

Article

Lightweight Monocular Relative Pose Estimation of Spacecraft via Scale-Adaptive Multi-Task Learning

Zhiwei Hu , Yijie Zhang, Bowen Hou * and Guangzhen Yao

College of Science, National University of Defense Technology, Changsha 410073, China;
huzhiwei24@nudt.edu.cn (Z.H.); zyj_nudt@nudt.edu.cn (Y.Z.); yaoguangzhen@nudt.edu.cn (G.Y.)

* Correspondence: houbowen12@nudt.edu.cn

Abstract

Accurate monocular pose estimation of spacecraft is essential for on-orbit servicing and proximity operations, yet this remains challenging because of large variations in the target scale and limited onboard computational resources. To address these challenges, this paper proposes SAPose, a lightweight geometry-guided pose-estimation framework that integrates multi-task keypoint prediction with geometric pose recovery. The proposed network jointly performs spacecraft detection and semantic keypoint localization using a stage-aware lightweight backbone and a scale-adaptive feature fusion structure, thereby enhancing geometric feature representation across different target scales. For distant spacecraft occupying only a small portion of the image, an adaptive coarse-to-fine inference strategy selectively activates ROI-based secondary refinement to recover weakened structural details while avoiding unnecessary computation for medium- and large-scale targets. In addition, confidence-ranked keypoint selection and object-space nonlinear refinement are employed to improve the stability and accuracy of pose recovery. Experiments on the Spacecraft Pose Estimation Dataset (SPEED) and the Spacecraft Keypoint Dataset (SKD) demonstrate that SAPose achieves competitive pose estimation accuracy with a compact model size and efficient inference, providing a practical balance between accuracy and computational cost for monocular spacecraft pose estimation.

Keywords: spacecraft pose estimation; monocular vision; keypoint localization; multi-task learning; coarse-to-fine inference

1. Introduction

With the expansion of space activities and the steady growth of the on-orbit spacecraft population, accurate pose estimation for space targets has become increasingly important for autonomous rendezvous, close-proximity inspection, on-orbit servicing, and active debris removal [1]. In these close-proximity scenarios, the servicing spacecraft must rely on onboard perception systems to estimate the relative state of the target in real time. Among available sensing modalities, monocular cameras are widely adopted for spaceborne relative navigation and on-orbit servicing due to their low power consumption, compact form factor, and cost-effectiveness [2]. However, spaceborne visual perception remains challenging because large target-scale variations, complex illumination conditions, cluttered backgrounds, and limited onboard computational resources can significantly degrade image quality and feature representation. Under such conditions, traditional geometric feature-based matching methods often suffer from unreliable feature extraction and correspondence matching, which can cause divergence or failure in subsequent pose estimation [3].



Academic Editor: Xiaodong Shao

Received: 22 July 2026

Revised: 1 September 2026

Accepted: 2 September 2026

Published: 7 September 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

To address these limitations, convolutional neural network (CNN)-based approaches have attracted significant attention because of their strong representation capability [4]. Early studies predominantly adopted end-to-end direct regression paradigms. For example, SPN, proposed by Sharma et al. [5], attempts to learn a direct mapping from images to pose parameters. Direct pose regression does not expose intermediate geometric correspondences, which limits geometric interpretability. Moreover, rotation resides on the non-Euclidean manifold ($SO(3)$), whereas network outputs are typically represented in Euclidean space. This mismatch can introduce representation bias, which in turn may hinder convergence during training and reduce generalization [6]. Motivated by these issues, recent work has increasingly shifted toward hybrid frameworks: deep learning is used for feature extraction, while pose estimation is performed by integrating classical geometric constraints and optimization, thereby improving robustness while maintaining geometric consistency under challenging imaging conditions.

Many hybrid methods follow a multi-stage cascaded pipeline. First, an object detector localizes the target's 2D bounding box in the full field of view. Next, the region of interest (ROI) is normalized via cropping and resizing, and a dedicated keypoint regressor predicts predefined semantic landmarks. Finally, the relative pose is recovered by establishing 2D–3D correspondences and solving the Perspective-n-Point (PnP) problem, often combined with robust estimation and nonlinear refinement. Existing work demonstrates that this paradigm can achieve high accuracy while remaining interpretable. For example, Chen et al. [7] proposed a Faster R-CNN + HRNet cascade and achieved state-of-the-art performance in the ESA Pose Estimation Challenge. In addition, keypoint confidence filtering and uncertainty modeling, as explored by Piazza [8] and Zhong [9], have been shown to improve stability and robustness under occlusion, high-contrast illumination, and low-SNR conditions.

Despite their high accuracy, multi-stage cascaded architectures are difficult to deploy under the stringent size, weight, and power (SWaP) constraints of on-orbit platforms. Since detection and keypoint regression are usually performed by separate networks or sequential inference stages, these methods introduce considerable computational cost, memory access overhead, and end-to-end latency. To improve deployment efficiency, multi-task learning (MTL) has been increasingly explored for lightweight single-stage spacecraft pose estimation [10,11]. In an MTL framework, related tasks such as target detection and keypoint localization are optimized jointly through shared feature representations. The detection task provides global semantic cues that help suppress background interference, while the keypoint-regression task encourages the network to preserve local structural information required for geometric pose recovery. For example, Hu et al. [12] used segmentation masks to suppress unreliable predictions, and He et al. [13] adopted a shared encoder to simplify the pose-regression pipeline for onboard deployment. By sharing visual features and avoiding repeated high-resolution inference, single-stage multi-task methods can substantially reduce computational redundancy and improve inference efficiency.

Although single-stage models are more efficient, they often suffer noticeable accuracy degradation in long-range small-target scenarios. This is because cascaded methods rely on crop-and-resize to normalize the ROI before performing keypoint regression, whereas single-stage models lack this mechanism and must predict directly on global, low-resolution feature maps. As a result, their capacity to represent fine-grained details for small targets is limited, leading to a clear drop in keypoint localization accuracy. Moreover, lightweight networks are more prone to losing shallow-level details under aggressive downsampling, and the inherently low pixel occupancy of small targets makes it even harder to preserve the high-frequency texture and geometric cues required for precise keypoint localization. Accordingly, preserving fine geometric detail for small targets without sacrificing single-stage efficiency remains a key challenge.

The main contributions of this work are summarized as follows:

1. A lightweight indirect pose-estimation framework is developed for space targets. SAPose integrates spacecraft detection and 2D semantic keypoint regression into a unified multi-task network, providing reliable 2D–3D correspondences for subsequent geometric pose recovery with reduced computational redundancy.
2. A scale-adaptive multi-task network is proposed to improve keypoint localization under target-scale variations. The stage-aware lightweight backbone and multi-level feature-fusion structure enhance structural feature representation while maintaining a compact model size.
3. An occupancy-triggered adaptive coarse-to-fine inference strategy is developed for long-range small-target scenarios. By selectively performing secondary ROI-based inference only for extremely small targets, the proposed method improves keypoint localization while preserving inference efficiency.

The remainder of this paper is organized as follows. Section 2 reviews related work on spacecraft pose estimation and lightweight network design. Section 3 details the proposed SApOse framework, including the stage-aware lightweight backbone, the multi-level feature-fusion neck, the multi-task detection and keypoint heads, and the adaptive coarse-to-fine inference strategy. Section 4 presents extensive experiments on the SPEED and SKD benchmarks, including comparisons against state-of-the-art methods and ablation studies validating each component. Section 5 discusses error characteristics, background robustness, computational efficiency, and limitations. Section 6 concludes this paper and outlines directions for future work.

2. Related Work

2.1. Direct and Indirect Spacecraft Pose Estimation

Direct methods estimate the 6-DoF pose from the input image without explicit intermediate geometric representations. SPN [5] first demonstrated end-to-end pose regression for non-cooperative spacecraft. Subsequent work has improved direct estimation through multi-task auxiliary supervision (SPNv2 [14]), transformer-based feature extraction and bidirectional fusion (SPTN [15]), and balanced loss design (URSONet [16]). Because direct regression produces pose parameters without explicit intermediate correspondences, it provides less geometric interpretability and can be more sensitive to large attitude variations and domain shift.

Indirect geometric methods instead predict intermediate visual representations and recover the pose through classical solvers. Among them, keypoint-based approaches are most widely adopted: predefined 2D landmarks are associated with a known 3D spacecraft model and solved via PnP. Chen et al. [7] proposed a Faster R-CNN + HRNet cascade that achieved state-of-the-art performance in the ESA Pose Estimation Challenge. Piazza et al. [8] combined CNN-based keypoint prediction with geometric optimization for robust monocular relative pose estimation. More recent studies further strengthen the keypoint-PnP paradigm through confidence-based filtering [9], uncertainty-aware correspondence selection [17], and self-training with robust outlier rejection [18]. By explicitly enforcing 2D–3D projection consistency, indirect methods introduce geometric constraints that are absent from pure direct regression, at the cost of a more complex inference pipeline.

2.2. Lightweight and Scale-Adaptive Network Design

To reduce the computational overhead of cascaded pipelines, lightweight multi-task architectures have been explored for onboard spacecraft perception. SPNv2 jointly optimizes detection, segmentation, and keypoint localization through a shared multi-scale encoder [19]. Hu et al. [12] use segmentation masks to suppress unreliable keypoint predictions within a single-stage framework. Bechini and Lavagna integrate ROI estimation and

keypoint regression in one compact network [18]. Their method also uses confidence-based keypoint rejection and conditionally invokes a second ROI inference based on first-pass outputs [18]. Conditional second-pass inference and confidence filtering have therefore been explored previously; SApOse differs in combining an occupancy-based trigger with minimum-cardinality confidence-ranked selection and the stage-aware/SAFPN visual front end. He et al. [13] adopt a shared encoder to simplify the pose-regression pipeline for onboard deployment. These designs demonstrate that feature sharing across correlated tasks can substantially reduce redundant computation compared with sequential inference.

