

## Article

# Multimodal Fusion and Dynamic Resource Optimization for Robust Cooperative Localization of Low-Cost UAVs

Hongfu Liu \*, Yajing Fu, Yangyang Ma and Wanpeng Zhang \*

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China

\* Correspondence: liuhongfu@nudt.edu.cn (H.L.); wpzhang@nudt.edu.cn (W.Z.)

## Highlights

### What are the main findings?

- Proposes a novel collaborative localization algorithm integrating a cross-modal attention mechanism to fuse vision, radar, and lidar data, significantly enhancing robustness in occluded and adverse weather conditions.
- Proposes a dynamic resource optimization framework using integer linear programming, enabling real-time allocation of computational and communication resources to prevent node overload and improve system efficiency.

### What are the implications of the main findings?

- Demonstrates superior performance in realistic simulations, significant improvements in positioning accuracy, resource efficiency, and fault recovery, demonstrating strong potential for applications in complex tasks.
- Provides a practical, low-cost system solution validated in complex scenarios, establishing a viable pathway for the engineering deployment of robust UAV swarms.

## Abstract

To overcome the challenges of low positioning accuracy and inefficient resource utilization in cooperative target localization by unmanned aerial vehicles (UAVs) in complex environments, this paper presents a cooperative localization algorithm that integrates multimodal data fusion with dynamic resource optimization. By leveraging a cross-modal attention mechanism, the algorithm effectively combines complementary information from visual, radar, and lidar sensors, thereby enhancing localization robustness under occlusions, poor illumination, and adverse weather conditions. Furthermore, a real-time resource scheduling model based on integer linear programming is introduced to dynamically allocate computational and communication resources, which mitigates node overload and minimizes resource waste. Experimental evaluations in scenarios including maritime search and rescue, urban occlusions, and dynamic resource fluctuations show that the proposed algorithm achieves significant improvements in positioning accuracy, resource efficiency, and fault recovery, demonstrating strong potential for applications in complex tasks, demonstrating its potential as a viable solution for low-cost UAV swarm applications in complex environments.

**Keywords:** multi-modal data fusion; dynamic resource optimization; cooperative target localization; unmanned aerial vehicles (UAVs); low-cost hardware



Academic Editor: Fei Gao

Received: 28 September 2025

Revised: 9 November 2025

Accepted: 19 November 2025

Published: 26 November 2025

**Citation:** Liu, H.; Fu, Y.; Ma, Y.; Zhang, W. Multimodal Fusion and Dynamic Resource Optimization for Robust Cooperative Localization of Low-Cost UAVs. *Drones* **2025**, *9*, 820. <https://doi.org/10.3390/drones9120820>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license

(<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the rapid development of unmanned systems and sensor technology, collaborative target localization using Unmanned Aerial Vehicles (UAVs) has become one of the core supporting technologies in critical fields such as search and rescue [1,2], marine monitoring [3,4], and urban security [5]. Single UAV systems have inherent limitations in terms of perception range, payload capacity, and environmental adaptability—for instance, single-machine visual sensors are prone to losing target features in complex outdoor occlusion scenarios, and radar sensors experience significant degradation in ranging accuracy in strong clutter environments at sea, making it difficult to meet high-precision localization demands in complex scenarios [4,6]. In contrast, multi-UAV collaborative localization, with its significant advantages of wide coverage, high system redundancy, and strong task adaptability, has gradually become a research hotspot in this field by overcoming the performance bottlenecks of single machines through complementary multi-node sensor data and coordinated resource scheduling [7,8].

In specific application areas, relevant research has achieved phased results: In search and rescue missions, Rudol et al. [2] proposed a multi-drone visual detector collaborative framework, which generates geographic location saliency maps by fusing detection results from multiple drones. This approach improved the efficiency of locating missing persons by over 40% in outdoor low-light and partially obstructed scenarios, effectively addressing the issue of insufficient robustness in single-drone visual localization. In the field of marine monitoring, Akram et al. [4] proposed a UAV-assisted Unmanned Surface Vehicle (USV) visual localization scheme to address the limitations of Global Navigation Satellite System (GNSS) signals in marine environments. Through visual data interaction between UAVs and USVs, they managed to keep the USV localization error within 1 m even when GPS signals were interfered with or obstructed by waves, successfully solving the technical challenge of GNSS signal susceptibility in marine scenarios. In low-cost cluster application scenarios, Li et al. [9] constructed a system based on Commercial Off-The-Shelf (COTS) drones, utilizing hierarchical control and Simultaneous Localization and Mapping (SLAM) technology. They achieved collaborative operations of a 10-node drone swarm under hardware conditions equipped only with Raspberry Pi edge computing units, providing a feasible path for the low-cost widespread application of swarm localization technology. Ultra-wideband technology has emerged as a powerful solution for positioning in GNSS-denied environments [10]. The aforementioned research and practical applications indicate that multi-drone collaborative localization has become an important technological direction for addressing complex task requirements.

However, the current multi-drone collaborative target localization technology still faces two core challenges that restrict its performance improvement and engineering implementation in complex scenarios:

Firstly, the construction and optimization of the multimodal data fusion mechanism are insufficient, affecting positioning accuracy in complex environments [11]. With the development of sensor technology, drone swarms can be equipped with various types of sensors, and the multimodal data can complement each other. However, existing fusion methods have not fully leveraged their advantages. For instance, the performance of visual detectors declines under video signal compression and transmission, as well as in complex operating conditions [12]. Although preliminary fusion of visual data and data links has been achieved, other modal data have not been integrated, leading to significant positioning deviations in complex scenarios, which makes it difficult to meet high-precision task requirements. Furthermore, traditional fusion methods rely on manually designed features, making it challenging to adapt to the heterogeneity and dynamics of multimodal data, thereby limiting the improvement of positioning performance [13].

Secondly, the dynamic resource optimization mechanism in multi-drone collaboration has defects, leading to low system efficiency and significant resource waste [14]. Multi-drone collaborative positioning requires coordinated scheduling of computational and communication resources, while drone hardware resources are limited and communication links are susceptible to interference, necessitating dynamic scheduling to match resources with task demands [15,16]. Existing research has preliminarily considered constraints on computational resources and communication bandwidth, but has not achieved real-time resource adjustments under dynamic task changes and environmental interference, resulting in potential computational overload or insufficient bandwidth when tasks or environments change. Current network adaptive frameworks lack the necessary dynamism and flexibility in resource allocation in multi-task switching scenarios. In practical applications, failure to dynamically allocate resources can lead to computational overload or idleness in certain nodes, or data transmission delays exceeding requirements, thereby restricting the overall collaborative efficiency of multi-drone systems.

To address the two core issues in multi-drone collaborative target localization mentioned above, this study aims to develop a collaborative localization algorithm that integrates multimodal data fusion and dynamic resource optimization. By leveraging the deep complementary fusion of multimodal data, the algorithm enhances localization accuracy in complex environments, while dynamic resource scheduling achieves a balance between system efficiency and resource utilization. The research findings hold significant theoretical significance and practical application value. The contributions of this paper are as follows:

- A three-level multimodal feature extraction network is proposed, incorporating a cross-modal attention (CMA) mechanism to address the poor adaptability of heterogeneous sensor data fusion. To tackle the differences in heterogeneous sensor data, a hierarchical feature processing architecture is designed: the modality adaptation layer standardizes the dimensions of radar, lidar data, and visual images through format conversion; the shared feature layer employs a switchable backbone network, introducing the CMA mechanism to enhance the complementarity among modalities; the task-specific layer adjusts modality weights based on scene requirements, resolving issues such as “information mismatch,” resulting in a 35–40% reduction in multimodal fusion localization errors compared to traditional methods.
- Establish a multi-stage collaborative verification mechanism for multiple drones to ensure data consistency and positioning accuracy. Design a comprehensive collaborative verification scheme: In the result association stage, image feature matching and Ultra-Wideband (UWB) ranging are used to determine whether the detection results point to the same target, eliminating mismatches; in the weighted fusion stage, the trace of the extended Kalman filter (EKF) covariance matrix is calculated to assess positioning reliability, generating weights to obtain the final coordinates of the target; in the anomaly removal stage, Grubbs’ test is employed to eliminate anomalous positioning results, and UWB technology is utilized to correct the drone’s attitude, ensuring spatial alignment of data among multiple drones. This mechanism reduces the mean absolute positioning error (mAPE) of collaborative positioning among multiple drones by 45% compared to the non-verification scheme, with an anomaly removal rate exceeding 98%.
- Design a dynamic resource optimization scheme to achieve real-time adaptation of models and tasks, ensuring continuity and efficiency in positioning. To address the issue of limited resources for drones, a hybrid offline–online closed-loop mechanism is constructed: in the offline phase, a model library is built; in the real-time phase, distributed sensing monitors hardware load and communication bandwidth, dynamically scheduling based on an Integer Linear Programming (ILP) model: switching

to lightweight models under high hardware load, enabling encoding and feature dimensionality reduction when bandwidth is insufficient, and migrating tasks in the event of node failures. This scheme improves resource utilization by 20–30% and reduces positioning interruption time by over 50% in node failure scenarios compared to traditional solutions.

## 2. Related Work

### 2.1. Cooperative Positioning Technology for Drones

Although continuous, steady, and precise positioning is crucial for enabling autonomous operation of robots and aircraft in complex environments [17]. The collaborative positioning technology for drones, as a key research direction in the field of multi-agent systems, achieves high-precision spatiotemporal localization of dynamic targets through efficient interaction of perception information among multiple nodes and deep collaboration of computational resources. Compared to single-machine positioning systems, this technology effectively overcomes inherent shortcomings such as limited perception range, insufficient environmental adaptability, and weak fault tolerance, providing core technical support for collaborative operations in complex scenarios [9]. Its technological development has evolved from distributed independent decision-making to centralized global optimization, and then to hybrid intelligent collaboration, reflecting the collaborative innovation and development of sensor technology, communication protocols, and intelligent algorithms.

#### 2.1.1. Single Drone Positioning System

As the fundamental unit of a collaborative positioning network, the performance of a standalone positioning system directly affects the reference accuracy of the collaborative system. Currently, standalone positioning technology has gradually expanded from single-sensor solutions to multi-source fusion solutions, with significant differences in adaptability among different schemes in complex scenarios, which directly impacts the underlying data quality of the collaborative positioning system.

In traditional standalone positioning schemes, the GNSS/IMU integrated navigation architecture utilizes GNSS to provide absolute position references, while the IMU performs trajectory estimation over short periods, forming a complementary positioning mode. However, in urban occlusion scenarios (such as densely built city canyon areas), GNSS signals are easily obstructed by buildings, leading to positioning interruptions. Multipath effects can cause GNSS positioning errors to reach several tens of meters [18], while the cumulative error of the IMU grows exponentially over time, typically exceeding 10% of the traveled distance within 60 s. In marine fog scenarios, GNSS signals are interfered with by atmospheric refraction, significantly reducing stability, and the cumulative error issue of the IMU is further exacerbated, making it difficult to meet the requirements for long-term continuous positioning [9].

The Visual SLAM-based positioning method constructs an environmental feature point cloud using a sequence of images captured by a camera, achieving self-pose estimation with centimeter to decimeter level positioning accuracy in GNSS-denied scenarios [19]. Mainstream algorithms such as ORB-SLAM3 enhance the robustness of positioning in dynamic environments by fusing visual features with IMU data [9]. However, in marine scenes with strong light reflections, the specular reflection from the water surface leads to a severe loss of image texture information, resulting in a feature matching success rate drop of over 70%. In unevenly lit environments such as urban shade or tunnels, the number of extracted feature points decreases by more than 50%, and the computational

complexity grows non-linearly with the map size, making it challenging to meet the real-time operational demands of resource-constrained micro-drone platforms [20].

In recent years, significant progress has been made in non-visual sensor positioning technology. Thermal imaging sensors utilize the temperature gradient difference between the target and the background, maintaining a target detection rate of over 85% even in low-light environments; however, their spatial resolution is relatively low (typically below  $640 \times 512$  pixels), resulting in positioning accuracy of only 2–3 m. Millimeter-wave radar has the capability to penetrate adverse weather conditions such as fog, rain, and dust, providing distance measurement accuracy on the order of 0.5 m. However, its angular resolution is usually below  $5^\circ$ , making precise positioning difficult in urban dense building areas or nearshore monitoring scenarios [16].

