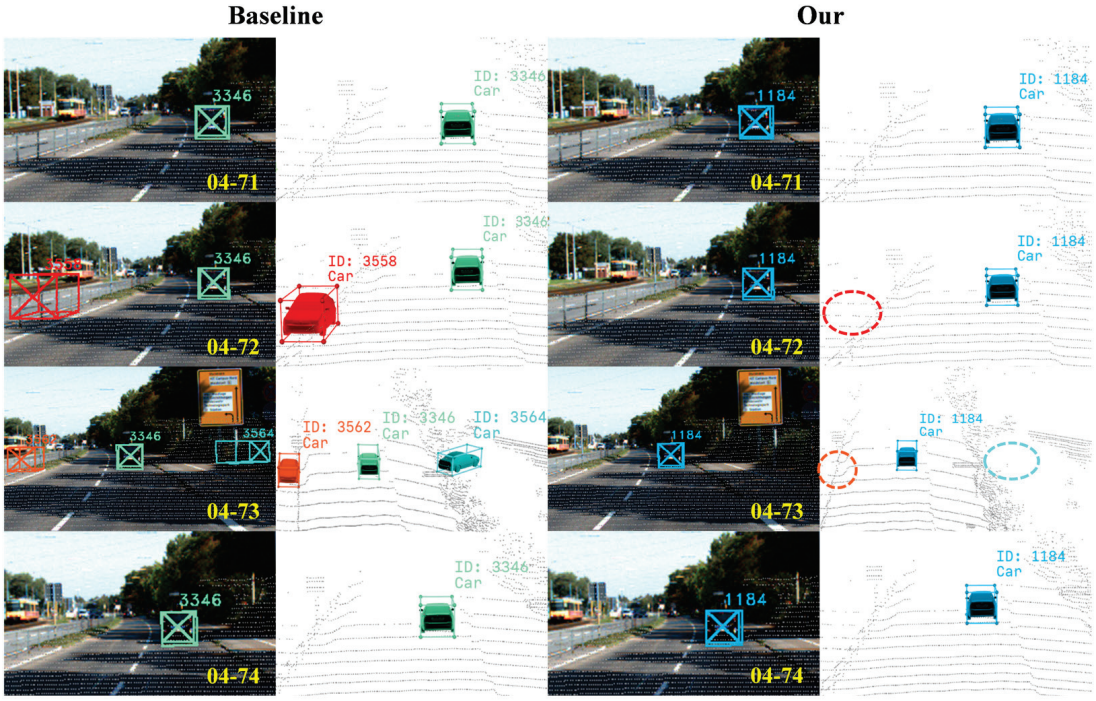


Figure 7. Visual comparison of ghost trajectories. The baseline model produces multiple ghost trajectories (IDs 3558, 3562, 3564) in frames 72 and 73, whereas our proposed method avoids such issues.

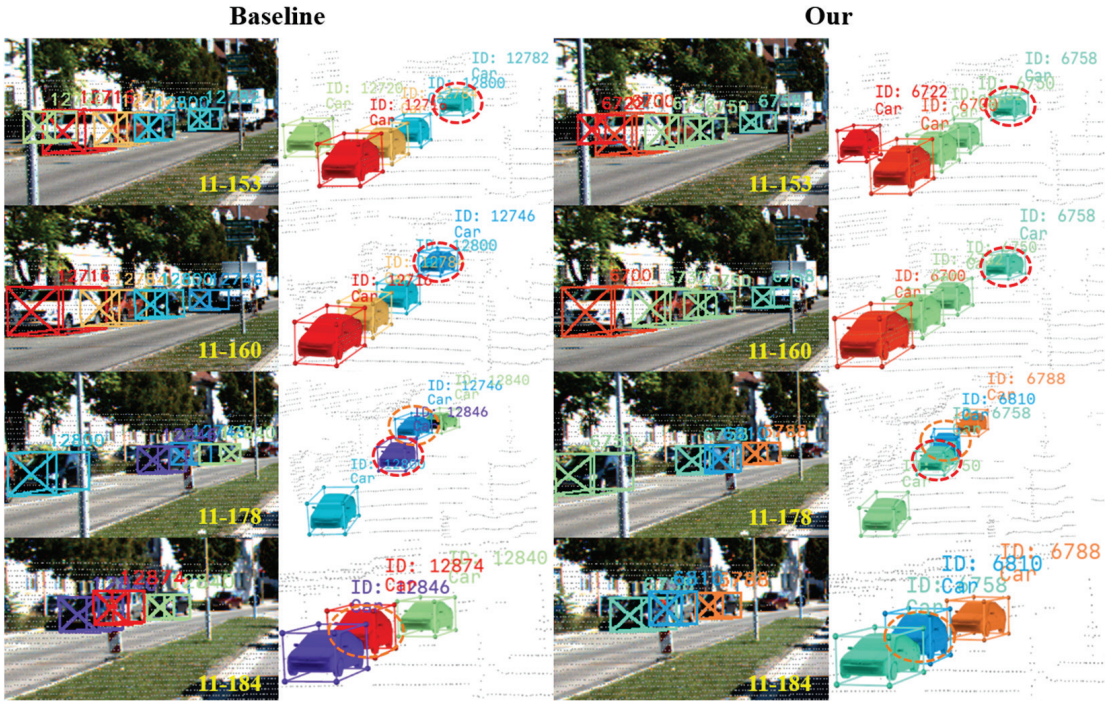


Figure 8. Visual comparison of data association. The cascade strategy used by the baseline model leads to ID switches during data association (e.g., ID 12782 changes to 12746 and then to 12874), while our method maintains identity consistency throughout the sequence.

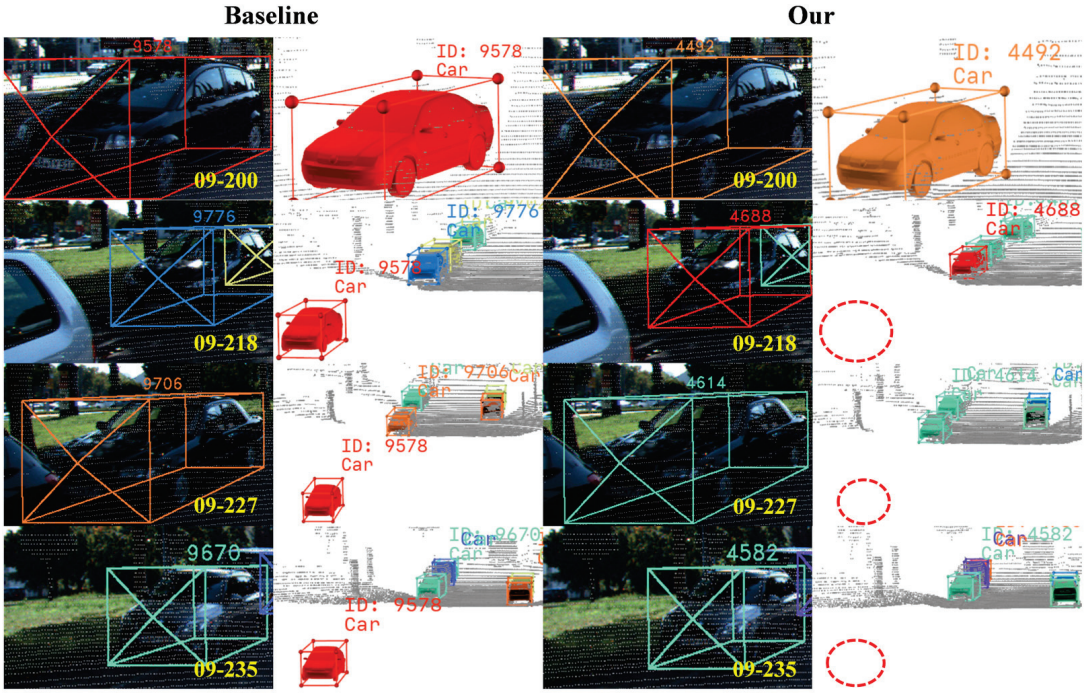


Figure 9. Visual comparison of trajectory retention. The baseline model incorrectly retains the trajectory of an object that has left the scene (ID 9578). Our method intelligently identifies and promptly terminates such trajectories.

As shown in the left visualizations of Figure 8, the object with ID 12782 in frame 153 undergoes an identity switch in frame 160, where its ID changes to 12746. Subsequently, the object with ID 12746 in frame 160 is reassigned a new ID, 12846, by frame 178. Finally, the object identified as 12746 in frame 178 experiences another identity switch in

frame 184, acquiring ID 12874. In contrast, the right visualizations demonstrate that the proposed method exhibits no identity switches. This robustness is primarily attributed to the proposed AMSLA module. This module not only reduces the risk of incorrect matching through its adaptive matching space strategy but also incorporates more stable determining factors to construct a more discriminative association matrix, thereby further enhancing the accuracy of data association.

As shown in the left visualization of Figure 9, the baseline model fails to distinguish between objects that have exited the scene and those that are temporarily occluded. It uniformly retains all unmatched trajectories for a fixed number of frames, leading to the excessive retention of an exited object (ID 9578) and causing duplicate matching in subsequent frames. In contrast, the right visualization demonstrates that the proposed method effectively avoids duplicate matching for exited objects. This capability primarily stems from the MFCTM module, which comprehensively evaluates object states through scene boundary awareness, motion trend analysis, and observation distance assessment. This integrated approach enables timely deletion of unmatched trajectories corresponding to exited objects while prioritizing the retention of those associated with temporarily occluded objects.

We also conduct a visual analysis of the pedestrian tracking results on the KITTI dataset to further validate the comprehensive performance of the proposed method on the pedestrian category. We selected typical complex scenarios containing multiple objects with severe occlusion, and performed a visual comparison between the tracking results of our method and the official KITTI ground truth. This comparison objectively demonstrates the effectiveness and stability of our method in pedestrian tracking tasks. As depicted in Figure 10, the left side shows the visualization of the ground truth annotations, while the right side presents the visual output of the tracking results generated by our proposed method.

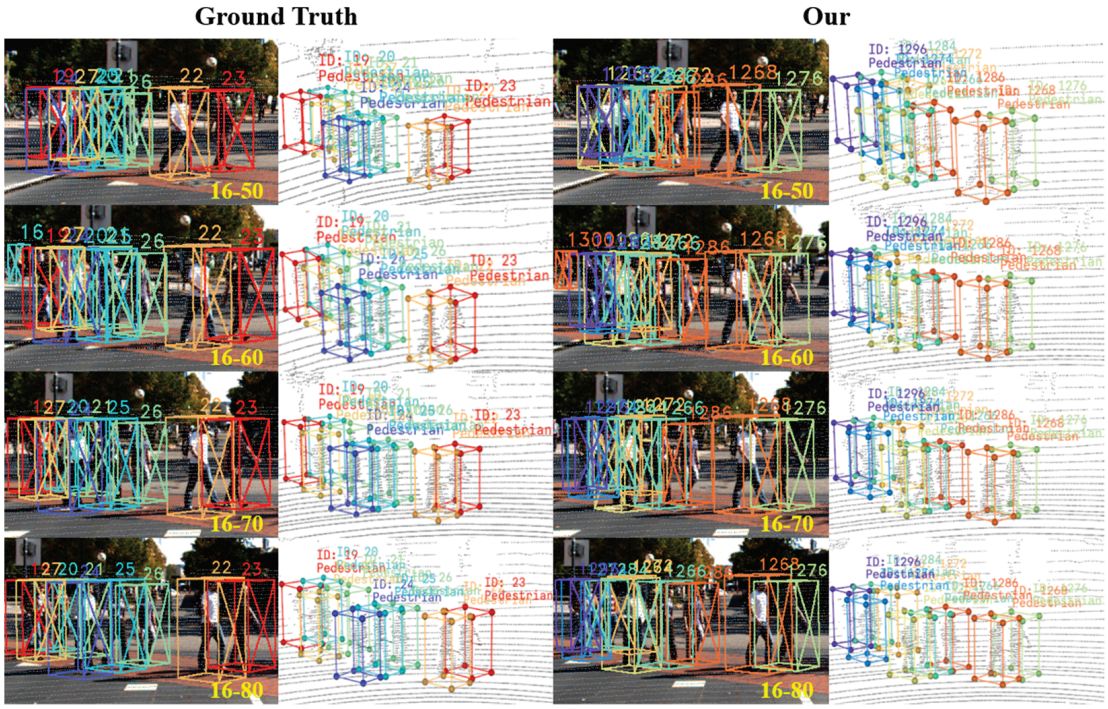


Figure 10. Visual comparison of pedestrian tracking results with ground truth. In dense scenarios, the comparison with ground truth across consecutive frames shows that our method does not produce any ID switches, effectively validating its tracking stability under challenging conditions.

In Figure 10, we select 40 consecutive frames of tracking results from high-complexity scenes for visualization. A horizontal analysis reveals that the proposed method produces no ghost trajectories, thereby validating the effectiveness of the CIGTS module. Furthermore, a longitudinal analysis across the 40-frame sequence demonstrates the complete absence of identity switches or incorrect associations, achieving continuous and accurate tracking of multiple objects. This confirms the efficacy of the AMSLA module. From a holistic perspective, the visualization results substantiate that the proposed 3D MOT method effectively addresses critical challenges including false positives, ghost trajectories, long-term occlusions, and identity switches, thereby enabling more stable and robust object tracking.

5. Conclusions

This paper proposes a lightweight 3D multi-object tracking method via collaborative camera and LiDAR sensors. It aims to address challenges including false detections and subsequent ghost trajectories caused by varying illumination conditions and dynamic backgrounds, poor tracking real-time performance due to inefficient data association, and difficulties in distinguishing object states and identity switches resulting from occlusions. By introducing an inverse normalization confidence enhancement mechanism and a hierarchical virtual trajectory management strategy, our method significantly improves the ability to distinguish true objects from false detections. Furthermore, we design a lightweight association module featuring adaptive matching space and multi-discriminant factor fusion, which reduces computational overhead while enhancing matching accuracy. Additionally, a multi-factor collaborative decision-making mechanism for trajectory management is constructed to achieve precise discrimination between occluded objects and those exiting the scene. Experimental results on the KITTI dataset demonstrate that our method achieves excellent performance in tracking multiple object categories, including cars and pedestrians, and outperforms existing state-of-the-art methods, particularly on higher-order metrics such as HOTA, AssA, and IDSW.

Nevertheless, the proposed method still has certain limitations. For instance, the filtered low-quality detections might exhibit positional inaccuracies due to factors such as minor jitter. To maintain a lightweight design and align with mainstream research practices, we do not perform positional regression correction on these detections. However, in extreme cases, where a low-quality detection initially exhibits a significant positional error and remains persistently occluded in subsequent frames without updates, the accumulation of errors may eventually affect the accuracy of data association. Future work will focus on optimizing low-quality detections and enhancing association robustness. Specifically, we plan to perform position regression on low-quality detections to obtain more precise object location information, and to design an association strategy based on the possible states of objects in real-world scenarios, thereby further enhancing the robustness and practicality of the proposed method under complex environments.

Author Contributions: Conceptualization, D.F. and H.L.; methodology, D.F. and H.L.; software, D.F. and Z.L.; validation, D.F. and H.L.; investigation, Z.L.; data curation, Z.L.; writing—original draft preparation, D.F. and H.L.; writing—review and editing, D.F. and H.L.; visualization, D.F. and H.L.; supervision, H.L. and Z.L.; project administration, D.F. and Z.L.; funding acquisition, D.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Research and Development Program of Shandong, grant number 2024CXGC010110.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in the multi-object tracking task of the KITTI project. These data were derived from the following resources available in the public domain: http://www.cvlibs.net/datasets/kitti/eval_tracking.php (accessed on 26 March 2025).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Pang, Z.; Li, Z.; Wang, N. SimpleTrack: Understanding and Rethinking 3D Multi-Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 680–696.
2. Wang, X.; Fu, C.; Li, Z.; Lai, Y.; He, J. DeepFusionMOT: A 3D Multi-Object Tracking Framework Based on Camera-LiDAR Fusion with Deep Association. *IEEE Robot. Autom. Lett.* **2022**, *7*, 8260–8267. [CrossRef]
3. Ding, S.; Schneider, L.; Cordts, M.; Gall, J. ADA-Track: End-to-End Multi-Camera 3D Multi-Object Tracking with Alternating Detection and Association. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 15184–15194.
4. Wu, H.; Han, W.; Wen, C.; Li, X.; Wang, C. 3D Multi-Object Tracking in Point Clouds Based on Prediction Confidence-Guided Data Association. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 5668–5677. [CrossRef]
5. Zhu, Z.; Nie, J.; Wu, H.; He, Z.; Gao, M. MSA-MOT: Multi-Stage Association for 3D Multimodality Multi-Object Tracking. *Sensors* **2022**, *22*, 8650. [CrossRef] [PubMed]
6. Wang, L.; Zhang, X.; Qin, W.; Li, X.; Gao, J.; Yang, L.; Li, Z.; Li, J.; Zhu, L.; Wang, H.; et al. CAMO-MOT: Combined Appearance-Motion Optimization for 3D Multi-Object Tracking with Camera-LiDAR Fusion. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 11981–11996. [CrossRef]
7. Wang, X.; Fu, C.; He, J.; Wang, S.; Wang, J. StrongFusionMOT: A Multi-Object Tracking Method Based on LiDAR-Camera Fusion. *IEEE Sens. J.* **2023**, *23*, 11241–11252. [CrossRef]
8. Kim, H.; Lee, H.J.; Lee, Y.; Lee, J.; Kim, H.; Koh, Y.J. GRAE-3DMOT: Geometry Relation-Aware Encoder for Online 3D Multi-Object Tracking. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 11–15 June 2025; pp. 11697–11706.
9. Li, H.; Liu, H.; Du, Z.; Chen, Z.; Tao, Y. MCCA-MOT: Multimodal Collaboration-Guided Cascade Association Network for 3D Multi-Object Tracking. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 974–989. [CrossRef]
10. Li, P.; Jin, J. Time3D: End-to-End Joint Monocular 3D Object Detection and Tracking for Autonomous Driving. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 3885–3894.
11. Kini, J.; Mian, A.; Shah, M. 3DMODT: Attention-Guided Affinities for Joint Detection & Tracking in 3D Point Clouds. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 841–848.
12. Ding, S.; Rehder, E.; Schneider, L.; Cordts, M.; Gall, J. 3DMOTFormer: Graph Transformer for Online 3D Multi-Object Tracking. In Proceedings of the 2023 IEEE International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 9784–9794.
13. Zhang, C.; Zhang, C.; Guo, Y.; Chen, L.; Happold, M. MotionTrack: End-to-End Transformer-Based Multi-Object Tracking with LiDAR-Camera Fusion. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 151–160.
14. Kim, A.; Ošep, A.; Leal-Taixé, L. EagerMOT: 3D Multi-Object Tracking Via Sensor Fusion. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 11315–11321.
15. Wang, X.; Fu, C.; He, J.; Huang, M.; Meng, T.; Zhang, S.; Zhang, C. You Only Need Two Detectors to Achieve Multi-Modal 3D Multi-Object Tracking. *arXiv* **2023**, arXiv:2304.08709.
16. Li, X.; Liu, D.; Wu, Y.; Wu, X.; Zhao, L.; Gao, J. Fast-Poly: A Fast Polyhedral Algorithm for 3D Multi-Object Tracking. *IEEE Robot. Autom. Lett.* **2024**, *9*, 10519–10526. [CrossRef]
17. Papais, S.; Ren, R.; Waslander, S. SWTrack: Multiple Hypothesis Sliding Window 3D Multi-Object Tracking. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 4939–4945.

18. Guo, G.; Zhao, S. 3D Multi-Object Tracking with Adaptive Cubature Kalman Filter for Autonomous Driving. *IEEE Trans. Intell. Veh.* **2023**, *8*, 512–519. [CrossRef]
19. Zhang, K.; Liu, Y.; Mei, F.; Jin, J.; Wang, Y. Boost Correlation Features with 3D-MiIoU-Based Camera-LiDAR Fusion for MODT in Autonomous Driving. *Remote Sens.* **2023**, *15*, 874. [CrossRef]
20. Pieroni, R.; Specchia, S.; Corno, M.; Savaresi, S.M. Multi-Object Tracking with Camera-LiDAR Fusion for Autonomous Driving. In Proceedings of the 2024 European Control Conference (ECC), Stockholm, Sweden, 25–28 June 2024; pp. 2774–2779.
21. Nagy, M.; Khonji, M.; Dias, J.; Javed, S. DFR-FastMOT: Detection Failure Resistant Tracker for Fast Multi-Object Tracking Based on Sensor Fusion. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 827–833.
22. Jin, J.; Zhang, J.; Zhang, K.; Wang, Y.; Ma, Y.; Pan, D. 3D Multi-Object Tracking with Boosting Data Association and Improved Trajectory Management Mechanism. *Signal Process.* **2024**, *218*, 12. [CrossRef]
23. Sadjadpour, T.; Li, J.; Ambrus, R.; Bohg, J. ShaSTA: Modeling Shape and Spatio-Temporal Affinities for 3D Multi-Object Tracking. *IEEE Robot. Autom. Lett.* **2024**, *9*, 4273–4280. [CrossRef]
24. Doll, S.; Hanselmann, N.; Schneider, L.; Schulz, R.; Enzweiler, M.; Lensch, H.P.A. S.T.A.R.-Track: Latent Motion Models for End-to-End 3D Object Tracking with Adaptive Spatio-Temporal Appearance Representations. *IEEE Robot. Autom. Lett.* **2024**, *9*, 1326–1333. [CrossRef]
25. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
26. Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P.; Geiger, A.; Leal-Taixé, L.; Leibe, B. HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 548–578. [CrossRef]
27. Bernardin, K.; Stiefelhagen, R. Evaluating Multiple Object Tracking Performance: The Clear MOT Metrics. *Eurasip. J. Image Video Process.* **2008**, *2008*, 246309. [CrossRef]
28. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 770–779.
29. Ren, J.; Chen, X.; Liu, J.; Sun, W.; Pang, J.; Yan, Q.; Xu, L. Accurate Single Stage Detector Using Recurrent Rolling Convolution. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5420–5428.
30. Mendonça, D.; Barros, T.; Premebida, C.; Nunes, U.J. Seg2Track-SAM2: SAM2-based Multi-object Tracking and Segmentation for Zero-shot Generalization. *arXiv* **2025**, arXiv:2509.11772.
31. Claasen, P.J.; de Villiers, J.P. One Homography is All You Need: IMM-based Joint Homography and Multiple Object State Estimation. *arXiv* **2024**, arXiv:2409.02562.
32. Jiang, C.; Wang, Z.; Liang, H.; Wang, Y. A Novel Adaptive Noise Covariance Matrix Estimation and Filtering Method: Application to Multiobject Tracking. *IEEE Trans. Intell. Veh.* **2024**, *9*, 626–641. [CrossRef]
33. Im, E.; Jee, C.; Lee, J.K. Spb3DTracker: A Robust LiDAR-Based Person Tracker for Noisy Environment. *arXiv* **2024**, arXiv:2408.05940.
34. He, J.; Fu, C.; Wang, X.; Wang, J. 3D Multi-Object Tracking Based on Informatic Divergence-Guided Data Association. *Signal Process.* **2024**, *222*, 10. [CrossRef]
35. Wang, X.; Fu, C.; He, J.; Huang, M.; Meng, T.; Zhang, S.; Zhou, H.; Xu, Z.; Zhang, C. A Multi-Modal Fusion-Based 3D Multi-Object Tracking Framework with Joint Detection. *IEEE Robot. Autom. Lett.* **2025**, *10*, 532–539. [CrossRef]
36. Ninh, P.P.; Kim, H. CollabMOT Stereo Camera Collaborative Multi Object Tracking. *IEEE Access* **2024**, *12*, 21304–21319. [CrossRef]
37. Yan, Z.; Feng, S.; Li, X.; Zhou, Y.; Xia, C.; Li, S. S3MOT: Monocular 3D Object Tracking with Selective State Space Model. *arXiv* **2025**, arXiv:2504.18068. [CrossRef]
38. Miah, M.; Bilodeau, G.A.; Saunier, N. Learning Data Association for Multi-Object Tracking Using Only Coordinates. *Pattern Recognit.* **2025**, *160*, 11. [CrossRef]
39. Gong, Y.; Chen, M.; Liu, H.; Gao, Y.; Yang, L.; Wang, N.; Song, Z.; Ma, H. Stable at Any Speed: Speed-Driven Multi-Object Tracking with Learnable Kalman Filtering. *arXiv* **2025**, arXiv:2508.00358.
40. Tokmakov, P.; Jabri, A.; Li, J.; Gaidon, A. Object Permanence Emerges in a Random Walk Along Memory. *arXiv* **2022**, arXiv:2204.01784.
41. Liao, P.; Yang, F.; Wu, D.; Liu, B.; Zhang, X. MO-YOLO: End-to-End Multiple-Object Tracking Method with YOLO and Decoder. *arXiv* **2023**, arXiv:2310.17170.
42. Zhou, T.; Ye, Q.; Luo, W.; Ran, H.; Shi, Z.; Chen, J. APPTracker+: Displacement Uncertainty for Occlusion Handling in Low-Frame-Rate Multiple Object Tracking. *Int. J. Comput. Vis.* **2025**, *133*, 2044–2069. [CrossRef]

43. Jiang, C.; Wang, Z.; Liang, H.; Tan, S. A Fast and High-Performance Object Proposal Method for Vision Sensors: Application to Object Detection. *IEEE Sens. J.* **2022**, *22*, 9543–9557. [CrossRef]
44. Xu, L.; Huang, Y. Rethinking Joint Detection and Embedding for Multiobject Tracking in Multiscenario. *IEEE Trans. Ind. Inform.* **2024**, *20*, 8079–8088. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

MemLoTrack: Enhancing TIR Anti-UAV Tracking with Memory-Integrated Low-Rank Adaptation

Jae Kwan Park [1] and Ji-Hyeong Han [2,*]

[1] Department of Defense AI Convergence Engineering, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea; 16102241@seoultech.ac.kr
[2] Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea
* Correspondence: jhhan@seoultech.ac.kr

Abstract

Tracking small, fast-moving unmanned aerial vehicles (UAVs) in thermal infrared (TIR) imagery is a significant challenge due to low-resolution targets, Dynamic Background Clutter, and frequent occlusions. To address this, we introduce MemLoTrack, a novel one-stream Vision Transformer tracker that integrates a memory mechanism into a parameter-efficient LoRA framework. MemLoTrack enhances a baseline tracker (LoRAT) with two key components: (i) a gated First-In, First-Out (FIFO) memory bank (MB) for temporal context aggregation and (ii) a lightweight Memory Attention Layer (MAL) for effective information retrieval. A key component of our method is a selective memory update policy, which commits a frame to the memory bank only when it satisfies both a classification confidence threshold (τ) and a Kalman filter-based motion consistency check. This gating mechanism robustly prevents memory contamination due to distractors, occlusions, and reappearance events. Our training is highly efficient, updating only the LoRA adapters, MAL, and prediction head while the pretrained DINOv2 backbone remains frozen. Evaluated on the challenging Anti-UAV410 benchmark, MemLoTrack ($L_{\text{mem}} = 7, \tau = 0.8$) achieves an AUC of 63.6 and a State Accuracy (SA) of 64.0, representing a significant improvement over the LoRAT baseline by +1.4 AUC and +1.5 SA. Compared to the state-of-the-art method FocusTrack, MemLoTrack demonstrates superior robustness with higher AUC (63.6 vs. 62.8) and SA (64.0 vs. 63.9), while trading lower precision (P/P-Norm) scores. Furthermore, MemLoTrack operates at 153 FPS on a single RTX 4070 Ti SUPER, demonstrating that parameter-efficient fine-tuning with a selective memory mechanism is a powerful and deployable strategy for real-time Anti-UAV tracking in demanding TIR environments.