However, lightweight single-stage networks are more susceptible to feature degradation under aggressive downsampling, particularly for long-range small targets with low pixel occupancy. Multi-scale feature fusion is a common remedy. SPTN employs a pyramid-style transformer structure for multi-level representation learning [15]. ADSAN introduces adaptive dual-stream aggregation to enhance both local geometry and cross-level semantic interaction [20]. Chen et al. [10] explore single-stage homography-based keypoint regression with geometric constraints for planar spacecraft structures. For extremely small targets, lightweight single-pass predictions can still lose fine structural detail. Prior conditional second-pass schemes use ROI re-inference to increase the effective target resolution [18]. SApOse therefore focuses on an occupancy-based trigger that limits secondary inference to resolution-limited cases and explicitly evaluates the resulting accuracy–efficiency trade-off.

3. Materials and Methods

3.1. Framework Overview

As illustrated in Figure 1, the proposed framework consists of an offline geometric preparation stage and an online pose-estimation stage. In the offline stage, a sparse 3D landmark prior of the target spacecraft is reconstructed from a small set of manually annotated reference images with known relative poses. This step follows the standard landmark-based pose-estimation setting and is used only to provide fixed 3D keypoint coordinates and automatically generated 2D supervision labels.

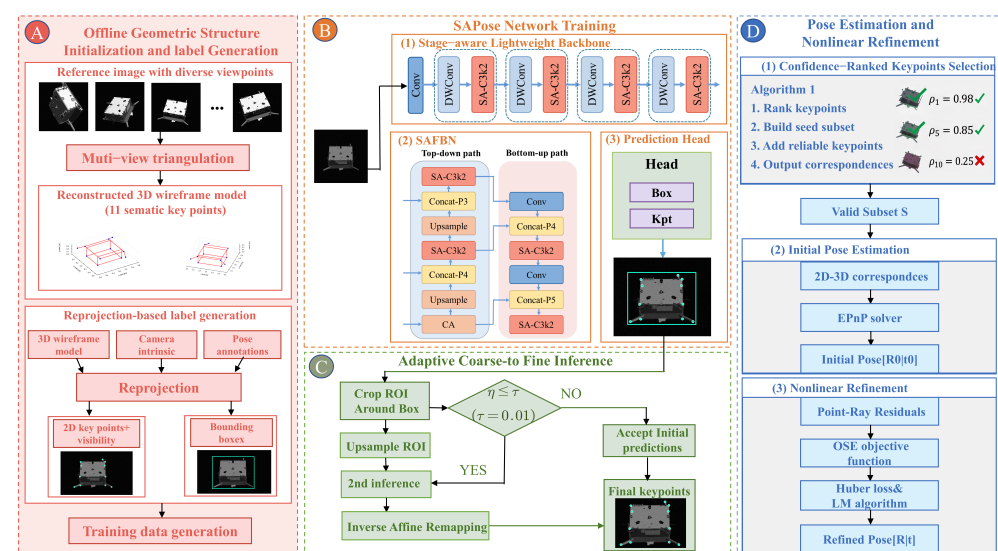


Figure 1. Overall pipeline of the proposed indirect pose-estimation framework: (A) offline geometric structure initialization and label generation; (B) SApOse network training; (C) adaptive coarse-to-fine inference; (D) pose estimation and nonlinear refinement.

In the online stage, the input monocular image is processed by SApOse, a unified single-stage multi-task network for spacecraft detection and keypoint localization. To improve localization accuracy for distant small-scale targets, an adaptive coarse-to-fine inference

strategy is further introduced. Finally, the 6-DoF pose is recovered from the predicted 2D–3D correspondences using PnP initialization followed by nonlinear refinement.

3.2. Coordinate Systems and Relative Pose Definition

To describe the imaging geometry of the target spacecraft, the following three Cartesian coordinate systems are defined, as illustrated in Figure 2.

Body Coordinate System (Frame B): This frame is strictly attached to the target spacecraft, with its origin located at the geometric center. In this study, the z_B -axis is defined as perpendicular to the solar panel plane, while the x_B - and y_B -axes form an orthogonal basis within the body plane.

Camera Coordinate System (Frame C): The origin of this frame is located at the optical center of the camera. The z_C -axis points forward along the optical axis, and the x_C - y_C plane is parallel to the image plane.

Image Coordinate System (Frame I): The origin is located at the top-left corner of the image. This frame is used to represent the 2D projected position of a keypoint, denoted by $u = [u, v]^T$ in pixels.

The pose of the spacecraft relative to the camera is described by a rotation matrix $R_{CB} \in \text{SO}(3)$ and a translation vector $t \in \mathbb{R}^3$. For any 3D semantic keypoint P_B defined in the body frame, its corresponding coordinate P_C in the camera frame can be expressed via a rigid transformation:

$$P_C = R_{CB}P_B + t \quad (1)$$

Based on the pinhole camera model, the point P_C is projected onto the 2D image plane according to the perspective projection equation:

$$s\tilde{u} = KP_C = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} P_C \quad (2)$$

where $\tilde{u} = [u, v, 1]^T$ represents the homogeneous pixel coordinates, s is the scale factor, and K is the camera intrinsic matrix containing the focal lengths (f_x, f_y) and the principal point (c_x, c_y). These equations establish the complete geometric mapping from 3D space to the 2D image, providing the theoretical basis for subsequent pose estimation using the Perspective-n-Point (PnP) algorithm.

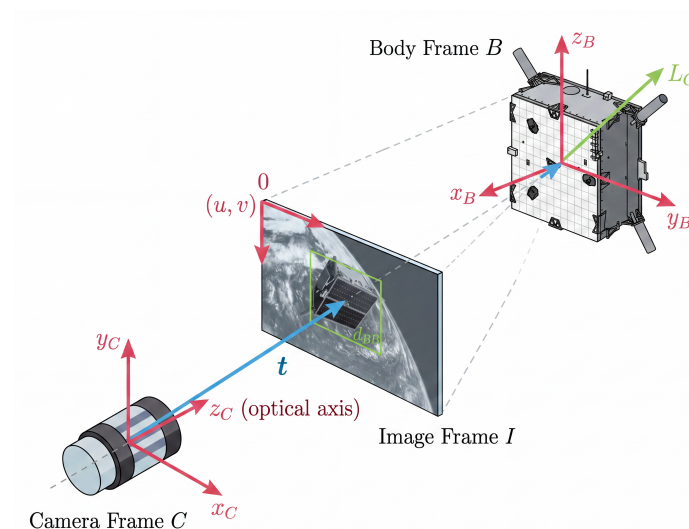


Figure 2. Coordinate systems and relative-pose geometry. B : target body frame; C : camera frame centered at the optical center; I : image plane.

3.3. Geometric Structure Initialization and Label Generation

Since the SPEED dataset provides relative pose annotations but does not include a canonical 3D keypoint model of the target spacecraft, we construct a sparse wireframe prior through an offline geometric initialization step. A total of 11 semantic landmarks are manually defined, including the eight visually prominent spacecraft corners and the center points at the tips of three antenna structures, following the principle that the selected landmarks should correspond to strong and repeatable visual features. As shown in Figure 3, these landmarks are annotated on a small number of training images with diverse viewpoints and known poses. The annotations are used only to reconstruct the fixed sparse 3D keypoint model, which is subsequently reprojected to generate 2D supervision labels for the remaining training images.

The 11 landmarks were selected using four criteria: (1) visual distinctiveness, so that each point corresponds to a recognizable physical structure; (2) spatial coverage, so that the correspondences are distributed over the spacecraft rather than concentrated in one local region; (3) geometric diversity, so that the set provides a wide baseline and redundant constraints for PnP-based pose recovery; and (4) viewpoint repeatability, so that the same semantic point can be identified consistently across observations. The eight body-corner landmarks provide broad coverage of the main spacecraft structure, while the three antenna-tip landmarks add asymmetric and spatially separated constraints outside the main body. The resulting 11-point set balances geometric redundancy against keypoint-prediction complexity. Section 4.6.4 evaluates alternative landmark configurations, partial landmark occlusion, and perturbations of the manually initialized annotations.

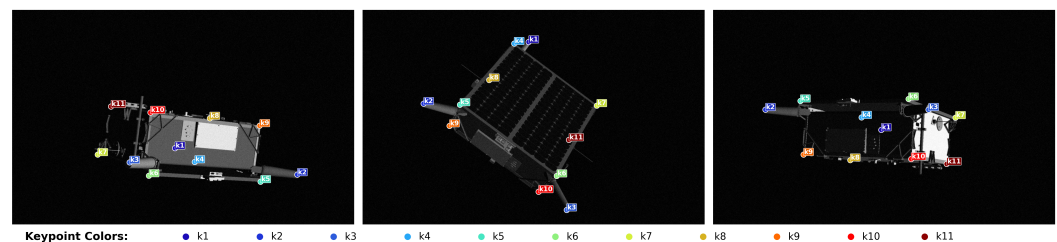


Figure 3. Reference images for semantic landmark annotation. Numbers 1–11 denote the 11 predefined semantic landmarks used to construct the sparse 3D spacecraft model.