Current mainstream multi-source fusion single-machine solutions (such as vision + IMU + radar, LiDAR + millimeter-wave radar + data link) significantly enhance adaptability in complex environments by integrating the advantages of different sensors. In urban environments with complex occlusions, visual sensors provide target texture features for category recognition, IMUs can temporarily compensate for positioning deviations during GNSS signal interruptions, and radar can penetrate building obstructions to provide target distance information [21]. The positioning accuracy after fusion improves by 40–60% compared to single solutions, effectively alleviating positioning challenges caused by urban building occlusions and multipath effects. In marine environments with multiple interferences, LiDAR can capture high-precision target contour information, millimeter-wave radar can resist interference from sea wave clutter and fog/rain, and the data link assists in correcting sensor synchronization deviations. After fusion, the positioning error can be controlled within 1 m, addressing the performance shortcomings of single sensors under strong light reflections and meteorological interferences in marine environments [22].

However, existing multi-source fusion schemes largely rely on manually set fixed weight distributions for fusion results, making it difficult to dynamically adapt to changes in scenarios (such as transitions from foggy to sunny conditions in the ocean, or differences between peak and off-peak traffic in urban environments). Additionally, the computational complexity of fusion is relatively high, with execution times on embedded platforms (such as Raspberry Pi) often exceeding 200 milliseconds, which fails to meet real-time positioning requirements and requires further optimization [23].

### 2.1.2. Multi-UAV Positioning System

Through multi-sensor fusion, we can make up for the shortcomings of single-machine sensors [24]. Multi-Robot Cooperative Localization (MRCL), a pivotal technology in Multi-Robot Cooperative Systems (MRCS), is gaining increasing popularity due to its enhanced accuracy and robustness through information exchange and sharing [25]. The core of the multi-drone collaborative positioning architecture lies in optimizing the design of information exchange and decision-making mechanisms [8]. Currently, there are three typical architectures. The centralized architecture adopts a “data upload–centralized processing–instruction distribution” working mode, using a ground control station as the central node to execute global optimization algorithms, such as the probabilistic fusion framework proposed by Rudol et al [2]. This framework integrates multi-drone visual detection results, controlling the positioning error within a range of 3 m in outdoor search and rescue scenarios. However, this architecture has a single point of failure risk, and the communication bandwidth requirement increases linearly with the number of drones. When the number of nodes exceeds 8, the data transmission delay will exceed the 1-s threshold, making it difficult to apply in densely clustered scenarios such as urban security monitoring with more than 20 nodes.

The distributed architecture implements decentralized decision-making based on a peer-to-peer communication protocol, with each node achieving consensus filtering through the exchange of local information. Taking the AirSwarm system as an example, it utilizes Raspberry Pi as the edge computing unit and synchronizes multi-robot SLAM results through the ROS topic mechanism. At a scale of 10 nodes, the system maintains a response latency of 0.8 s. However, the distributed filtering algorithm of this architecture struggles to eliminate cumulative errors, leading to a positioning drift of 5–8 m after prolonged operation. In long-distance collaborative scenarios such as ocean monitoring over 5 km, unstable communication links further exacerbate error accumulation, affecting positioning accuracy.

The hybrid architecture combines the advantages of centralized and distributed architectures, employing a layered processing model of “edge computing + cloud optimization”: drone nodes are responsible for local feature extraction and preliminary fusion, while ground stations perform global pose optimization and consistency verification. This architecture achieves a good balance between communication overhead and positioning accuracy, enabling 0.5-m level positioning accuracy and a system latency of 1.2 s at a scale of 20 nodes. It can address the local real-time computing needs in urban environments with multiple occlusions, while also reducing error accumulation in long-distance marine coordination through global optimization, making it the mainstream technical route for large-scale drone swarm positioning.

Ultra-Wideband (UWB) provides absolute, drift-free positioning at the cost of requiring external infrastructure. Conversely, vision and LiDAR systems generate detailed relative odometry and maps but suffer from cumulative drift, a limitation shared by Inertial Measurement Units (IMUs) which, despite providing high-frequency motion prediction, drift rapidly. Consequently, sensor fusion methodologies, including Visual-Inertial Odometry (VIO) and UWB-aided LiDAR SLAM, have emerged to synergistically combine these technologies. These methods effectively leverage absolute updates to eliminate drift and integrate high-frequency data to ensure smooth output. Table 1 presents a detailed comparative analysis.

**Table 1.** Performance Comparison of Visual, UWB, LIDAR, and Sensor Fusion for 2D/3D Positioning.

| Technology                        | Accuracy                                 | Range/Coverage                                                  | Update Rate                                                           | Robustness & Environmental Sensitivity                                                                                                      | Scalability                                                                                                                                          |
|-----------------------------------|------------------------------------------|-----------------------------------------------------------------|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Visual (Monocular/Stereo VO/SLAM) | Medium-High (cm-dm level)                | Short (<10–20 m). Limited by field of view and feature quality. | Medium (10–30 Hz). Limited by computational load of image processing. | Low. Highly sensitive to lighting conditions (low light, glare), textureless environments, and dynamic obstacles. Prone to drift over time. | Medium. Requires significant processing per robot. Inter-robot loop closure can improve scalability but adds communication overhead.                 |
| UWB (Ultra-Wideband)              | Medium (10–30 cm) under good conditions. | Long (up to 100–200 m in LOS).                                  | Very High (100–1000 Hz). Low latency.                                 | Medium-High. Robust to RF interference and multipath in theory, but performance can degrade in dense NLOS (Non-Line-of-Sight) conditions.   | High. The system is inherently scalable; adding more tags has minimal impact on infrastructure, though network congestion can occur.                 |
| LIDAR                             | Very High (cm-level)                     | Medium-Long (up to 100–200 m for high-end models).              | Medium-High (5–20 Hz for 3D LiDAR).                                   | High. Robust to lighting conditions. Performance can degrade in presence of smoke, dust, rain, or highly reflective/absorbent surfaces.     | Low-Medium. Each robot typically requires its own LiDAR. Inter-robot loop closure is complex. Dense multi-robot environments can cause interference. |

Table 1. Cont.

| Technology    | Accuracy                                                    | Range/Coverage                                                   | Update Rate                                                                             | Robustness & Environmental Sensitivity                                                                                        | Scalability                                                                                                     |
|---------------|-------------------------------------------------------------|------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| Sensor Fusion | Very High (cm-level, often drift-free with aiding sensors). | Depends on the primary sensor (e.g., UWB range or Visual range). | High. IMU provides high-rate data between primary sensor updates, smoothing the output. | Very High. Mitigates individual sensor weaknesses, e.g., IMU counters visual drift; UWB anchors prevent LiDAR odometry drift. | Medium-High. Depends on the fusion architecture. Centralized fusion is complex; decentralized is more scalable. |

## 2.2. Multimodal Data Fusion

Multimodal data fusion technology integrates complementary information from heterogeneous sensors to construct a more robust positioning model, essentially addressing the differences in representation dimensions, spatiotemporal characteristics, and noise distributions among different modal data. With the diversified development of sensor technology, the fusion hierarchy has gradually expanded from traditional decision-level fusion to encompass full-stack solutions that include feature-level and data-level fusion, forming a multi-scale and multi-level fusion technology system [26].

In terms of fusion hierarchy and methods, data-level fusion directly processes raw sensor data, preserving complete information. Taking visual-radar fusion as an example, an association matrix is formed through camera extrinsic parameter calibration. This method theoretically achieves optimal positioning accuracy but has high requirements for sensor synchronization and calibration, with time synchronization errors needing to be within 1 ms and spatial calibration errors not exceeding  $0.5^\circ$ . Otherwise, performance degrades, and it demands significant computational resources, making real-time operation on embedded platforms challenging [27]. Feature-level fusion extracts features from various modalities to integrate information and is a research hotspot. Traditional methods manually design features and combine Kalman filtering to estimate multimodal states, resulting in low computational complexity but limited feature expressiveness, leading to a decline in positioning accuracy of over 40% in dynamic scenes. Deep learning methods automatically learn features, and cross-modal fusion networks can adaptively allocate weights, improving positioning accuracy by 30% to 50% [28]. However, they rely on large-scale labeled data and have computational demands 8 to 10 times greater than traditional methods. Decision-level fusion integrates the decision results from various modalities and has strong engineering practicality. For instance, the SFQ method analyzes positioning results to determine fusion weights, providing stable outputs when there are significant differences in sensor performance, with computational overhead only 1/5 that of feature-level fusion. However, it is limited by the accuracy of single-modal decisions, making it difficult to break through performance barriers.

Multimodal fusion faces core challenges that require an in-depth analysis of marine search and rescue and urban security scenarios to support subsequent method design [29]. Modal heterogeneity is a central issue, as there are significant structural differences in visual, millimeter-wave radar, and LiDAR data in marine scenarios, while urban scenarios face data quality issues with visual images, radar, and laser point clouds, exacerbating fusion difficulties. Temporal and spatial synchronization demands that multimodal data be strictly aligned in terms of timestamps and spatial coordinate systems. In marine scenarios, factors such as drone posture affect calibration errors and time synchronization, while in urban scenarios, high-rise buildings obstruct and impact the accuracy of temporal and spatial synchronization. The issue of environmental adaptability focuses on the robustness of fusion in complex scenarios; in marine environments, wave clutter affects data quality, while in urban settings, electromagnetic interference increases packet loss rates and matching

error rates. Existing fusion methods lack adaptive adjustment mechanisms, making it difficult to ensure positioning stability.

In response to the aforementioned challenges, existing solutions have made some progress, such as the spatial projection method that converts 3D radar point clouds into 2D depth maps, the feature mapping method that uses kernel functions to project heterogeneous features into a unified high-dimensional space, and the tensor decomposition method that directly processes multimodal tensor data to preserve modality correlations. However, these methods experience an information loss of 15–30% in marine and urban scenarios. For dynamic scenes, the time delay compensation algorithm based on extended Kalman filtering can correct synchronization errors within 50 ms, and the spatial calibration method based on SLAM can update sensor extrinsic in real-time to adapt to changes in drone posture. Nevertheless, in marine environments with strong interference and complex urban occlusions, achieving synchronization accuracy that meets high-precision positioning requirements remains challenging. An adaptive weighting mechanism can dynamically adjust the fusion coefficients based on sensor confidence levels, raising the radar weight to above 0.8 in scenarios of visual failure. A multimodal switching strategy selects the optimal sensor combination through a scene recognition model; for instance, activating thermal imaging sensors in low-light environments can increase the positioning success rate by 60%. However, existing strategies are often designed based on single scene features, making it difficult to adapt to the complex environmental changes in marine and urban settings, necessitating further optimization.

### 2.3. Dynamic Resource Optimization

Dynamic resource optimization technology is a core element for the efficient operation of multi-drone collaborative positioning systems. Its essence lies in achieving real-time and precise matching of task requirements and resource supply under the constraints of computational resources (CPU/GPU power) and communication resources (bandwidth/latency) [30]. This technology integrates multidisciplinary theories such as combinatorial optimization, game theory, and adaptive control to construct an integrated solution for task scheduling and resource allocation, optimizing system resource configuration from both computational and communication dimensions [31].

#### 2.3.1. Optimization of Computing Resources

The core objective is to address the task allocation challenge across heterogeneous hardware platforms, primarily achieved through two strategies: task offloading and algorithm selection [32]. In terms of task offloading, the cloud-edge collaborative architecture and edge computing each have their characteristics: the cloud-edge collaborative solution offloads computation-intensive object detection tasks to cloud GPUs, resulting in a frame rate increase of 3–5 times, but introduces a transmission delay of 200–500 ms, making it difficult to meet real-time requirements; edge computing, on the other hand, offloads some tasks to edge nodes such as Raspberry Pi, performing localized SLAM front-end processing through the AirSwarm system, effectively reducing communication volume by 60% while maintaining a processing speed of 15 fps [33]. The algorithm selection mechanism achieves dynamic adaptation by establishing a mapping relationship between algorithms and computing resources: the Rudol team constructs a performance database through offline testing, selecting the SSD MobileNet v2 algorithm (30 fps, mAP 0.72) for the Intel NUC platform, and deploying Faster R-CNN (15 fps, mAP 0.89) on the NVIDIA Tesla V100 platform; the online adaptive strategy dynamically switches algorithms by real-time monitoring of CPU utilization (threshold 70%) and task latency (threshold 100 ms), ensuring system stability under fluctuating computing power scenarios [34].