Keywords: anti-UAV; thermal infrared (TIR); single-object tracking (SOT); low-rank adaptation (LoRA); memory attention layer (MAL); gated memory bank (MB); vision transformer (ViT); DINOv2

1. Introduction

Thermal infrared (TIR) single-object tracking (SOT) plays a crucial role in Anti-UAV systems, ensuring continuous perimeter surveillance and airspace security, especially under low illumination and adverse weather conditions [1,2]. Despite remarkable advances in recent years, TIR tracking remains highly challenging due to the low resolution and texture deficiency of aerial targets, frequent Thermal Crossover events, and abrupt viewpoint variations arising from platform motion [2,3]. In such complex conditions, local search-based

trackers often fail when the target undergoes occlusion or abrupt motion, whereas global re-detection strategies, though more resilient, incur a substantial computational cost [4–6]. The Anti-UAV410 benchmark explicitly exposes these long-term robustness limitations through its challenging TIR sequences and attributes, including Thermal Crossover (TC), Dynamic Background Clutter (DBC), and occlusion (OC) [2].

Transformer-based one-stream trackers, which perform joint encoding of the template (z) and search region (x), have emerged as a dominant SOT paradigm by enabling early feature interaction [6–9]. Concurrently, parameter-efficient fine-tuning (PEFT) methods, notably Low-Rank Adaptation (LoRA), allow large Vision Transformer (ViT) backbones to be adapted with minimal computational cost by freezing most weights [10,11]. This efficiency is highly desirable for resource-constrained Anti-UAV systems. For instance, LoRAT [12] applies LoRA modules to a ViT backbone, demonstrating a path toward balancing tracking accuracy with computational throughput.

However, existing local trackers, including LoRAT, exhibit two major failure modes when evaluated on the challenging Anti-UAV410 dataset. First, abrupt camera motion can easily shift the small UAV target completely outside the predefined search region. Second, and more critically, the absence of a robust temporal memory mechanism causes identity drift during occlusion, out-of-view situations, and severe background clutter [3,13]. Although several approaches attempt to enhance robustness through adaptive or focused sampling, the fundamental issue of preserving long-term temporal coherence remains largely unresolved.

To overcome these limitations, we propose MemLoTrack, a LoRA-based, one-stream ViT tracker that incorporates an explicit, dual-gated memory bank (MB). Our approach freezes a high-capacity DINOv2 ViT encoder and trains only the low-rank adapters, a lightweight Memory Attention Layer (MAL) and an MLP-based prediction head [14]. The core of our innovation lies in the memory update policy: the feature tokens of a given frame are written to the MB only if two conditions are met: (i) the classification confidence exceeds a threshold τ, and (ii) a Kalman filter-based motion consistency test is passed. This dual-gating mechanism effectively prevents memory pollution from distractors and ambiguous observations. At inference, the MAL cross-attends the current search region tokens to the curated memory tokens, thereby mitigating drift with minimal computational overhead. Our design is informed by prior work showing that gated, attention-based memory policies significantly outperform naive fixed-window buffers in challenging tracking scenarios [13].

The main contributions of this work are summarized as follows:

- **Memory-Integrated PEFT Tracker:** We introduce a parameter-efficient ViT architecture featuring a Memory Attention Layer (MAL) that leverages long-range temporal context while preserving the efficiency of LoRA-based fine-tuning.
- **Dual-Gated Memory Write Policy:** During inference, a compact FIFO memory bank is governed by a novel dual-gated policy that admits frames only upon passing both a confidence check and a Kalman-based motion consistency check. This strategy robustly mitigates memory contamination under occlusion (OC), out-of-view (OV), and Dynamic Background Clutter (DBC) situations.
- **State-of-the-Art Robustness on Anti-UAV410:** MemLoTrack achieves state-of-the-art (SOTA) performance on the Anti-UAV410 benchmark, attaining an AUC of 63.6 and an SA of 64.0. These metrics, which reflect overall success and re-acquisition capabilities, establish a new benchmark for tracking robustness. While FocusTrack [3] demonstrates superior precision-based metrics (P/P-Norm), our method presents a notable advancement in robustness. This advancement is primarily attributed to our dual-gated memory policy, which effectively curates the temporal context, prevents memory contamination, and significantly enhances target re-acquisition during critical

failure modes such as occlusion (OC), out-of-view (OV), and Dynamic Background Clutter (DBC).

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 details the proposed architecture. Section 4 presents the experimental setup, ablation studies, and results.

2. Related Work

We review three categories of prior work most pertinent to MemLoTrack: (i) transformer based single-object tracking (SOT) with parameter-efficient adaptation; (ii) Anti-UAV tracking in thermal infrared (TIR), including benchmarks and protocol details; and (iii) memory-augmented tracking and motion-aware memory selection that motivate our memory bank (MB) and Memory Attention Layer (MAL) design.

2.1. Transformer-Based SOT and Parameter-Efficient Adaptation

One-stream ViT trackers and LoRA-based fine-tuning. Transformer backbones have become standard in modern SOT because self-attention enables long-range dependency modeling and early fusion of template–search tokens. One-stream trackers concatenate template and search before multi-head self-attention, improving interaction compared with two-stream variants [3,6,8,15,16]. LoRAT [12] operationalizes low-rank adaptation (LoRA) for this one-stream paradigm: it freezes the ViT backbone [14], inserts LoRA into attention blocks, introduces token-type embeddings to disambiguate template vs. search tokens, and replaces the convolutional head with a lightweight MLP head to avoid inductive-bias conflicts under PEFT [10,12,17–19]. LoRAT demonstrates competitive accuracy–efficiency trade-offs on several SOT benchmarks, including LaSOT [20], LaSOT-Ext/TrackingNet [21], GOT-10k [22], and TNL2k [23]. Notably, LoRAT-B-224 achieves high throughput, with certain variants reaching up to 603 fps on a NVIDIA RTX 4090, illustrating that adapter-based one-stream tracking is both practical and accurate [10,12].

2.2. Anti-UAV Tracking in TIR: Benchmarks, Protocols, and Specialized Designs

Benchmarks and evaluation protocol. Anti-UAV300 (RGB–T) [1] and Anti-UAV410 (TIR-only) [2] are the benchmarks for vision-based Anti-UAV tracking. Anti-UAV [1,2] defines the State Accuracy (SA) metric that integrates IoU with target-visibility flags (predict "present/absent") and also reports Success AUC and precision; Anti-UAV410 follows the same core protocol—Success AUC and precision and adopts SA for comparability. Normalized precision (P-Norm) is frequently reported by subsequent works for completeness but is not used as the official ranking metric in the Anti-UAV410 paper. Both benchmarks provide attribute-wise evaluation—Thermal Crossover (TC), Dynamic Background Clutter (DBC), Fast Motion (FM), occlusion (OC), Out-of-View (OV), and Scale Variation (SV)—and size categories, enabling fine-grained analysis of failure modes [1,2].

RGB vs. TIR characteristics in Anti-UAV tracking. While RGB trackers capitalize on detailed visual features available in the visible spectrum, they remain inherently vulnerable to environmental factors such as sudden illumination shifts and adverse weather. TIR tracking overcomes these photometric limitations but introduces unique challenges: the target UAV is often an indistinct, textureless blob, frequently merging with background thermal patterns. This inherent ambiguity, exacerbated by sensor noise and Thermal Crossover, suggests that the tracking paradigm should deviate from appearance-heavy matching. As a result, successful Anti-UAV trackers must combine per-frame appearance cues with temporally aggregated evidence and motion priors to resolve target identity under dynamic clutter and long-range localization difficulties, regardless of modality. In this work, we therefore adopt a unified one-stream ViT backbone augmented with an

explicit memory bank and a Kalman-based motion prior, and we empirically show in Section 4 that this memory-integrated temporal design yields clear robustness gains on the TIR-only Anti-UAV410 and Anti-UAV300 IR benchmarks, while offering only limited benefits on RGB Anti-UAV data, reflecting its specialization for thermal infrared Anti-UAV tracking.

Vision-based Anti-UAV benchmarks and pipelines. Beyond the scope of Anti-UAV410, vision-based Anti-UAV tracking has been extensively explored on RGB and RGB-T benchmarks [1,24]. These studies typically employ detection and tracking cascade pipelines that couple high-performance UAV detectors with multi-object trackers. A key advantage of this framework is the ability to trigger global re-detection whenever tracker confidence drops, thereby recovering lost targets. In contrast to this detection-centric approach, a parallel line of research focuses on lightweight SOT models based on Transformer backbones that operate within restricted search regions instead of performing full-frame scanning. MemLoTrack aligns with this latter paradigm. It integrates temporal memory and a Kalman prior into a single-object Transformer tracker to maintain trajectories without relying on external detectors. Consequently, our experimental design applies this identical framework to both a standard RGB Anti-UAV benchmark and the TIR-only Anti-UAV410 dataset. While the former provides comparative validation against competitive local trackers, the latter is dedicated to rigorous attribute-wise analyses and ablations to stress-test robustness against tiny, low-contrast targets and challenging occlusions.

Task characteristics and domain-tailored trackers. Anti-UAV410 emphasizes long-range imaging of tiny UAVs in natural scenes (forests, mountains, lakes), increasing background clutter and target ambiguity; the dataset explicitly highlights DBC and tiny-size prevalence, making local tracker design challenging [2]. Within local tracker lines, a recent Anti-UAV method (FocusTrack) demonstrates that self-adaptive search-region adjustment (SRA) and hierarchical attention-to-mask aggregation can substantially narrow the gap to global re-detection while preserving throughput—evidence that local tracking can be competitive on Anti-UAV410 if its field of view is adaptively controlled [3]. On the TIR-specialized side, FFTR augments a ViT-style encoder with frequency-channel attention and a template-reconstruction mechanism; by training compact adapter-like modules rather than fully fine-tuning the backbone, FFTR shows strong results on PTB-TIR [25], LSOTB-TIR [26], and Anti-UAV410, underscoring the value of domain priors and lightweight temporal updating for IR data [27].

2.3. Memory-Augmented Tracking and Motion-Aware Memory Selection

Streaming memory for video perception. External key–value memory is a long-standing mechanism to maintain identity through occlusions and appearance changes. SAM 2 adopts a streaming memory architecture that stores past frame embeddings and performs memory attention for promptable video segmentation; ablations indicate that direct memory embeddings with appropriate positional encoding strike a favorable speed–accuracy balance [28]. These insights motivate our MB–MAL design while we target SOT rather than segmentation.

Motion-aware Single-Object Tracking (SOT). Recent SOT frameworks have also started to model target motion explicitly at the sequence level. SwinTrack [9] introduces a dedicated motion token that summarizes past trajectories and injects it into the Transformer backbone, improving robustness under Fast Motion. ODTrack [29] learns dense temporal tokens over consecutive search regions so that motion patterns are embedded into the token sequence and updated online. ARTrack [16] treats tracking as an autoregressive trajectory prediction problem, tokenizing past bounding boxes and feeding them as spatio-temporal prompts to a Transformer decoder. These methods demonstrate that mixing appearance

features with motion-aware tokens is an effective way to stabilize long-term trajectories. Our MemLoTrack is complementary to this line: instead of replacing the localization head with an autoregressive decoder, we augment a strong one-stream PEFT-based tracker with a gated memory bank and a Kalman prior tailored to TIR Anti-UAV410 [2].

Motion-aware memory selection. Selective write policies are crucial: naive FIFO buffers (recent-N windows) admit low-quality frames under occlusion, Fast Motion, or viewpoint changes, which degrades long-term memory. Building on SAM 2's streaming memory, SAMURAI replaces this fixed-window FIFO with a hybrid scoring strategy that combines mask affinity, objectness, and kalman-based motion cues when deciding which frames to store—yielding higher success and precision without task-specific fine-tuning [13]. Motivated by these findings, we apply dual gating—(i) a classification confidence filter and (ii) a Kalman-based motion consistency check—before writing to the memory bank (MB), curbing error propagation from irregular updates and motion-induced drift.

Positioning of MemLoTrack. Within this landscape, MemLoTrack starts from a one-stream ViT tracker with LoRA confined to self-attention (PEFT) and adds an explicit MAL between the encoder and head. During training, the model samples seven memory frames between the template and the current search region frame to align optimization with the memory-augmented inference pipeline; at inference, a FIFO MB stores post-embedding tokens and supplies K, V to MAL, but only after dual gating. Our experiments on Anti-UAV410 follow the official success/precision/SA protocol (we additionally report P-Norm) [1,2].

3. Method

3.1. Backbone and Trainable Scope

MemLoTrack is a one-stream ViT tracker that jointly encodes the template z and the search frame x_t. The backbone is a frozen DINOv2 ViT backbone (patch size $p = 14$, embedding width $d = 768$). Parameter-efficient adaptation is achieved by inserting LoRA adapters (rank $r = 64$) into all linear projections of attention and MLP blocks. The set of trainable components consists of (1) the LoRA adapters, (2) a Memory Attention Layer (MAL) positioned between the encoder and the MLP head, (3) learned token-type embeddings (template foreground/background/search region/memory frames), and (4) a lightweight MLP-only prediction head. During training, we jointly encode (z, x_t) to obtain the search queries X_t. We then sample seven memory frames that lie between z and x_t. Each sampled frame is converted to tokens using the same patch embedding as the search, an absolute positional table on the token grid, and the memory type embedding. These memory frame tokens bypass the encoder and are stacked as keys/values for MAL, while X_t serves as queries. Neither a memory bank nor gates are used during training. The head operates on the token grid of the search region and produces a classification map and four distances per cell. Figure 1 summarizes this process.

3.2. Inputs and Tokens

Each training instance consists of a template $z \in \mathbb{R}^{3\times112\times112}$, a search frame $x_t \in \mathbb{R}^{3\times224\times224}$, and L_{mem} memory frames $\{m_i\}$ at the search resolution. Let $\text{PE}(\cdot)$ denote the frozen ViT patch embedding and $P^{(a,b)} \in \mathbb{R}^{(ab)\times d}$ an absolute 2D positional table shared across sources. We use four learnable token-type embeddings: $e_{\text{t}}, e_{\text{b}}, e_{\text{s}}, e_{\text{m}} \in \mathbb{R}^d$.

Input Token Formulation. The final input tokens are constructed by combining the patch embedding $\text{PE}(\cdot)$, the shared positional embedding $P^{(i,j)}$ for grid position (i, j), and the appropriate token-type embedding. Let $T^{(i,j)}$, $S^{(i,j)}$, and $M^{(i,j)}$ be the image patches at grid position (i, j) for the template, search, and memory inputs, respectively. The resulting tokens are formally defined as follows:

$$E_T^{(i,j)} = \mathrm{PE}(T^{(i,j)}) + P^{(i,j)} + \begin{cases} e_{\mathrm{t}}, & \text{if patch } (i,j) \text{ is target (t)} \\ e_{\mathrm{b}}, & \text{otherwise (b)} \end{cases} \tag{1}$$

$$E_S^{(i,j)} = \mathrm{PE}(S^{(i,j)}) + P^{(i,j)} + e_{\mathrm{s}} \tag{2}$$

$$E_M^{(i,j)} = \mathrm{PE}(M^{(i,j)}) + P^{(i,j)} + e_{\mathrm{m}} \tag{3}$$

These individual tokens are then flattened to form the template sequence $T_z \in \mathbb{R}^{B\times N_z\times d}$ and the search sequence $T_{x_t} \in \mathbb{R}^{B\times N_x\times d}$, where $N_z = (H_z/p)(W_z/p)$ and $N_x = (H_x/p)(W_x/p)$. The memory tokens (from Equation (3)) are processed separately and do not enter the main encoder.

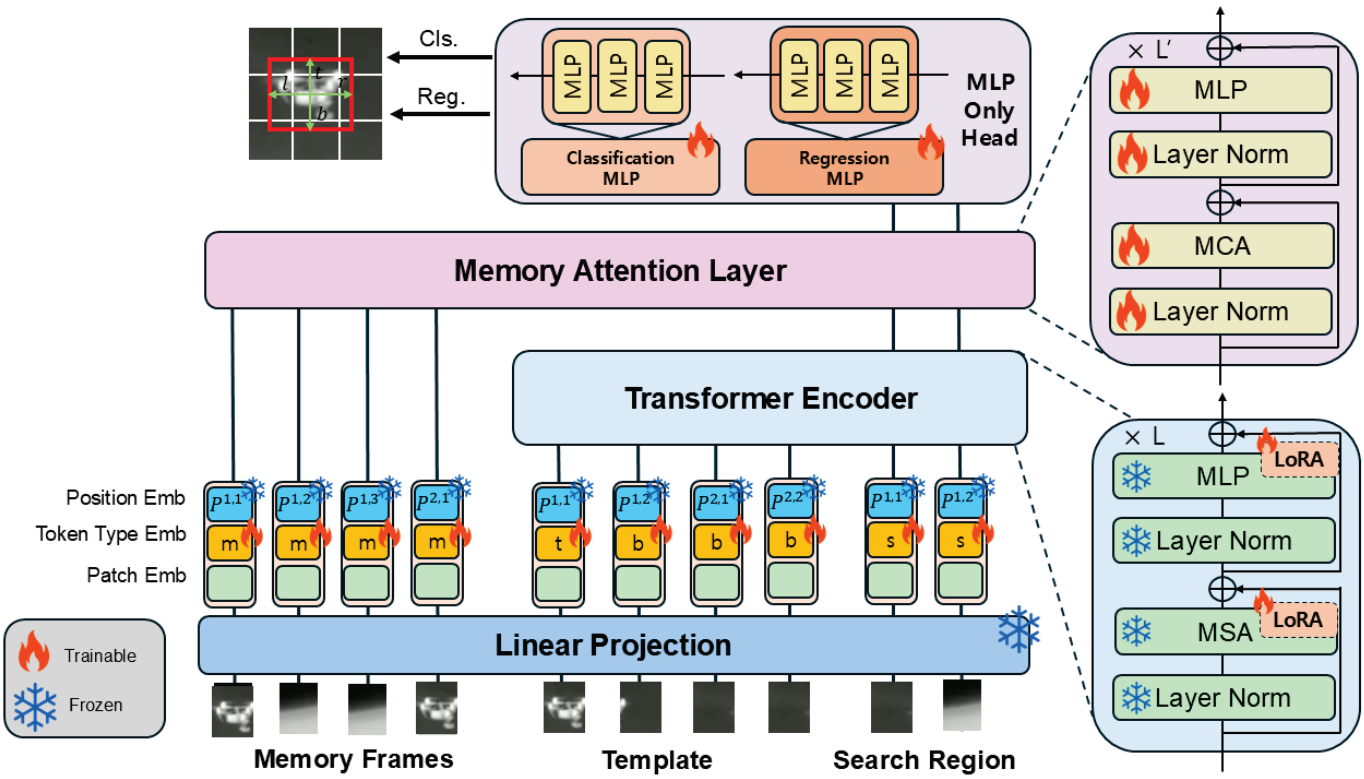


Figure 1. Training pipeline. The template z and the current search x_t are jointly encoded (frozen DINOv2 + LoRA). We sample seven memory frames from the interval between z and x_t and feed their post-embedding tokens to MAL as keys/values. No runtime memory bank is maintained during training.

Encoder output. Concatenating T_z and T_{x_t} and passing them through the encoder yields the search queries $X_t \in \mathbb{R}^{B\times N_x\times d}$ that are later conditioned by MAL.

Notation for memory tokens. To avoid confusion with the confidence threshold τ, we use the following convention throughout the paper: (i) the current frame's MB write candidate is $E_t \in \mathbb{R}^{B\times N_x\times d}$; (ii) any past entry already stored in the MB is written as $E_{t'}$. The stacked memory at time t is $M_t = \mathrm{concat}(E_{t'_1}, \ldots, E_{t'_{n_t}}) \in \mathbb{R}^{B\times N_m\times d}$ with $N_m = n_t \cdot N_x$ and $n_t \le L_{\mathrm{mem}}$.

3.3. Training-Time Memory Frame Sampling

To align the training process with our inference-time memory mechanism, we adopt a streaming-based memory sampling strategy inspired by recent work in large-scale video understanding, such as SAM 2 [28]. For each training instance, a template frame z at time t_z and a current search frame x at t_x are first sampled. Subsequently, we determine the set of memory frames based on the temporal interval between t_z and t_x, as illustrated in Figure 2.

Specifically, the sampling logic branches into two cases depending on the interval length relative to the memory budget L_{mem}: (1) If the interval is sufficient ($|t_x - t_z| > L_{\text{mem}}$), we randomly sample L_{mem} distinct frames from the intermediate period to cover diverse temporal contexts. (2) If the interval is insufficient ($|t_x - t_z| \leq L_{\text{mem}}$), we utilize all available intermediate frames and pad the remaining slots by duplicating the last valid frame (m_{pad}). Each selected frame is tokenized and directly supplied as keys and values to the MAL.