Let $\tilde{\mathbf{u}}_j^{(i)} = [u_j^{(i)}, v_j^{(i)}, 1]^T$ denote the annotated image coordinate of the j -th landmark in the i -th reference image. Given the camera intrinsic matrix $\mathbf{K}$ and the relative pose annotation $[\mathbf{R}_{CB}^{(i)} | \mathbf{t}^{(i)}]$, the corresponding projection matrix is defined as

$$\mathbf{M}^{(i)} = \mathbf{K}[\mathbf{R}_{CB}^{(i)} | \mathbf{t}^{(i)}]. \quad (3)$$

The 3D position of each landmark is estimated by standard linear triangulation. Specifically, for the j -th landmark, the matrix $\mathbf{A}_j$ is constructed by stacking the DLT constraints from all selected views:

$$\mathbf{A}_j = \begin{bmatrix} u_j^{(1)} m_3^{(1)} - m_1^{(1)} \\ v_j^{(1)} m_3^{(1)} - m_2^{(1)} \\ \vdots \\ u_j^{(n)} m_3^{(n)} - m_1^{(n)} \\ v_j^{(n)} m_3^{(n)} - m_2^{(n)} \end{bmatrix}, \quad (4)$$

where $\mathbf{m}_r^{(i)}$ denotes the r -th row of $\mathbf{M}^{(i)}$. The homogeneous 3D coordinate is obtained as the right singular vector of $\mathbf{A}_j$ associated with the smallest singular value. After homoge-

neous normalization, the reconstructed landmark is represented as $\mathbf{P}_j = [X_j, Y_j, Z_j]^T$. The complete sparse model is denoted by

$$\mathcal{P}_B = \{\mathbf{P}_j\}_{j=1}^m, \quad (5)$$

where $m = 11$ in this work. The reconstructed sparse wireframe model is shown in Figure 4.

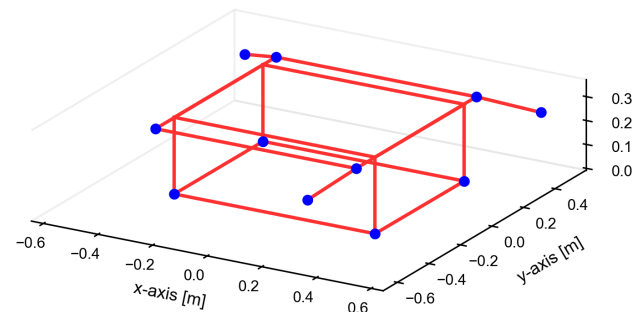


Figure 4. Reconstructed sparse 3D wireframe model with 11 landmarks.

The reconstructed 3D model $\mathcal{P}_B$ is then used to automatically generate training labels by reprojection. For a training image with known pose $[\mathbf{R}_{CB}|\mathbf{t}]$, each 3D landmark is projected as

$$\tilde{\mathbf{u}}_j \sim K[\mathbf{R}_{CB}|\mathbf{t}]\tilde{\mathbf{P}}_j. \quad (6)$$

The keypoint label is encoded as

$$\mathbf{k} = \{(u_j, v_j, \mu_j)\}_{j=1}^m, \quad (7)$$

where μ_j denotes the image-frame validity indicator. Specifically, $\mu_j = 1$ if the projected landmark lies inside the image boundary and $\mu_j = 0$ otherwise. This indicator is used to exclude out-of-frame landmarks during training.

The bounding box label $\mathbf{b} = [c_x, c_y, w, h]^T$ is generated from the valid in-frame projected landmarks. Let $x_{\min}$, $x_{\max}$, $y_{\min}$, and $y_{\max}$ denote the extremal coordinates of all valid in-frame landmarks. The box is enlarged by a scale factor $\lambda = 1.1$:

$$\mathbf{b} = \left[\frac{x_{\min} + x_{\max}}{2}, \frac{y_{\min} + y_{\max}}{2}, \lambda(x_{\max} - x_{\min}), \lambda(y_{\max} - y_{\min}) \right]^T. \quad (8)$$

This offline procedure produces a fixed 3D geometric prior and consistent 2D keypoint annotations for training SApOSE and for subsequent PnP-based pose recovery.

3.4. Scale-Adaptive Architecture for Spacecraft Pose Estimation

SAPOSE adopts a unified single-stage multi-task architecture for spacecraft detection and keypoint localization, which serves as the visual front-end for subsequent spacecraft pose estimation. The overall framework follows a backbone–neck–head paradigm, as illustrated in Figure 5, where the backbone is responsible for lightweight hierarchical feature extraction, the neck performs scale-adaptive multi-level feature fusion, and the prediction heads jointly perform spacecraft detection and 2D keypoint localization. The predicted bounding boxes and keypoints provide the visual measurements for subsequent geometric pose recovery.

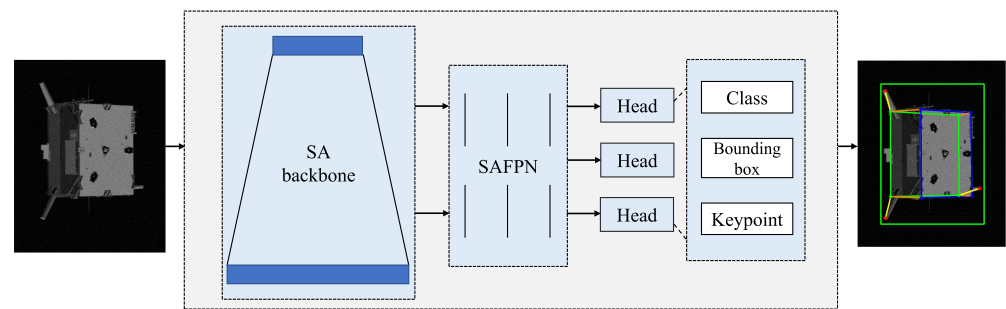


Figure 5. Overall SApOse architecture. A stage-aware lightweight backbone extracts hierarchical features, SAFP fuses multi-scale features, and task-specific heads predict the spacecraft class, bounding box, and 2D keypoints.

To balance keypoint localization accuracy and computational efficiency, we design a stage-aware lightweight backbone based on the proposed SA-C3k2 module, as shown in Figure 6. The backbone starts with an initial convolutional layer and contains four hierarchical stages. Each stage consists of a depthwise convolution (DWConv) layer followed by an SA-C3k2 block. The output feature maps from Stages 1 through 4 are denoted as $(C_1 \times \frac{H}{4} \times \frac{W}{4})$, $(C_2 \times \frac{H}{8} \times \frac{W}{8})$, $(C_3 \times \frac{H}{16} \times \frac{W}{16})$, and $(C_4 \times \frac{H}{32} \times \frac{W}{32})$, respectively. A final convolutional layer is applied after Stage 4 for feature refinement.

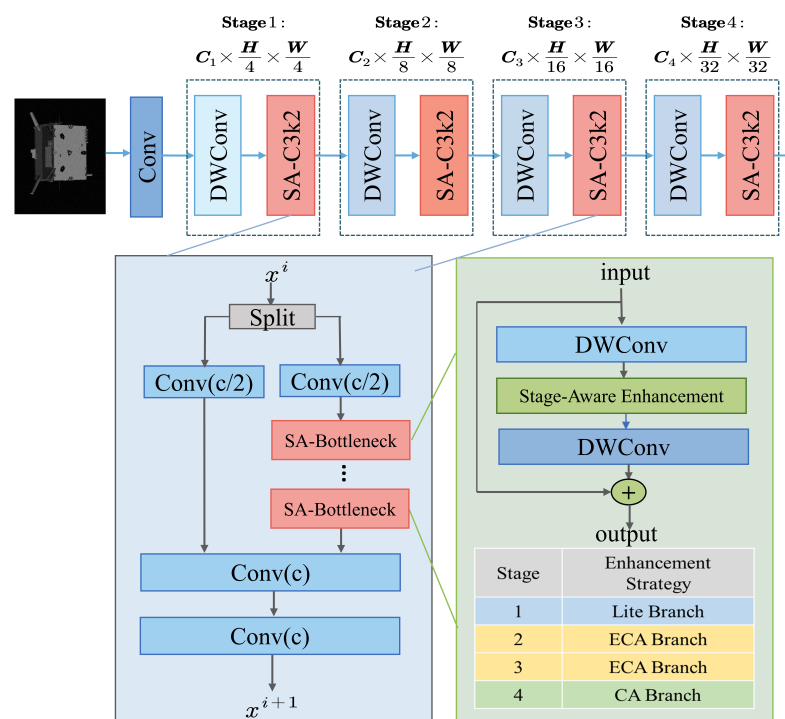


Figure 6. Stage-aware lightweight backbone. Each stage comprises a DWConv layer and an SA-C3k2 block; SA-C3k2 uses dual branches with stacked SA-Bottlenecks and stage-specific enhancement.

The SA-C3k2 block follows a dual-branch structure. Given an input feature x^i , it is split into two branches and transformed by convolutional layers with $c/2$ output channels. One branch is directly propagated, while the other is processed by stacked SA-Bottleneck modules. The two branches are then fused through convolutional layers to generate the output feature x^{i+1} .

As shown in Figure 6, each SA-Bottleneck consists of two DWConv layers, a stage-aware enhancement module, and a residual connection. Different enhancement branches are adopted at different stages: a Lite Branch for Stage 1, ECA Branches for Stages 2 and 3,

and a CA Branch for Stage 4. This design enables lightweight multi-scale feature extraction for spacecraft detection and keypoint localization.