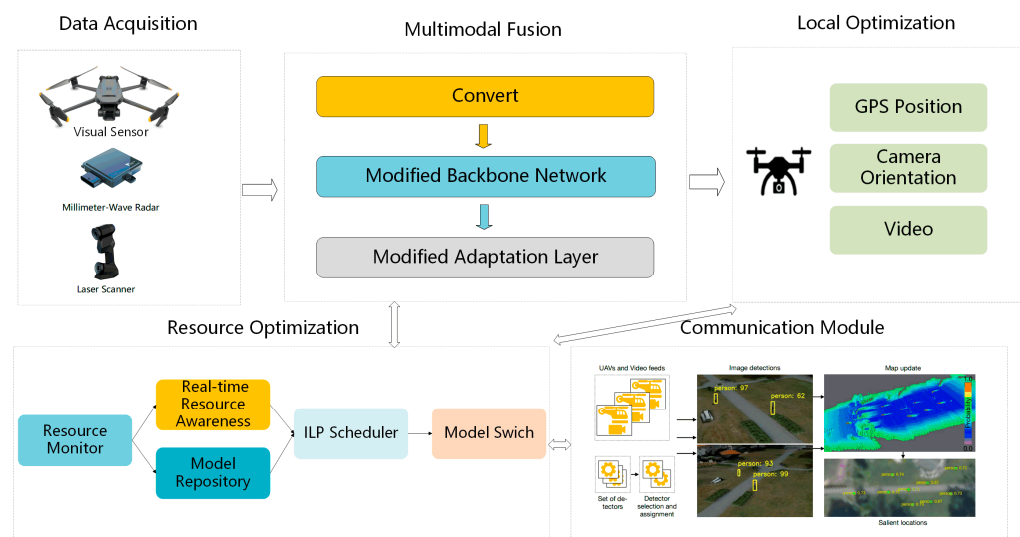
### 2.3.2. Communication Resource Optimization

The goal of communication resource optimization is to enhance data transmission efficiency under limited bandwidth, primarily involving data compression and transmission scheduling technologies [35]. In terms of data compression, H.265 video coding reduces the bitrate by 50% compared to H.264 while maintaining 720 p resolution (ITU-T, 202X), although it incurs a 5–10% loss in object detection accuracy. Feature compression technology utilizes PCA to reduce a 1024-dimensional feature vector to 256 dimensions, achieving a 75% compression rate while retaining over 90% of the discriminative information, making it suitable for feature-level fusion scenarios. Transmission scheduling strategies implement dynamic bandwidth allocation based on data priority and link quality: the AirSwarm system uses UDP to transmit high-priority control commands and positioning results, while TCP is used for map data, automatically reducing image resolution when bandwidth is insufficient; multi-channel multiplexing technology reduces data conflicts through frequency band separation, lowering the packet loss rate from 15% to below 5% in a 10-node scenario. The synergistic application of the aforementioned computational and communication resource optimization technologies significantly enhances the resource utilization efficiency and operational stability of multi-drone collaborative positioning systems.

## 3. Research Methods

### 3.1. Overall Framework

The multi-drone collaborative target localization framework proposed in this study is an integrated collaborative architecture that combines multimodal data processing, dynamic resource scheduling, and localization result optimization. It aims to address the issues of insufficient localization accuracy and low system resource utilization in complex environments. As shown in Figure 1, its five core modules interact through standardized interfaces and a closed-loop feedback mechanism, guaranteeing efficient and stable operation.



**Figure 1.** Overall Framework for Multi-Drone Cooperative Target Localization.

The data acquisition and preprocessing module provides foundational data for the framework, obtaining raw data from multimodal sensors. It suppresses noise through preprocessing, completes data time synchronization and spatial registration, and outputs standardized heterogeneous data, while also providing references for resource scheduling.

The multimodal data fusion module is the core computing unit that integrates the advantages of different modality sensors, receives standardized data, and completes feature

extraction, fusion, and preliminary positioning based on resource adaptation strategies, while feeding back relevant information and transmitting preliminary results.

The dynamic resource optimization allocation module regulates system resources, achieving real-time matching of computing and communication resources. It constructs model solutions by acquiring information from various nodes, issues scheduling instructions, and dynamically adjusts strategies.

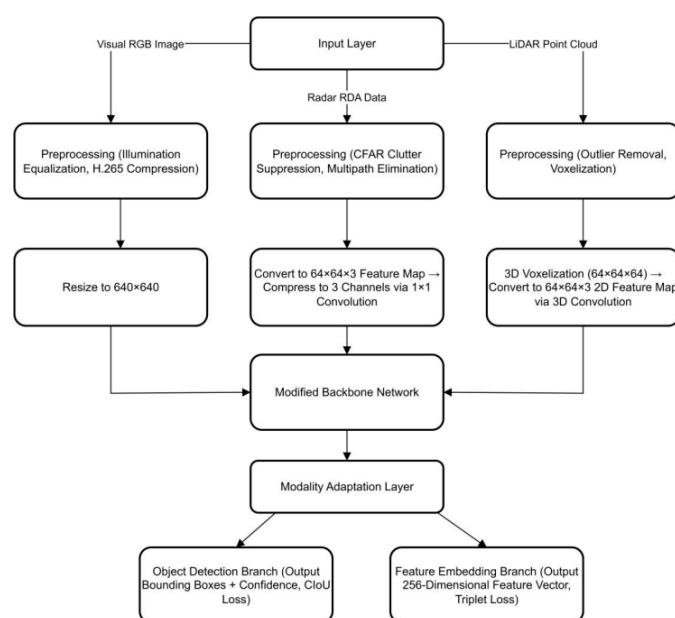
The positioning result optimization and collaborative output module enhances positioning accuracy by receiving preliminary results, processing them through filtering and collaborative verification, and outputting high-precision results, while also providing feedback information to assist resource scheduling.

The collaborative control and communication interaction module serves as the link for collaborative operation, running through the entire process to achieve pose synchronization, data transmission, and command distribution, while establishing a fault-tolerant mechanism to ensure system robustness.

Each module interacts through data feedback to form a whole, achieving high-precision and high-efficiency collaborative target localization of multiple drones in complex environments, addressing core challenges.

### 3.2. Multimodal Data Fusion Structure

The core objective of multimodal data fusion is to integrate the advantages of three typical sensors: vision, radar, and laser scanning, to address the performance shortcomings of a single modality in complex environments—such as the lack of robustness of visual sensors in low-light or occluded scenarios, the low positioning accuracy of radar at close range, and data loss in laser scanning under dense occlusion. As illustrated in Figure 2, this framework follows a progressive logic, with each stage designed to closely align with sensor application experiences and data processing strategies.



**Figure 2.** Multimodal Fusion Architecture for Heterogeneous Sensors.

#### 3.2.1. Data Collection Phase

In the data collection phase, three types of complementary sensors are selected, with the following configurations and characteristics: The visual sensor uses a high-definition camera (e.g., DJI Zenmuse P1) to capture RGB images (resolution  $1920 \times 1080$ , frame rate 30 fps), suitable for identifying target appearance features, but it needs to address the

issue of uneven lighting caused by strong light reflection from the ocean and shadows from urban buildings; the millimeter-wave radar is a miniaturized airborne model (e.g., TI AWR1843), which outputs target distance, speed, and azimuth data (ranging from 0.2 m to 400 m, with an accuracy of  $\pm 0.1$  m), adaptable for all-weather (rain and fog) and long-distance detection scenarios, focusing on handling multipath interference at close range ( $< 10$  m); the laser scanner employs a low-cost solid-state LiDAR (e.g., Velodyne VLP-16), generating 3D point clouds (point cloud density of 300 points/m<sup>2</sup>, ranging accuracy of  $\pm 2$  cm), used for extracting target contours and depth information, while needing to alleviate data voids under dense occlusions (e.g., between urban buildings). The dataset comprises approximately 18,000 aligned visual, radar, and LiDAR samples. All data were annotated through manual and semi-automated procedures, with labeled targets including vehicles, pedestrians, and other objects. The dataset was randomly split into training, validation, and test sets at a ratio of 70%/15%/15% to ensure unbiased evaluation. For model training, we employed the Adam optimizer with an initial learning rate of 0.001 and used Smooth L1 Loss as the regression objective function. During data preprocessing, domain-specific normalization was applied to all modal inputs: visual images underwent mean subtraction and standard deviation normalization, while radar RDA matrices and LiDAR voxel grids were subjected to min-max normalization to compress value ranges to the interval (0, 1) thereby accelerating model convergence.

### 3.2.2. Preprocessing and Modal Adaptation

The preprocessing stage designs differentiated processes based on the noise characteristics of different modalities: visual data employs the Multi-Scale Retinex with Color Restoration (MSRCR) algorithm for illumination equalization, which demonstrates better adaptability to strong light reflection scenarios in the ocean compared to Single-Scale Retinex (SSR) and basic MSR. The specific parameter settings include Gaussian filter radii of 15, 80, and 250, with weight coefficients set to 1/3. Additionally, the H.265 video compression scheme is utilized (adaptively adjusting between bitrates of 500 kbps to 20 Mbps) to achieve denoising and bandwidth adaptation. Radar data is processed using the Constant False Alarm Rate (CFAR) algorithm to remove ground/ocean surface clutter, and multipath reflection signals are eliminated based on Range-Doppler (RD) spectral analysis. Laser point clouds are filtered using statistical methods to remove outliers and downsampled to 100 points/m<sup>2</sup> using voxel grid filtering to reduce computational overhead. Simultaneously, time synchronization of the three modalities is achieved based on the drone's GPS timestamps (with an error of  $< 1$  ms). To address interference issues with GPS timestamps in urban canyons and long-distance ocean scenarios, a UWB (Ultra-Wideband) ranging-assisted time calibration scheme is implemented—calculating relative time differences through the time of flight (TOF) of UWB signals between drone nodes, correcting GPS timestamp deviations, and ensuring that the time synchronization error is stably controlled within 5 ms. Spatial registration is completed using sensor intrinsic parameters (camera focal length, radar field of view) to ensure that the data alignment accuracy meets the requirements for subsequent fusion.

To fully exploit the discriminative features of various modalities, a three-level network structure is constructed: “Modality Adaptation Layer–Shared Feature Layer–Task-Specific Layer,” integrating the real-time advantages of CNN detectors (SSD, Faster R-CNN) with YOLOv5. When considering inference speed and model compactness, YOLOv5 offers a more favorable operating point. The Modality Adaptation Layer is designed with dedicated conversion modules for different modality input formats: the visual modality directly inputs preprocessed RGB images (resized to  $640 \times 640$ , meeting YOLOv5 input requirements); the radar modality converts Range-Doppler-Angle (RDA) data into a three-dimensional

feature map of “Range-Angle-Intensity” ( $64 \times 64 \times 3$ ), compressing the channel number to 3 through  $1 \times 1$  convolution; the laser modality voxelizes the target point cloud clusters (voxel size  $0.1 \text{ m} \times 0.1 \text{ m} \times 0.1 \text{ m}$ ), generating a three-dimensional voxel feature map ( $64 \times 64 \times 64$ ), which is then transformed into a 2D feature map ( $64 \times 64 \times 3$ ) through 3D convolution, achieving dimensional consistency with the visual and radar feature maps.

### 3.2.3. Shared Feature Layer

The shared feature layer adopts a switchable backbone network architecture, achieving adaptive selection of multiple backbone networks through real-time feedback from a dynamic resource optimization module. It encompasses four types of networks: CSPDarknet53 (the native backbone of YOLOv5s), ResNet-50, EfficientNet-B4, and MobileNetV3-Large, and uniformly introduces a Cross-Modal Attention (CMA) mechanism to enhance the ability to capture key features. Additionally, all backbone networks replace the convolution kernels in the 3rd, 5th, and 7th layers with Deformable Convolutions (DCNv2) to adapt to target deformations, integrate the CMA module at the neck to dynamically allocate modal contribution weights, and extend the output scales of the Feature Pyramid Network (FPN) from  $8 \times 8$  to  $128 \times 128$  to improve small target detection accuracy. The task-specific layer features a dual-branch structure for object detection and feature embedding, utilizing CIoU loss function and Triplet Loss to optimize localization accuracy and feature discriminability. The performance characteristics of each network are as follows: CSPDarknet53 balances detection speed and localization accuracy, suitable for most balanced scenarios; ResNet-50 excels in feature extraction, suitable for high-precision localization needs; EfficientNet-B4 offers superior parameter control while ensuring feature expression capability; MobileNetV3-Large is a lightweight architecture, reducing computational load by over 60% compared to traditional networks, suitable for resource-constrained scenarios. Cross-modal feature fusion, combined with attention mechanisms and probabilistic models, enhances robustness. Following [36], first, During the training phase, the confidence of each modality is calculated, where the visual modality confidence  $C_{\text{vis}} = 0.6 \times \text{IOU} + 0.4 \times \left(1 - \frac{\text{image entropy}}{\text{maximum entropy value}}\right)$  (IOU is the intersection over union of the object detection candidate box and the ground truth box, image entropy reflects clarity, and the maximum entropy value is taken as 8, the theoretical maximum at 256 grayscale levels), radar modality confidence  $C_{\text{rad}} = 0.7 \times \frac{\text{SNR}}{\text{SNR}_{\text{max}}} + 0.3 \times \left(1 - \frac{\text{distance error}}{\text{maximum allowable error}}\right)$  (SNR is the signal-to-noise ratio of the radar echo,  $\text{SNR}_{\text{max}}$  is taken as 50 dB, and the maximum allowable distance is defined as  $C_{\text{las}} = 0.5 \times \frac{\text{point cloud density}}{\text{standard density}} + 0.5 \times \text{segmentation accuracy}$ , where standard density is set at  $300 \text{ points/m}^2$ , and the segmentation accuracy is defined as the matching ratio between the laser point cloud and the ground-truth target contour).