During training, these seven frames act as an ephemeral memory. They are used only within the current mini-batch and are not stored in a persistent buffer over time or across batches. We fix the number of sampled memory frames to seven so that the MAL is always exposed to the same context length as the default inference-time budget $L_{\text{mem}} = 7$. In contrast, during inference, MemLoTrack maintains a FIFO memory bank of size L_{mem} that accumulates post-embedding tokens E_t over time whenever the dual gates are satisfied. Rather than explicitly unrolling this recurrent bank during training, we follow the streaming formulation of SAM 2 [28] and train the MAL on fixed-size snippets of intermediate frames that share the same embedding format as the runtime memory tokens.

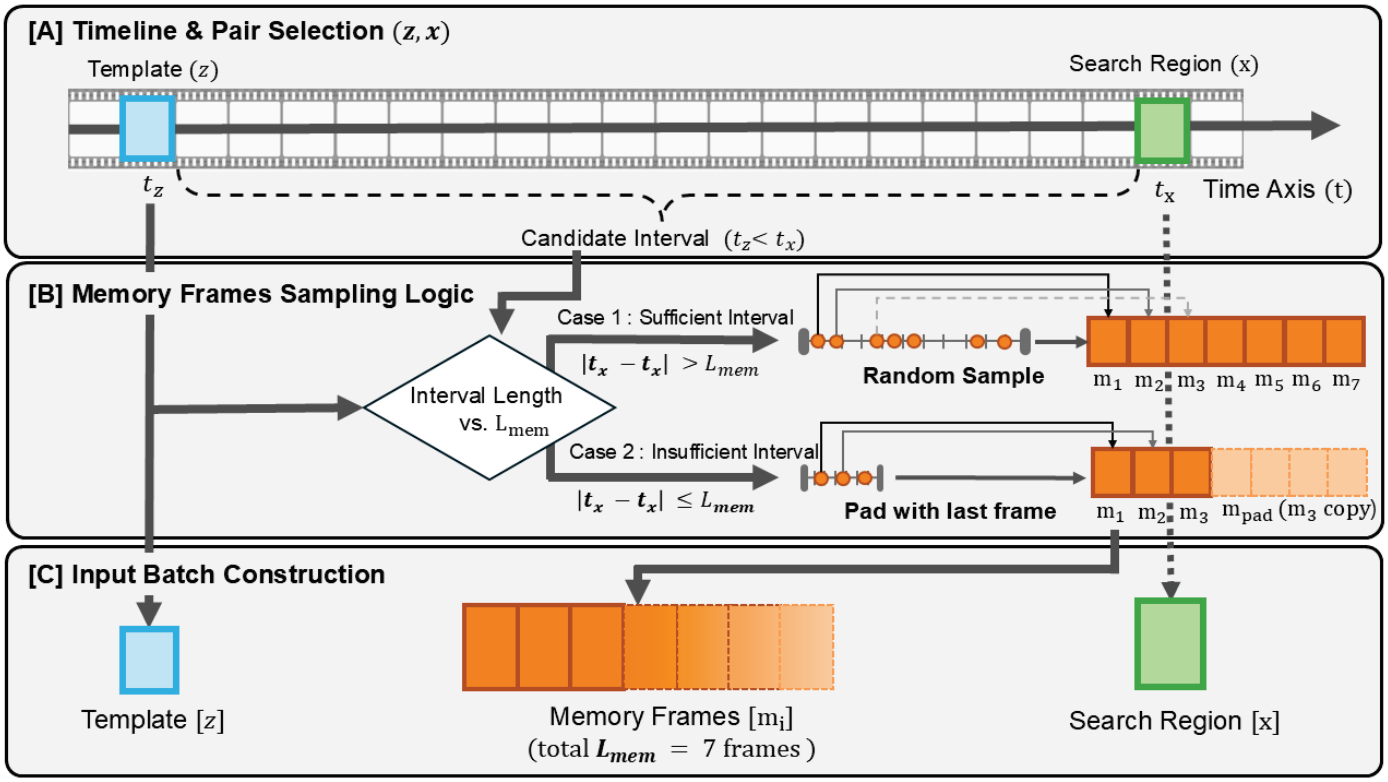


Figure 2. Memory frame sampling strategy during training. The pipeline illustrates (**A**) the selection of a template (z) and a search region (x) on the timeline, (**B**) the memory frame sampling logic that applies random sampling for sufficient intervals or pads with the last frame for insufficient ones, and (**C**) the construction of the final input batch with a fixed memory length L_{mem}.

3.4. Memory Attention Layer (MAL)

MAL performs cross-attention from the search queries X_t to the stacked memory tokens M_t and outputs a memory-conditioned representation:

$$\text{Attn}(X, M) = \text{softmax}\Big(\frac{(XW^Q)(MW^K)^\top}{\sqrt{d_h}}\Big)(MW^V)\,W^O. \tag{4}$$

If the memory stack is empty, such as at the beginning of a sequence, the MAL is bypassed and the query tokens X_t are sent directly to the MLP head.

3.5. Inference-Time Memory Bank (MB) with Dual Gating

At inference, we maintain a FIFO memory bank of capacity L_{mem}. After processing frame t, two gates are applied to decide whether to write the current entry: (i) a confidence gate that accepts if the maximum classification score s_t exceeds a threshold τ; and (ii) a motion consistency gate based on a Kalman prediction that accepts if the Mahalanobis distance between the predicted box and the decoded box is below a chi-square threshold. If a frame is accepted, its tokens E_t are first spatially aligned to the current query grid via an affine transformation and then added to the bank. This alignment ensures that memory tokens and current queries share a canonical coordinate system.

$$E_t = \text{Align}_t(\text{PE}(x_t)) + P_{16\times16} + e_{\text{mem}} \in \mathbb{R}^{B\times N_x\times d}. \tag{5}$$

If the memory bank is full, the oldest entry is evicted. When the bank is non-empty, its tokens (M_t) serve as MAL keys/values for the next frame; if it is empty, MAL is bypassed. Alignment uses the crop-to-crop affine transform (axis-aligned scaling and translation) between that frame's search window and the current one; if unavailable, we fall back to an identity warp. This ensures that memory tokens and current queries share the same canonical 16×16 grid.

Dual gating policy. Let $\mathbf{p}_t \in [0,1]^{N_x}$ denote the classification scores over the $N_x = 16 \times 16$ spatial locations at frame t, and define

$$s_t = \max_{1\le i\le N_x} \mathbf{p}_t(i).$$

The confidence gate accepts frame t if

$$s_t \ge \tau,$$

where $\tau \in (0,1)$ is an inference-time threshold selected on the Anti-UAV410 validation split.

For motion consistency, we maintain a constant-velocity Kalman filter per target with 6-D state vector:

$$\mathbf{x}_t = [c_x, c_y, v_x, v_y, \log w, \log h]^\top \in \mathbb{R}^6,$$

where (c_x, c_y) and (w, h) are the box center and size, and measurement vector

$$\mathbf{z}_t = [c_x, c_y, \log w, \log h]^\top \in \mathbb{R}^4$$

is obtained from the decoded box at frame t. Given the predicted state $(\hat{\mathbf{x}}_t, \hat{\mathbf{P}}_t)$ before seeing $\mathbf{z}_t$, the innovation and its covariance are

$$\mathbf{y}_t = \mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_t, \qquad \mathbf{\Sigma}_t = \mathbf{H}\hat{\mathbf{P}}_t\mathbf{H}^\top + \mathbf{R},$$

where $\mathbf{H} \in \mathbb{R}^{4\times6}$ selects the $(c_x, c_y, \log w, \log h)$ components of the state, and $\mathbf{R} \in \mathbb{R}^{4\times4}$ is the measurement covariance. The Mahalanobis distance between prediction and measurement is then

$$d_t^2 = \mathbf{y}_t^\top \mathbf{\Sigma}_t^{-1} \mathbf{y}_t.$$

The Kalman motion gate accepts the measurement if

$$d_t^2 < \lambda,$$

where $\lambda > 0$ is a chi-square threshold (initially set to the 95%th percentile for four degrees of freedom). In addition, we discard clearly implausible measurements whose IoU with

the predicted box is too small, whose scale ratio is outside a reasonable range, or whose center displacement exceeds a predefined velocity bound; these simple geometric checks are applied before evaluating d_t^2.

A frame is finally written to the memory bank only if it passes *both* gates, i.e., $s_t \geq \tau$ and $d_t^2 < \lambda$, together with the geometric consistency checks above. This dual gating policy prevents low-confidence or kinematically inconsistent frames from polluting the memory while keeping the mechanism lightweight for real-time deployment.

3.6. Loss and Optimization

The trainable components of our model—specifically, the LoRA adapters, the Memory Attention Layer (MAL), and the prediction head—are optimized jointly in an end-to-end manner. The total loss function $\mathcal{L}$ is formulated as a weighted sum of a binary cross-entropy (BCE) term for classification and a Generalized Intersection over Union (GIoU) term for bounding box regression:

$$\mathcal{L} = \lambda_{\text{cls}} \cdot \text{BCE}\big(\mathbf{S}_t, \mathbf{S}_t^\star\big) \; + \; \lambda_{\text{box}} \cdot \big(1 - \text{GIoU}(\widehat{\mathbf{B}}_t, \mathbf{B}_t^\star)\big). \tag{6}$$

For both training and inference, the template and search region sizes are configured to $z = 112 \times 112$ and $x = 224 \times 224$, respectively. The MB capacity is fixed at $L_{\text{mem}} = 7$ during inference to maintain consistency with the seven memory frames sampled per instance during training.

3.7. Prediction Head and Targets

The head is a lightweight MLP applied per token on the 16×16 token grid produced by the encoder (each token corresponds to a 14×14 pixel region in the 224×224 search window). For every grid cell, it outputs a classification score $\mathbf{S}_t$ and four distances (ℓ, t, r, b). Box formation converts these distances into a rectangle around the cell center (c_x, c_y):

$$x_L = c_x - \ell, \qquad x_R = c_x + r, \qquad y_T = c_y - t, \qquad y_B = c_y + b. \tag{7}$$

Targets $(\mathbf{S}_t^\star, \mathbf{B}_t^\star)$ are assigned on a center-based (anchor-free) grid.

3.8. Inference Procedure

The procedure follows Figure 3. (i) Form the search queries X_t by jointly encoding (z, x_t). (ii) If the memory bank is non-empty, apply MAL with bank tokens as K, V to obtain $\widetilde{X}_t$; otherwise bypass MAL and set $\widetilde{X}_t = X_t$. (iii) Decode the classification map and the bounding box. (iv) Apply the confidence gate using the maximum score. (v) Apply the motion consistency gate using the Kalman-based Mahalanobis test. (vi) If either gate fails, discard the frame (no write). If both pass, align to the current query grid and write the tokens E_t to the FIFO bank. (vii) Proceed to the next frame, where the bank supplies K, V to MAL.

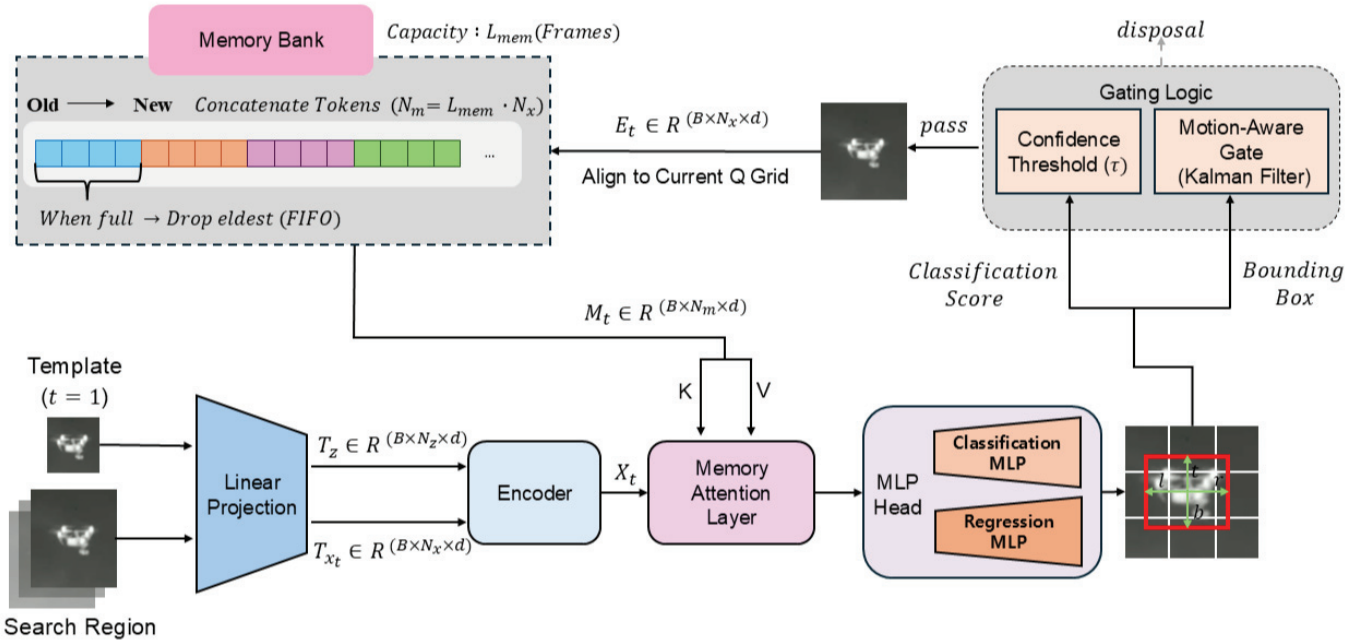


Figure 3. Inference pipeline with runtime memory bank. After head decoding, two gates are evaluated. Only if both pass, the frame is aligned to the current query grid and its post-embedding tokens E_t are written into the FIFO memory bank (capacity L_{mem}; drop oldest when full). On subsequent frames, bank tokens serve as K, V for MAL; if the bank is empty, MAL is bypassed.

4. Experiments and Results

4.1. Anti-UAV410 Protocol and Metrics

We follow the Anti-UAV410 [2] one-pass evaluation (OPE) protocol on thermal infrared sequences, which include attribute annotations for Thermal Crossover (TC), Dynamic Background Clutter (DBC), Fast Motion (FM), occlusion (OC), Out-of-View (OV), and Scale Variation (SV), as well as target-size buckets (Tiny, Small, Medium, and Normal). Performance is quantified using four standard metrics. First, Success AUC assesses overall robustness, defined as the area under the success rate–IoU curve, where the success rate represents the proportion of frames satisfying a given Intersection-over-Union (IoU) threshold. Second, precision (P) measures localization accuracy based on the 20-pixel criterion, specifically calculating the percentage of frames where the center location error relative to the ground truth is less than 20 pixels. Third, to mitigate scale sensitivity, we report Normalized Precision (P_{Norm}), which normalizes the center location error by the ground truth bounding box size prior to thresholding. Finally, we employ State Accuracy (SA) to address the target visibility fluctuations inherent in UAV tracking. Unlike conventional metrics, SA explicitly incorporates the "target present/absent" status. It evaluates IoU-based localization for visible targets while assessing the tracker's presence prediction accuracy for absent ones. This is formulated as follows:

$$SA = \frac{1}{T}\sum_{t=1}^{T}(\text{IoU}_t \cdot \delta(v_t > 0) + p_t \cdot (1 - \delta(v_t > 0))), \tag{8}$$

where T denotes the sequence length, IoU_t is the IoU with the ground truth, v_t represents the ground truth visibility flag, and p_t is the predicted presence probability. We adopt two evaluation regimes:non-training (using pre-trained weights) and re-training (fine-tuned on Anti-UAV410). Unless stated otherwise, input resolutions are $z = 112 \times 112$ and $x = 224 \times 224$. For inference, the memory bank capacity is fixed at seven frames to match

the training configuration (see Section 3.3), with a memory update confidence threshold of $\tau = 0.8$ determined via validation.

4.2. Anti-UAV300 Training and Evaluation

We extended our evaluation to the Anti-UAV300 IR benchmark [1]. All evaluated trackers were fine-tuned on the Anti-UAV300 IR train split following the protocol described in Section 4.1. As shown in Table 1, MemLoTrack consistently outperforms the strong LoRAT-B [12] (66.4 AUC, 64.8 SA). Specifically, the configuration with $MB = 11$ improves performance to 66.9 AUC and 66.6 SA. This enhancement stems from the memory-integrated adaptation pipeline, where PEFT-based fine-tuning and confidence-motion gating jointly adapt the model to dataset-specific temporal patterns. Notably, the gains are more pronounced in State Accuracy (SA) than in AUC. This indicates that the proposed method primarily bolsters temporal consistency and visibility handling while retaining the robust single-frame localization capabilities inherited from LoRAT-B [12].

Table 1. Performance comparison of trackers on the Anti-UAV300 IR test set. The metrics include Area Under Curve (AUC), precision (P), and State Accuracy (SA).

Method	AUC	P	SA
ATOM [30]	49.2	68.0	50.0
TransT [31]	51.3	68.1	52.1
Super DiMP [32]	55.8	77.2	56.7
STARK [33]	58.2	76.8	59.1
GlobalTrack [4]	63.3	85.5	64.3
LoRAT-B [12]	66.4	85.6	64.8
MemLoTrack (MB = 7)	66.6	86.3	66.4
MemLoTrack (MB = 11)	66.9	86.7	66.6

4.3. Overall Comparison on Anti-UAV410 and Cross Dataset Evaluation

Table 2 reports the Anti-UAV410 test set comparison of MemLoTrack against recent state-of-the-art trackers under both non-training (pre-trained) and re-training (fine-tuned) regimes. Under the re-training setting, MemLoTrack (MB= 7, $\tau = 0.8$, $x = 224 \times 224$) achieves AUC 63.6, P 82.7, P-Norm 79.8, and SA 64.0. Relative to FocusTrack, MemLoTrack improves AUC (62.8 $\rightarrow$ 63.6) and SA (63.9 $\rightarrow$ 64.0) while achieving lower scores on P (86.2 vs. 82.7) and P-Norm (82.8 vs. 79.8). Compared with the LoRAT-B baseline at the same resolution (AUC 62.2, P 80.9, P-Norm 78.1, SA 62.5), MemLoTrack yields consistent gains across all four metrics (+1.4 AUC, +1.8 P, +1.7 P-Norm, +1.5 SA).

For a detailed breakdown, Figures 4 and 5 present the attribute-wise Success AUC and precision (at 20 px) comparisons, respectively, for DBC, TC, SV, OC, OV, and FM. Similarly, Figures 6 and 7 summarize the Success AUC and precision results stratified by target size (Tiny, Small, Medium, Normal). As illustrated in Figures 4 and 5, MemLoTrack yields its most significant gains in Success AUC on TC and DBC sequences compared with LoRAT-B and FocusTrack. Figure 5 further reveals that our method slightly trades distance-based precision on FM, OC, and OV for improvements in overall AUC and SA, a behavior consistent with the robustness-oriented design detailed in Section 5. In the size-wise breakdown (Figures 6 and 7), MemLoTrack matches or surpasses FocusTrack across Tiny, Small, Medium, and Normal categories. This highlights that the proposed memory mechanism is particularly beneficial when tracking tiny UAVs with ambiguous thermal signatures.

Cross-modality evaluation on DUT Anti-UAV RGB tracking dataset. To assess cross-modality generalization, we evaluated the models on the DUT Anti-UAV RGB dataset [24]

using weights pre-trained on general SOT benchmarks, including LaSOT [20], GOT10k [22], TrackingNet [21], and COCO2017 [34]. As shown in Table 3, LoRAT-B [12] outperforms MemLoTrack. This result is expected given the domain shift; unlike thermal imagery characterized by low contrast and Thermal Crossover, RGB targets are texture-rich and visually distinct, enabling robust discrimination based on spatial features alone. In this context, the additional temporal memory offers limited advantage and may introduce unnecessary smoothing. These findings, juxtaposed with the significant gains on the Anti-UAV410 TIR benchmark [2], indicate that our memory-integrated pipeline is inherently specialized for thermal infrared tracking, whereas strong single-frame baselines are more suitable for texture-rich RGB domains.

Table 2. Anti-UAV410 test set comparison. In each block, the best per-column score is **bold** and the second best is underlined. Upper block: without training on Anti-UAV410; lower block: after re-training on Anti-UAV410.

Method	Size	Anti-UAV410			SA
		AUC	P	P_{norm}	
Without training on Anti-UAV410					
ETTrack [35]	256	41.5	59.7	54.8	41.6
GRM [15]	256	42.3	58.5	55.1	42.2
MixFormerV2-S [7]	224	45.6	64.1	60.0	46.1
ARTrack [16]	256	48.2	67.2	62.9	48.5
TransT [31]	256	48.2	67.7	64.1	48.9
JointNLT [36]	320	48.4	69.0	64.5	48.9
PromptVT [37]	320	50.5	71.5	65.6	51.2
SeqTrack [8]	256	52.2	73.8	70.0	52.9
LoRAT-B [12]	224	**57.2**	77.1	**74**	**57.1**
MemLoTrack (Ours)	224	57.1	**77.2**	73.8	56.6
After re-training on Anti-UAV410					
TCTrack [38]	287	41.1	60.4	56.0	41.6
SwinTrack-Tiny [9]	224	53.0	71.4	68.1	53.1
OSTrack [6]	256	53.7	73.9	70.9	54.7
ToMP50 [32]	288	54.1	73.8	70.2	55.1
ToMP101 [32]	288	54.2	75.0	70.5	55.1
ROMTrack [39]	256	54.7	74.5	71.7	55.7
SwinTrack-Base [9]	384	55.9	76.4	72.3	55.7
Stark-ST101 [33]	320	56.2	78.5	74.6	57.1
ZoomTrack [40]	256	58.4	81.2	77.4	59.4
AiATrack [41]	320	58.6	82.3	78.0	59.6
MixFormerV2-B [7]	288	58.7	80.5	76.8	59.6
DropTrack [42]	256	59.2	82.2	78.2	60.2
LoRAT-B [12]	224	62.2	80.9	78.1	62.5
FocusTrack [3]	256	62.8	**86.2**	**82.8**	63.9
MemLoTrack (Ours)	224	**63.6**	82.7	79.8	**64**

Table 3. Comparison with trackers on the DUT Anti-UAV [24] RGB tracking dataset.