To handle substantial target-scale variations caused by changes in observation distance, a scale-adaptive multi-level feature fusion structure is constructed based on the P3, P4, and P5 feature maps, as shown in Figure 7. The neck adopts a top-down and bottom-up feature aggregation strategy to progressively fuse spatial details and semantic information across different levels. Specifically, P3 preserves high-resolution spatial information for capturing fine-grained keypoint cues of distant small-scale spacecraft, P4 provides a balanced representation for medium-scale targets, and P5 delivers stronger semantic information for large-scale spacecraft under close-range observation. Based on the fused multi-level features, the prediction head performs two tasks: spacecraft detection and keypoint regression. The detection branch predicts bounding boxes and target confidence scores for object-level localization, while the keypoint regression branch estimates predefined structural keypoints to provide geometric constraints for pose estimation.

To jointly optimize the spacecraft detection and geometric structure estimation tasks, the total loss function $\mathcal{L}_{total}$ is formulated as a weighted combination of five independent sub-terms:

$$\mathcal{L}_{total} = \lambda_{box}\mathcal{L}_{box} + \lambda_{dfl}\mathcal{L}_{dfl} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{pose}\mathcal{L}_{pose} + \lambda_{kobj}\mathcal{L}_{kobj} \quad (9)$$

where $\lambda_{(\cdot)}$ denotes the weights used to balance the relative contributions of the sub-task losses during multi-task training.

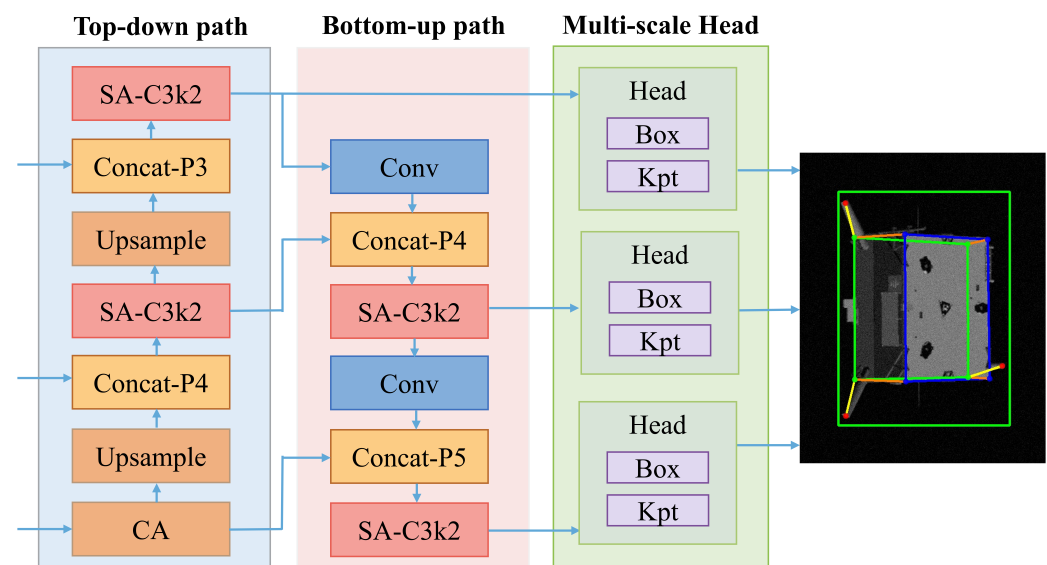


Figure 7. SApse neck and head structure. SAFPn fuses P3–P5 through top-down and bottom-up paths for small-, medium-, and large-scale targets. Right: C3k2, upsampling, and convolution blocks.

Bounding Box Regression Loss. To account for substantial variations in spacecraft aspect ratio across different viewing angles, the Complete Intersection over Union (CIoU) [21] is adopted to compute the bounding box loss $\mathcal{L}_{box}$. For the predicted bounding box $\mathbf{b}$ and the ground-truth bounding box $\mathbf{b}^{gt}$, it is defined as follows:

$$\mathcal{L}_{box} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v \quad (10)$$

where IoU represents the Intersection over Union, $\rho(\cdot)$ is the Euclidean distance between the center points of the two boxes, and c is the diagonal length of the smallest enclosing

rectangle. The parameter v measures the consistency of the aspect ratio, and α is a balancing coefficient; both are calculated as follows:

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2, \quad \alpha = \frac{v}{(1 - \text{IoU} + v)} \quad (11)$$

where w, h and w^{gt}, h^{gt} represent the width and height of the predicted and ground-truth boxes, respectively.

Distribution Focal Loss. Distribution Focal Loss (DFL) [22] is used to represent bounding-box boundary locations with discrete probability distributions. Given the ground-truth label y and its adjacent integers y_i, y_{i+1} (satisfying $y_i \leq y \leq y_{i+1}$), along with the corresponding predicted probabilities S_i, S_{i+1} , $\mathcal{L}_{dfl}$ is defined as follows:

$$\mathcal{L}_{dfl} = -((y_{i+1} - y) \log(S_i) + (y - y_i) \log(S_{i+1})) \quad (12)$$

Object Classification Loss. The classification loss $\mathcal{L}_{cls}$ uses binary cross-entropy (BCE) loss to distinguish foreground spacecraft targets from the deep space background.

Keypoint Regression Loss. The keypoint regression branch adopts Object Keypoint Similarity (OKS) [23] as the loss metric to achieve scale-invariant supervision. For N keypoints, the loss function is expressed as follows:

$$\mathcal{L}_{pose} = 1 - \frac{1}{N_{\text{valid}}} \sum_{i=1}^N \exp\left(-\frac{d_i^2}{2s^2k_i^2}\right) \cdot \mathbb{I}(\mu_i > 0) \quad (13)$$

where $N_{\text{valid}} = \sum_i \mathbb{I}(\mu_i > 0)$ is the number of valid in-frame keypoints, and $\mathbb{I}(\cdot)$ is the indicator function. d_i is the Euclidean distance between the predicted and ground-truth keypoints, s is the target scale factor, and k_i is the normalization constant for each keypoint.

Keypoint Validity Loss. $\mathcal{L}_{kobj}$ uses binary cross-entropy to supervise keypoint validity prediction within the image frame:

$$\mathcal{L}_{kobj} = -\frac{1}{N} \sum_{i=1}^N (\mu_i \log(\hat{\mu}_i) + (1 - \mu_i) \log(1 - \hat{\mu}_i)) \quad (14)$$

where μ_i is the ground-truth image-frame validity label and $\hat{\mu}_i$ is the predicted validity confidence of the i -th keypoint.

3.5. Adaptive Coarse-to-Fine Inference Strategy

Although the high-resolution P3 branch improves small-target representation, precise keypoint localization remains difficult when the spacecraft occupies only a very small image region because repeated downsampling weakens fine structural cues. We therefore introduce a conditional coarse-to-fine (CTF) inference strategy that applies ROI refinement only to extremely small targets.

Given an input image with size $W \times H$, SAPose first performs a standard full-image inference and predicts the spacecraft bounding box and keypoints. The relative target scale is measured by the bounding-box occupancy ratio:

$$\eta = \frac{A_{\text{bbox}}}{A_{\text{img}}} = \frac{w \cdot h}{W \cdot H}, \quad (15)$$

where w and h are the width and height of the predicted bounding box. Samples with $\eta < \tau$ enter ROI refinement, and τ is selected in Section 4.2 according to the accuracy–efficiency trade-off.

If $\eta \geq \tau$, the first-pass predictions are used directly. Otherwise, an ROI centered on the initial detection is cropped from the original image. For the final operating point $\tau = 0.010$, the ROI extends the first-pass bounding box by a fixed per-side factor $p = 2$. Let $B = (x_1, y_1, x_2, y_2)$, with $b_w = x_2 - x_1$ and $b_h = y_2 - y_1$; the crop is

$$\begin{aligned} x_1^c &= \max(0, x_1 - pb_w), & y_1^c &= \max(0, y_1 - pb_h), \\ x_2^c &= \min(W, x_2 + pb_w), & y_2^c &= \min(H, y_2 + pb_h). \end{aligned} \quad (16)$$

The cropped ROI is resized to 512×512 using the same preprocessing as the first pass and processed once more by SApPose. Enlarging the target's relative image area provides greater spatial support for fine keypoint localization. The CTF branch is non-iterative: η is evaluated only from the first-pass detection, so at most one additional network forward pass is performed.

Let (o_x, o_y) be the top-left coordinate of the cropped ROI in the original image, and let $\mathbf{p}'_k = [p'_x, p'_y]^\top$ denote a refined keypoint in the resized ROI. With resizing factors (s_x, s_y) from the original ROI to the network input, the keypoint is mapped back to full-image coordinates as

$$\mathbf{p}_k = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \frac{p'_x}{s_x} + o_x \\ \frac{p'_y}{s_y} + o_y \end{bmatrix}. \quad (17)$$

The first-pass prediction also serves as the fallback. If no valid first-pass box is available, ROI refinement is skipped; if the second pass does not return valid refined keypoints, the valid first-pass keypoints are retained. Thus, only targets satisfying $\eta < \tau$ incur the additional inference cost. The overall process is illustrated in Figure 8.