During inference, modality confidences are derived from temporal predictions to avoid ground truth dependency:  $C_{\text{vis}}$  uses the IoU between predicted boxes across frames,  $C_{\text{rad}}$  is based on the consistency of predicted positions, and  $C_{\text{las}}$  relies on the alignment of consecutive point cloud predictions.

The three types of confidence are normalized using the Softmax f modality are weighted and summed according to their respective weights, followed by batch normalization (Batch Norm) and ReLU activation function to enhance nonlinear expressiveness.

### 3.3. Dynamic Resource Optimization Allocation Algorithm

Dynamic resource optimization allocation aims to address the core issue of mismatched computational and communication resources with task demands in multi-drone collaborative positioning. By establishing an overall closed-loop mechanism, it achieves global optimization of positioning model selection, data transmission strategies, and task loads,

with a focus on ensuring hardware load balancing, efficient bandwidth utilization, and task continuity in the event of node failures.

### 3.3.1. Resource Real-Time Awareness Mechanism

The real-time resource perception mechanism employs a distributed sensing framework, utilizing ROS topic communication to collect the hardware status, communication quality, and task requirements of drone nodes in real-time, providing a decision-making basis for resource scheduling. In terms of computational resource perception, tools such as `htop` and `nvidia-smi` are used to monitor the CPU usage (threshold set at 80%, exceeding this is considered high load), GPU usage (threshold set at 75%), and memory usage (threshold set at 90%) of onboard computing units (e.g., Jetson AGX Orin, Raspberry Pi 4B) in real-time, quantifying the remaining computational resources. Additionally, the hardware types (e.g., high-end: Intel Core i7-7567U + NVIDIA GTX 1070, low-end: ARM Cortex-A72 of Raspberry Pi 4B) are recorded to provide a hardware basis for model adaptation. Communication resource perception employs the “Round Trip Time (RTT)” method to periodically measure the communication bandwidth between drones and between drones and ground stations every second, while also calculating the packet loss rate (greater than 5% is deemed an unstable link) and transmission delay (greater than 100 ms is considered high latency) over a 10-s period. Based on real-time bandwidth, data compression strategies are pre-calculated (e.g., H.265 encoding with a compression ratio of 10:1 is enabled when bandwidth < 5 Mbps; H.264 encoding with a compression ratio of 5:1 is enabled when bandwidth > 10 Mbps) to prevent bandwidth overflow. Node status and task requirement perception is monitored through a heartbeat mechanism to check the online status of drone nodes; if three consecutive heartbeat packets are not received, the node is deemed faulty. At the same time, the requirements for positioning tasks are analyzed, such as a positioning accuracy of <1 m for maritime search and rescue and a delay of <200 ms for moving targets, which are mapped to a “precision-real-time” priority (levels 1 to 5), with high-priority tasks receiving resource allocation first.

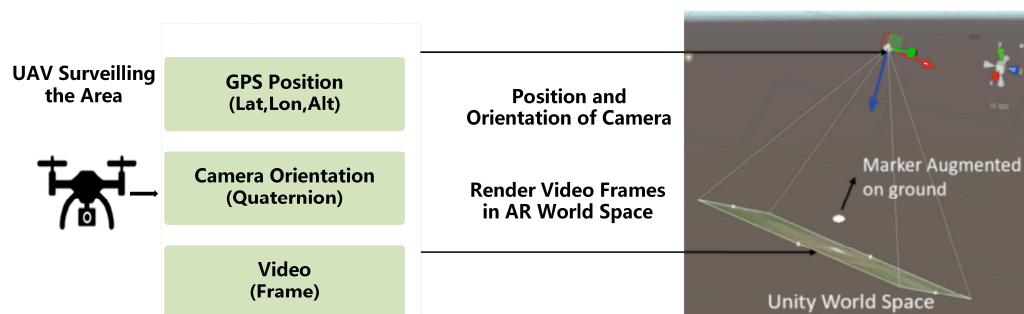
### 3.3.2. Offline Model Library Construction

The offline model library construction (detector pool) conducts systematic testing of the performance of various localization models on different hardware platforms in advance, creating a multi-dimensional indexed model library to provide data support for online model selection. The candidate model screening covers localization models with different accuracy-resource consumption trade-offs, such as high-precision models like Faster R-CNN ResNet-101 (suitable for resource-rich scenarios, such as ground stations or Jetson AGX Orin); balanced models like YOLOv5s (adapted for medium-resource nodes, such as drones equipped with Jetson Nano); and lightweight models like SSD MobileNet-v2 and YOLOv5n (targeted at resource-constrained nodes, such as drones equipped only with Raspberry Pi 4B). Offline testing primarily evaluates the model’s detection performance through two key metrics: Average Precision (AP@0.5, AP@0.5:0.95) and localization error (mean and standard deviation). To ensure comprehensive validation, testing employs two distinct dataset types: benchmark performance comparison uses standard datasets like COCO-person to establish baseline performance and enable fair comparisons with existing studies, while domain adaptation evaluation utilizes self-collected real-world datasets containing various practical challenges that serve as the key basis for assessing the model’s actual application performance. As well as resource consumption, where the CPU/GPU usage, single-frame execution time (e.g., Faster R-CNN execution time on Raspberry Pi > 500 ms, SSD MobileNet-v2 only requires 80 ms), and data output volume (e.g., Faster R-CNN outputs a 1024-dimensional feature vector, SSD outputs 512 dimensions) are recorded



### 3.4. Optimization of Positioning Results and Collaborative Output

As shown in Figure 4, the optimization of positioning results is centered on “high-precision filtering + multi-drone collaborative verification.” It corrects nonlinear errors through Extended Kalman Filtering (EKF) and combines a multi-drone collaborative verification mechanism of “result association–weighted fusion–anomaly elimination” to eliminate positioning deviations between nodes. Meanwhile, it utilizes Ultra-Wideband (UWB) technology to correct the drone’s own posture, ensuring temporal and spatial alignment of multi-drone data, ultimately generating a visual significance map.



**Figure 4.** Schematic Diagram of Recognition and Localization Output.

#### 3.4.1. EKF Correction

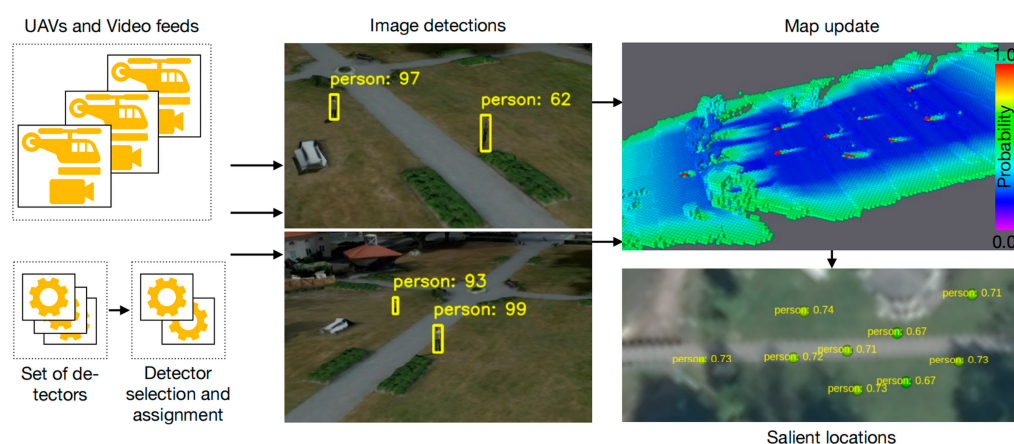
Firstly, to address the nonlinear characteristics of multimodal fusion results, an EKF model is constructed to achieve dynamic optimization of the target state (position, velocity), providing high-precision foundational data for subsequent collaborative verification.

Firstly, to address the nonlinear characteristics of multimodal fusion results, an EKF model is constructed to achieve dynamic optimization of the target state (position, velocity), providing high-precision foundational data for subsequent collaborative verification. The state vector  $\mathbf{x}_k = [x_k, y_k, z_k, v_{x_k}, v_{y_k}, v_{z_k}]^T$  contains the target’s 3D position ( $x, y, z$ ) and velocity ( $v_x, v_y, v_z$ ); the measurement vector  $\mathbf{z}_k = [u_k, v_k, r_k, z_{lk}]^T$  includes visual projection coordinates ( $u, v$  with an error of 0.5 pixels), radar distance ( $r$  with an error of 0.1 m), and laser depth ( $z_l$  with an error of 0.05 m). In terms of the nonlinear measurement functions, the visual projections are given by  $u = f_x \cdot (x/z) + c_x$  and  $v = f_y \cdot (y/z) + c_y$  (where  $f_x, f_y$  are the camera focal lengths, and  $c_x, c_y$  are the image centers), the radar distance is  $r = \sqrt{x^2 + y^2 + z^2}$ , and the laser depth is  $z_l = z$ . In the EKF filtering process, during the prediction phase, the state prediction value is calculated as  $\hat{\mathbf{x}}_{k|k-1} = \mathbf{F} \cdot \hat{\mathbf{x}}_{k-1|k-1}$  and the covariance prediction value as  $\mathbf{P}_{k|k-1} = \mathbf{F} \cdot \mathbf{P}_{k-1|k-1} \cdot \mathbf{F}^T + \mathbf{Q}$  (where  $\mathbf{F}$  is the state transition matrix and  $\mathbf{Q}$  is the process noise matrix with diagonal elements [0.008, 0.008, 0.008, 0.0008, 0.0008, 0.0008]). During the update phase, the Jacobian matrix  $\mathbf{H}_k$  is computed (by taking the partial derivative of the measurement function at  $\hat{\mathbf{x}}_{k|k-1}$ ), and the Kalman gain is calculated as  $\mathbf{K}_k = \mathbf{P}_{k|k-1} \cdot \mathbf{H}_k^T \cdot (\mathbf{H}_k \cdot \mathbf{P}_{k|k-1} \cdot \mathbf{H}_k^T + \mathbf{R})^{-1}$ , where “ $\mathbf{R}$ ” is the measurement noise matrix with diagonal elements [0.25, 0.25, 0.01, 0.0025]. The state is then updated as  $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \cdot (\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}))$  and the covariance as  $\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \cdot \mathbf{H}_k) \cdot \mathbf{P}_{k|k-1}$ , outputting the filtered positioning results.