Method	Size	AUC	P	P_{Norm}
SiamFC [43]	255	38.1	62.3	52.6
ATOM [30]	288	57.4	83.0	75.8
DiMP [44]	288	57.8	83.1	75.6
TransT [31]	256	58.6	83.2	76.5
LoRAT-B [12]	224	64.1	83.1	87.7
MemLoTrack (MB = 7)	224	60.8	78.6	83.4
MemLoTrack (MB = 11)	224	63	81.5	86.2

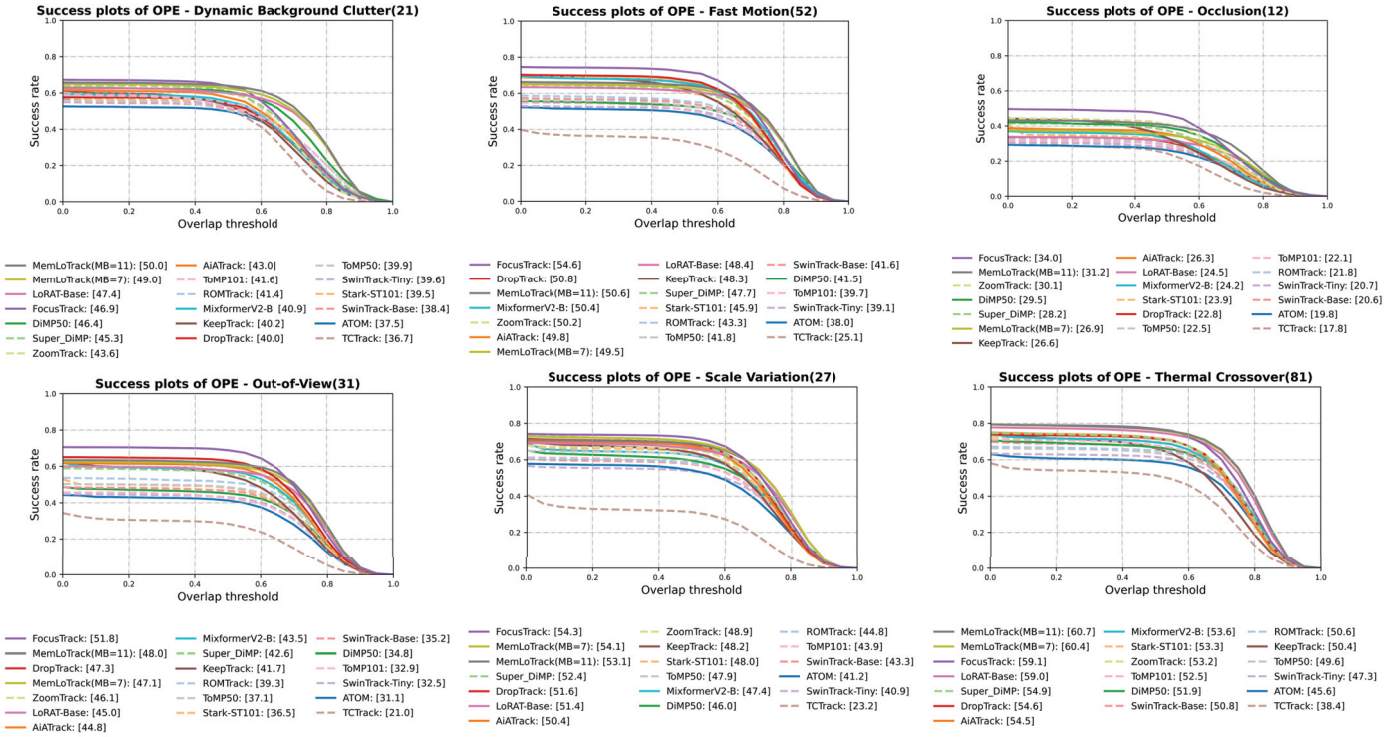


Figure 4. Attribute-wise Success AUC on Anti-UAV410 (DBC, TC, SV, OC, OV, FM).

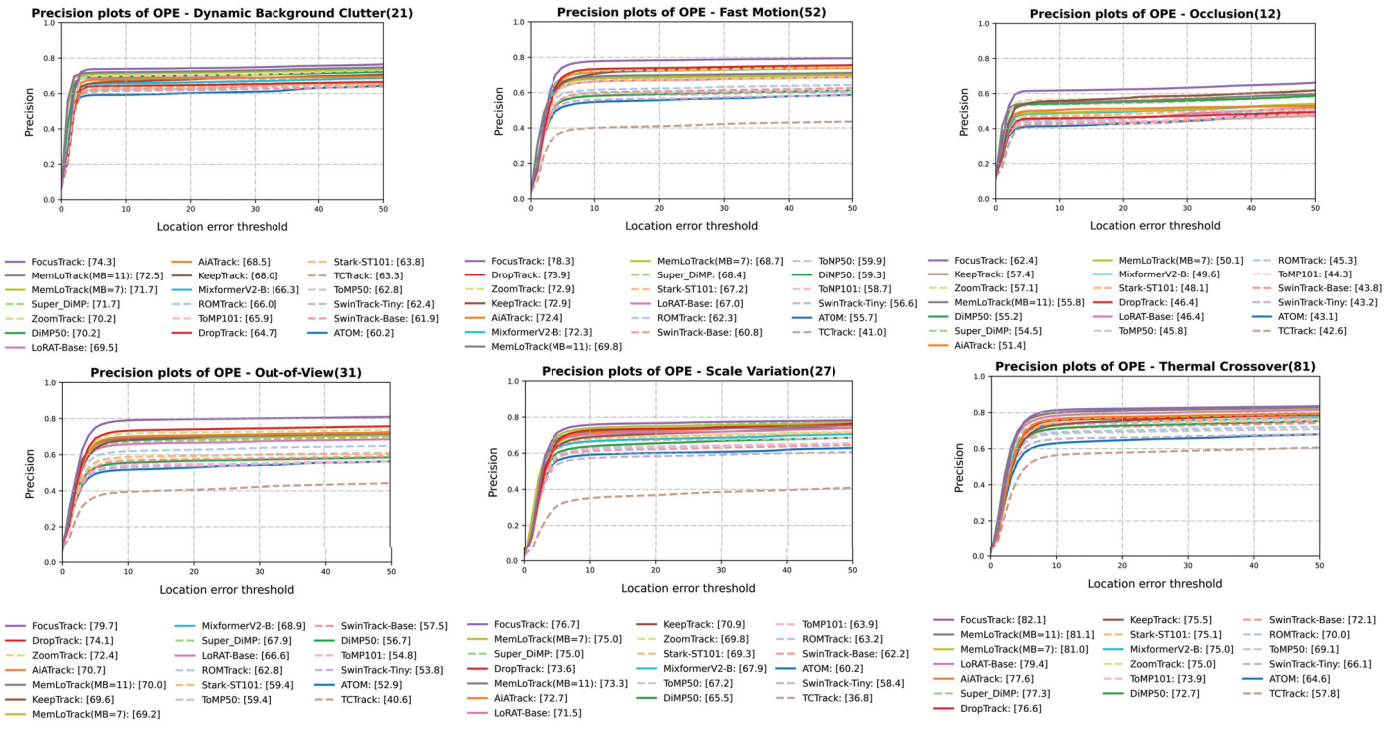


Figure 5. Attribute-wise precision on Anti-UAV410 (DBC, TC, SV, OC, OV, FM).

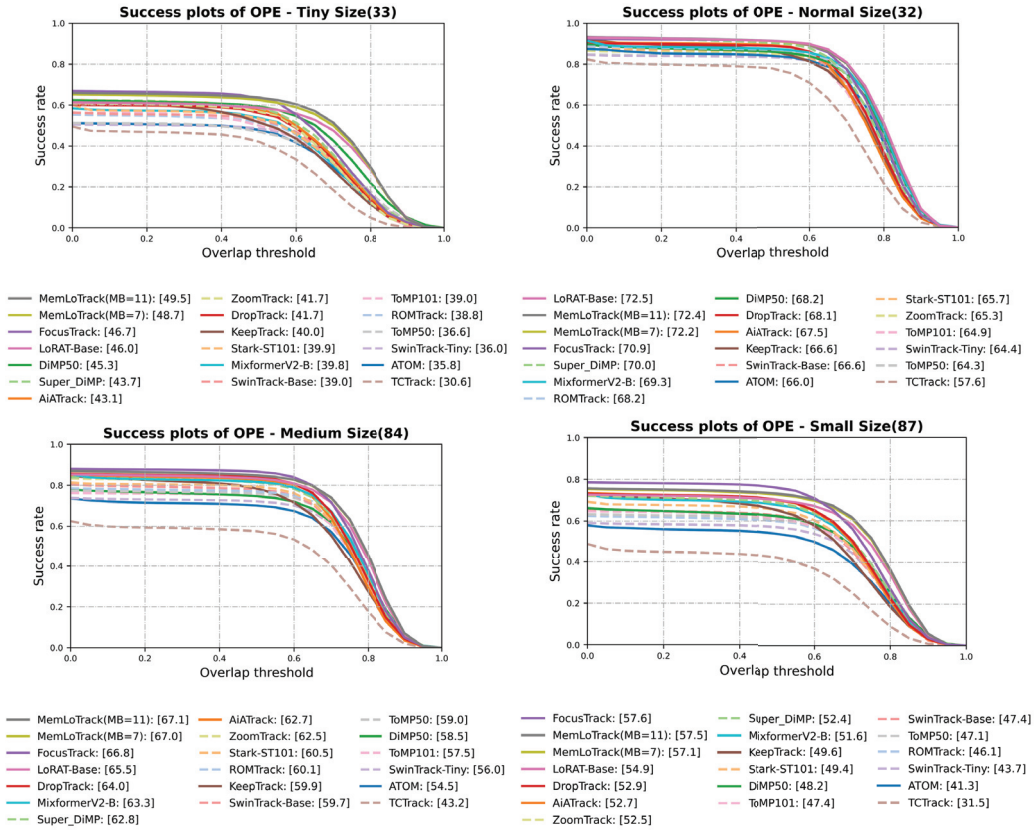


Figure 6. Target-size Success AUC on Anti-UAV410 (Tiny, Small, Medium, Normal).

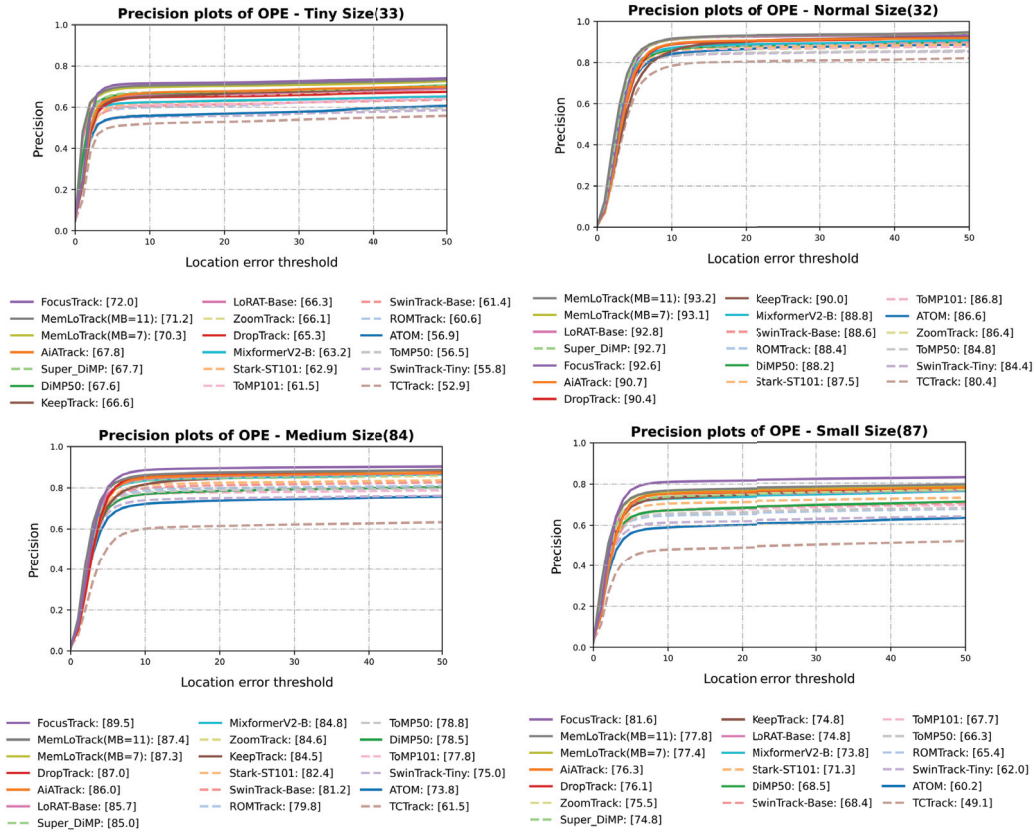


Figure 7. Target-size precision on Anti-UAV410 (Tiny, Small, Medium, Normal).

4.4. Attribute-Wise Analysis

Comparison with external trackers. With $MB = 7$, MemLoTrack demonstrates its strongest performance on the TC and DBC attributes and performs competitively across all target-size buckets (Tiny/Small/Medium/Normal), whereas FocusTrack [3] exhibits superior performance on FM, OC, OV, and SV. This pattern aligns with the aggregate outcome summarized in Section 4.3: MemLoTrack achieves higher AUC and SA but attains lower precision on attributes dominated by abrupt motion, long occlusions, or view losses. In cross-tracker comparisons, FocusTrack's advantages do not extend to all size buckets; MemLoTrack matches or exceeds it across Tiny/Small/Medium/Normal, while FocusTrack retains its advantage on FM/OC/OV/SV. Specifically, Figures 4 and 5 show that MemLoTrack falls behind on attributes involving rapid displacement (FM) and target loss (OC, OV). This is largely a consequence of the restricted field of view inherent to the 224×224 resolution of the local search window, which hampers the re-acquisition of targets that move out of frame quickly. Furthermore, the linear motion assumption of the Kalman filter is not ideal for abrupt maneuvers, limiting the tracker's ability to follow erratic UAV flight paths. This limitation regarding kinematic priors is empirically substantiated by our ablation study. Under the FM category, the Kalman-only policy (AUC 48.5) underperforms the visual only Conf-only policy (AUC 48.8), suggesting that kinematic priors can be detrimental when motion is erratic. Moreover, relaxing these constraints (Loose, $\lambda = 16.0$) yields higher FM performance (AUC 49.7) compared to the Strict setting ($\lambda = 4.0$, AUC 49.2), confirming that rigid adherence to the Kalman prior hinders tracking effectiveness during sudden acceleration or rotation.

4.5. Ablations: Memory and Thresholding

Effect of MB size. As established in Section 4.1, our default inference configuration uses $MB = 7$ to align with the 7-frame sampling strategy used during training. Table 4 analyzes the empirical impact of this design choice by comparing it against other bank sizes ($MB \in \{1, 3, 7, 11\}$) and the $MB = OFF$ baseline (all at $\tau = 0.8$). The baseline ($MB = OFF$) achieves a strong 62.9 AUC and runs at 424 fps. Enabling memory ($MB = 1$) provides an immediate boost (e.g., +0.6 AUC). Notably, our design-consistent MB = 7 setting also achieves peak performance across AUC (63.6), P (82.7), and P-Norm (79.8), offering the best balance. Increasing the capacity further to $MB = 11$ slightly raises State Accuracy (SA = 64.3), but this configuration exhibits a small drop in AUC and P compared to $MB = 7$. This pattern suggests a mild trade-off. A larger memory bank improves robustness on challenging frames, which is reflected in SA, while providing no additional benefit in per-frame localization accuracy as measured by AUC and P. Moreover, the differences in global AUC and P across $MB \in \{1, 3, 7\}$ remain very small, indicating that fine-grained tuning of the memory bank size is not the primary source of the observed gains. Instead, the practical advantage of the memory mechanism becomes clearer when contrasting $MB =$ OFF with $MB > 0$ and examining attribute-wise behavior. As summarized in our attribute-wise results, all $MB > 0$ configurations consistently outperform the memory bank-free baseline under challenging attributes such as OC, OV, and DBC, confirming that the memory bank primarily functions as a robustness mechanism against severe tracking disruptions rather than monotonically improving average AUC and P.

Table 4. Overall performance of MemLoTrack with different memory bank (MB) sizes; the best per-column score is **bold**. (fps measured on an NVIDIA GeForce RTX 4070 Ti SUPER.)

Variant	AUC	P	P-Norm	SA	fps
MB = OFF	62.9	81.8	79.2	63.3	424
MB = 1	63.5	82.5	**79.8**	63.5	248
MB = 3	63.4	82.4	79.7	63.7	207
MB = 7	**63.6**	**82.7**	**79.8**	64.0	153
MB = 11	62.5	81.1	78.5	**64.3**	127

Attribute-wise trends across MB. While the overall metrics in Table 4 suggest that MB sizes 1, 3, and 7 yield similar AUC, P, and P-Norm values, the attribute-wise analysis in Table 5 together with the ON/OFF comparison in Table 6 reveals that the memory bank mainly benefits rare but critical failure modes such as occlusion, out-of-view, and Dynamic Background Clutter. In these attributes, switching from $MB = OFF$ to $MB = ON$ yields substantial gains (e.g., OC: +8.2 AUC/+11.4 P), whereas frames from easier segments dominate the dataset-level averages and thus dilute the apparent differences among $MB = 1$, 3, and 7. From a practical standpoint, $MB = 7$ offers the best trade-off between overall accuracy and robustness to such challenging events, while remaining consistent with the training-time memory sampling strategy.

Table 5 details the attribute-level response to MB size (at $\tau = 0.8$). As bank size increases from $MB = 7$ to $MB = 11$, performance on challenging attributes improves significantly: **OC** gains +4.3 AUC (26.9 $\rightarrow$ 31.2) and +5.7 P (50.1 $\rightarrow$ 55.8). OV, DBC, and FM also see modest gains. This confirms that a larger memory bank enhances robustness to target loss. However, this comes at the cost of performance on SV, which drops by 1.0 AUC (54.1 $\rightarrow$ 53.1). Performance on Medium and Normal targets peaks at $MB = 3$. This trade-off analysis, combined with the overall metrics in Table 4, solidifies $MB = 7$ as the optimal default.

Table 5. Attribute-wise success (AUC) and precision (P) for MemLoTrack (ours) with different memory bank (MB) sizes.

Attribute	MB = 1		MB = 3		MB = 7		MB = 11	
	AUC	P	AUC	P	AUC	P	AUC	P
Thermal Crossover	59.7	79.7	60.0	80.2	60.5	81.0	60.7	81.1
Out-of-View	46.2	67.6	45.6	66.8	47.1	69.2	48.0	70.0
Scale Variation	53.2	73.4	52.7	72.9	54.1	75.0	53.1	73.3
Fast Motion	49.5	68.3	48.8	67.6	49.5	68.7	50.6	69.8
Occlusion	26.9	50.1	26.9	50.0	26.9	50.1	31.2	55.8
Dynamic Background Clutter	47.9	69.8	47.4	69.1	49.0	71.7	50.0	72.5
Tiny Size	47.8	68.8	46.8	67.5	48.7	70.3	49.5	71.2
Small Size	56.3	76.2	56.7	76.8	57.1	77.4	57.5	77.8
Medium Size	66.7	86.9	67.4	88.0	67.0	87.3	67.1	87.4
Normal Size	72.5	93.3	73.4	94.8	72.2	93.1	72.4	93.2

Gated memory vs. no memory. At $\tau = 0.8$ and $MB = 7$, enabling the memory bank and Memory Attention Layer yields consistent gains over the no-memory setting (Table 6). The largest deltas (ON$-$OFF) appear on challenging attributes: OC (+8.2 AUC/+11.4 P), OV (+4.1/+5.9), DBC (+6.5/+9.5), and SV (+3.3/+4.7). Other attributes, such as FM, show marginal changes (-0.1 AUC/-0.1 P), while performance on Medium and Normal size targets remains stable. At the dataset level, enabling the memory bank ($MB = ON$) improves all metrics (AUC/P/P-Norm/SA) from 62.9/81.8/79.2/63.3 to 63.6/82.7/79.8/64.0.

Table 6. Attribute-wise comparison at $\tau = 0.8$. MB = ON uses MB size 7; MB = OFF treats the Memory Attention Layer as a no-op. All numbers are percentages. Δ denotes ON−OFF.

Attribute	AUC			P		
	OFF	ON	Δ (ON-OFF)	OFF	ON	Δ (ON-OFF)
Thermal Crossover	59.7	60.5	+0.7	79.7	81.0	+1.3
Out-of-View	43.0	47.1	+4.1	63.3	69.2	+5.9
Scale Variation	50.7	54.1	+3.3	70.3	75.0	+4.7
Fast Motion	49.6	49.5	−0.1	68.8	68.7	−0.1
Occlusion	18.8	26.9	+8.2	38.6	50.1	+11.4
Dynamic Background Clutter	42.5	49.0	+6.5	62.2	71.7	+9.5
Tiny Size	43.5	48.7	+5.1	62.8	70.3	+7.5
Small Size	56.3	57.1	+0.8	76.2	77.4	+1.2
Medium Size	66.9	67.0	+0.1	87.2	87.3	+0.1
Normal Size	72.1	72.2	+0.1	92.8	93.1	+0.3

Threshold sweep. We ablate the inference-time confidence gate, sweeping $\tau \in [0.1, 0.9]$ on the validation split with MB = 7. Tightening τ from 0.8 to 0.9 collapses memory writes from 38,439 to 101 (Table 7), effectively bypassing the Memory Attention Layer and forcing single-frame operation. Despite this severe under-provisioning, the performance at $\tau = 0.9$ remains the second-highest (Figure 8), underscoring the tracker's strong single-frame capabilities. Nevertheless, $\tau = 0.8$ provides the optimal balance of filtering and coverage for peak performance across all metrics and is thus adopted as our default.

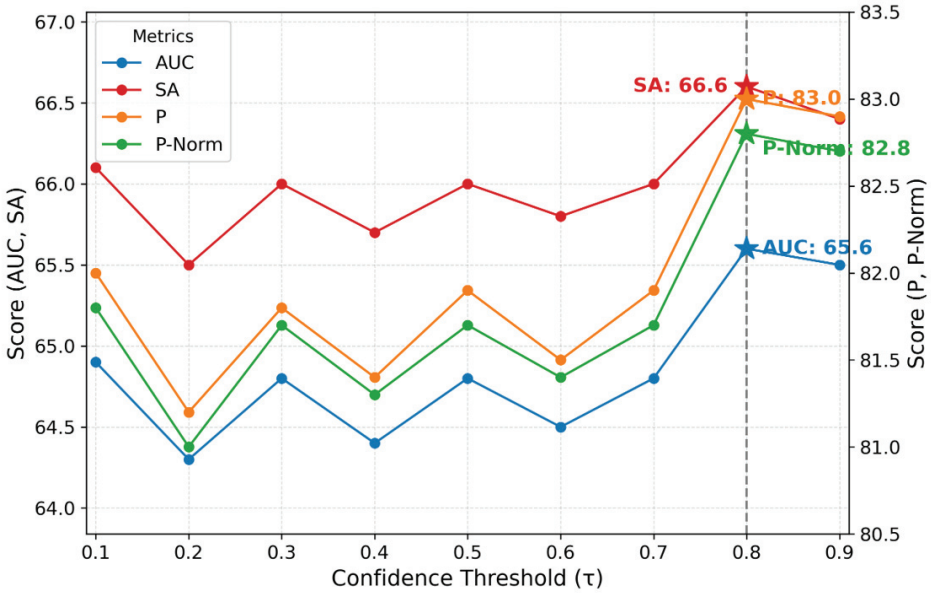


Figure 8. Threshold sweep over $\tau \in [0.1, 0.9]$ with MB = 7, evaluated on the validation dataset; all metrics peak at $\tau = 0.8$.

Kalman-based motion gating and threshold sensitivity. To quantify the sensitivity of the proposed motion gate and disentangle the roles of confidence and motion cues, we conducted controlled ablations as summarized in Table 8. The baseline employs an adaptive dual-gating policy with a chi-square threshold of $\lambda = 9.21$. We compared this against three threshold variants: Strict ($\lambda = 4.0$), Loose ($\lambda = 16.0$), and Fixed. Specifically, the Fixed variant is explicitly implemented as a standard non-adaptive Mahalanobis logic; it removes the adaptive scaling mechanism and relies solely on a constant threshold ($\lambda = 9.21$). We further evaluated two structural ablations: Kalman-only (motion gating

without confidence) and Conf-only (confidence gating without motion constraints). All models utilized the same backbone and a fixed memory bank size of $MB = 7$ to ensure fair comparison. The results indicate that the baseline configuration yields the highest State Accuracy (64.0). The threshold variants, including the non-adaptive Fixed model, exhibit only marginal degradation (63.1–63.7), suggesting that the method is robust to hyperparameter fluctuations provided λ is within a reasonable range. In contrast, structural changes lead to significant performance drops. Kalman-only (63.0 SA) underperforms on Tiny targets and SV due to the lack of appearance validation, while Conf-only (62.8 SA) struggles to filter distractors in DBC and OC scenarios without kinematic guidance. These findings confirm that the synergistic combination of confidence and motion priors is critical for robustness in the Anti-UAV410 benchmark.

Table 7. Effect of update threshold (τ) on the total number of memory bank updates for MB = 7 across the validation set.

Threshold (τ)	Total Memory Bank Updates
0.1	42,648
0.2	42,105
0.3	41,532
0.4	40,911
0.5	40,250
0.6	39,683
0.7	39,017
0.8	38,439
0.9	101

Table 8. Effect of Kalman-based motion gating variants on overall State Accuracy (SA) and attribute-wise AUC/precision on Anti-UAV410. All variants share the same backbone and memory bank size ($MB = 7$, $\tau = 0.8$). Only the gating policy differs.