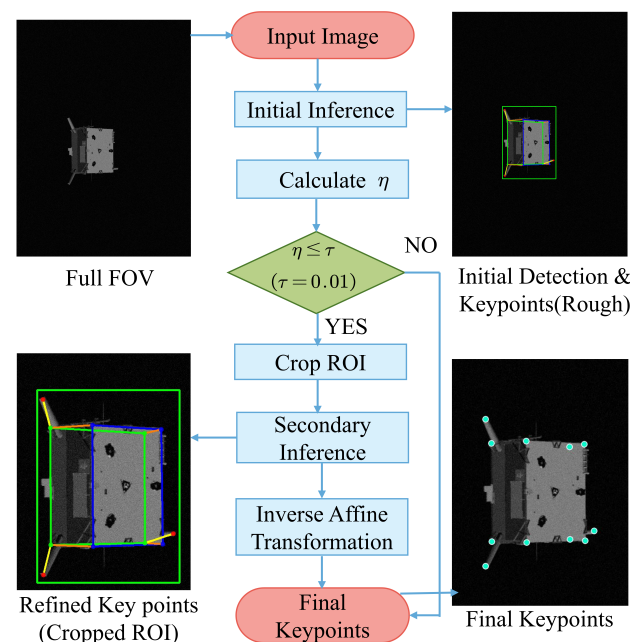


Figure 8. Conditional coarse-to-fine inference. Targets with $\eta < \tau$ undergo at most one ROI refinement pass, and the refined keypoints are mapped back to the full image. If refinement is unavailable or invalid, the valid first-pass prediction is retained.

3.6. Pose Estimation and Nonlinear Refinement

Given the reconstructed 3D wireframe model $\mathcal{P}_B$ with $m = 11$ keypoints and the predicted 2D image coordinates $\mathbf{u}_j$, the pose is recovered from 2D–3D correspondences. To

reduce the influence of low-confidence keypoints, we first construct a confidence-ranked correspondence subset.

Let ρ_j denote the confidence of the j -th keypoint, and let $\mathcal{L}$ denote the keypoint indices sorted such that $\rho_{\mathcal{L}_1} \geq \dots \geq \rho_{\mathcal{L}_m}$. The valid subset is

$$\mathcal{S} = \{\mathcal{L}_1, \dots, \mathcal{L}_{K_{\min}}\} \cup \{\mathcal{L}_k \mid k > K_{\min}, \rho_{\mathcal{L}_k} \geq \bar{\rho}\}. \quad (18)$$

This guarantees at least $K_{\min}$ correspondences while excluding additional low-confidence predictions.

An initial pose $Pose_{\text{initial}} = [R_0 \mid t_0]$ is obtained from $\mathcal{S}$ using EPnP. Nonlinear refinement then optimizes the six-dimensional parameter vector

$$\theta = \begin{bmatrix} \mathbf{r}^\top & \mathbf{t}^\top \end{bmatrix}^\top, \quad (19)$$

where $\mathbf{r} \in \mathbb{R}^3$ is an axis-angle rotation vector and $\mathbf{t} \in \mathbb{R}^3$ is the translation. The rotation matrix is obtained through the Rodrigues/exponential map,

$$R(\mathbf{r}) = \text{Rodrigues}(\mathbf{r}) = \exp([\mathbf{r}]_\times), \quad (20)$$

which keeps $R(\mathbf{r})$ on $\text{SO}(3)$ throughout refinement.

The pose is refined by minimizing the object-space point-to-ray error (OSE) [24]. For each $j \in \mathcal{S}$, the unit viewing ray is

$$\mathbf{v}_j = \frac{\mathbf{K}^{-1} \tilde{\mathbf{u}}_j}{\|\mathbf{K}^{-1} \tilde{\mathbf{u}}_j\|}, \quad \tilde{\mathbf{u}}_j = [\mathbf{u}_j^\top, 1]^\top. \quad (21)$$

The refinement objective is

$$L_{\text{ose}} = \sum_{j \in \mathcal{S}} \mathcal{L}_\delta(\|(R(\mathbf{r})P_j + \mathbf{t}) \times \mathbf{v}_j\|), \quad (22)$$

where P_j is the corresponding 3D landmark. Because $\mathbf{v}_j$ is a unit vector, the residual magnitude equals the distance from the transformed 3D point to its viewing ray.

For robust optimization, define

$$\mathbf{e}_j = (R(\mathbf{r})P_j + \mathbf{t}) \times \mathbf{v}_j, \quad r_j = \|\mathbf{e}_j\|. \quad (23)$$

A Huber penalty with $\delta = 0.02$ is used:

$$\mathcal{L}_\delta(r_j) = \begin{cases} \frac{1}{2}r_j^2, & r_j \leq \delta, \\ \delta\left(r_j - \frac{1}{2}\delta\right), & r_j > \delta. \end{cases} \quad (24)$$

The robust problem is solved by iteratively reweighted least squares (IRLS) with Levenberg–Marquardt (LM). At each outer iteration,

$$w_j = \begin{cases} 1, & r_j \leq \delta, \\ \frac{\delta}{r_j}, & r_j > \delta, \end{cases} \quad \tilde{\mathbf{e}}_j = \sqrt{w_j} \mathbf{e}_j, \quad (25)$$

and LM minimizes the stacked weighted residuals $\tilde{\mathbf{e}}_j$. The weights are recomputed after each pose update until convergence or the preset iteration limit is reached.

Unlike conventional reprojection refinement, which minimizes the image-plane discrepancy $\|\pi(RP_j + t) - u_j\|$, OSE minimizes the 3D point-to-ray distance. The distinction therefore lies in the geometric residual rather than in the LM optimizer.

The refined pose is finally obtained as $Pose_{\text{refined}} = [R | t]$.

4. Results

4.1. Experimental Settings

To evaluate the accuracy, cross-geometry adaptability, synthetic-to-real transfer behavior, and computational efficiency of the proposed method, experiments were conducted on two public spacecraft pose-estimation datasets with different target geometries and imaging characteristics. The datasets, implementation details, statistical protocol, and evaluation metrics are described below.

Datasets. Two public spacecraft pose-estimation datasets were adopted for evaluation, namely, SPEED and SKD. The SPEED dataset was used as the primary benchmark, while the SKD dataset was employed to evaluate the adaptability of the proposed framework to spacecraft with different structural geometries.

The **SPEED** dataset, introduced for the ESA Kelvins Pose Estimation Challenge, consists of photorealistic grayscale images of the Tango spacecraft. The public release contains 12,000 labeled synthetic training images and 2998 synthetic test images with withheld annotations, all at a resolution of 1920×1200 pixels. The spacecraft appears against either a deep-space or an Earth background, with substantial variations in target scale, attitude, illumination, and observation distance, as illustrated in Figure 9. In addition, the public release includes 300 real test images without publicly available pose labels and five labeled real-image examples. The five labeled examples are used in this study only for a small-sample out-of-domain evaluation.

Since the ground-truth annotations of the official synthetic SPEED test set are not publicly available, the 12,000 labeled synthetic images were randomly divided into training, validation, and testing subsets in a 10:1:1 ratio, yielding 10,000, 1000, and 1000 images, respectively. The training subset was used for network optimization, the validation subset was used exclusively for hyperparameter selection, and the held-out testing subset was used only for final quantitative evaluation.

For the synthetic-to-real evaluation, the five labeled real images were kept separate from model training and hyperparameter selection. Models trained only on synthetic SPEED images were directly evaluated on these real images without fine-tuning or adaptation.

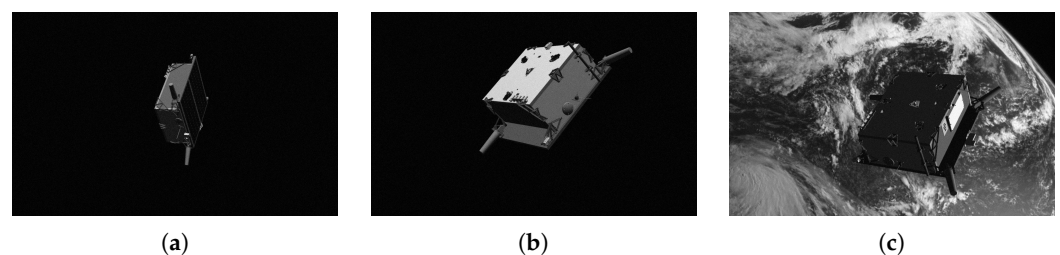


Figure 9. Examples from SPEED under varying illumination, background, distance, and viewpoint. (a) img001491. (b) img005179. (c) img014352.

The **SKD**, publicly available through SelectDataset, contains simulated images of three spacecraft models, namely, Satellite01, Satellite02, and Satellite03. The three targets exhibit different body geometries and appendage configurations, providing variations in structural appearance across the dataset. In total, the dataset includes 90,000 images annotated with ground-truth 6-DoF poses under different illumination and background conditions. For each spacecraft model, the images were divided into training, validation, and test subsets

using a 3:1:1 ratio. SAPose was trained and evaluated separately for each spacecraft model using the same network architecture and training configuration. Representative images of the three spacecraft models are shown in Figure 10.

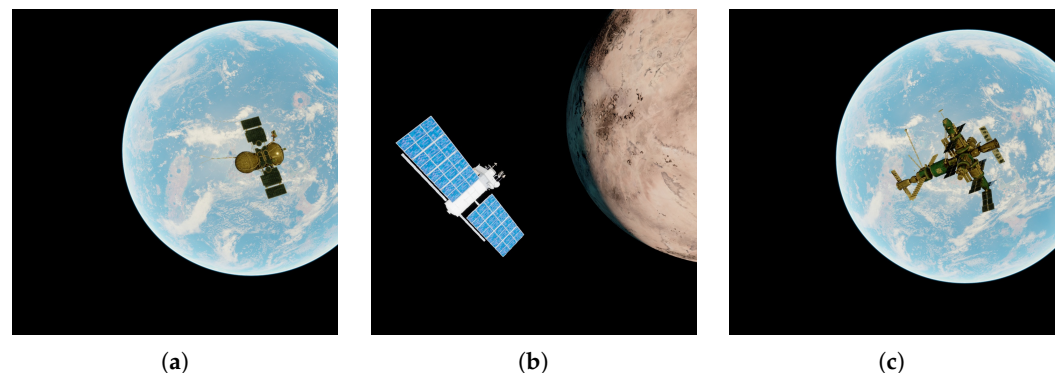


Figure 10. Spacecraft models in SKD. (a) Satellite01. (b) Satellite02. (c) Satellite03.