#### 3.4.2. Multi-Drone Verification

As shown in Figure 5, the interaction and verification of positioning results among multiple drone nodes eliminate single-point positioning deviations and ensure data consistency across multiple drones. At the same time, UWB technology is introduced to correct the drone’s own posture, ensuring temporal and spatial alignment of the data. In the result association step, each drone employs image feature matching algorithms (SIFT or ORB) to extract image features (such as SIFT feature points and ORB descriptors) from

the detected target area, recording the position, scale, and orientation information of the feature points. The FLANN matcher is used to calculate the matching degree of target features among different drones (e.g., the number of matched feature points/total number of feature points), with a similarity threshold set at 0.7; if the matching degree  $\geq 0.7$ , it is determined to be the same target. Additionally, by combining UWB ranging between drones (error < 10 cm) and GPS timestamps (synchronization error < 1 ms), mismatches with excessive temporal and spatial distances (e.g., spatial distance > 1 m, time difference > 500 ms) are excluded, further enhancing association accuracy. During the weighted fusion, the trace of the covariance matrix after EKF filtering,  $\text{Tr}(\mathbf{P}\{\mathbf{k}|\mathbf{k}\})$ , is used as an indicator to calculate the positioning credibility  $C_i = 1/\text{Tr}(\mathbf{P}_i)$  (where  $\mathbf{P}_i$  is the EKF covariance matrix of the  $i$ -th drone); the smaller  $\text{Tr}(\mathbf{P})$ , the more stable the positioning error and the higher the credibility. The credibility is converted into normalized weights using the Softmax function  $v_i = \frac{\exp(C_i)}{\sum_{i=1}^n \exp(C_i)}$ , “where “ $n$ ” is the number of drones participating in the fusion), satisfying  $\sum w_i = 1$ . Finally, the target coordinates are obtained through weighted calculations:  $x_{\text{final}} = \sum w_i \cdot x_i$ ,  $y_{\text{final}} = \sum w_i \cdot y_i$ ,  $z_{\text{final}} = \sum w_i \cdot z_i$  (where  $x_i, y_i, z_i$  are the positioning results of the  $i$ -th drone). Anomaly removal employs the Grubbs test (significance level  $\alpha = 0.05$ ); first, the mean  $\mu$  of the fusion results is calculated (e.g., mean in the  $x$  direction  $\mu_x = \frac{\sum x_i}{n}$ ) and the standard deviation  $\sigma$  (e.g., standard deviation in the  $x$  direction  $\sigma_x = \sqrt{\frac{\sum (x_i - \mu_x)^2}{n-1}}$ ) and then the Grubbs statistic  $G = \frac{\max |x_i - \mu_x|}{\sigma_x}$  is computed. The critical value table for the Grubbs test is consulted (determined by  $n$  and  $\alpha$ ); if  $G > G_{\text{critical value}}$ , the  $x_i$  is deemed an outlier and removed. After removing outliers, the credibility and weights of the remaining drones are recalculated, and the weighted fusion step is repeated until no outliers remain. In the UWB posture correction phase, the relative pose (roll, pitch, yaw deviations) between drones is measured every 10 s using the UWB module and compared with the ground truth (OptiTrack system, accuracy  $\pm 0.1$  mm) to determine the posture error  $\Delta\theta$ . OptiTrack system was exclusively used for benchmark testing and sensor calibration in a controlled indoor environment. This occurred specifically during indoor calibration/validation experiments. The IMU parameters and camera extrinsics of the drones are then adjusted based on  $\Delta\theta$ , keeping the posture error within  $0.1^\circ$  to avoid shifts in the positioning coordinate system due to drone posture deviations, thereby ensuring spatial consistency of the multi-drone data.



**Figure 5.** Schematic of Collaborative Output.

### 3.4.3. Three-Dimensional Visualization

Finally, the positioning results after collaborative verification are converted into a visualized 3D grid probability map, providing intuitive support for subsequent tasks such as rescue path planning. A 3D grid with a resolution of  $0.25 \text{ m} \times 0.25 \text{ m} \times 0.25 \text{ m}$  is

constructed to cover the monitored area, where each grid cell (voxel) stores the probability of target presence  $P$ . The grid probability is updated based on Bayes' theorem:  $P(\text{voxel}|z_1:k) = \frac{P(z_k|\text{voxel}) \cdot P(\text{voxel}|z_1:k-1)}{P(z_k)}$ , where  $P(z_k|\text{voxel})$  is the measurement likelihood (calculated from the EKF measurement residual, with smaller residuals indicating higher likelihood). Grid cells with  $p \geq 0.7$  are filtered as candidate target areas, and the OctoMap tool is used to generate a 3D significance map, marking the target's 3D coordinates  $x_{final}$ ,  $y_{final}$ ,  $z_{final}$  and confidence  $(1 - \text{Tr}(P_{final})/\text{Tr}(P_{max}))$ , where  $\text{Tr}(P_{final})$  is the final EKF covariance trace and  $\text{Tr}(P_{max})$  is the maximum allowable trace), which is exported in JSON format to the ground control station (e.g., QGroundControl) to support real-time viewing and task scheduling. 4 Experimental Design and Result Analysis

## 4. Experimental Design and Results

### 4.1. Experimental Platform and Environment Design

The construction of the experimental platform is a key aspect of the research on multi-drone collaborative target localization technology, directly affecting the reliability and practicality of the experimental results. The experimental platform consists of three parts: hardware system, software framework, and scenario setup. The hardware selection comprehensively considers classic configurations and engineering practical needs, while the software is developed modularly based on the ROS architecture. The scenario design fully covers typical complex environments for target localization, ensuring that the experimental conditions are highly aligned with actual applications.

#### 4.1.1. Hardware System

In terms of hardware systems, the drone swarm utilizes three DJI Matrice 300 RTK drones, each equipped with a multimodal sensor suite. The visual sensor is the DJI Zenmuse P1 full-frame camera (resolution  $6016 \times 4016$ , frame rate 20 fps, focal length 24 mm), which supports RAW format image capture to cope with the strong light reflection environment at sea. The millimeter-wave radar is the TI AWR1843 automotive radar (ranging from 0.2 m to 400 m, ranging accuracy  $\pm 0.1$  m, azimuth angle resolution  $3^\circ$ ), mounted on a custom bracket under the drone's belly to avoid obstruction by the airframe. The laser scanner is the Velodyne VLP-16 solid-state LiDAR (point cloud density 300 points/m<sup>2</sup>, ranging accuracy  $\pm 2$  cm, vertical field of view  $\pm 15^\circ$ ), designed with a lightweight configuration based on classic selection to control costs.

The configuration of the computing and communication module is as follows: each drone is equipped with an NVIDIA Jetson AGX Orin (64GB RAM, 12-core ARM Cortex-A78AE CPU, Reference Document 1) as the onboard computing unit, responsible for real-time preprocessing and feature extraction of multimodal data; the ground control center utilizes an Intel Core i9-13900K CPU + NVIDIA RTX 4090 GPU workstation for resource scheduling optimization and fusion of positioning results; the communication link employs a dual-link redundancy design of 4G + WiFi (WiFi follows the IEEE 802.11ac protocol with a bandwidth of 1.3Gbps; 4G uses industrial-grade modules with a latency of  $\leq 50$  ms), ensuring the stability of data transmission.

The truth value acquisition device uses a Trimble R12 GNSS RTK receiver (positioning accuracy  $\pm 1$  cm) to collect ground target truth values, recording the target's 3D coordinates at a frequency of 10 Hz. Simultaneously, an OptiTrack Motion Capture system (equipped with 6 Prime 13 W cameras, positioning accuracy  $\pm 0.1$  mm) is used to record the UAV's pose for sensor extrinsic parameter calibration and localization deviation assessment.

#### 4.1.2. Software Framework

The software framework is developed based on the ROS Noetic architecture and is divided into five layers: data acquisition layer, preprocessing layer, fusion layer, resource scheduling layer, and result visualization layer. The data acquisition layer achieves multi-sensor data synchronization (time synchronization error  $< 1$  ms) through custom ROS topics; the preprocessing layer integrates Retinex illumination equalization, CFAR radar clutter suppression, and point cloud statistical filtering algorithms; the fusion layer implements improved YOLOv5 feature extraction and cross-modal attention fusion; the resource scheduling layer utilizes Google OR-Tools for ILP optimization solving; and the result visualization layer employs RViz and OctoMap for real-time display of 3D saliency maps.

The experiment selects three types of benchmark algorithms for comparison: the first is single visual localization (YOLOv5 + EKF), which utilizes only visual camera data, detecting targets through YOLOv5 and optimizing the localization results via EKF; the second is resource-unoptimized multimodal localization (traditional weighted fusion + fixed detector), which employs equal-weight fusion of visual, radar, and laser data, using Faster R-CNN as a fixed detector without dynamic resource allocation; the third is distributed collaborative localization (CCM-SLAM), which achieves distributed map construction and localization based on monocular vision without introducing multimodal data.

#### 4.1.3. Experimental Scenario Design

The experimental scenario design encompasses three typical environments, with parameter settings referenced from relevant literature and combined with practical application needs. The marine search and rescue scenario is selected in the  $2\text{ km} \times 2\text{ km}$  sea area near the Wanshan Islands in Zhuhai, Guangdong, featuring 5 stationary targets (simulating individuals who have fallen into the water, equipped with RTK locators) and 2 moving targets (simulating drifting vessels, with speeds of 1–3 m/s). This scenario includes environmental conditions such as bright sunlight (noon at 12:00, illumination intensity of 100,000 lux), overcast weather (illumination intensity of 20,000 lux), and light fog (visibility of 500 m), to test the algorithm's performance under strong reflections, low light, and adverse weather conditions. The urban occlusion scenario is selected from the Qianhai Free Trade Zone building complex in Shenzhen (including 10 buildings ranging from 50 to 100 m in height), setting up 8 stationary personnel targets simulating urban surveillance, distributed in occluded positions such as building gaps, under tree shade, and at street intersections, to test the algorithm's robustness in densely occluded and multi-target overlapping scenarios. The resource dynamic variation scenario is based on the marine search and rescue scenario, dynamically adjusting WiFi link bandwidth (2 Mbps, 5 Mbps, 10 Mbps, 20 Mbps) and drone computational load (simulating CPU loads of 30%, 60%, and 90%) to test the adaptability of the resource dynamic allocation algorithm.

#### 4.2. Experimental Indicators and Evaluation Methods

This study constructs a quantitative evaluation index system from three core dimensions: positioning accuracy, resource utilization efficiency, and collaborative robustness, to comprehensively and accurately characterize the overall performance of the algorithm in complex scenarios.

In the experimental data processing, each scenario was repeated 10 times, with each experiment lasting 30 min, resulting in the collection of approximately 18,000 frames of target localization data. To ensure data validity and analysis accuracy, outliers were removed using the Grubbs test (significance level  $\alpha = 0.05$ ). A one-way ANOVA was conducted to examine the differences in algorithm performance across different scenarios (with a significance level set at 0.05), and Tukey's HSD test was employed for post hoc

multiple comparisons to determine whether the performance differences between the proposed algorithm and the baseline algorithm were statistically significant. All experimental data analysis was performed in Python 3.10.6, using the Pandas library (version 1.5.3) for data processing, and the Matplotlib library (version 3.7.1) for visualization, ensuring the intuitiveness and accuracy of the experimental results.

#### 4.2.1. Positioning Accuracy

In the dimension of positioning accuracy evaluation, absolute positioning error (APE), localization recall precision (LRP), and temporal consistency error (TCE) are selected as core indicators. Absolute positioning error (APE) is measured by calculating the Euclidean distance between the algorithm's output three-dimensional coordinates  $(x_{\text{pred}}, y_{\text{pred}}, z_{\text{pred}})$  and the ground truth coordinates  $(x_{\text{gt}}, y_{\text{gt}}, z_{\text{gt}})$  obtained through Real-Time Kinematic (RTK) differential technology, with the formula  $\text{APE} = \sqrt{(x_{\text{pred}} - x_{\text{gt}})^2 + (y_{\text{pred}} - y_{\text{gt}})^2 + (z_{\text{pred}} - z_{\text{gt}})^2}$ . The mean (mAPE), standard deviation ( $\sigma_{\text{APE}}$ ), and maximum value (maxAPE) are statistically analyzed to quantitatively characterize the concentration and stability of positioning accuracy. Localization recall precision (LRP) is evaluated by calculating the optimal LRP error (oLRP) and its components ( $\text{oLRP}_{\text{IoU}}$ ,  $\text{oLRP}_{\text{FP}}$ ,  $\text{oLRP}_{\text{FN}}$ ), where  $\text{oLRP}_{\text{IoU}} = \frac{1}{N_{\text{TP}}} \sum (1 - \text{IoU}_i)$  is used to represent local- and deviation ( $N_{\text{TP}}$  is the number of true positive detections, and  $\text{IoU}_i$  is the intersection over union for the  $i$ -th true positive detection),  $\text{oLRP}_{\text{FP}} = \frac{N_{\text{FP}}}{|Y_s|}$  represents the false positive rate ( $N_{\text{FP}}$  is the number of false positive detections, and  $Y_s$  is the detection set filtered by confidence threshold  $s$ ), and  $\text{oLRP}_{\text{FN}} = \frac{N_{\text{FN}}}{|X|}$  represents the false negative rate ( $N_{\text{FN}}$  is the number of false negative detections, and  $X$  is the ground truth target set). These three components work together to provide a comprehensive evaluation of the algorithm's positioning accuracy and target detection completeness. Temporal consistency error (TCE) is used to characterize the stability of positioning in dynamic target tracking scenarios, with the calculation formula  $\text{TCE} = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (\text{APE}_i - \text{mAPE})^2}$ , where  $\text{APE}_i$  is the absolute positioning error of the  $i$ -th frame in a continuous sequence of 10 frames, and mAPE is the average absolute positioning error of that sequence. A smaller TCE value indicates that the positioning results are more stable over time.