	Ours (dual, $\lambda = 9.21$)	**Strict ($\lambda = 4.0$)**	**Loose ($\lambda = 16.0$)**	**Fixed (No Adapt.)**	**Kalman-Only**	**Conf-Only**
State Accuracy (SA)						
SA	64.0	63.1	63.7	63.1	63.0	62.8
Success AUC						
DBC	49.0	47.1	47.4	47.1	47.5	46.0
FM	49.5	49.2	49.7	49.2	48.5	48.8
OC	26.9	26.4	27.0	26.4	26.9	24.3
OV	47.1	45.8	46.3	45.8	44.7	45.0
SV	54.1	53.2	53.6	53.2	50.9	51.5
TC	60.4	59.1	59.6	59.1	59.0	58.7
Tiny	48.7	47.6	47.8	47.6	45.6	46.1
Small	57.1	55.8	56.4	55.8	55.7	55.5
Medium	67.0	66.1	66.7	66.1	66.7	66.4
Normal	72.2	72.4	72.6	72.4	71.7	71.9
Precision (P)						
DBC	71.7	68.7	69.2	68.7	69.4	67.1
FM	68.7	68.0	68.7	68.0	67.1	67.5
OC	50.1	49.5	50.3	49.5	50.3	46.4
OV	69.2	67.0	67.7	67.0	65.6	66.0
SV	75.0	73.6	74.1	73.6	70.1	71.0
TC	81.0	79.1	79.9	79.0	78.9	78.3
Tiny	70.3	68.6	69.0	68.6	65.7	66.5
Small	77.4	75.7	76.5	75.7	75.3	75.1
Medium	87.3	86.2	87.0	86.2	87.0	86.4
Normal	93.1	93.3	93.6	93.3	92.1	92.5

4.6. Qualitative Evaluation

Visualization setup. Figures 9 and 10 compare three inference settings of our tracker—$MB = OFF, MB = 7$, and $MB = 11$—together with the LoRAT-B baseline and FocusTrack against the ground truth (GT) across six canonical Anti-UAV410 attributes (FM, SV, OC, TC, OV, DBC). For each attribute, three representative frames are shown with matched zoomed crops to expose fine localization differences. This configuration enables a side-by-side comparison between a single-frame PEFT baseline without explicit temporal memory, a spatially adaptive state-of-the-art tracker, and our memory-augmented variants.

Effect of memory under occlusion and out-of-view. When memory is disabled, the tracker is prone to drift during long or partial occlusions and often latches onto distractors after re-entry. A similar drift behavior is observed for the LoRAT-B baseline, which has no explicit temporal memory, whereas FocusTrack generally re-detects onto the target more quickly after occlusion, consistent with its stronger OC and OV scores. Enabling the memory bank markedly stabilizes the trajectory: with $MB = 7$, the tracker reduces post-occlusion jitter and recovers the target with fewer frames; with $MB = 11$, re-acquisition becomes more reliable in prolonged or repeated occlusions. In out-of-view segments, $MB = 7$ suppresses background locks on re-entry, while $MB = 11$ further shortens re-localization lag, reflecting the benefit of longer temporal context. Overall, the qualitative panels show that the memory-enabled variants narrow much of the gap to FocusTrack while clearly improving over LoRAT-B and MB = OFF in OC and OV scenarios.

Handling of target absence. In sequences where the target truly leaves the search region (OV), the memory-enabled variants correctly maintain a "no-exist" state until the target reappears, avoiding spurious boxes in clutter. Compared with these variants, LoRAT-B and $MB = OFF$ tend to re-detect the target with a longer delay and exhibit more pronounced localization jitter after re-entry, reflecting the limitations of relying solely on single-frame evidence. FocusTrack and the memory-enabled MemLoTrack variants re-acquire the target more promptly and maintain a more consistent identity once it returns to the field of view. This behavior aligns with the intended presence-aware design of our tracker and is visible in the OV panels: the output remains empty during absence and resumes with a consistent identity upon reappearance.

Fast Motion and background dynamics. Under rapid displacement (FM), FocusTrack tends to produce the tightest and most responsive boxes, which reflects its quantitative advantage on this attribute, while the memory bank damps frame-to-frame overshoot, providing steadier box placement than $MB = OFF$. In the presence of Dynamic Background Clutter (DBC), memory helps discriminate transient background activations from target motion cues; the boxes from $MB = 7$ and $MB = 11$ adhere more tightly to GT than LoRAT-B and FocusTrack, particularly when water, foliage, or heat shimmer introduce confusing patterns.

Scale Variation and Thermal Crossover. For steep scale changes (SV), FocusTrack more aggressively follows changes in apparent size and often attains the tightest bounding boxes; $MB = 7$ offers a favorable balance between responsiveness and stability, yielding tighter boxes when altitude or apparent size shifts abruptly. Increasing the bank to $MB = 11$ can retain slightly outdated scale information in rare cases, trading a small amount of responsiveness for robustness to longer context. In Thermal Crossover (TC), both memory-enabled settings diminish target–background ambiguity relative to $MB = OFF$, LoRAT-B, and FocusTrack, with $MB = 11$ showing marginally cleaner boundaries in sustained crossover intervals.

Synthesis. Across attributes, the qualitative evidence supports the same pattern as our quantitative analyses: integrating a memory bank improves robustness to occlusion, out-of-view events, dynamic clutter, and thermal interference while stabilizing localization

under Fast Motion. FocusTrack remains the strongest competitor in scenarios dominated by rapid displacement and extended target loss, particularly FM, SV, OC, and OV, whereas MemLoTrack with $MB = 7$ or $MB = 11$ shows clearer advantages in TC and DBC sequences. The LoRAT-B baseline, which relies on single-frame evidence without explicit temporal memory, is especially vulnerable to long occlusions and out-of-view events and frequently exhibits drift or false locks in these cases. Between the two memory budgets, $MB = 7$ provides the best overall balance for precise, scale-responsive localization, whereas $MB = 11$ is preferable when prolonged occlusions or multi-cycle re-entries dominate. In all cases, memory-enabled tracking adheres more closely to GT than $MB = OFF$, confirming the utility of temporal aggregation in Anti-UAV settings.

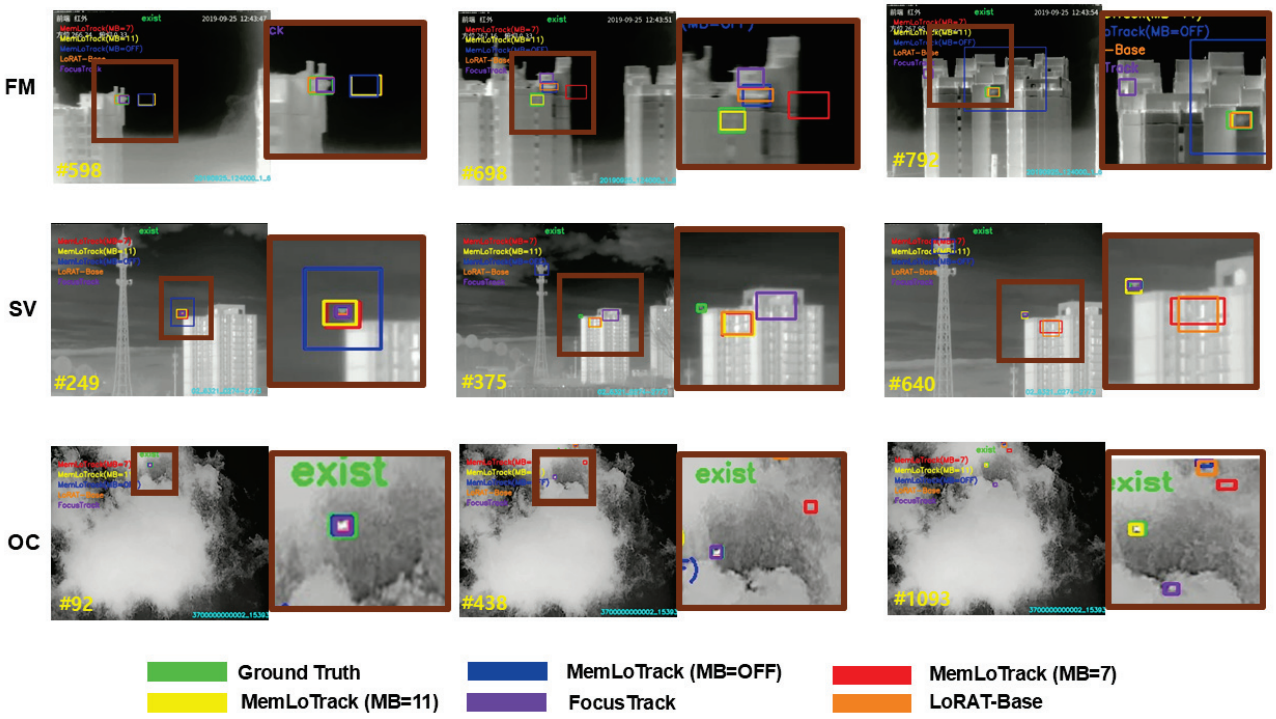


Figure 9. Qualitative comparison on sequences exhibiting Fast Motion (FM), Scale Variation (SV), and occlusion (OC). Outputs from MB = OFF, MB = 7, MB = 11, LoRAT-B, and FocusTrack are shown alongside the ground truth. Rows correspond to one sequence per attribute. Panels alternate between a full frame and a magnified crop; the crop on the right is taken from the region bounded by the brown rectangle in the panel immediately to its left. Inspect at high magnification for fine details.

4.7. Model Size and Computation

We summarize the compute budget of MemLoTrack in Table 9. "Total parameters" counts the entire tracker including the frozen DINOv2 backbone. "Trainable parameters" covers the LoRA adapters, token-type embeddings, the Memory Attention Layer, and the prediction head. "MACs" is the number of multiply-accumulate operations for one forward pass. To further evaluate the efficiency of our approach, we compare MemLoTrack with state-of-the-art trackers in Table 10. While the integration of the memory bank and MAL increases the computational load to 37.5 G MACs compared to the baseline LoRAT-B (30.0 G), MemLoTrack maintains a high inference speed of 153 FPS. Notably, our method is significantly faster than FocusTrack (28 FPS). Although FocusTrack reports lower theoretical MACs, its practical speed is constrained, likely due to the computational overhead of iterative Search Region Adjustment and mask refinement processes. These results demonstrate that MemLoTrack achieves a favorable balance between model complexity

and performance, ensuring the real-time capability essential for Anti-UAV applications while delivering robust tracking accuracy.

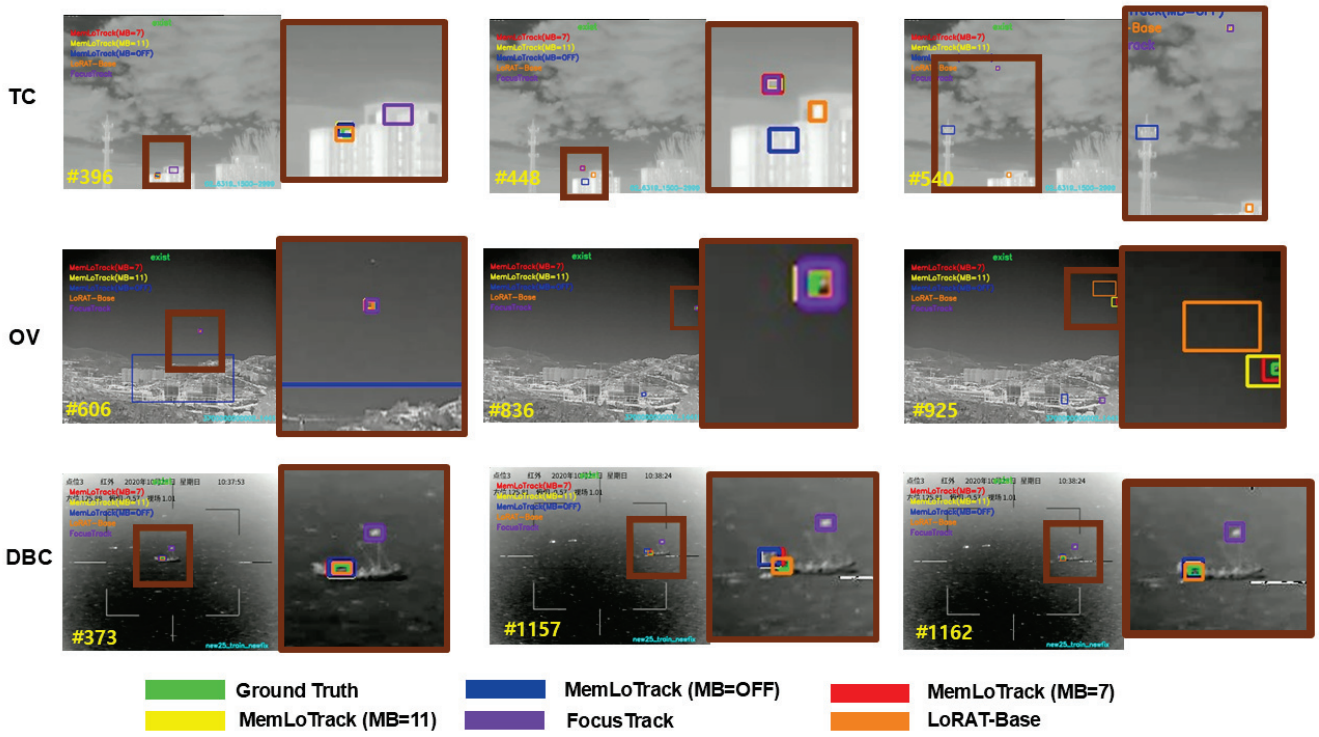


Figure 10. Qualitative comparison on sequences with Thermal Crossover (TC), out-of-view (OV), and Dynamic Background Clutter (DBC). Outputs from MB = OFF, MB = 7, MB = 11, LoRAT-B, and FocusTrack are shown alongside the ground truth. Rows correspond to one sequence per attribute. Panels alternate between a full frame and a magnified crop; the crop on the right is taken from the region bounded by the brown rectangle in the panel immediately to its left. Inspect at high magnification for fine details.

Table 9. Model size and computation of MemLoTrack.

Metric	Value
Total parameters (B)	0.12
Trainable parameters (M)	41.34
MACs (G)	37.5

Table 10. Comparison with state-of-the-art trackers in terms of computational complexity, inference speed, and tracking accuracy. The inference speeds (FPS) for LoRAT-B and MemLoTrack are measured on an NVIDIA GeForce RTX 4070 Ti Super GPU.

Tracker	Speed (fps)	MACs (G)	AUC	P	SA
OSTrack [6]	267	29.1	53.7	73.9	54.7
ROMTrack [39]	9	34.5	54.7	74.5	55.7
ZoomTrack [40]	276	29.1	58.4	81.2	59.4
DropTrack [42]	154	48.4	59.2	82.2	60.2
FocusTrack [3]	28	30.1	62.8	86.2	63.9
LoRAT-B [12]	700	30.0	62.2	80.9	62.5
MemLoTrack (MB = 7)	153	37.5	63.6	82.7	64.0

5. Discussion

Comparison to baseline. MemLoTrack is developed upon the LoRAT-B-224 architecture and evaluated at an identical input resolution ($x = 224 \times 224$). After fine-tuning on the Anti-UAV410 dataset, our model demonstrates consistent and significant performance improvements over the LoRAT-B baseline across all four standard metrics (e.g., AUC +1.4, P +1.8, P-Norm +1.7, SA +1.5; see Table 2).

Comparative analysis with state-of-the-art. Relative to FocusTrack [3], MemLoTrack exhibits a distinct trade-off between robustness and precision on the Anti-UAV410 benchmark. While it trails in distance-based precision (P: 82.7 vs. 86.2), MemLoTrack achieves superior success-oriented metrics, recording higher AUC and SA scores of 63.6 and 64.0 compared to FocusTrack's 62.8 and 63.9. This divergence reflects our design objective, which prioritizes tracking robustness over pixel-level proximity. We attribute these performance characteristics to the fundamental difference in architectural priors. FocusTrack leverages a spatial prior through Search Region Adjustment and Attention-to-Mask modules to refine localization. Conversely, MemLoTrack implements a temporal prior via a dual-gated FIFO memory and Kalman filtering. By strictly utilizing kinematically consistent frames to guide the Memory Attention Layer, our method excels in attributes such as DBC and TC and remains robust against Tiny targets. In contrast, the spatial adaptability of FocusTrack yields advantages in scenarios with rapid displacement, such as FM, OC, OV, and SV. Although the difference in input resolution (224 vs. 256) may contribute to the precision gap, the results ultimately highlight MemLoTrack as a robustness-oriented alternative that complements the spatial strategies of FocusTrack.

Effectiveness of the memory mechanism. The temporal memory is a key contributor to MemLoTrack's gains under challenging cases. With MB = 7 and $\tau = 0.8$, switching memory ON yields the largest OC delta versus MB = OFF (AUC +8.2; P +11.4), with OV and DBC also improving and FM essentially unchanged (Table 6). SA increases modestly, indicating that memory chiefly strengthens presence estimation and target reacquisition.

Varying the memory capacity reveals a trade-off between the benefits of long-term context (recency) and the risk of using outdated information (staleness). Dataset-level AUC and P peak at MB = 7 (Table 4), whereas within MemLoTrack, MB = 11 attains the best absolute OC (AUC 31.2 vs. 26.9 at MB = 7) and slightly higher OV, but lowers overall AUC/P and coincides with a small regression on SV (AUC 54.1→53.1; P 75.0→73.3) (Table 5). On the validation set, a threshold sweep selects $\tau = 0.8$ as optimal; $\tau = 0.9$ is too strict and degrades performance by delaying memory updates (Figure 8). For completeness, P-Norm is tied at MB = 1 and MB = 7 (79.8), while SA is highest at MB = 11 (64.3).

Practical implications and guidance. We use MB = 7 by design—mirroring the seven memory frames used during training—and set $\tau = 0.8$ via a validation sweep (Table 4). Prefer MB = 11 only when prolonged occlusion/re-entry dominates and a small loss in fine localization is tolerable. For high-speed maneuvering, memory alone offers limited benefit; compensate with higher input resolution and/or a lightweight motion prior. Throughput remains real-time in our setup (RTX 4070 Ti SUPER); for reference, 153 fps at MB = 7.

6. Conclusions

In this work, we presented MemLoTrack, a novel tracking architecture designed for long-horizon robustness. The architecture employs a frozen DINOv2 ViT backbone with parameter-efficient LoRA adapters and a lightweight Memory Attention Layer (MAL) to incorporate temporal context. To align the model with this memory-augmented process, a stateless memory frame sampling strategy is used during training. In contrast, during inference, the temporal context is supplied by a FIFO memory bank, which is critically governed by a dual-gated update policy using both a confidence threshold and a Kalman

filter-based motion consistency check. This design effectively curates a high-quality memory to mitigate drift during occlusion and target absence, all without the computational expense of online backbone updates.

On the challenging Anti-UAV410 benchmark, MemLoTrack demonstrates highly competitive performance, achieving the highest reported scores for the key success metrics of AUC (63.6) and SA (64.0). This result represents a significant improvement over the LoRAT-B architecture baseline (+1.4 AUC) and showcases a distinct advantage in overall tracking success when compared to spatially adaptive methods like FocusTrack. This robustness-focused profile, achieved at the cost of some localization precision, validates the effectiveness of our temporally grounded approach for mitigating tracking failures caused by target disappearance and heavy background clutter.

Our comprehensive ablation studies validate that these performance gains are directly attributable to the gated memory mechanism. The temporal context provides pronounced benefits in failure-prone scenarios, most notably under occlusion (+8.2 AUC, +11.4 P) and during out-of-view events, while leaving attributes like Fast Motion largely unaffected. Furthermore, our analysis reveals a clear trade-off between recency and staleness, identifying a memory budget of MB = 7 and a confidence gate of $\tau = 0.8$ as the optimal configuration for balancing peak performance with robustness.

Ultimately, MemLoTrack presents a compelling design that balances tracking robustness with computational efficiency. By demonstrating that a selective temporal memory can be integrated into a parameter-efficient framework to achieve real-time throughput ($\sim$153 fps on an NVIDIA RTX 4070 Ti SUPER) without online fine-tuning, our work offers a practical and deployable solution for high-stakes TIR tracking applications. It validates that for complex scenarios dominated by occlusion and clutter, a memory-augmented approach is a powerful strategy, highlighting a promising direction for developing more resilient and efficient visual trackers.

Author Contributions: Conceptualization, J.K.P.; Methodology, J.K.P. and J.-H.H.; Software, J.K.P.; Validation, J.K.P. and J.-H.H.; Formal analysis, J.K.P.; Investigation, J.K.P.; Resources, J.-H.H.; Data curation, J.K.P.; Writing—original draft, J.K.P.; Writing—review & editing, J.-H.H.; Supervision, J.-H.H.; Project administration, J.-H.H.; Funding acquisition, J.-H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by IITP (Institute of Information & Communications Technology Planning & Evaluation)—ITRC (Information Technology Research Center) grant funded by the Korea government (Ministry of Science and ICT) grant number IITP-2025-RS-2022-00156295.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: During the preparation of this manuscript, the authors used GPT5 and gemini-2.5-pro for the purposes of generating text and drafting the initial script to improve the overall langauge for the clarity and readability. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jiang, N.; Wang, K.; Peng, X.; Yu, X.; Wang, Q.; Xing, J.; Li, G.; Guo, G.; Zhao, J.; Han, Z. Anti-UAV: A Large Multi-Modal Benchmark for UAV Tracking. *arXiv* **2021**, arXiv:2101.08466.
2. Huang, B.; Li, J.; Chen, J.; Wang, G.; Zhao, J.; Xu, T. Anti-UAV410: A Thermal Infrared Benchmark and Customized Scheme for Tracking Drones in the Wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 2852–2865. [CrossRef] [PubMed]
3. Wang, Y.; Xu, T.; Li, J. FocusTrack: A Self-Adaptive Local Sampling Algorithm for Efficient Anti-UAV Tracking. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 1–14. [CrossRef]

4. Huang, L.; Zhao, X.; Huang, K. Globaltrack: A simple and strong baseline for long-term tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11037–11044.
5. Voigtlaender, P.; Luiten, J.; Torr, P.H.S.; Leibe, B. Siam R-CNN: Visual Tracking by Re-Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6577–6587.
6. Ye, B.; Chang, H.; Ma, B.; Shan, S. Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022.
7. Cui, Y.; Song, T.S.; Wu, G.; Wang, L. MixFormerV2: Efficient Fully Transformer Tracking. *arXiv* **2023**, arXiv:2305.15896.
8. Chen, X.; Peng, H.; Wang, D.; Lu, H.; Hu, H. Seqtrack: Sequence to sequence learning for visual object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 14572–14581.
9. Lin, L.; Fan, H.; Xu, Y.; Ling, H. SwinTrack: A Simple and Strong Baseline for Transformer Tracking. *arXiv* **2021**, arXiv:2112.00995.
10. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *ICLR* **2022**, *1*, 3.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998.
12. Lin, L.; Fan, H.; Zhang, Z.; Wang, Y.; Xu, Y.; Ling, H. Tracking Meets LoRA: Faster Training, Larger Model, Stronger Performance. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024.
13. Yang, C.Y.; Huang, H.W.; Chai, W.; Jiang, Z.; Hwang, J.N. SAMURAI: Adapting Segment Anything Model for Zero-Shot Visual Tracking with Motion-Aware Memory. *arXiv* **2024**, arXiv:2411.11922.
14. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. Dinov2: Learning robust visual features without supervision. *arXiv* **2023**, arXiv:2304.07193.
15. Gao, S.; Zhou, C.; Zhang, J. Generalized Relation Modeling for Transformer Tracking. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 18686–18695. [CrossRef]
16. Bai, Y.; Zhao, Z.; Gong, Y.; Wei, X. ARTrackV2: Prompting Autoregressive Tracker Where to Look and How to Describe. *arXiv* **2023**, arXiv:2312.17133.
17. Jia, M.; Tang, L.; Chen, B.C.; Cardie, C.; Belongie, S.; Hariharan, B.; Lim, S.N. Visual prompt tuning. In Computer Vision—ECCV 2022, Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 709–727.
18. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 2790–2799.
19. Liu, H.; Tam, D.; Muqeeth, M.; Mohta, J.; Huang, T.; Bansal, M.; Raffel, C.A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 1950–1965.
20. Fan, H.; Bai, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Harshit; Huang, M.; Liu, J.; et al. Lasot: A high-quality large-scale single object tracking benchmark. *Int. J. Comput. Vis.* **2021**, *129*, 439–461. [CrossRef]
21. Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; Ghanem, B. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 300–317.
22. Huang, L.; Zhao, X.; Huang, K. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1562–1577.
23. Wang, X.; Shu, X.; Zhang, Z.; Jiang, B.; Wang, Y.; Tian, Y.; Wu, F. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13763–13773.
24. Zhao, J.; Zhang, J.; Li, D.; Wang, D. Vision-based anti-uav detection and tracking. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25323–25334. [CrossRef]
25. Liu, Q.; He, Z.; Li, X.; Zheng, Y. PTB-TIR: A thermal infrared pedestrian tracking benchmark. *IEEE Trans. Multimed.* **2019**, *22*, 666–675.
26. Liu, Q.; Li, X.; He, Z.; Li, C.; Li, J.; Zhou, Z.; Yuan, D.; Li, J.; Yang, K.; Fan, N.; et al. LSOTB-TIR: A Large-Scale High-Diversity Thermal Infrared Object Tracking Benchmark. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020.
27. Liao, D.; Shu, X.; Li, Z.; Liu, Q.; Yuan, D.; Chang, X.; He, Z. Fine-Grained Feature and Template Reconstruction for TIR Object Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2025**, *35*, 9276–9286. [CrossRef]
28. Ravi, N.; Gabeur, V.; Hu, Y.T.; Hu, R.; Ryali, C.K.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. SAM 2: Segment Anything in Images and Videos. *arXiv* **2024**, arXiv:2408.00714.