Implementation Details. All experiments were conducted on a single NVIDIA GeForce RTX 4060 Laptop GPU (8 GB) with an Intel® Core™ i9-14900HX CPU (2.2 GHz) and 32 GB RAM. The software environment consisted of Windows 11, Python 3.11.7, CUDA 11.8, PyTorch 2.1.0, and Ultralytics 8.3.99. The model was trained using the hyperparameters summarized in Table 1. Unless otherwise stated, the same configuration was used for both SPEED and SKD.

Statistical Protocol. For the primary SPEED experiments involving trainable configurations, training was repeated using five random seeds (0–4), and the corresponding results are reported as the mean $\pm$ standard deviation (SD). The same seed assignments were used for paired method comparisons whenever applicable.

All inference-stage hyperparameters were determined using the SPEED validation subset before final evaluation on the held-out test subset. The correspondence-selection parameters were first examined with the occupancy threshold fixed at the nominal value $\tau = 0.010$. This analysis resulted in a confidence threshold of $\bar{\rho} = 0.85$ and a minimum number of retained keypoints of $K_{\min} = 8$. After fixing these two parameters, the sensitivity of the coarse-to-fine inference strategy to τ was evaluated separately over a range of occupancy thresholds. The resulting accuracy–efficiency analysis identified $\tau = 0.010$ as the operating point used in the subsequent experiments. The complete hyperparameter-selection procedure is presented in Section 4.2.

For inferential comparisons on SPEED, statistical testing was performed on the dataset-level metric obtained from each independently trained checkpoint, with the training run treated as the statistical unit. OKS-mAP was used as the primary endpoint for the keypoint comparisons of SAPose with RTMO-S and ProbPose-S, whereas the overall pose error E was used for the spacecraft pose comparisons with Bechini & Lavagna and SPN. Two-sided paired t -tests were performed across the five matched runs ($n = 5$, $df = 4$), and 95% confidence intervals were computed for the mean paired differences. Holm correction was applied separately to the two keypoint comparisons and the two pose comparisons.

Ablation, occlusion, and other secondary analyses are reported descriptively using the mean $\pm$ SD and effect magnitude rather than additional hypothesis tests. Unless otherwise stated, the final SAPose configuration uses $\bar{\rho} = 0.85$, $K_{\min} = 8$, nonlinear object-space refinement, and the conditional coarse-to-fine inference threshold $\tau = 0.010$.

Table 1. Training hyperparameter configuration.

Parameter	Value
Optimizer	AdamW
Initial Learning Rate	1×10^{-3}
Learning Rate Scheduler	Cosine Annealing
Batch Size	24
Epochs	250
Input Image Size	512×512
Momentum	0.88
Weight Decay	3.8×10^{-4}
Data Augmentation	Mosaic 0.9, Hue/Saturation/Brightness

Evaluation Metrics. Both intermediate keypoint localization and final 6-DoF pose estimation are evaluated. PCK@0.05 and OKS-mAP are used to measure 2D keypoint localization accuracy, whereas normalized translation error, rotation error, and their combined pose error are used to evaluate the recovered spacecraft pose.

PCK@0.05. A predicted keypoint is considered correct when its Euclidean localization error does not exceed 5% of the maximum dimension of the corresponding target bounding box. For K evaluated keypoints, PCK@0.05 is defined as

$$\text{PCK@0.05} = \frac{1}{K} \sum_{k=1}^K \delta \left(\frac{\|\mathbf{p}_k - \hat{\mathbf{p}}_k\|_2}{\max(w, h)} \leq 0.05 \right), \quad (26)$$

where $\mathbf{p}_k$ and $\hat{\mathbf{p}}_k$ denote the ground-truth and predicted coordinates of the k -th keypoint, respectively, and w and h denote the target bounding-box width and height.

OKS-mAP. Following the standard COCO keypoint evaluation protocol, Object Keypoint Similarity (OKS) is defined as

$$\text{OKS} = \frac{\sum_k \exp \left(-\frac{d_k^2}{2s^2\kappa_k^2} \right) \delta(v_k > 0)}{\sum_k \delta(v_k > 0)}, \quad (27)$$

where d_k denotes the localization error of the k -th keypoint, s represents object scale, κ_k is the keypoint-specific normalization factor, and v_k denotes keypoint visibility. OKS-mAP is obtained by averaging AP over OKS thresholds from 0.50 to 0.95.

6D Pose Evaluation Metrics. Following the SPEED evaluation protocol [25], the normalized translation error and rotation error for image i are defined as

$$E_t^{(i)} = \frac{\|\mathbf{t}_{gt}^{(i)} - \mathbf{t}_{est}^{(i)}\|_2}{\|\mathbf{t}_{gt}^{(i)}\|_2}, \quad (28)$$

and

$$E_q^{(i)} = 2 \arccos \left(\left| \left\langle \mathbf{q}_{est}^{(i)}, \mathbf{q}_{gt}^{(i)} \right\rangle \right| \right), \quad (29)$$

where $\mathbf{t}_{est}^{(i)}$ and $\mathbf{t}_{gt}^{(i)}$ are the estimated and ground-truth translation vectors, and $\mathbf{q}_{est}^{(i)}$ and $\mathbf{q}_{gt}^{(i)}$ are the corresponding unit quaternions.

For N test images, the reported pose metrics are

$$E_t = \frac{1}{N} \sum_{i=1}^N E_t^{(i)}, \quad E_q = \frac{1}{N} \sum_{i=1}^N E_q^{(i)}, \quad E = \frac{1}{N} \sum_{i=1}^N (E_t^{(i)} + E_q^{(i)}). \quad (30)$$

4.2. Hyperparameter Selection

The main inference-stage hyperparameters were determined on the SPEED validation subset before the comparative and ablation experiments. The confidence threshold $\bar{\rho}$ and the minimum number of retained keypoints $K_{\min}$ control the confidence-ranked correspondence selection strategy, whereas the target-occupancy threshold τ determines when the adaptive coarse-to-fine strategy activates ROI-based secondary inference. Since these parameters operate at different stages of the inference pipeline, they were examined sequentially. The correspondence-selection parameters were first evaluated with $\tau = 0.010$ used as a nominal operating value. After $\bar{\rho}$ and $K_{\min}$ were fixed, a separate sensitivity analysis was conducted for τ to determine the final accuracy–efficiency operating point.

Keypoint-selection hyperparameters. As described in Section 3.6, the confidence-ranked correspondence selection procedure is controlled by $\bar{\rho}$ and $K_{\min}$. The confidence threshold $\bar{\rho}$ determines whether additional keypoints are retained according to their predicted confidence, whereas $K_{\min}$ ensures that a minimum number of 2D–3D correspondences is available for pose recovery.

A grid search was conducted over different combinations of $\bar{\rho}$ and $K_{\min}$ while the coarse-to-fine occupancy threshold was fixed at the nominal value $\tau = 0.010$. For each candidate pair, the predicted keypoints were first ranked in descending order of confidence. The $K_{\min}$ highest-confidence keypoints were retained unconditionally, after which the remaining keypoints satisfying $\rho_j \geq \bar{\rho}$ were added to the correspondence set. The resulting correspondences were then provided to EPnP followed by nonlinear object-space refinement.

The mean overall pose error on the validation set was adopted as the optimization criterion:

$$\bar{E} = \frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} (E_t^{(i)} + E_q^{(i)}), \quad (31)$$

where N_{val} denotes the number of validation images.

The confidence threshold was varied over $\bar{\rho} \in \{0.50, 0.55, \dots, 0.95\}$, while the minimum subset size was varied over $K_{\min} \in \{4, 5, \dots, 9\}$. The lower bound of $K_{\min}$ was set to four because EPnP requires at least four 2D–3D correspondences to compute a pose solution. However, using only the minimum number of correspondences provides limited geometric redundancy and may increase sensitivity to keypoint-localization errors or unfavorable spatial configurations.

The grid-search results are presented in Figure 11. Rejecting low-confidence keypoints generally reduces the pose-estimation error by preventing unreliable correspondences from entering the geometric solver. However, an excessively high confidence threshold may discard accurately localized keypoints and reduce the geometric redundancy of the correspondence set. The minimum-cardinality constraint alleviates this problem by ensuring that a sufficient number of correspondences are retained even when only a limited number of predictions exceed the confidence threshold.

Among the evaluated combinations, $\bar{\rho} = 0.85$ and $K_{\min} = 8$ achieved the lowest mean validation error. With $\tau = 0.010$ fixed, this configuration obtained $\bar{E} = 0.0354$, averaged over the five training seeds (0–4), and was therefore adopted for the subsequent inference-stage analysis.

Coarse-to-fine occupancy threshold. After fixing the correspondence-selection parameters at $\bar{\rho} = 0.85$ and $K_{\min} = 8$, the sensitivity of the adaptive coarse-to-fine strategy to the occupancy threshold τ was evaluated using the same validation subset and the same five trained checkpoints corresponding to seeds 0–4. The threshold determines whether the ROI-based secondary inference stage described in Section 3.5 is activated. For a fixed set of first-pass predictions, increasing τ enlarges the set of samples satisfying $\eta < \tau$ and

therefore increases the frequency of secondary inference. Consequently, τ directly controls the trade-off between the accuracy improvement provided by ROI refinement and the additional computational cost of a second forward pass.