#### 4.2.2. Resource Utilization Efficiency

In the dimension of resource utilization efficiency assessment, three indicators are included: computing resource utilization rate, communication bandwidth utilization rate, and detector switching efficiency. The computing resource utilization rate is quantified by calculating the mean and peak values of the usage rates of the drone's onboard central processing unit (CPU) ( $U_{\text{CPU}}$ ), graphics processing unit (GPU) ( $U_{\text{GPU}}$ ), and random access memory (RAM) ( $U_{\text{RAM}}$ ). The communication bandwidth utilization rate is assessed by calculating the ratio of actual bandwidth consumption during data transmission ( $B_{\text{used}}$ ) to the maximum available bandwidth of the link ( $B_{\text{max}}$ ), represented as  $\eta_B = \frac{B_{\text{used}}}{B_{\text{max}}} \times 100\%$ . The detector switching efficiency is evaluated by measuring the average time taken for detector switching during resource dynamic adjustment ( $T_{\text{switch}}$ ), which includes the time taken for task offloading, inference computation, and feedback transmission) and the switching success rate ( $R_{\text{switch}} = \frac{\text{Number of successful switches}}{\text{Total number of switches}} \times 100\%$ ), enabling a quantitative assessment of the real-time performance and reliability of the resource allocation algorithm.

#### 4.2.3. Collaborative Robustness

In the dimension of collaborative robustness assessment, three indicators are used: node failure recovery time ( $T_{\text{recover}}$ ), multimodal failure adaptability ( $A_{\text{model}}$ ), and scenario adaptability score ( $S_{\text{adapt}}$ ). The node failure recovery time ( $T_{\text{recover}}$ ) is characterized by

simulating the failure of drone nodes (disconnection of communication links) and recording the time taken for the remaining nodes to re-establish collaborative positioning, which reflects the fault tolerance capability of the algorithm. Multimodal failure adaptability ( $A_{modal}$ ) is quantified by separately shutting down visual, radar, and laser sensors, and calculating the ratio of mAPE in single-modal failure scenarios to mAPE in normal scenarios, expressed as  $A_{modal} = \frac{mAPE_{single}}{mAPE_{normal}}$ , thereby quantifying the algorithm's adaptability to single-modal failures. The scenario adaptability score ( $S_{adapt}$ ) is calculated for different scenarios such as marine, urban, and resource dynamic changes by computing the standardized scores of mAPE,  $\eta_B$ , and  $T_{recover}$  (ranging from 0 to 1), and then summing them with weights of 0.4, 0.3, and 0.3, respectively. The formula is  $S_{adapt} = 0.4 \cdot \frac{mAPE_{norm}}{mAPE_{max}} + 0.3 \cdot \eta_B + 0.3 \cdot \left(1 - \frac{T_{recover}}{T_{max}}\right)$ , where  $mAPE_{norm}$  is the actual mean absolute positioning error of the scenario,  $mAPE_{max}$  is the maximum mean absolute positioning error of the baseline scenario, and  $T_{max}$  is the maximum node failure recovery time of the baseline scenario, collectively representing the algorithm's generalizability across different scenarios.

#### 4.3. Experimental Results and Analysis

The experimental results are presented by categorizing them into different scenarios, specifically analyzing the positioning accuracy, resource utilization efficiency, and collaborative robustness of the algorithm in marine search and rescue, urban occlusion, and dynamic resource change scenarios. By comparing with baseline algorithms, the effectiveness of multimodal fusion and dynamic resource optimization is validated. It should be noted that the target application scenarios of this study—maritime search and rescue (weak signal reflection over open water), urban occlusion (multipath effects caused by dense structures), and dynamic resource variations (fluctuating computational loads)—significantly exacerbate issues such as multipath effects, signal attenuation/occlusion, and dynamic interference, which are the primary factors contributing to the increased error. While existing studies achieving centimeter-level accuracy are typically conducted in well-controlled environments with clear line-of-sight and minimal interference, our system maintains a controllable level of error in real-world open scenarios, achieving an acceptable balance among reliability, resource efficiency, and accuracy. Our work thus provides a feasible solution for complex environments where traditional high-precision technologies struggle to perform stably. In subsequent research, we will further narrow the error margin in practical applications by optimizing sensor fusion strategies and dynamic resource scheduling mechanisms. Additionally, the advantages and shortcomings of the algorithm are analyzed in conjunction with experimental observations, providing direction for future improvements.

##### 4.3.1. Results of Marine Search and Rescue Scenario Experiments

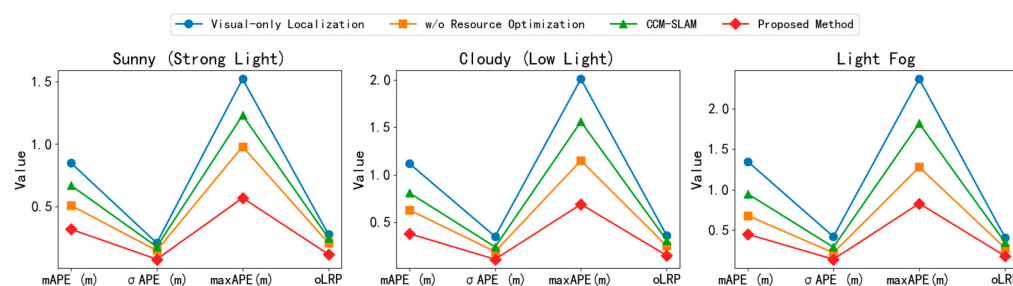
The core challenge in marine search and rescue scenarios is the impact of strong light reflection, low light conditions, and adverse weather on positioning accuracy. Experimental results indicate that the proposed algorithm maintains high positioning accuracy under various environmental conditions, significantly outperforming the baseline algorithms (based on data from 10 repeated experiments, verified by  $t$ -tests, the performance differences between the proposed algorithm and each baseline algorithm meet the criterion of  $p < 0.05$ , indicating statistical significance).

In terms of positioning accuracy, Table 2 presents a comparison of the positioning accuracy metrics of various algorithms under different environmental conditions (experimental conditions: 3 drones, 10 repeated experiments, each lasting 30 min). These results are visualized for clarity in Figure 6. In sunny and bright scenarios, the algorithm proposed in this study demonstrates outstanding performance, with a mean absolute positioning

error (mAPE) of 0.32 m and a standard deviation of absolute positioning error ( $\sigma$ APe) of 0.08 m. Compared to the single visual positioning algorithm (mAPE = 0.85 m), the resource-unoptimized multimodal positioning algorithm (mAPE = 0.51 m), and the CCM-SLAM algorithm (mAPE = 0.67 m), the mAPE values are reduced by 62.4%, 37.3%, and 52.2%, respectively. In low-light (overcast) scenarios, the mAPE of this algorithm is 0.38 m, still maintaining a low level; in contrast, the mAPE of the single visual positioning algorithm significantly rises to 1.12 m, primarily due to the effective compensation for the lack of visual features under low-light conditions through the deep integration of radar and laser data in this algorithm. In light fog scenarios, the mAPE of this algorithm is 0.45 m, outperforming the resource-unoptimized multimodal positioning algorithm (mAPE = 0.68 m), thanks to the dynamic resource allocation strategy that prioritizes radar detectors (average precision @ 0.5 = 0.89), thereby minimizing the interference of foggy conditions on visual detection.

**Table 2.** Comparison of Positioning Accuracy Metrics of Various Algorithms in Marine Search and Rescue Scenarios (3 Drones, 10 Repeated Experiments).

| Environmental Conditions | Algorithm                                            | mAPE (m) | $\sigma$ APe (m) | maxAPE (m) | oLRP |
|--------------------------|------------------------------------------------------|----------|------------------|------------|------|
| Sunny (Strong Light)     | Visual-only Localization                             | 0.85     | 0.21             | 1.52       | 0.28 |
|                          | Multi-modal Localization (w/o Resource Optimization) | 0.51     | 0.15             | 0.98       | 0.21 |
|                          | CCM-SLAM                                             | 0.67     | 0.18             | 1.23       | 0.25 |
|                          | Proposed Method                                      | 0.32     | 0.08             | 0.57       | 0.12 |
| Cloudy (Low Light)       | Visual-only Localization                             | 1.12     | 0.35             | 2.01       | 0.36 |
|                          | Multi-modal Localization (w/o Resource Optimization) | 0.63     | 0.19             | 1.15       | 0.25 |
|                          | CCM-SLAM                                             | 0.81     | 0.24             | 1.56       | 0.31 |
|                          | Proposed Method                                      | 0.38     | 0.11             | 0.69       | 0.15 |
| Light Fog                | Visual-only Localization                             | 1.35     | 0.42             | 2.37       | 0.41 |
|                          | Multi-modal Localization (w/o Resource Optimization) | 0.68     | 0.22             | 1.28       | 0.28 |
|                          | CCM-SLAM                                             | 0.95     | 0.29             | 1.82       | 0.35 |
|                          | Proposed Method                                      | 0.45     | 0.14             | 0.83       | 0.18 |



**Figure 6.** Performance Comparison of Algorithms for Marine Search and Rescue.

In terms of resource utilization efficiency, under sunny and bright conditions, the proposed algorithm's  $U_{CPU}$  average is 42%,  $U_{GPU}$  average is 38%, and  $U_{RAM}$  average is 55%, all of which are lower than the multimodal localization without resource optimization ( $U_{CPU} = 68\%$ ,  $U_{GPU} = 59\%$ ,  $U_{RAM} = 72\%$ ). This is because the dynamic resource allocation algorithm prioritizes the lightweight YOLOv5n detector (execution time 30 ms) based on lighting conditions, rather than consistently using Faster R-CNN (execution time 263 ms). The average communication bandwidth utilization  $\eta_B$  is 45%, which is lower than that of the multimodal localization without resource optimization ( $\eta_B = 78\%$ ), as the algorithm dynamically adjusts the video compression rate (H.265 compression ratio 10:1) based on bandwidth, reducing data transmission volume. In terms of detector switching efficiency, the proposed algorithm's  $T_{switch}$  average is 85 ms, with  $R_{switch}$  at 99%, enabling quick

adaptation to change in lighting conditions (such as sudden changes in illumination caused by cloud cover).

In terms of collaborative robustness, during the simulation of a drone failure (a total of 10 failure simulation experiments), the proposed algorithm achieved  $T_{\text{recover}} = 2.3$  s ( $\sigma = 0.3$  s), which is lower than that of CCM-SLAM ( $T_{\text{recover}} = 5.7$  s,  $\sigma = 0.6$  s). The  $t$ -test showed  $p < 0.05$ , indicating a significant performance difference between the two, attributed to the proposed algorithm's basis in distributed resource scheduling, allowing remaining nodes to quickly reallocate detection tasks. When a single modality fails, after shutting down the visual sensor, the proposed algorithm's mAPE = 0.62 m ( $A_{\text{modal}} = 1.63$ ); after shutting down the radar sensor, mAPE = 0.51 m ( $A_{\text{modal}} = 1.34$ ); and after shutting down the laser sensor, mAPE = 0.43 m ( $A_{\text{modal}} = 1.13$ ), all outperforming the multimodal positioning without resource optimization ( $A_{\text{modal}} = 2.18$  after shutting down the visual sensor), thereby validating the fault tolerance of multimodal fusion against single modality failures.

#### 4.3.2. Experimental Results of Urban Occlusion Scenarios

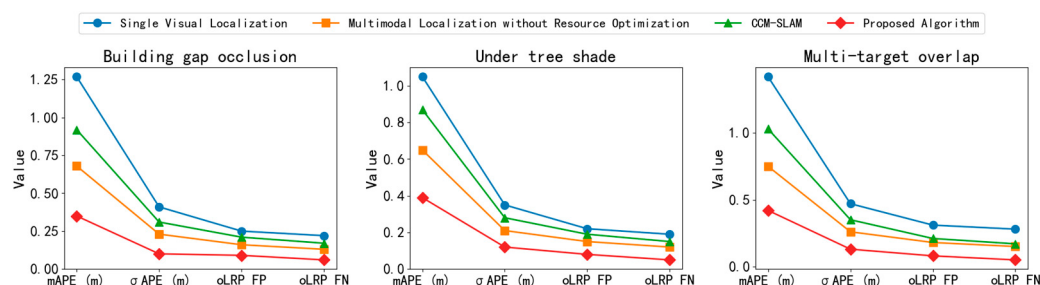
The core challenge of urban occlusion scenarios is the positioning deviation caused by building occlusion, overlapping multiple targets, and shadow interference. Experimental results indicate that the proposed algorithm effectively addresses the positioning difficulties in occluded scenarios through multimodal feature fusion and collaborative positioning (based on data from 10 repeated experiments, the  $t$ -test confirmed that the performance differences between the proposed algorithm and various baseline algorithms are statistically significant with  $p < 0.05$ ).