29. Zheng, Y.; Zhong, B.; Liang, Q.; Mo, Z.; Zhang, S.; Li, X. Odtrack: Online dense temporal token learning for visual tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; Volume 38, pp. 7588–7596.
30. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. Atom: Accurate tracking by overlap maximization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4660–4669.
31. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer Tracking. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8122–8131.
32. Mayer, C.; Danelljan, M.; Bhat, G.; Paul, M.; Paudel, D.P.; Yu, F.; Van Gool, L. Transforming Model Prediction for Tracking. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8721–8730. [CrossRef]
33. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning Spatio-Temporal Transformer for Visual Tracking. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10428–10437.
34. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 740–755.
35. Han, X.; Oishi, N.; Tian, Y.; Ucurum, E.; Young, R.; Chatwin, C.; Birch, P. ETTrack: Enhanced temporal motion predictor for multi-object tracking. *arXiv* **2024**, arXiv:2405.15755. [CrossRef]
36. Zhou, L.; Zhou, Z.; Mao, K.; He, Z. Joint Visual Grounding and Tracking with Natural Language Specification. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 23151–23160.
37. Zhang, M.; Zhang, Q.; Song, W.; Huang, D.; He, Q. PromptVT: Prompting for Efficient and Accurate Visual Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 7373–7385. [CrossRef]
38. Cao, Z.; Huang, Z.; Pan, L.; Zhang, S.; Liu, Z.; Fu, C. TCTrack: Temporal Contexts for Aerial Tracking. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 14778–14788.
39. Cai, Y.; Liu, J.; Tang, J.; Wu, G. Robust Object Modeling for Visual Tracking. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 9555–9566.
40. Kou, Y.; Gao, J.; Li, B.; Wang, G.; Hu, W.; Wang, Y.; Li, L. ZoomTrack: Target-aware Non-uniform Resizing for Efficient Visual Tracking. *arXiv* **2023**, arXiv:2310.10071.
41. Gao, S.; Zhou, C.; Ma, C.; Wang, X.; Yuan, J. AiATrack: Attention in Attention for Transformer Visual Tracking. *arXiv* **2022**, arXiv:/2207.09603. [CrossRef]
42. Wu, Q.; Yang, T.; Liu, Z.; Lin, W.; Wu, B.; Chan, A.B. DropMAE: Learning Representations via Masked Autoencoders with Spatial-Attention Dropout for Temporal Matching Tasks. *arXiv* **2023**, arXiv:2304.00571.
43. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 850–865.
44. Bhat, G.; Danelljan, M.; Gool, L.V.; Timofte, R. Learning discriminative model prediction for tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6182–6191.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

sensors

MDPI

Article

Outdoor Characterization and Geometry-Aware Error Modelling of an RGB-D Stereo Camera for Safety-Related Obstacle Detection

Pierluigi Rossi [1], Elisa Cioccolo [1,*], Maurizio Cutini [2], Danilo Monarca [1], Daniele Puri [3], Davide Gattamelata [3] and Leonardo Vita [3]

[1] Department of Agriculture and Forest Sciences (DAFNE), Tuscia University, 01100 Viterbo, Italy; pierluigi.rossi@unitus.it (P.R.); monarca@unitus.it (D.M.)
[2] CREA Consiglio per la Ricerca in Agricoltura e L'analisi Dell'economia Agraria (Research Centre for Engineering and Agro-Food Processing), Via Milano 43, 24047 Treviglio, Italy; maurizio.cutini@crea.gov.it
[3] Department of Technological Innovations and Safety of Plants, Products and Anthropic Settlements, Italian National Institute for Insurance Against Accidents at Work (INAIL), Via Roberto Ferruzzi 38/40, 00143 Rome, Italy; d.puri@inail.it (D.P.); d.gattamelata@inail.it (D.G.); l.vita@inail.it (L.V.)
* Correspondence: elisa.cioccolo@unitus.it

Highlights

- **What is the chance of detecting distance errors in RGB-D Cameras?**
  Depth camera errors can be predicted according to distance, angle of the target and light conditions;
- **How can these sources of measurement bias be considered?**
  A geometry-aware model was tested to provide depth measurement corrections in outdoor environments;
- **What is the behavior of the sensor across different distances?**
  Distance measurement errors grow with distance and angle up to 3.5 m with targets at 16 m;
- **What is the precision of the depth correction model developed in this research?**
  Depth correction models can achieve RMSE between 0.46 and 0.64 m, even at long distances.

Abstract

Stereo cameras, also known as depth cameras or RGB-D cameras, are increasingly employed in a large variety of machinery for obstacle detection purposes and navigation planning. This also represents an opportunity in agricultural machinery for safety purposes to detect the presence of workers on foot and avoid collisions. However, their outdoor performance at medium and long range under operational light conditions remains weakly quantified: the authors then fit a field protocol and a model to characterize the pipeline of stereo cameras, taking the Intel RealSense D455 as benchmark, across various distances from 4 m to 16 m in realistic farm settings. Tests have been conducted using a 1 square meter planar target in outdoor environments, under diverse illumination conditions and with the panel being located at 0°, 10°, 20° and 35° from the center of the camera's field of view (FoV). Built-in presets were also adjusted during tests, to generate a total of 128 samples. The authors then fit disparity surfaces to predict and correct systematic bias as a function of distance and radial FoV position, allowing them to compute mean depth and estimate a model of systematic error that takes depth bias as a function of distance, light conditions and FoV position. The results showed that the model can predict depth errors achieving a

good degree of precision in every tested scenario (RMSE: 0.46–0.64 m, MAE: 0.40–0.51 m), enabling the possibility of replication and benchmarking on other sensors and field contexts while supporting safety-critical perception systems in agriculture.

Keywords: agriculture; safety; RGB-D; depth cameras; obstacle detection

1. Introduction

Depth perception is one of the key abilities that allows both autonomous and semi-autonomous systems to move safely, plan their paths and understand their surroundings in most environments, playing a central role in what nowadays is defined as motion planning and obstacle avoidance. Over the past years, compact depth-sensing cameras—often referred to as RGB-D or stereo cameras—have become increasingly common thanks to their affordability, small size and ease of integration [1–3]: combining standard color red–green–blue (RGB) images with stereo triangulation, these devices provide real-time depth maps, making them useful across a large variety of applications, from mobile robots and logistics to driver-assistance systems and human–robot interaction [4,5].

Although these sensors are now widely used, their reliability in challenging environments is still questionable due to several aspects (weather, light, harsh terrain, etc.), and concerns arise when safety systems are solely based on such technology. In fact, most commercial products such as the Intel® RealSense™ D400 series [6] are designed mainly for indoor or short-range robotic use [7,8]. When exposed to outdoor lighting, where brightness can severely change across a sunny day, cloudy conditions and dark environments, their performance naturally tends to drop. This often results in noisy depth data or systematic measurement bias [9,10]. Such limitations are the reason behind safety concerns, particularly in agricultural contexts, where accurate depth information is essential for avoiding collisions and detecting workers on foot [11].

Stereo vision estimates depth by identifying matching points in two or more images taken from slightly different positions and then triangulating their disparities: active stereo cameras expand on this principle by projecting infrared or structured-light patterns, which help find matches even in low-texture areas [12,13]. However, both active and passive systems are vulnerable to strong sunlight, reflections, shadows and vegetation, all of which can distort the captured patterns and reduce accuracy [14].

A good example is the Intel RealSense D400 series, which uses a global-shutter stereo pair, an optional infrared emitter and an internal processing pipeline for depth estimation. The camera's Software Development Kit (SDK) includes several presets—such as default, high accuracy and high density—that balance precision, range and data density; however, their performance in outdoor conditions and outside of their standard range remains unpredictable and strongly context-dependent [15]. Prior research has shown, for example, that depth errors tend to increase toward the edges of the FoV [16], and that bright sunlight can amplify noise significantly [17]. Other studies confirm that ambient light and background texture can heavily influence the camera's effective sensing range [18], also showing how agricultural environments pose some of the toughest challenges for vision-based perception [19–21]. Fields and orchards are visually complex, full of dense vegetation, fine textures and constantly changing lighting conditions [22]. Machinery operating in these environments must also contend with dust, shadows from plants or tools and oblique sunlight, all of which create strong and unpredictable contrast variations [23]. These factors make it difficult for stereo cameras to maintain stable and accurate depth estimates, especially at medium-to-long distances (4–15 m), where measurement precision

naturally decreases, becoming a crucial factor for autonomous tractors or agricultural robots where the reliable detection of people and obstacles is a key safety aspect [24].

Depth cameras are a valid alternative to various sensing technologies that have been explored for agricultural safety: LiDAR sensors offer high accuracy at long ranges but are costly and provide lower resolution [25]. Ultrasonic sensors represent a low-cost solution but lack spatial precision. While purely visual systems based on RGB images can recognize humans or obstacles, they do not provide direct depth information [26]. Stereo cameras, therefore, present a promising solution, combining good accuracy, coverage and affordability when their errors can be properly characterized and corrected for real-world conditions.

Recent studies have started to address the need for the systematic characterization of stereo cameras in both lab and outdoor settings, given the limits of other technologies [27,28]. Moreover, while average error and standard deviation are common evaluation metrics, they do not fully describe how depth bias changes with distance or viewing angle. This is the reason why modeling such sources of bias is critical for predicting and correcting systematic errors, as well as for defining reliable confidence bounds that can inform safety-related decisions [29]. In addition, standards such as the International Organization for Standardization (ISO) 18497 [30] and ISO 25119 [31] define the performance requirements for systems that operate near humans, emphasizing the importance of measurable and validated sensor reliability; indeed RGB-D cameras, despite being highly attractive due to their affordable costs and high-resolution data, are limited by their performance in uncontrolled outdoor environments and in field deployments [32,33].

This work is built on previous research by presenting and validating a field-oriented protocol for testing stereo RGB-D cameras in realistic agricultural conditions. Using the Intel RealSense D455 as a case study, a series of controlled outdoor experiments were carried out to examine the effects of three key variables: (i) distance (4–16 m), (ii) field-of-view angle (0°, 10°, 20° and 35° off-center) and (iii) illumination time. A 1 m^2 planar target was used for the tests and placed in vegetation to reflect the visual complexity of contexts such as agriculture and forestry. The goal is to map condition-dependent trade-offs that matter to safety functions, and to provide an error model that can be used to correct raw depth or, at minimum, to inflate safety margins when detection confidence drops.

To address the aforementioned gap between previous research and the use of RGB-D sensors in outdoor activities, an Intel RealSense D455 has been employed as a study platform to be tested in 128 outdoor trials that jointly vary three key factors, namely, distance, FoV and light conditions. The scope of the research is to estimate an interpretable, geometry-aware error surface that would also allow us to quantify the practical contribution of the built-in laser emitter projector of the camera at various ranges. The resulting model would provide valuable guidance for safety-related obstacle detection systems on activities that are highly affected by variability in environmental conditions, for instance, those involving agricultural machinery.

The protocol deliberately follows the camera's built-in functions so that the resulting model stays anchored to the physics of stereo vision while remaining practical for field use, and it is meant to be applied to other cameras as well, since it is not just meant to identify the best calibration for the RGB-D sensor model that has been employed for the tests but to provide a model-agnostic methodology. Tests have therefore been organized to emphasize the effects of camera presets to obtain a compact and calibrated model that links depth bias to viewing geometry and illumination, which are elements that are directly linked to risk assessment models as well and system configuration settings as countermeasures.

2. Materials and Methods

Starting from the aforementioned state of the art on the use of RGB-D cameras for safety purposes, the authors selected an Intel RealSense D455 (Cupertino, CA, USA) as a reference, due to its widespread use in several applications and research studies [34]. Since, from a systems engineering perspective, understanding how perception uncertainty evolves with lighting and distance would allow adaptive safety margins, the authors intended to perform specific tests to evaluate the effects of environmental conditions on visibility and the performances of the sensor at different ranges and angles. As already stated in previous research, distance estimation is affected by an error which might severely limit their applications for safety purposes, given the chance of mistakenly assessing objects further than away than they effectively are. This is a major issue when the safety of vehicles depends on knowing the relative position of obstacles along their trajectory and when the braking distance is higher than the available space, generating the need for a model which is able to provide the error associated with a specific environmental condition and prevent dangerous optimistic estimations.

The research protocol developed for this research follows the camera's native SDK processing pipeline and evaluates several built-in presets to gather several measurements in different configurations, using a disparity surface to identify and model systematic depth bias as a function of both range and angle of the surface in the Field of View (FoV). The purpose was to develop a model that could be used either to correct these biases directly or to estimate adaptive safety margins when confidence in sensor data is low, aiming to bridge the gap between controlled laboratory testing and the real-world outdoor deployment of stereo cameras. This will contribute to the development of standardized methods for evaluating depth cameras' perception reliability, which is an essential step toward the safe and certified use of depth cameras in agricultural machinery.

2.1. Methodology for Depth Error Calibration

To do so, the authors designed a research protocol that takes into account both physics and concepts that lay behind the depth estimation methods, starting from how the distance of a pixel is estimated by the camera [35,36], as shown in Figure 1.

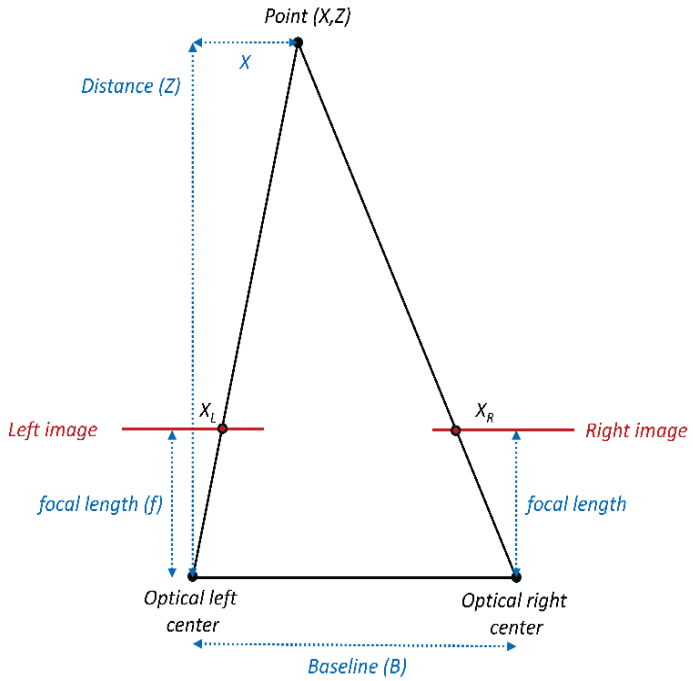


Figure 1. Disparity estimation method in depth cameras.

The ratio between the distance of the point from the origin (X) and the point's distance (Z) equals the ratio of the position of the point in the left image (XL) and focal length (f). Equation (1) shows how XL can therefore be related to focal length and the point's distance:

$$\frac{X}{Z} = \frac{X_L}{f} \qquad X_L = f\frac{X}{Z} \tag{1}$$

The same equation can be set up for the position of the point in the right image (XR), but of course the baseline's length (B) must be subtracted from the X coordinate of the observed point. Equation (2) provides the same information regarding XL for XR as well.

$$\frac{X-B}{Z} = \frac{X_R}{f} \qquad X_R = f\frac{X-B}{Z} \tag{2}$$

Having obtained XL and XR, disparity (d) can be calculated as the difference among these two values. The result is shown in Equation (3):

$$d = X_L - X_R \qquad d = f\left(\frac{X}{Z} - \frac{X-B}{Z}\right) = f\frac{B}{Z} \tag{3}$$

From that information, distance (Z) can be calculated as follows in Equation (4):

$$Z = \frac{fB}{d} \tag{4}$$

This method is used to estimate depth in RGB-D cameras starting from a disparity calculated from two optical sensors of known focal length and baseline. To understand how these equations are affected by measurement errors, considerations must be found on derivative estimations on disparity in Equation (5).

$$\frac{\partial Z}{\partial d} = -\frac{fB}{d^2} \qquad \Delta Z \approx \left|\frac{\partial Z}{\partial d}\right|\Delta d = \frac{fB}{d^2}\Delta d = \frac{Z^2}{fB}\Delta d \tag{5}$$

Equation (5) provides the insights that are required to understand how distance error is generated and what would be required by a model to provide this kind of information. Given that the error is affected by the square of distance and of disparity, with the last being related to the angle at which the pixel is located in the frame, a model that considers these effects has been tested and has been defined, as shown in Equation (6), with the inclusion of both the assessed distance and the angle of the point plus several calibration weights and their relative interactions.

$$(Z,\ \alpha) = w_0 + w_1 Z + w_2 Z^2 + w_3 tan\ \alpha + w_4 (tan\ \alpha)^{\ 2} + w_5\ Z\ tan\ \alpha \tag{6}$$

This model aims to consider a series of parameters that vary according to the situation:

- an offset calibration value (w_0) has been considered;
- scale errors (w_1) related to the assessed distance;
- the quadratic growth of bias given by noisy measurements (w_2Z^2);
- a metric related to the FoV angle α, in a form (tan α) that aims to adhere to the X/Z ratio used in disparity, with its calibration weight (w_3);
- the quadratic growth of bias given by extreme angle measurements, together with its specific weight (w_4);
- overlapping effects of distances and angles, again with their weight (w_5).

This choice on the mathematical model relies on the will to remain strongly linked to the physics of depth estimations and to leave some flexibility; a greater number of

parameters could lead to better results based on test data, but also in overfitting the model to the camera type and field conditions. Moreover, since the model does not aim to interpret relative coefficient magnitudes nor apply scale-sensitive penalties, feature normalization is not required for unbiased estimation or prediction in this specific operational setting. The authors also assume zero-mean and mutually uncorrelated errors with respect to the weights and in relation to the values used as possible predictors (Z and tan α). Since the dataset consists of independent distance–angle trials, any time or spatial correlation tests would not be directly applicable, so focus has been given to an interpretable geometry-aware error model.

2.2. Experimental Design

Tests were run at the Experimental Farm of Tuscia University located in Viterbo, Italy. The Selected area was an environment that is common to many outdoor applications, namely, a plain lawn field of approximately 0.15 ha in size. Field layout, including the panel used for tests, is shown in Figure 2.



(**a**) (**b**)

Figure 2. Field setup for detection tests. Placeholders were set on the grass as distance reference (**a**), while the gray panel was set parallel to the depth camera at each test point (**b**).

Having defined the field layout, points were selected to perform the tests. A set of four points were selected at distances of 4 m, 8 m, 12 m and 16 m. For each of these points, measurements were repeated in the following conditions:

- 0 degrees, namely, with the camera right in front of the panel. In this condition, the panel will fall exactly in the center of the frame;
- 10° angle, keeping the camera parallel to the panel, but with it being visible slightly on the right from the center of the frame;
- 20° angle, with the same camera conditions of the previous scenarios but slightly more toward the edge of the frame;
- 35° angle, which means that the panel is nearly on the edge of the frame, but is still fully visible at every distance.

The layout of the test points is shown in Figure 3.

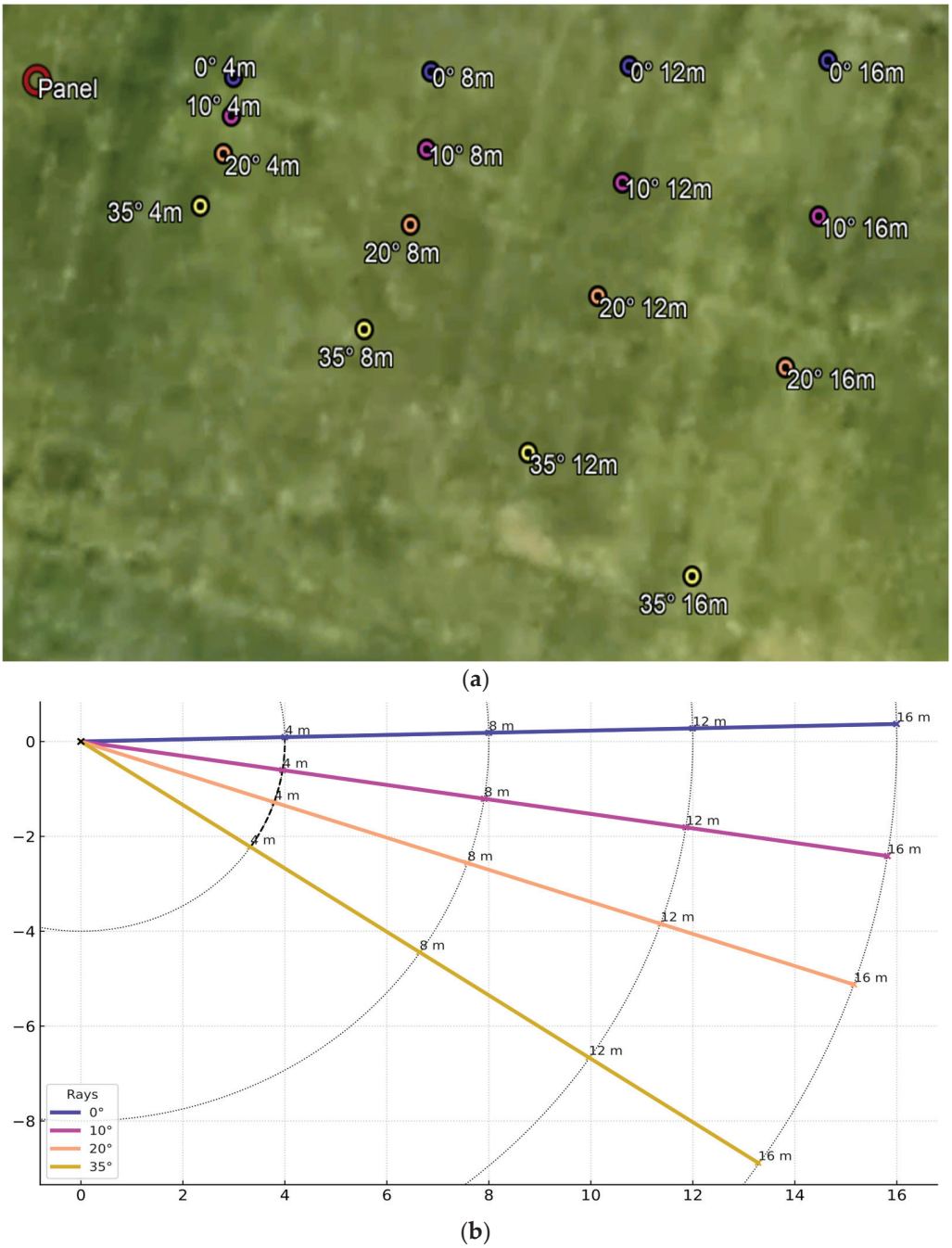


Figure 3. Test point positions on the map (**a**) and the criteria used to identify them (**b**).