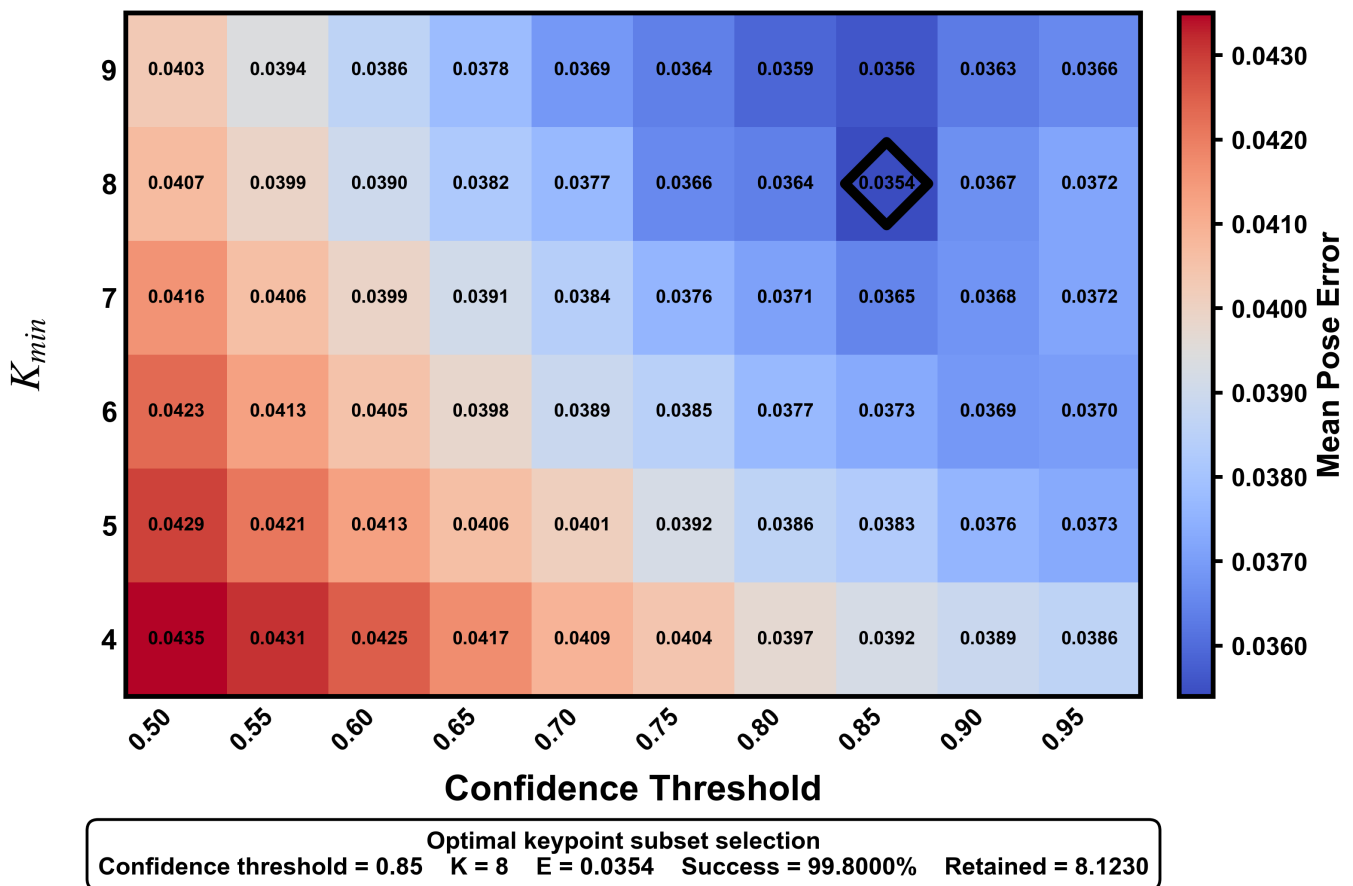


Figure 11. Validation mean pose error over keypoint-selection hyperparameters with $\tau = 0.010$. The axes denote confidence threshold $\bar{\rho}$ and minimum retained keypoints K_{min} ; the black diamond marks the selected optimum, $\bar{\rho} = 0.85$ and $K_{min} = 8$.

Five candidate values, $\tau \in \{0.0050, 0.0075, 0.0100, 0.0125, 0.0150\}$, were evaluated for each of the five checkpoints. For every candidate threshold, the overall validation pose error was first computed independently for each seed and then averaged across the five runs:

$$\bar{E}_{val}(\tau) = \frac{1}{5} \sum_{s=0}^4 E_{val}^{(s)}(\tau), \quad (32)$$

where $E_{val}^{(s)}(\tau)$ denotes the overall pose error obtained on the validation subset using the checkpoint trained with seed s . PCK@0.05 was used as a complementary keypoint-localization metric, while the secondary-inference trigger ratio and end-to-end latency were used to characterize the additional inference cost. The resulting validation statistics are summarized in Table 2.

The sensitivity analysis shows a clear change in the accuracy–efficiency trade-off around $\tau = 0.0100$. Increasing τ from 0.0050 to 0.0100 substantially improves pose-estimation accuracy: PCK@0.05 increases from 96.82% to 98.55%, while the overall pose error decreases from 0.0513 to 0.0354. Over the same range, the secondary-inference trigger ratio increases from 15.1% to 33.2%, and the mean latency increases from 10.74 ms to 12.50 ms.

Table 2. Validation-set sensitivity analysis of the target-occupancy threshold τ . Results are reported as the mean $\pm$ SD over five training seeds (0–4), with $\bar{\rho} = 0.85$ and $K_{\min} = 8$ fixed throughout the sweep.

τ	Trigger Ratio (%)	PCK@0.05 $\uparrow$	E_t $\downarrow$	E_q $\downarrow$	E $\downarrow$	Latency (ms) $\downarrow$
0.0050	15.1 $\pm$ 0.5	96.82 $\pm$ 0.24	0.0124 $\pm$ 0.0009	0.0389 $\pm$ 0.0025	0.0513 $\pm$ 0.0031	10.74 $\pm$ 0.27
0.0075	24.3 $\pm$ 0.7	97.96 $\pm$ 0.18	0.0099 $\pm$ 0.0007	0.0327 $\pm$ 0.0020	0.0426 $\pm$ 0.0025	11.61 $\pm$ 0.30
0.0100	33.2 $\pm$ 0.8	98.55 $\pm$ 0.14	0.0077 $\pm$ 0.0005	0.0277 $\pm$ 0.0012	0.0354 $\pm$ 0.0015	12.50 $\pm$ 0.34
0.0125	42.5 $\pm$ 1.0	98.61 $\pm$ 0.13	0.0073 $\pm$ 0.0005	0.0278 $\pm$ 0.0013	0.0351 $\pm$ 0.0016	13.39 $\pm$ 0.37
0.0150	51.0 $\pm$ 1.2	98.63 $\pm$ 0.12	0.0072 $\pm$ 0.0006	0.0278 $\pm$ 0.0014	0.0350 $\pm$ 0.0017	14.24 $\pm$ 0.40

Note: Bold values indicate the selected operating point. $\uparrow$ indicates that a higher value is better, whereas $\downarrow$ indicates that a lower value is better.

Further increasing τ provides only marginal accuracy gains. From $\tau = 0.0100$ to 0.0150, PCK@0.05 increases by only 0.08 percentage points and the overall pose error decreases by 0.0004, whereas the trigger ratio increases from 33.2% to 51.0% and the mean latency rises from 12.50 ms to 14.24 ms. The additional reduction in pose error is also smaller than the cross-seed variation observed at these operating points, indicating that the accuracy improvement beyond $\tau = 0.0100$ is limited relative to the additional inference cost.

Accordingly, $\tau = 0.010$ was adopted as the final operating point, providing a favorable balance between pose-estimation accuracy and computational efficiency. At this threshold, ROI-based secondary inference is activated when the predicted spacecraft bounding box occupies less than 1% of the input-image area. Together with $\bar{\rho} = 0.85$ and $K_{\min} = 8$, this setting defines the inference configuration used in the subsequent comparative and ablation experiments. The 33.2% trigger ratio and 12.50 ms latency reported here are validation-set means; the corresponding held-out SPEED test values reported later are 33.4% and 12.55 ms, respectively.

4.3. Comparison Results on SPEED

The SPEED comparison is organized into four parts: (1) a controlled comparison with lightweight keypoint estimation methods to evaluate 2D localization accuracy, (2) a controlled comparison with spacecraft pose-estimation methods to evaluate pose recovery performance, (3) a reference-level comparison with published SPEED leaderboard results, and (4) a qualitative comparison on representative scenarios.

4.3.1. Controlled Comparison with Keypoint Estimation Methods

Table 3 compares SAPose with representative recent lightweight keypoint estimation methods. All baseline methods were adapted to the spacecraft keypoint estimation task by modifying the output head to predict the same 11 semantic keypoints, and retrained from scratch using the same data split, input resolution (512×512), optimizer, learning rate schedule, and data augmentation strategy. Other training hyperparameters followed the respective original papers. At test time, the same conditional coarse-to-fine (CTF) inference policy with $\tau = 0.010$ was applied to SAPose and all retrained baseline methods, ensuring that the comparison was conducted under a common inference protocol.