In terms of positioning accuracy, Table 3 compares the positioning accuracy metrics of various algorithms under urban occlusion scenarios (experimental conditions: 3 drones, 10 repeated experiments, each lasting 30 min), and the corresponding results are depicted more intuitively in Figure 7. In the building gap occlusion scenario (target located between two buildings, visual occlusion rate of 40%), the proposed algorithm achieved mAPE = 0.35 m, which is a 72.4% reduction compared to single visual positioning (mAPE = 1.27 m). The main reason is that the laser point cloud captured the contour features of the target, while radar data provided distance information, compensating for the feature loss caused by visual occlusion. In the shaded scene (uneven visual illumination, target located in the shade), the proposed algorithm achieved mAPE = 0.39 m, outperforming resource-optimized multimodal positioning (mAPE = 0.65 m), as the cross-modal attention mechanism enhanced the weight of radar and laser features. In the multi-target overlapping scenario (3 targets partially overlapping in the visual image, overlap rate of 30%), the proposed algorithm achieved oLRP<sub>FP</sub> = 0.08 and oLRP<sub>FN</sub> = 0.05, which are lower than those of CCM-SLAM (oLRP<sub>FP</sub> = 0.21, oLRP<sub>FN</sub> = 0.17). This is because multimodal data provided richer target features, reducing false positives and missed detections.

In terms of collaborative robustness, the proposed algorithm achieves a scene adaptability score of  $S_{\text{adapt}} = 0.87$ , which is higher than that of single visual localization ( $S_{\text{adapt}} = 0.42$ ), resource-free optimized multimodal localization ( $S_{\text{adapt}} = 0.65$ ), and CCM-SLAM ( $S_{\text{adapt}} = 0.58$ ), indicating superior overall performance of the algorithm in urban occlusion scenarios. Regarding node fault recovery, during simulations of two drone failures (a total of 10 fault simulation experiments), the proposed algorithm has a recovery time of  $T_{\text{recover}} = 3.8$  s ( $\sigma = 0.4$  s) while maintaining an mAPE of 0.61 m. In contrast, CCM-SLAM, due to the lack of multimodal data support, has a recovery time of  $T_{\text{recover}} = 8.5$  s ( $\sigma = 0.8$  s) and an mAPE of 1.32 m. A  $t$ -test shows  $p < 0.05$ , indicating a significant performance difference between the two, thereby validating the collaborative robustness of the proposed algorithm.

**Table 3.** Comparison of Positioning Accuracy Metrics of Various Algorithms under Urban Occlusion Scenarios (3 Drones, 10 Repeated Experiments).

| Occlusion Type         | Algorithm                                             | mAPE (m) | $\sigma$ APE (m) | oLRP_FP | oLRP_FN |
|------------------------|-------------------------------------------------------|----------|------------------|---------|---------|
| Building gap occlusion | Single Visual Localization                            | 1.27     | 0.41             | 0.25    | 0.22    |
|                        | Multimodal Localization without Resource Optimization | 0.68     | 0.23             | 0.16    | 0.13    |
|                        | CCM-SLAM                                              | 0.92     | 0.31             | 0.21    | 0.17    |
|                        | Proposed Algorithm                                    | 0.35     | 0.10             | 0.09    | 0.06    |
| Under tree shade       | Single Visual Localization                            | 1.05     | 0.35             | 0.22    | 0.19    |
|                        | Multimodal Localization without Resource Optimization | 0.65     | 0.21             | 0.15    | 0.12    |
|                        | CCM-SLAM                                              | 0.87     | 0.28             | 0.19    | 0.15    |
|                        | Proposed Algorithm                                    | 0.39     | 0.12             | 0.08    | 0.05    |
| Multi-target overlap   | Single Visual Localization                            | 1.42     | 0.47             | 0.31    | 0.28    |
|                        | Multimodal Localization without Resource Optimization | 0.75     | 0.26             | 0.18    | 0.15    |
|                        | CCM-SLAM                                              | 1.03     | 0.35             | 0.21    | 0.17    |
|                        | Proposed Algorithm                                    | 0.42     | 0.13             | 0.08    | 0.05    |

**Figure 7.** Performance Comparison of Algorithms for Urban Occlusion Positioning.

#### 4.3.3. Experimental Results of Resource Dynamic Variation Scenarios

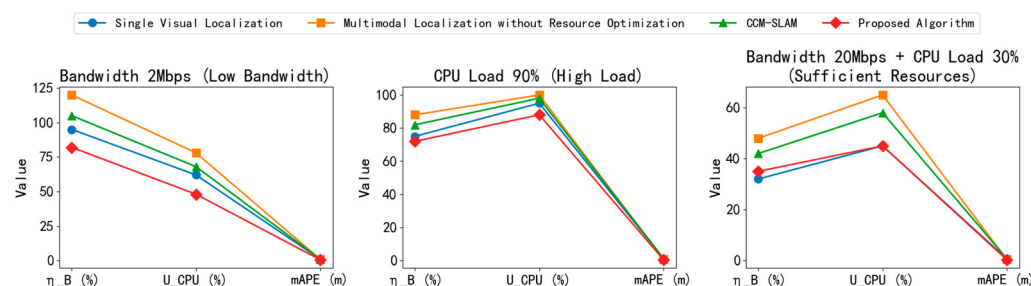
The core challenge of dynamic resource variation scenarios is the imbalance in resource allocation caused by fluctuations in communication bandwidth and changes in computational load. Experimental results indicate that the proposed algorithm achieves precise matching of resources to task requirements through dynamic resource optimization (based on data from 10 repeated experiments, the performance differences between the proposed algorithm and various benchmark algorithms were validated by *t*-tests, with all differences satisfying  $p < 0.05$ , indicating statistical significance).

In terms of resource utilization efficiency analysis, Table 4 compares the resource utilization efficiency indicators of various algorithms under dynamic resource variation scenarios (experimental conditions: 3 drones, 10 repeated experiments, each lasting 30 min), and the corresponding results are depicted more intuitively in Figure 8. In the low bandwidth scenario of 2Mbps, the proposed algorithm achieves  $\eta_B = 82\%$ , using the lightweight SSD MobileNet-v2 detector (execution time 30 ms), with mAPE = 0.51 m. In contrast, the resource-unoptimized multimodal localization, which relies on the fixed use of Faster R-CNN, results in  $\eta_B = 120\%$  (bandwidth overflow). To ensure data transmission, the system automatically activates the packet loss retransmission mechanism, but approximately 15% of packets are still lost, leading to incomplete localization data and an eventual mAPE of 0.89 m. Although localization results can be obtained, the accuracy significantly decreases. In the high load scenario with CPU utilization at 90%, the proposed algorithm achieves  $U_{CPU} = 88\%$  by offloading some computational tasks to the ground control center, resulting in mAPE = 0.43 m. Conversely, the resource-unoptimized multimodal localization suffers from CPU overload, causing the detection frame rate to drop from 20 fps to 8 fps, with mAPE = 0.76 m. In the resource-abundant scenario of 20Mbps bandwidth and 30% CPU

load, the proposed algorithm selects the high-precision Faster R-CNN detector, achieving  $mAPE = 0.29$  m, with resource utilization efficiency  $\eta_B = 35\%$  and  $U_{CPU} = 45\%$ , thus balancing accuracy and efficiency.

**Table 4.** Comparison of Resource Utilization Efficiency Indicators of Various Algorithms under Dynamic Resource Variation Scenarios (3 Drones, 10 Repeated Experiments).

| Resource State                                                | Algorithm                                                | $\eta_B$ (%) | $U_{CPU}$ (%) | Detector Type    | mAPE (m) |
|---------------------------------------------------------------|----------------------------------------------------------|--------------|---------------|------------------|----------|
| Bandwidth 2 Mbps<br>(Low Bandwidth)                           | Single Visual Localization                               | 95           | 62            | YOLOv5s          | 0.98     |
|                                                               | Multimodal Localization<br>without Resource Optimization | 120          | 78            | Faster R-CNN     | 0.89     |
|                                                               | CCM-SLAM                                                 | 105          | 68            | Monocular SLAM   | 1.05     |
|                                                               | Proposed Algorithm                                       | 82           | 48            | SSD MobileNet-v2 | 0.51     |
| CPU Load 90%<br>(High Load)                                   | Single Visual Localization                               | 75           | 95            | YOLOv5n          | 1.12     |
|                                                               | Multimodal Localization<br>without Resource Optimization | 88           | 100           | Faster R-CNN     | 0.76     |
|                                                               | CCM-SLAM                                                 | 82           | 98            | Monocular SLAM   | 1.23     |
|                                                               | Proposed Algorithm<br>(Task Migration)                   | 72           | 88            | YOLOv5s          | 0.43     |
| Bandwidth<br>20 Mbps + CPU Load 30%<br>(Sufficient Resources) | Single Visual Localization                               | 32           | 45            | YOLOv5l          | 0.65     |
|                                                               | Multimodal Localization<br>without Resource Optimization | 48           | 65            | Faster R-CNN     | 0.41     |
|                                                               | CCM-SLAM                                                 | 42           | 58            | Monocular SLAM   | 0.72     |
|                                                               | Proposed Algorithm                                       | 35           | 45            | Faster R-CNN     | 0.29     |



**Figure 8.** Performance Comparison of Algorithms in Dynamic Resource Utilization.

In terms of positioning accuracy and robustness, under dynamically changing resource scenarios, the proposed algorithm achieves an average  $mAPE$  of  $0.41$  m ( $\sigma = 0.11$  m), which is lower than all benchmark algorithms. Additionally,  $\sigma APE = 0.11$  m indicates that the algorithm maintains stable positioning accuracy even during resource fluctuations. The detector switching success rate  $R_{switch}$  is  $99\%$ , with a switching time  $T_{switch}$  of  $85$  ms ( $\sigma = 12$  ms), allowing for rapid adaptation to changes in resource status. The scene adaptability score  $S_{adapt}$  is  $0.91$ , higher than other scenarios, validating the effectiveness of the dynamic resource optimization algorithm.

## 5. Conclusions

To address the challenges of low positioning accuracy and inefficient resource utilization in collaborative target localization by unmanned aerial vehicles (UAVs) in complex environments, this paper proposes a cooperative localization algorithm integrating multimodal fusion and dynamic resource optimization. A technical framework is established under the theme of “multimodal data fusion–dynamic resource optimization–precise localization output.”

First, a standardized preprocessing workflow is designed to handle heterogeneous data from visual, radar, and lidar sensors, achieving data synchronization and normalization. Specific network layers are employed to extract modality-specific features, and an attention-based fusion mechanism is introduced to overcome the limitations of individual sensing modalities.

For resource optimization, the algorithm incorporates a real-time resource-aware strategy combined with an offline-evaluated detector pool to select the optimal detector and transmission path. An integer linear programming model is applied to dynamically adjust the allocation of computational and communication resources, thereby reducing bandwidth consumption and improving resource efficiency.

To ensure precise localization output, multimodal fused features are fed into an extended Kalman filter for state prediction and update, reducing temporal inconsistency errors. A multi-UAV collaborative verification mechanism is further introduced to maintain consensus across the swarm. Finally, a 3D grid-based probability map is generated, compatible with ground control station software, to provide localization support for downstream tasks.

A comparative analysis of related studies against our proposed framework is summarized in Table 5. While prior research has made significant progress in specific areas—such as low-cost system design [9], UAV-USV cooperation [4], and visual collaboration [2]—several gaps remain prominent. Firstly, many studies rely on a limited set of homogeneous sensors, which restricts their robustness in diverse and challenging environments like marine or occluded urban settings. Secondly, the critical aspect of dynamic resource optimization under computational and communication constraints is largely unaddressed, limiting the practical scalability of these systems. Furthermore, none of the cited works incorporate a dedicated, multi-stage collaborative verification mechanism to ensure data consistency and mitigate outlier effects among UAVs, which is crucial for high-integrity positioning. Our study is designed to address these identified limitations comprehensively.