For each of the 16 test points, a series of camera configurations were tested, mixing the high accuracy and high density mode with its laser emitter enabled or disabled. These are presets available in the Intel RealSense D4xx firmware, where high accuracy aims to provide better results for fewer pixels, while high density tends to provide the highest number of depth estimations in the frame; the laser emitter, instead, aims to support depth estimations in the short range to cover poor light conditions. The combinations that were tested, therefore, are as follows:

- High accuracy mode, laser emitter on;
- High accuracy mode, laser emitter off;
- High density mode, laser emitter on;

- High density mode, laser emitter off.

This experimental configuration leads to 64 diverse tests that had to be performed for each light condition scenario. To cover both good and poor light conditions, these tests were performed on a sunny day in September 2025 at 12:00 pm with approximately 64,000 lux of environmental light and on a cloudy day, still in September 2025, approximately at sunset (7:00 pm) with 14,000 lux of environmental light. Hence, these configurations resulted in 128 different tests (16 points, with 4 camera presets and in 2 different scenarios). The acquisition was performed by making the camera warm up for at least 10 min, while in the meantime verifying the proper inclination of the gray panel. As each test started, a .bag file was generated by running RGB and 2D depth captures for at least 5 s from the RealSense Viewer application for Windows 11 version 2.56.5. Frame resolution was set to 1280 × 720 pixels for both color and depth frames, each at 30 frames per second of acquisition rate.

2.3. Data Processing and Model Fitting

Data generated by .bag files was analyzed to identify depth parameters that could be useful for the calibration of the error on each test. The algorithm worked according to this flow of processing:

- Manual selection of the panel area in the frames. A python code reads all the bag files that are found in the folder in which it is located;
- The algorithm processes information such as number of pixels, hole rate of invalid depth values in the panel area, average depth, standard deviation of depth, number of frames and intrinsics;
- A comma separated value (csv) file is generated with all the metrics, along with a JSON (JavaScript Object Notation) file that stores data on the bounding boxes that have been defined on the panel area for each .bag file.

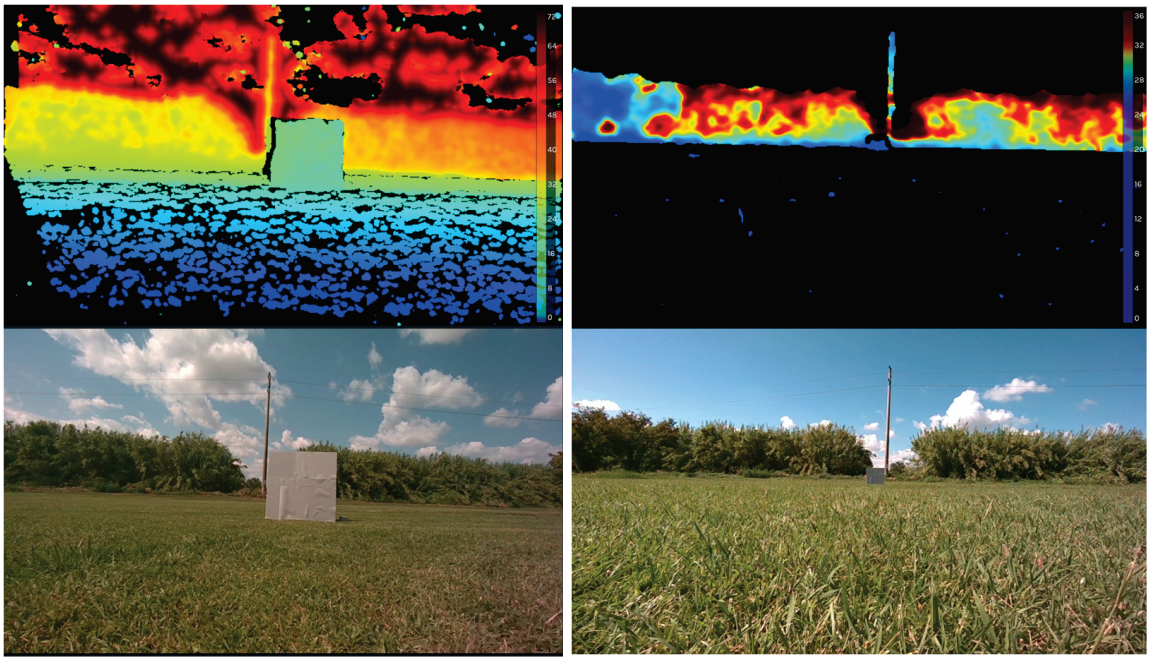
After having moved all the measurements from .bag files into a .csv file, the authors processed data to obtain features related to the error of distance compared with ground truth. Given the angle, laser emitter condition, high accuracy/high density settings, a ridge regression (or Tikhonov regularization) model, provided by the package scikit-learn of Python 3.11, was performed on the aforementioned Equation (6) to obtain weights for each scenario. The predicted bias provides corrected depth values according to Equation (6) and generated the following:

- A comparison of errors, for the same environmental and setting conditions, across distances for each test;
- Plots with approximate surfaces of areas sharing the same error according to distance and angle;
- A representation of the raw distance data with ideal condition (ground truth) and with the corrected values generated by the distance error model.

Lastly, data was saved regarding Root Mean Square Error (RMSE), Mean Squared Error (MAE) and weights of the regression model into another JSON file for subsequent usage in the field. The expected outcome of the tests is to verify error differences among test points and light conditions, with the additional feature of verifying the effect of the presence of the laser emitter to support depth readings, resulting in the possibility of analyzing each scenario separately (16 tests for 8 scenarios) or evaluating the possibility of merging the laser emitter on/off scenarios together (32 tests for 4 scenarios) to make a more robust regression model if the effects of the laser emitter in open fields are negligible.

3. Results

For each test, the authors had the possibility of identifying the test panel in the log file which presented both color and depth frames. Since data has been collected at different distances, the size of the panel area changed in pixel size across tests, leading to smaller detection boxes as the distance increased. The impact of such effects is also visible in far objects that fall beyond the maximum detection range of the camera, since they will most likely be identified as being located at the maximum available distance that the RGB-D sensor can provide, which is usually 64 m. An example of data visualization is shown in Figure 4.



(**a**) (**b**)

Figure 4. Example of test frames at close distance, high density preset (**a**) and far distance, high precision preset (**b**). Colors indicate depth from blue (close) to red (far), while black indicates no distance assessments.

Figure 4 already shows some of the classic effects of the two presets, high accuracy (HA) and high density (HD), since they provide a tradeoff between precision in depth measurements and the quantity of available measurements on each frame. For instance, the high density preset will provide a high quantity of data even on the ground around the target panel and of far elements, while the high precision mode will not provide much information on the ground surface while also reducing the number of measurements that will be available in the target area. Despite this not being an issue in close-range measurements, it becomes critical at long ranges, since the area of the target will be smaller and therefore will likely contain a limited number of valid values.

Results are therefore presented as depth estimations on each of the 128 tests, divided into scenarios, plus the error model estimation based on light conditions, distance and angle. Data for each scenario will also be analyzed afterwards to determine the effects of the laser emitter that is a built-in feature of the camera, which is a useful tool for indoor distance measurements but which might not be very effective in outdoor daylight conditions, given the distance of the targets and the presence of strong external lights, compared with indoor conditions.

3.1. Depth Distance Estimations

The outcome of field tests is a wide set of data divided into light conditions, panel angle position and distance from the camera. For each scenario, results are presented for the four presets. Tests on depth estimation in daylight with the laser off are shown in Table 1, where the data is shown with mean and median distances, standard deviation on the plane surface of the 2D panel, the hole rate and number of valid pixels, which also considers possible abnormal depth values which have been discarded.

Table 1. Depth distance results in daylight conditions with laser emitter off. HA and HD stand for high accuracy and high density presets.

Distance	Position	Preset	Frames	Mean Distance	Median Distance	Plane Std Dev	Hole Rate	Valid Pixels
4 m	0°	HA	293	3.993	3.989	0.048	37.73%	62.27%
8 m	0°	HA	232	7.728	7.723	0.141	35.12%	64.88%
12 m	0°	HA	180	11.846	11.382	0.107	67.56%	32.44%
16 m	0°	HA	144	16.581	15.042	0.137	55.60%	44.40%
4 m	0°	HD	294	4.005	3.984	0.145	6.59%	93.41%
8 m	0°	HD	203	7.993	7.718	0.216	11.26%	88.74%
12 m	0°	HD	241	13.221	11.514	0.181	9.50%	90.50%
16 m	0°	HD	217	17.457	15.434	0.175	3.97%	96.03%
4 m	10°	HA	268	3.894	3.910	0.035	95.22%	4.78%
8 m	10°	HA	300	7.646	7.617	0.092	66.86%	33.14%
12 m	10°	HA	277	11.268	11.236	0.106	61.16%	38.84%
16 m	10°	HA	247	14.459	14.372	0.096	47.18%	52.82%
4 m	10°	HD	291	4.374	3.909	0.169	75.43%	24.57%
8 m	10°	HD	283	9.472	8.426	0.170	37.61%	62.39%
12 m	10°	HD	271	11.609	11.296	0.153	10.32%	89.68%
16 m	10°	HD	219	17.843	15.088	0.152	5.91%	94.09%
4 m	20°	HA	300	3.205	3.231	0.018	99.57%	0.43%
8 m	20°	HA	300	7.189	7.221	0.031	96.62%	3.38%
12 m	20°	HA	300	10.573	10.573	0.038	94.48%	5.52%
16 m	20°	HA	276	15.137	15.136	0.044	94.00%	6.00%
4 m	20°	HD	300	3.837	3.459	0.180	81.74%	18.26%
8 m	20°	HD	300	8.405	7.314	0.144	62.13%	37.87%
12 m	20°	HD	295	14.982	12.826	0.153	41.37%	58.63%
16 m	20°	HD	300	19.256	17.286	0.128	37.36%	62.64%
4 m	35°	HA	195	3.304	3.296	0.047	72.84%	27.16%
8 m	35°	HA	206	6.955	6.957	0.117	34.66%	65.34%
12 m	35°	HA	290	9.897	9.812	0.066	85.28%	14.72%
16 m	35°	HA	300	14.001	13.856	0.044	90.80%	9.20%
4 m	35°	HD	239	3.354	3.311	0.169	37.96%	62.04%
8 m	35°	HD	300	7.002	6.951	0.173	7.38%	92.62%
12 m	35°	HD	300	14.932	10.862	0.228	34.80%	65.20%
16 m	35°	HD	289	14.345	14.294	0.181	20.99%	79.01%

Daylight test conditions without laser emitter support showed a progressively greater error as both the distances and angle of the panel position in the frame rise. In addition, the percentage of valid pixels (hole rate) changed significantly across the high accuracy and high density presets: despite the fact that lowered hole rate values were expected from the high density mode, the number of valid pixels in the plane area around the target in the frames was lower than expected.

Tests have also been run to compare outcomes with the laser emitter on and off, both in order to evaluate possible depth assessment improvements and to increase valid pixels. The results of these daylight tests that have been run with the laser on are shown in Table 2.

Table 2. Depth distance results in daylight conditions with laser emitter on.

Distance	Position	Preset	Frames	Mean Distance	Median Distance	Plane Std Dev	Hole Rate	Valid Pixels
4 m	0°	HA	300	3.994	3.990	0.049	38.05%	61.95%
8 m	0°	HA	184	7.717	7.715	0.144	36.01%	63.99%
12 m	0°	HA	195	11.524	11.454	0.136	39.91%	60.09%
16 m	0°	HA	206	15.818	15.078	0.140	54.81%	45.19%
4 m	0°	HD	249	4.052	3.983	0.242	7.41%	92.59%
8 m	0°	HD	191	7.900	7.714	0.212	8.60%	91.40%
12 m	0°	HD	169	13.130	11.457	0.194	7.89%	92.11%
16 m	0°	HD	175	17.700	15.555	0.169	4.11%	95.89%
4 m	10°	HA	234	3.897	3.914	0.039	95.28%	4.72%
8 m	10°	HA	263	7.650	7.611	0.100	60.93%	39.07%
12 m	10°	HA	212	11.165	11.175	0.100	57.42%	42.58%
16 m	10°	HA	249	14.774	14.528	0.108	53.11%	46.89%
4 m	10°	HD	300	4.550	3.908	0.174	75.61%	24.39%
8 m	10°	HD	276	8.654	8.419	0.172	44.14%	55.86%
12 m	10°	HD	271	11.519	11.261	0.155	9.84%	90.16%
16 m	10°	HD	212	17.850	15.059	0.148	6.19%	93.81%
4 m	20°	HA	300	3.324	3.402	0.023	99.02%	0.98%
8 m	20°	HA	272	7.151	7.165	0.028	97.43%	2.57%
12 m	20°	HA	270	10.517	10.554	0.037	92.45%	7.55%
16 m	20°	HA	262	20.547	19.924	0.078	94.55%	5.45%
4 m	20°	HD	300	3.896	3.484	0.177	82.26%	17.74%
8 m	20°	HD	300	8.822	7.499	0.149	62.61%	37.39%
12 m	20°	HD	300	13.596	12.846	0.137	40.50%	59.50%
16 m	20°	HD	281	19.235	17.265	0.144	33.75%	66.25%
4 m	35°	HA	187	3.308	3.300	0.046	74.05%	25.95%
8 m	35°	HA	300	6.957	6.954	0.112	37.24%	62.76%
12 m	35°	HA	266	9.887	9.845	0.101	70.29%	29.71%
16 m	35°	HA	300	13.958	13.561	0.050	89.93%	10.07%
4 m	35°	HD	208	3.353	3.308	0.171	40.76%	59.24%
8 m	35°	HD	271	6.983	6.957	0.170	6.86%	93.14%
12 m	35°	HD	300	13.397	10.390	0.207	36.50%	63.50%
16 m	35°	HD	300	14.873	14.565	0.187	19.34%	80.66%

The same tests have been performed at sunset, both with the laser emitter off and on, as performed in daylight conditions. The results are shown in Tables 3 and 4, where similar patterns observed for daylight have been confirmed in that environmental condition as well.

Distance estimations, as tested in sunset light conditions, started to be less precise as light is reduced; the absence of the laser emitter might be responsible for a decrease in precision at close distances but the assessments at far distances started to be significant as the angle of the target panel from the camera increased. It should be noted, however, that even the presence of the active laser emitter at maximum power did not change the result or improve depth readings, as was already observed in previous daylight tests with the laser emitter on. This is shown in the following table, which reports the last run of 32 tests in the given field positions from the target panel. The combination of the high accuracy preset and the reduced presence of external light played a key role in generating an even

lower percentage of valid pixels in the target area across frames if compared with previous daylight tests.

For each test, further analyses were performed to investigate the influence of laser emitter conditions on depth readings, given that in outdoor environments the effect of this camera enhancement could have been limited. This part of the method is meant to verify if there are significant differences among the data and decide whether to perform error regression model estimations separately or not. The results are shown in Table 5, as mean results of the differences in the two corresponding readings for each test point.

No significant differences have been highlighted among the use of the laser emitter in the explored field tests; therefore, the upcoming analyses of the precision of the depth sensor at each condition will be explored as datasets of 32 readings (four distances at four angles each) in four scenarios, which are daylight with high accuracy and high density presets plus sunset scenario with high accuracy and high density presets.

Table 3. Depth distance results in sunset conditions with laser emitter off.

Distance	Position	Preset	Frames	Mean Distance	Median Distance	Plane Std Dev	Hole Rate	Valid Pixels
4 m	0°	HA	300	3.664	3.667	0.037	99.32%	0.68%
8 m	0°	HA	300	7.692	7.583	0.036	98.46%	1.54%
12 m	0°	HA	300	10.990	10.913	0.048	95.16%	4.84%
16 m	0°	HA	300	17.890	17.905	0.057	91.38%	8.62%
4 m	0°	HD	300	3.728	3.695	0.205	68.40%	31.60%
8 m	0°	HD	300	8.312	7.711	0.203	39.59%	60.41%
12 m	0°	HD	300	13.932	12.642	0.197	28.96%	71.04%
16 m	0°	HD	300	17.632	17.754	0.186	18.25%	81.75%
4 m	10°	HA	300	3.627	3.639	0.026	98.41%	1.59%
8 m	10°	HA	274	7.509	7.489	0.029	97.77%	2.23%
12 m	10°	HA	281	13.398	13.393	0.039	98.55%	1.45%
16 m	10°	HA	300	18.516	18.425	0.048	98.12%	1.88%
4 m	10°	HD	300	3.900	3.696	0.202	53.95%	46.05%
8 m	10°	HD	300	8.649	7.564	0.167	50.61%	49.39%
12 m	10°	HD	277	12.849	12.903	0.180	47.36%	52.64%
16 m	10°	HD	300	18.761	17.936	0.192	29.15%	70.85%
4 m	20°	HA	233	4.502	4.464	0.065	97.94%	2.06%
8 m	20°	HA	300	8.720	8.713	0.031	97.44%	2.56%
12 m	20°	HA	300	13.078	13.979	0.019	99.59%	0.41%
16 m	20°	HA	300	18.392	18.353	0.074	99.16%	0.84%
4 m	20°	HD	298	4.535	4.451	0.204	51.85%	48.15%
8 m	20°	HD	300	8.551	8.457	0.175	44.95%	55.05%
12 m	20°	HD	300	16.312	13.217	0.190	26.07%	73.94%
16 m	20°	HD	251	18.578	18.575	0.142	22.44%	77.56%
4 m	35°	HA	300	3.067	3.046	0.017	96.41%	3.59%
8 m	35°	HA	283	6.295	6.296	0.065	89.21%	10.79%
12 m	35°	HA	300	9.123	9.131	0.037	90.83%	9.17%
16 m	35°	HA	282	12.588	12.605	0.061	76.99%	23.01%
4 m	35°	HD	300	3.193	3.170	0.108	44.66%	55.34%
8 m	35°	HD	300	6.548	6.391	0.155	22.65%	77.35%
12 m	35°	HD	300	10.816	9.505	0.163	38.71%	61.29%
16 m	35°	HD	300	12.679	12.839	0.139	7.78%	92.22%

Table 4. Depth distance results in sunset conditions with laser emitter on.

Distance	Position	Preset	Frames	Mean Distance	Median Distance	Plane Std Dev	Hole Rate	Valid Pixels
4 m	0°	HA	300	3.644	3.647	0.035	99.26%	0.74%
8 m	0°	HA	300	7.674	7.661	0.051	97.96%	2.04%
12 m	0°	HA	300	11.038	10.979	0.047	94.92%	5.08%
16 m	0°	HA	300	17.945	17.887	0.052	95.22%	4.78%
4 m	0°	HD	300	3.710	3.686	0.210	68.47%	31.53%
8 m	0°	HD	300	8.691	7.701	0.213	40.24%	59.76%
12 m	0°	HD	300	13.721	12.641	0.192	28.38%	71.62%
16 m	0°	HD	300	17.588	17.684	0.180	16.49%	83.51%
4 m	10°	HA	298	3.633	3.632	0.037	97.55%	2.45%
8 m	10°	HA	300	7.707	7.577	0.035	97.00%	3.00%
12 m	10°	HA	300	13.348	13.326	0.030	98.81%	1.19%
16 m	10°	HA	300	18.252	18.120	0.054	95.82%	4.18%
4 m	10°	HD	300	3.897	3.698	0.205	54.77%	45.23%
8 m	10°	HD	300	8.696	7.533	0.168	52.84%	47.16%
12 m	10°	HD	291	12.812	12.930	0.161	45.28%	54.72%
16 m	10°	HD	300	18.843	17.645	0.182	29.03%	70.97%
4 m	20°	HA	300	4.539	4.419	0.033	98.11%	1.89%
8 m	20°	HA	300	8.965	8.754	0.036	97.44%	2.56%
12 m	20°	HA	284	13.895	13.866	0.035	99.46%	0.54%
16 m	20°	HA	223	18.469	18.575	0.097	99.11%	0.89%
4 m	20°	HD	267	4.515	4.454	0.204	52.25%	47.75%
8 m	20°	HD	300	8.435	8.434	0.172	44.56%	55.44%
12 m	20°	HD	280	15.576	13.025	0.182	27.50%	72.50%
16 m	20°	HD	300	18.986	18.715	0.156	22.07%	77.93%
4 m	35°	HA	300	3.098	3.095	0.015	97.52%	2.48%
8 m	35°	HA	300	6.282	6.283	0.063	90.69%	9.31%
12 m	35°	HA	300	9.119	9.132	0.030	91.73%	8.27%
16 m	35°	HA	279	12.613	12.612	0.062	77.38%	22.62%
4 m	35°	HD	300	3.192	3.174	0.109	46.96%	53.04%
8 m	35°	HD	296	6.698	6.397	0.157	23.54%	76.46%
12 m	35°	HD	265	10.843	9.528	0.150	40.99%	59.01%
16 m	35°	HD	296	12.690	12.718	0.149	8.80%	91.20%

Table 5. Mean difference in measurements among light conditions with laser emitter on and off.

Light Condition	Δ Distance (Mean)	Δ Distance (Median)	Δ Plane Std Dev	Δ Hole Rate	Δ Valid Pixels
Daylight	−0.060	−0.151	−0.007	0.013	−0.013
Sunset	−0.034	0.018	0.000	−0.003	0.003

3.2. Distance Error Comparison Among Scenarios

Test data has been compared across scenarios to verify how the error changed as the position of the panel went towards the edge of the frame. An important feature that needed to be assessed was the tendency of the error to vary as the distance of the panel from the camera increased. The results have hence been divided into each one of the four scenarios and depth estimation errors are shown in Figure 5.

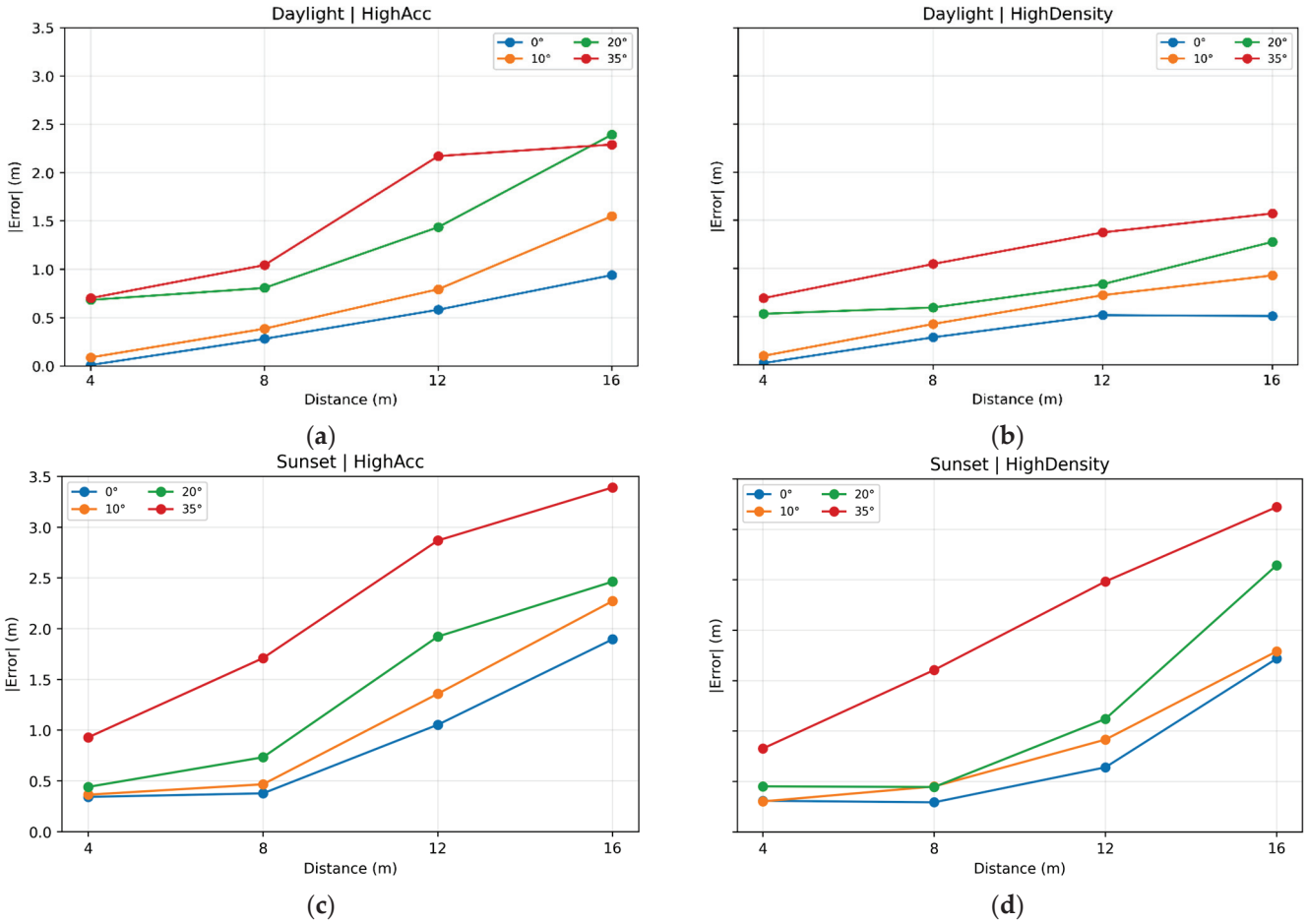


Figure 5. Error variability in each scenario: daylight and high accuracy preset (**a**), daylight and high density preset (**b**), sunset and high accuracy preset (**c**), sunset and high density preset (**d**), divided by positioning angle of the panel.