**Table 5.** Comparative Analysis of UAVs Cooperative Target Localization Approaches.

| Study/Paper            | Core Methodology                                                                         | Sensor Type(s)                 | Study Area and Test Environment                     | Localization Accuracy                                   | Multi-Stage Collaborative Verification                    |
|------------------------|------------------------------------------------------------------------------------------|--------------------------------|-----------------------------------------------------|---------------------------------------------------------|-----------------------------------------------------------|
| Li et al. [9]          | Hierarchical control & SLAM with COTS hardware.                                          | Vision, IMU                    | Controlled/Outdoor                                  | Not explicitly quantified (focus on system feasibility) | No                                                        |
| Akram et al. [4]       | UAV-USV visual data interaction for GNSS-denied localization.                            | Vision, GNSS                   | Marine                                              | <1.0 m                                                  | No                                                        |
| Rudol et al. [2]       | Fusion of visual detection results from multiple UAVs.                                   | Vision                         | Outdoor, partially occluded                         | Qualitative improvement (40% efficiency gain)           | No                                                        |
| Typical UWB-based [10] | UWB ranging for relative positioning.                                                    | UWB, IMU                       | GNSS-denied Indoor/Outdoor                          | ~0.1–0.3 m                                              | No                                                        |
| Our Proposed Method    | Multimodal fusion with CMA, dynamic resource optimization, and multi-stage verification. | Vision, LiDAR, Radar, UWB, IMU | Marine, Urban Occlusion, Dynamic Resource Scenarios | <0.5 m (mAPE); 45% error reduction vs. baseline         | Yes (Data association, EKF-weighted fusion, Grubbs' test) |

Experimental results demonstrate that the proposed algorithm effectively reduces localization error and enhances resource utilization in complex environments. Theoretically, this study refines the framework for multimodal fusion and resource-aware UAV cooperation. From an engineering perspective, the algorithm is implemented on low-cost hardware using open-source frameworks, lowering the barrier to deployment and supporting scalable applications. Future research will focus on further improving algorithmic

performance and extending its applicability across domains, thereby contributing reliable technical support for UAV swarm systems in challenging real-world scenarios.

Owing to the current limitations in experimental scale, the primary contribution of our present work lies in establishing a feasible system architecture and a preliminary software foundation for large-scale applications. Testing for network congestion, transmission latency, and large-scale fault recovery are explicitly identified as key directions for our future research.

**Author Contributions:** Conceptualization, H.L. and W.Z.; methodology, H.L.; software, Y.F.; validation, H.L. and Y.F.; formal analysis, Y.M.; investigation, H.L. and Y.F.; data curation, H.L. and Y.F.; writing—original draft preparation, H.L. and W.Z.; writing—review and editing, H.L. and Y.M.; visualization, Y.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation (NNSF) of China under Grant No. 62173336.

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

**Acknowledgments:** The authors would like to express their sincere appreciation to those who provided assistance throughout the course of this research.

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

## References

1. Bakirci, M.; Ozer, M.M. Post-disaster area monitoring with swarm UAV systems for effective search and rescue. In Proceedings of the 2023 10th International Conference on Recent Advances in Air and Space Technologies (RAST), Istanbul, Turkey, 7–9 June 2023; pp. 1–6.
2. Rudol, P.; Doherty, P.; Wzorek, M.; Sombattheera, C. UAV-Based Human Body Detector Selection and Fusion for Geolocated Saliency Map Generation. *arXiv* **2024**, arXiv:2408.16501. [\[CrossRef\]](#)
3. Manley, J.E.; Puzzuoli, D.; Taylor, C.; Stahr, F.; Angus, J. A New Approach to Multi-Domain Ocean Monitoring: Combining UAS with USVs. In Proceedings of the OCEANS 2025 Brest, Brest, France, 16–19 June 2025; pp. 1–5.
4. Akram, W.; Yang, S.; Kuang, H.; He, X.; Din, M.U.; Dong, Y.; Lin, D.; Seneviratne, L.; He, S.; Hussain, I. Long-Range Vision-Based UAV-Assisted Localization for Unmanned Surface Vehicles. *arXiv* **2024**, arXiv:2408.11429.
5. Hildmann, H.; Kovacs, E. Using unmanned aerial vehicles (UAVs) as mobile sensing platforms (MSPs) for disaster response, civil security and public safety. *Drones* **2019**, *3*, 59. [\[CrossRef\]](#)
6. Liu, X.; Wen, W.; Hsu, L.-T. GLIO: Tightly-coupled GNSS/LiDAR/IMU integration for continuous and drift-free state estimation of intelligent vehicles in urban areas. *IEEE Trans. Intell. Veh.* **2023**, *9*, 1412–1422. [\[CrossRef\]](#)
7. Zheng, Y.; Li, L.; Lin, W.; Liang, W.; Du, Q.; Han, Z. Resource Allocation Based on Optimal Transport Theory in ISAC-Enabled Multi-UAV Networks. *arXiv* **2024**, arXiv:2410.02122.
8. Bu, S.; Bi, Q.; Dong, Y.; Chen, L.; Zhu, Y.; Wang, X. Collaborative Localization and Mapping for Cluster UAV. In Proceedings of the 4th 2024 International Conference on Autonomous Unmanned Systems, Shenyang, China, 19–21 September 2024; Springer: Singapore, 2025; pp. 313–325.
9. Li, X.; Xu, K.; Liu, F.; Bai, R.; Yuan, S.; Xie, L. AirSwarm: Enabling Cost-Effective Multi-UAV Research with COTS Drones. *arXiv* **2025**, arXiv:2503.06890.
10. Shule, W.; Almansa, C.M.; Queralta, J.P.; Zou, Z.; Westerlund, T. UWB-based localization for multi-UAV systems and collaborative heterogeneous multi-robot systems. *Procedia Comput. Sci.* **2020**, *175*, 357–364. [\[CrossRef\]](#)
11. Wu, L. Vehicle-based vision–radar fusion for real-time and accurate positioning of clustered UAVs. *Nat. Rev. Electr. Eng.* **2024**, *1*, 496. [\[CrossRef\]](#)
12. Wang, X.; Peng, Y.; Shen, C. Efficient Feature Fusion for UAV Object Detection. *arXiv* **2025**, arXiv:2501.17983. [\[CrossRef\]](#)
13. Irfan, M.; Dalai, S.; Trslic, P.; Riordan, J.; Dooly, G. LSAF-LSTM-Based Self-Adaptive Multi-Sensor Fusion for Robust UAV State Estimation in Challenging Environments. *Machines* **2025**, *13*, 130. [\[CrossRef\]](#)
14. Peng, C.; Wang, Q.; Zhang, D. Efficient dynamic task offloading and resource allocation in UAV-assisted MEC for large sport event. *Sci. Rep.* **2025**, *15*, 11828. [\[CrossRef\]](#)
15. Alqefari, S.; Menai, M.E.B. Multi-UAV task assignment in dynamic environments: Current trends and future directions. *Drones* **2025**, *9*, 75. [\[CrossRef\]](#)

16. Qamar, R.A.; Sarfraz, M.; Rahman, A.; Ghauri, S.A. Multi-criterion multi-UAV task allocation under dynamic conditions. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 101734. [\[CrossRef\]](#)
17. Yang, T.; Wang, S.; Li, X.; Zhang, Y.; Zhao, H.; Liu, J.; Sun, Z.; Zhou, H.; Zhang, C.; Xu, K. LD-SLAM: A Robust and Accurate GNSS-Aided Multi-Map Method for Long-Distance Visual SLAM. *Remote Sens.* **2023**, *15*, 4442.
18. Weng, D.; Chen, W.; Ding, M.; Liu, S.; Wang, J. Sidewalk matching: A smartphone-based GNSS positioning technique for pedestrians in urban canyons. *Satell. Navig.* **2025**, *6*, 4. [\[CrossRef\]](#)
19. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [\[CrossRef\]](#)
20. Ramasubramanian, K.; Ginsburg, B. AWR 1243 sensor: Highly integrated 76–81-GHz radar front-end for emerging ADAS applications. In *Texas Instruments White Paper*; Texas Instruments: Dallas, TX, USA, 2017.
21. Gao, L.; Xia, X.; Zheng, Z.; Ma, J. GNSS/IMU/LiDAR fusion for vehicle localization in urban driving environments within a consensus framework. *Mech. Syst. Signal Process.* **2023**, *205*, 110862. [\[CrossRef\]](#)
22. Qu, S.; Cui, J.; Cao, Z.; Qiao, Y.; Men, X.; Fu, Y. Position estimation method for small drones based on the fusion of multisource, multimodal data and digital twins. *Electronics* **2024**, *13*, 2218. [\[CrossRef\]](#)
23. Kwon, H.; Pack, D.J. A robust mobile target localization method for cooperative unmanned aerial vehicles using sensor fusion quality. *J. Intell. Robot. Syst.* **2012**, *65*, 479–493. [\[CrossRef\]](#)
24. Guan, W.; Huang, L.; Wen, S.; Yan, Z.; Liang, W.; Yang, C.; Liu, Z. Robot Localization and Navigation Using Visible Light Positioning and SLAM Fusion. *J. Light. Technol.* **2021**, *39*, 7040–7051. [\[CrossRef\]](#)
25. Yan, Z.; Guan, W.; Wen, S.; Huang, L.; Song, H. Multirobot Cooperative Localization Based on Visible Light Positioning and Odometer. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 7004808. [\[CrossRef\]](#)
26. Huang, K.; Shi, B.; Li, X.; Li, X.; Huang, S.; Li, Y. Multi-Modal Sensor Fusion for Auto Driving Perception: A Survey. *arXiv* **2022**, arXiv:2202.02703. [\[CrossRef\]](#)
27. Li, S.; Chen, S.; Li, X.; Zhou, Y.; Wang, S. Accurate and automatic spatiotemporal calibration for multi-modal sensor system based on continuous-time optimization. *Inf. Fusion* **2025**, *120*, 103071. [\[CrossRef\]](#)
28. Lai, H.; Yin, P.; Scherer, S. Adafusion: Visual-lidar fusion with adaptive weights for place recognition. *IEEE Robot. Autom. Lett.* **2022**, *7*, 12038–12045. [\[CrossRef\]](#)
29. Queralta, J.P.; Taipalmaa, J.; Pullinen, B.C.; Sarker, V.K.; Gia, T.N.; Tenhunen, H.; Gabbouj, M.; Raitoharju, J.; Westerlund, T. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *IEEE Access* **2020**, *8*, 1000–1010. [\[CrossRef\]](#)
30. Chen, Y.; Inaltekin, H.; Gorlatova, M. AdaptSLAM: Edge-assisted adaptive SLAM with resource constraints via uncertainty minimization. In Proceedings of the IEEE INFOCOM 2023—IEEE Conference on Computer Communications, New York City, NY, USA, 17–20 May 2023; pp. 1–10.
31. He, Y.; Xie, J.; Hu, G.; Liu, Y.; Luo, X. Joint optimization of communication and mission performance for multi-UAV collaboration network: A multi-agent reinforcement learning method. *Ad Hoc Netw.* **2024**, *164*, 103602. [\[CrossRef\]](#)
32. Ben Ali, A.J.; Kouroshli, M.; Semenova, S.; Hashemifar, Z.S.; Ko, S.Y.; Dantu, K. Edge-SLAM: Edge-assisted visual simultaneous localization and mapping. *ACM Trans. Embed. Comput. Syst.* **2022**, *22*, 1–31. [\[CrossRef\]](#)
33. Koubâa, A.; Ammar, A.; Alahdab, M.; Kanhouh, A.; Azar, A.T. Deepbrain: Experimental evaluation of cloud-based computation offloading and edge computing in the internet-of-drones for deep learning applications. *Sensors* **2020**, *20*, 5240. [\[CrossRef\]](#)
34. Liu, X.; Wen, S.; Zhao, J.; Qiu, T.Z.; Zhang, H. Edge-assisted multi-robot visual-inertial SLAM with efficient communication. *IEEE Trans. Autom. Sci. Eng.* **2024**, *22*, 2186–2198. [\[CrossRef\]](#)
35. Hu, Y.; Peng, J.; Liu, S.; Ge, J.; Liu, S.; Chen, S. Communication-efficient collaborative perception via information filling with codebook. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 15481–15490.
36. Seçkin, A.Ç.; Karpuz, C.; Özek, A. Feature matching based positioning algorithm for swarm robotics. *Comput. Electr. Eng.* **2018**, *67*, 807–818. [\[CrossRef\]](#)

**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.