In each scenario, the tendency of the error appears to linearly increase after 8 m, which is the limit of the expected functionality of the camera model to keep most of the errors below 0.5 or 1 m; beyond that threshold, errors rise and reach up to 3.5 m of error in the case of poor visibility or far distances, such as 16 m, which is twice the optimal range of the camera.

3.3. Ridge Regression Model for Distance Errors

The 32 tests for each of the four scenarios have been employed to run a ridge regression model to estimate the weights of Equation (6) that are necessary for the error prediction model. This aspect can show only slight differences in some weights, while it could have greater differences in others, such as those that refer to parameters that depend on the angle of the target. The result of the regression model on such weights for each scenario is therefore shown in Table 6, where RMSE and MAE are computed against the ground truth distances for each condition (daylight and sunset) and preset (HA and HD).

From the regression model, it is possible to see a high variability between daylight and sunset conditions, plus the expected change in the weights that affect the angle of the target. More specifically, despite RMSE and MAE remaining stable, the model shows a major impact of the W_3 and W_4 weights associated with the angle of the panel in the frame. Other parameters, such as W_5, associated with the variability given by both distance and angle, remain stable within light conditions in both the high accuracy and high density presets.

Table 6. Weights and regression errors for each scenario.

Light	Preset	W_0	W_1	W_2	W_3	W_4	W_5	RMSE	MAE
Daylight	HA	2.543	−0.714	0.035	−0.837	−1.630	0.054	0.635	0.509
	HD	0.484	−0.217	0.010	2.606	−5.857	0.054	0.541	0.459
Sunset	HA	−0.493	−0.211	0.019	12.683	−17.073	−0.358	0.550	0.491
	HD	−0.278	−0.179	0.017	9.834	−13.237	−0.358	0.461	0.401

3.4. Depth Error Estimation Based on Angle and Distance

The depth estimation model was employed to determine the depth error based on the angle and distance from the panel. The error can be represented by a surface on which it remains constant. Figure 6 shows the performance of the model in estimating the error in the four scenarios.

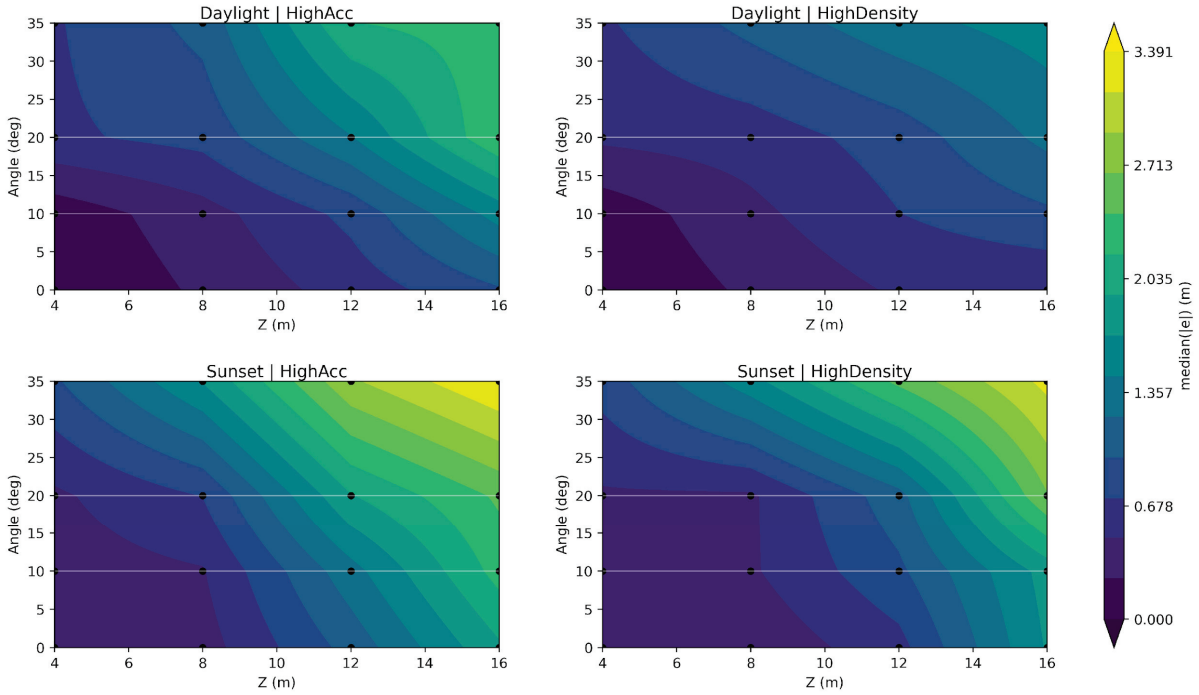


Figure 6. Depth error estimation according to the regression model for each scenario.

The iso-error surfaces in the previous figure show a consistent pattern across scenarios: the error, in fact, grows with range as expected but also intensifies as the target moves off-axis, producing steeper contours toward the edges of the FoV. Under daylight conditions, high density increases spatial coverage (i.e., more usable pixels), but of course it does not prevent angular bias; on the other hand, high accuracy yields slightly tighter surfaces in the central region while showing comparable off-axis behavior. In sunset light conditions, the angle-driven component becomes more prominent, with wider areas showing a higher degree of error, confirming that illumination modulates how strongly viewing geometry can generate an impact on depth stability. The plot also provides an intuitive representation of possible safety-critical zones in which the depth correction model can have a greater impact.

To better illustrate how this information can be translated into field use, it can be useful to compare raw depth readings with bias-corrected values from the regression model, as shown in Figure 7.

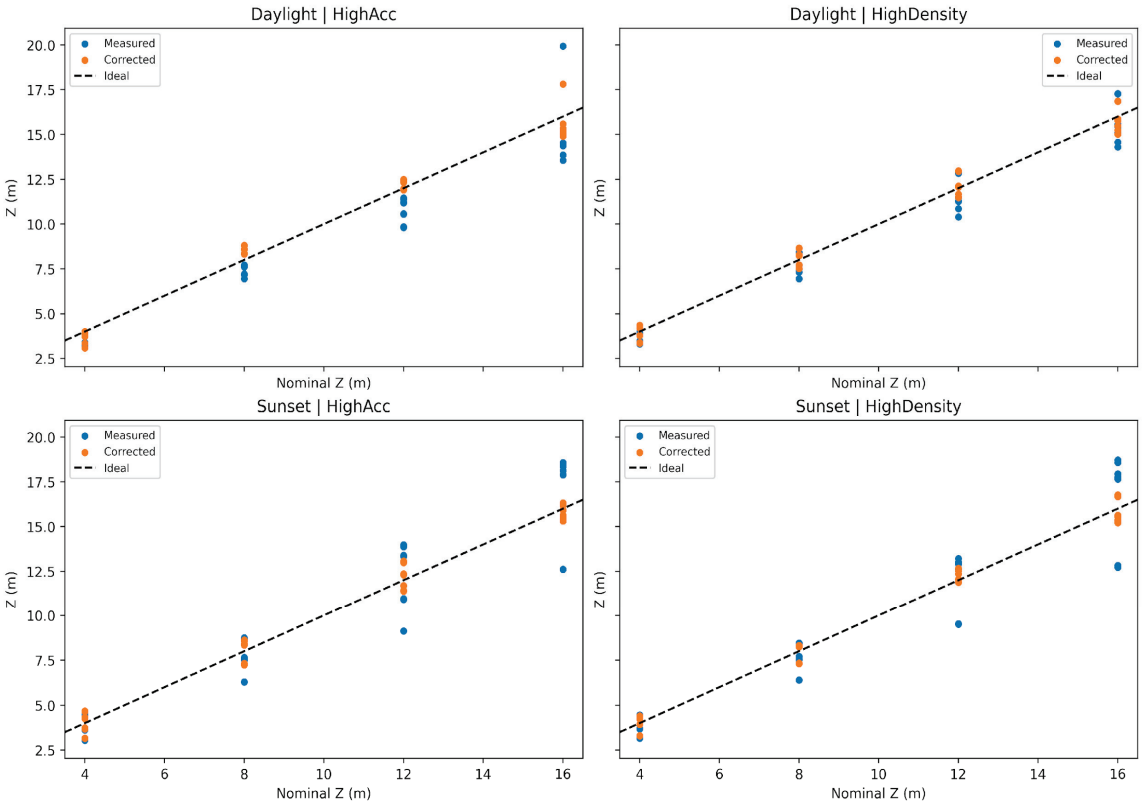


Figure 7. Comparison between raw depth estimation and model's correction.

The compact ridge regression model captured the observed behavior with stable errors across scenarios. Terms linked to viewing angles emerged as especially influential under dimmer light, while interaction terms between range and angle helped explain the rapid growth of bias off-axis. The correction attenuates the systematic trend visible in the raw data, especially when overestimation or underestimation accumulates off-axis, bringing the observed distances closer to the nominal layout.

4. Discussion

For safety applications on agricultural machinery, the findings shown in this research translate into concrete design and integration guidance for both autonomous and manned vehicles, with the latter as an alert device or as a tool that provides extra visibility in blind spots. Detection zones that matter for braking and collision avoidance should be placed as close as possible to the optical axis, and system logic should assume greater uncertainty as targets move laterally or fall near the practical range limit. When illumination drops, planners should increase error margins or require other detection methods, since the angular component of the error becomes more prominent. Preset selection can be tuned to prioritize coverage when needed, but it will not replace geometry-aware handling of uncertainty. Embedding the proposed bias model in the decision stack allows distance-based rules to operate on corrected values or to adapt thresholds dynamically when conditions predict an increased risk of false negatives.

A consideration on the outcome of the trials is that the outer FoV sectors consistently exhibit larger errors, especially at medium-to-long ranges, suggesting that relying on peripheral areas for safety-critical operations should be avoided and, where possible, the camera should be configured so that the typical region of interest falls within the central area of FoV or at least should not exceed the 35° angle. For machines requiring wider coverage, dual cameras can be used to mutually cover edge regions, reducing the chances of abnormal

distance assessments. On the software side, conservative gating can improve safety buffers, especially for peripheral areas of FoV at long range while keeping fairly good distance assessments in the central sector and at short range. Moreover, since daylight reduces the laser emitter effects on measurements, these recommendations can be considered to be applicable independently of the presence of any sort of laser projector on the camera.

The extremely low outdoor impact of the onboard emitter suggests that, in sunlight and against vegetation, passive stereo remains the dominant mechanism, with limited headroom for active assistance at the tested ranges. This supports a broader strategy that combines model-based correction with operational constraints and, where feasible, fusing complementary sensors to stabilize perception under challenging illumination. It can also be stated that, from a practical point of view, the developed model enables two complementary strategies, namely, applying the corrected distances from obstacles directly in motion planning (i) and alerting or inflating risk thresholds according to the degree of depth measurements' uncertainty (ii). In both cases, the model can transform static readings driven by fixed presets into a geometry-aware and light-aware depth perception tool.

The limitations of this work are related to the use of a single device class, a controlled yet realistic background, and quasi-static scenes. Studies should, in future, repeat the same scientific protocol to broaden scene diversity, including articulated targets and dynamics to compare across stereo baselines and sensing technologies. Extending the protocol to moving platforms and publishing harmonized datasets would enable fair cross-device benchmarking and help translate model corrections into standardized safety envelopes usable across equipment and environments. Even within these bounds, the proposed approach offers a practical path to quantify, visualize and mitigate condition-dependent depth errors where it matters most: in the field, under real light, with actionable consequences for machine safety.

5. Conclusions

This study demonstrates that outdoor distance estimation with a stereo RGB-D camera is strongly conditioned by where an object sits in the FoV and how the scene is illuminated, with central regions yielding the most stable results and edge placement introducing systematic bias. This aspect is critical for the use of RGB-D cameras as part of safety devices or safety systems in working environments that are characterized by outdoor harsh weather conditions and by non-standardized obstacles that are typical, for instance, of agricultural contexts. The camera's preset selection influences the balance between coverage and precision, yet does not remove the angular dependence of error, while the active projector offers little benefit at typical working ranges in outdoor applications. Instead, a compact and geometry-informed regression can capture these trends and enable bias-aware correction, turning raw depth into distances that align more faithfully with the physical scene, therefore allowing us to determine a better distance estimation based on environmental conditions.

Taken together, these findings argue for safety pipelines that explicitly manage field-of-view placement, monitor validity density and apply bias surfaces during risk evaluation rather than treating depth as uniformly reliable across the image. The proposed protocol and modeling workflow are readily transferable, providing a practical basis for acceptance testing, sensor placement and periodic recalibration in open-air agricultural settings. Embedding these steps into system integration can reduce optimistic distance readings in critical zones and support more dependable human–machine separation, with the purpose of creating a standardized and replicable practice across platforms and sites and allowing the possibility for future works to perform cross-model comparisons as well as testing

additional challenging environmental scenarios such as night activities, operations under canopies or weather conditions such as rain and fog.

Author Contributions: Conceptualization, L.V., D.G. and D.P.; methodology, P.R., E.C. and M.C.; software, P.R.; validation, E.C. and D.M.; formal analysis, E.C.; investigation, P.R. and E.C.; resources, E.C. and P.R.; data curation, E.C. and P.R.; writing—original draft preparation, P.R. and E.C.; writing—review and editing, E.C.; visualization, L.V., D.G. and D.P.; supervision, D.M. and L.V.; project administration, D.M.; funding acquisition, D.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research is funded by the Italian National Institute for Insurance against Accidents at Work (INAIL) project called "Obstacle detection and tracking system for fixed and moving obstacles in agriculture" (Bando di Ricerca in Collaborazione BRiC 2022 ID04 SIRTRAck), CUP ID J83C23000020001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Raw data (.bag files) from which datasets presented in this study have been derived are available on request due to their size.

Acknowledgments: This research is part of the activities included in the "Rome Technopole" Project (ECS 0000024), which is part of the Next Generation EU programme, Italy's "Piano Nazionale di Ripresa e Resilienza (PNRR)", mission 4, component 2, investment no. 1.5.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FoV	Field of View
HA	High Accuracy
HD	High Density
RGB-D	Red, Green, Blue, Depth
RMSE	Root Mean Squared Error
MAE	Mean Squared Error
ISO	International Standard Organization
CSV	Comma Separate Values
JSON	JavaScript Object Notation
2D	Two-dimensional

References

1. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004. [CrossRef]
2. Birchfield, S.; Tomasi, C. Depth Discontinuities by Pixel-to-Pixel Stereo. *Int. J. Comput. Vis.* **1999**, *35*, 269–293. [CrossRef]
3. Rançon, F.; Keresztes, B.; Deshayes, A.; Tardif, M.; Abdelghafour, F.; Fontaine, G.; Da Costa, J.-P.; Germain, C. Designing a Proximal Sensing Camera Acquisition System for Vineyard Applications: Results and Feedback on 8 Years of Experiments. *Sensors* **2023**, *23*, 847. [CrossRef]
4. Park, J.-M.; Lee, J.-W. Improved Stereo Matching Accuracy Based on Selective Backpropagation and Extended Cost Volume. *Int. J. Control. Autom. Syst.* **2022**, *20*, 2043–2053. [CrossRef]
5. Herrera, D.; Kannala, J.; Heikkilä, J. Joint Depth and Color Camera Calibration with Distortion Correction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2058–2064. [CrossRef]
6. Keselman, L.; Woodfill, J.I.; Grunnet-Jepsen, A.; Bhowmik, A. Intel RealSense Stereoscopic Depth Cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1–10. [CrossRef]

7. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361. [CrossRef]
8. Vit, A.; Shani, G. Comparing RGB-D Sensors for Close-Range Outdoor Agricultural Phenotyping. *Sensors* **2018**, *18*, 4413. [CrossRef]
9. Birchfield, S.; Tomasi, C. A Pixel Dissimilarity Measure that is Insensitive to Image Sampling. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 401–406. [CrossRef]
10. Skoczeń, M.; Ochman, M.; Spyra, K.; Nikodem, M.; Krata, D.; Panek, M.; Pawłowski, A. Obstacle detection system for agricultural mobile robot application using rgb-d cameras. *Sensors* **2021**, *21*, 5292. [CrossRef]
11. Hübner, P.; Hou, J.; Iwaszczuk, D. Evaluation of Intel RealSense D455 Camera Depth Estimation for Indoor SLAM Applications. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *48*, 1207–1214. [CrossRef]
12. Servi, M.; Profili, A.; Furferi, R.; Volpe, Y. Comparative Evaluation of Intel RealSense D415, D435i, D455 and Microsoft Azure Kinect DK Sensors for 3D Vision Applications. *IEEE Access* **2024**, *12*, 115919–115932. [CrossRef]
13. Upadhyay, A.; Zhang, Y.; Koparan, C.; Rai, N.; Howatt, K.; Bajwa, S.; Sun, X. Advances in Ground Robotic Technologies for Site-Specific Weed Management in Precision Agriculture: A Review. *Comput. Electron. Agric.* **2024**, *225*, 109363. [CrossRef]
14. Muñoz-Bañón, M.; Candelas, F.A.; Torres, F. Targetless Camera–LiDAR Calibration in Unstructured Environments. *IEEE Access* **2020**, *8*, 143692–143705. [CrossRef]
15. Aby, G.R.; Issa, S.F. Safety of Automated Agricultural Machineries: A Systematic Literature Review. *Safety* **2023**, *9*, 13. [CrossRef]
16. Wang, T.; Chen, B.; Zhang, Z.; Li, H.; Zhang, M. Applications of Machine Vision in Agricultural Robot Navigation: A Review. *Comput. Electron. Agric.* **2022**, *198*, 107085. [CrossRef]
17. Kurtser, P.; Lowry, S. RGB-D Datasets for Robotic Perception in Site-Specific Agricultural Operations–A Survey. *Comput. Electron. Agric.* **2023**, *212*, 108035. [CrossRef]
18. Camplani, M.; Salgado, L. Depth–Color Fusion Strategy for 3-D Scene Modeling With Kinect. *IEEE Trans. Cybern.* **2013**, *43*, 1560–1571. [CrossRef]
19. Qiu, R.; Zhang, M.; He, Y. Field estimation of maize plant height at jointing stage using an RGB-D camera. *Crop. J.* **2022**, *10*, 1274–1283. [CrossRef]
20. Le Francois, E.; Griffiths, A.D.; McKendry, J.J.D.; Chen, H.; Li, D.D.-U.; Henderson, R.K.; Herrnsdorf, J.; Dawson, M.D.; Strain, M.J. Combining time of flight and photometric stereo imaging for 3D reconstruction of discontinuous scenes. *Opt. Lett.* **2021**, *46*, 3612–3615. [CrossRef]
21. Milella, A.; Marani, R.; Petitti, A.; Reina, G. In-field high throughput grapevine phenotyping with a consumer-grade depth camera. *Comput. Electron. Agric.* **2019**, *156*, 293–306. [CrossRef]
22. Zbontar, Ž.; LeCun, Y. Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. *J. Mach. Learn. Res.* **2016**, *17*, 1–32. Available online: http://jmlr.org/papers/v17/15-535.html (accessed on 4 December 2025).
23. Wan, H.; Ou, Y.; Guan, X.; Zhou, Z.; Luo, X. Review of the Perception Technologies for Unmanned Agricultural Machinery Operating Environment. *Trans. Chin. Soc. Agric. Eng.* **2024**, *40*, 1–18. [CrossRef]
24. *ISO 18497:2018*; Agricultural Machinery and Tractors—Safety of Highly Automated Agricultural Machines—Principles for Design. ISO: London, UK, 2018. Available online: https://www.iso.org/standard/62659.html (accessed on 4 December 2025).
25. Carfagni, M.; Furferi, R.; Governi, L.; Santarelli, C.; Servi, M.; Uccheddu, F.; Volpe, Y. Metrological and Critical Characterization of the Intel D415 Stereo Depth Camera. *Sensors* **2019**, *19*, 489. [CrossRef]
26. Qiu, Z.; Martínez-Sánchez, J.; Arias-Sánchez, P.; Rashdi, R. External Multi-Modal Imaging Sensor Calibration: A Review. *Inf. Fusion* **2023**, *101*, 806. [CrossRef]
27. Monarca, D.; Rossi, P.; Alemanno, R.; Cossio, F.; Nepa, P.; Motroni, A.; Gabbrielli, R.; Pirozzi, M.; Console, C.; Cecchini, M. Autonomous Vehicles Management in Agriculture with Bluetooth Low Energy (BLE) and Passive Radio Frequency Identification (RFID) for Obstacle Avoidance. *Sustainability* **2022**, *14*, 9393. [CrossRef]
28. Rossi, P.; Mangiavacchi, P.L.; Monarca, D.; Cecchini, M. Smart Machinery and Devices for the Reduction of Risks from Human-machine Interference: A review. In *International Conference on Safety, Health and Welfare in Agriculture and Agro-food Systems*; Springer International Publishing: Cham, Germany, 2020.
29. Lyu, X.; Liu, S.; Qiao, R.; Jiang, S.; Wang, Y. Camera, LiDAR, and IMU Spatiotemporal Calibration: Methodological Review and Research Perspectives. *Sensors* **2025**, *25*, 5409. [CrossRef]
30. *ISO 18497-1*; Agricultural Machinery and Tractors—Safety of Partially Automated, Semi-Autonomous and Autonomous Machinery Part 1: Machine Design Principles and Vocabulary. ISO: London, UK, 2024. Available online: https://www.iso.org/standard/82684.html (accessed on 4 December 2025).
31. *ISO 25119*; Tractors and Machinery for Agriculture and Forestry—Safety-Related Parts of Control Systems. ISO: London, UK, 2018. Available online: https://www.iso.org/standard/69025.html (accessed on 4 December 2025).

32. Pal, A.; Leite, A.C.; From, P.J. A Novel End-to-End Vision-Based Architecture for Agricultural Human–Robot Collaboration in Fruit Picking Operations. *Rob. Auton. Syst.* **2024**, *172*, 104567. [CrossRef]
33. Nebiker, S.; Meyer, J.; Blaser, S.; Ammann, M.; Rhyner, S. Outdoor Mobile Mapping and AI-Based 3D Object Detection with Low-Cost RGB-D Cameras: The Use Case of On-Street Parking Statistics. *Remote. Sens.* **2021**, *13*, 3099. [CrossRef]
34. Gnanamurthy, P.; Krishnan, P.; Gehlot, C.; Joshi, J. Evaluation of Intel® RealSense D455 for 3D Vision Applications. *IEEE Access* **2024**. Available online: https://ieeexplore.ieee.org/ (accessed on 4 December 2025).
35. Kytö, M.; Nuutinen, M.; Oittinen, P. Method for Measuring Stereo Camera Depth Accuracy Based on Stereoscopic Vision. In *Three-Dimensional Imaging, Interaction, and Measurement*; SPIE: Bellingham, WA, USA, 2011. [CrossRef]
36. Zaremba, A.; Nitkiewicz, S. Distance Estimation with a Stereo Camera and Accuracy Analysis. *Appl. Sci.* **2024**, *14*, 11444. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Sensors Editorial Office
E-mail: sensors@mdpi.com
www.mdpi.com/journal/sensors



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Guest Editor. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editor and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.





Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-9116-0