SAPose attains the highest keypoint accuracy in this controlled comparison with a 9.4 M-parameter model, reaching 98.58 ± 0.13 PCK@0.05 and 94.03 ± 0.14 OKS-mAP. Relative to ProbPose-S (24.0 M), SAPose improves PCK by 0.96 percentage points and OKS-mAP by 0.87 percentage points while using fewer than 40% of the parameters. Relative to the similarly sized RTMO-S (9.9 M), the gains are 1.63 and 1.95 percentage points, respectively. ViTPose-S is lower by 2.20 percentage points in PCK and 2.89 percentage points in OKS-mAP under the same SPEED split and input-resolution protocol. These

results describe the behavior of the tested implementations under this controlled setting and should not be interpreted as a general limitation of Transformer architectures.

Table 3. Controlled comparison with recent lightweight keypoint estimation methods on the SPEED test subset ($n = 1000$). The architecture column identifies CNN, Transformer-based, and hybrid representatives. Trainable methods are reported as the mean $\pm$ SD over five independent runs, and the best entry per column is bolded. ViTPose-S provides a Transformer-based controlled baseline, while RTMO-S and ProbPose-S provide lightweight- and accuracy-oriented reference points, respectively.

Method	Architecture	Source	Params (M)	PCK@0.05 $\uparrow$	OKS-mAP $\uparrow$
YOLO-Pose-S	CNN	CVPRW 2022 [26]	~ 10.0	92.32 ± 0.09	85.27 ± 0.11
KAPAO-S	CNN-based	ECCV 2022 [27]	12.6	94.71 ± 0.14	88.06 ± 0.11
ViTPose-S	Vision Transformer	NeurIPS 2022 [28]	~ 22.0	96.38 ± 0.13	91.13 ± 0.14
RTMO-S	Hybrid/lightweight	CVPR 2024 [29]	9.9	96.95 ± 0.12	92.08 ± 0.10
ProbPose-S	Probabilistic/hybrid	CVPR 2025 [30]	~ 24.0	97.62 ± 0.15	93.16 ± 0.14
SAPose (Ours)	CNN + geometry	–	9.4	98.58 ± 0.13	94.03 ± 0.14

Note: $\uparrow$ indicates that a higher value is better. OKS-mAP is the primary keypoint endpoint for inferential analysis; significance for SAPose vs. RTMO-S and SAPose vs. ProbPose-S is reported in Table 4.

To quantify whether the two primary keypoint improvements are consistent across independent training runs, Table 4 reports paired tests on the dataset-level OKS-mAP obtained from the five matched seeds. The training run, rather than the individual test image, is used as the statistical unit. SAPose improves OKS-mAP over RTMO-S by a mean of 1.95 percentage points (95% CI [1.70, 2.20]) and over ProbPose-S by 0.87 percentage points (95% CI [0.56, 1.18]). Both primary comparisons remain statistically significant after Holm correction.

Table 4. Statistical analysis of the pre-specified primary keypoint comparisons on SPEED. OKS-mAP is computed once per trained checkpoint, and two-sided paired t -tests are performed across five matched training seeds ($n = 5$, $df = 4$). Positive differences indicate higher OKS-mAP for SAPose. Holm-adjusted p -values are reported for the two-comparison family.

Comparison	Mean Δ OKS-mAP (pp)	95% CI	p_{adj}
SAPose vs. RTMO-S	+1.95	[+1.70, +2.20]	5.1×10^{-5}
SAPose vs. ProbPose-S	+0.87	[+0.56, +1.18]	1.4×10^{-3}

4.3.2. Comparison with Representative Spacecraft Pose-Estimation Methods

Table 5 compares SAPose with representative spacecraft-specific pose-estimation methods covering direct pose regression, Transformer-based estimation, graph-based modeling, and keypoint-plus-geometry pipelines. Reproducible methods are retrained using the same SPEED data split, input resolution, and evaluation protocol. For reproducible keypoint-based methods, the same geometric pose recovery protocol is used where applicable to reduce differences caused by the downstream solver. Methods for which an equivalent controlled implementation is unavailable are reported using published results and marked with †.

Recent Transformer and hybrid approaches are included to broaden the comparison beyond conventional CNN baselines. In particular, Ye et al. represents a Transformer-based direct-regression approach, while ADSAN and Bechini & Lavagna represent hybrid learned-feature/geometric pipelines. Together with the controlled ViTPose-S experiment in Table 3, these comparisons provide complementary reference points for Transformer-based and hybrid architectures. Literature-only entries are used for contextual comparison and are excluded from statistical significance testing.

Table 5. Controlled results for representative spacecraft-specific pose-estimation methods on the SPEED held-out labeled subset ($n = 1000$), with additional literature-only reference values reported under their original protocols. Reproducible methods are reported as the mean $\pm$ SD over five independent training runs. The best reproducible result in each error column is bolded.

Method	Architecture (Paradigm)	Source	Params (M)	$E_t \downarrow$	$E_q \downarrow$	$E \downarrow$
SPN	CNN (direct regression)	TAES 2020 [5]	25.6	0.0218 ± 0.0015	0.0953 ± 0.0033	0.1171 ± 0.0036
Song et al. [†]	CNN (direct regression, FPGA)	Aerospace 2025 [31]	17.4	0.053	0.0942	0.1472
Balance-URSONet	CNN (direct regression)	Aerospace 2025 [16]	11.2	0.0421 ± 0.0010	0.1219 ± 0.0019	0.1640 ± 0.0021
Ye et al. [†]	Transformer (direct regression)	Remote Sens. 2023 [32]	186	0.0440	0.0275	0.0715
StructGCN [†]	GCN/hybrid (end-to-end)	CJA 2026 [33]	35.1	0.011	0.0276	0.0386
ADSAN [†]	Hybrid (keypoint + PnP)	Remote Sens. 2024 [20]	104.3	0.027	0.093	0.120
Bechini & Lavagna	Hybrid CNN + geometry (keypoint + PnP)	Acta Astro. 2025 [18]	11.5	0.0083 ± 0.0009	0.0296 ± 0.0021	0.0379 ± 0.0023
SAPose (Ours)	CNN + geometry (multi-task keypoint + PnP)	–	9.4	0.0075 ± 0.0004	0.0277 ± 0.0013	0.0352 ± 0.0014

Note: $E = E_t + E_q$ is computed per run before summarizing mean and SD. The downward arrow ($\downarrow$) indicates that lower values are better. [†] entries are literature-only and excluded from inferential statistics; E is the primary pose endpoint in Table 6.

Among the reproducible methods evaluated under the common protocol, SAPose achieves the lowest mean translation, rotation, and overall pose errors, with $E_t = 0.0075 \pm 0.0004$, $E_q = 0.0277 \pm 0.0013$, and $E = 0.0352 \pm 0.0014$. Compared with Bechini & Lavagna, the closest reproducible keypoint-plus-geometry baseline, SAPose reduces the mean translation error by approximately 9.6%, the rotation error by 6.4%, and the overall error by 7.1%, while reducing the parameter count from 11.5 M to 9.4 M.

The controlled direct-regression baselines exhibit larger errors under the same SPEED split. SPN obtains an overall error of 0.1171, while Balance-URSONet obtains 0.1640. Literature-only methods are included to indicate the broader accuracy range reported for spacecraft pose estimation, but direct numerical differences involving these entries are not interpreted as controlled performance gains because their original experimental conditions are not fully matched to the present evaluation.

To evaluate whether the principal improvements are consistent across independent training runs, two complementary comparisons are used for statistical analysis. Bechini & Lavagna represents the closest reproducible keypoint-plus-geometry competitor, whereas SPN provides a representative reproducible direct-regression baseline. The overall pose error E is used as the single statistical endpoint to avoid separate hypothesis tests on its translation and rotation components.

Two-sided paired t -tests are performed across the five matched training seeds ($n = 5$, $df = 4$), and 95% confidence intervals are computed for the mean paired differences. Holm correction is applied to the two-comparison family. The results are summarized in Table 6.

Both controlled comparisons show negative paired differences in the overall pose error. Relative to Bechini & Lavagna, SAPose achieves a mean reduction of 0.0027 in E , with the 95% confidence interval remaining below zero. A substantially larger difference is observed relative to SPN ($\Delta E = -0.0819$). The two comparisons remain statistically

significant after Holm correction, indicating that the observed improvements are consistent across the five matched training runs rather than being driven by a single training seed. Together with the mean $\pm$ SD results in Table 5, these tests provide statistical support for the primary controlled spacecraft-pose comparisons without extending hypothesis testing to literature-only methods or secondary experimental analyses.

Table 6. Statistical analysis of the two primary spacecraft-pose comparisons on SPEED. Differences are defined as SAPose minus the corresponding baseline; therefore, negative values indicate lower overall pose error for SAPose. Two-sided paired t -tests are performed across five matched training runs ($n = 5$, $df = 4$), with Holm-adjusted p -values.

Comparison	Mean ΔE	95% CI	p_{adj}
SAPose vs. Bechini & Lavagna	−0.0027	[−0.0049, −0.0005]	0.026
SAPose vs. SPN	−0.0819	[−0.0859, −0.0779]	1.1×10^{-6}

4.3.3. Reference Comparison with SPEED Leaderboard

For broader context, Table 7 lists representative results from the official SPEED leaderboard. Since SApOSE is evaluated on the held-out labeled subset (1000 images) rather than the official hidden test set (2998 images with withheld annotations), this comparison is intended solely as a reference and *no ranking conclusion is drawn*.

Because the evaluation sets differ, the leaderboard values are used only to contextualize the numerical scale of the pose metric and model size; cross-row differences are not interpreted as controlled performance gains or ranking evidence. SApOSE yields $E = 0.0352$ with 9.4 M parameters on the held-out labeled subset, while the official hidden-test entries span $E = 0.0094$ – 0.0571 with substantially different model sizes and evaluation conditions.

Table 7. Reference comparison with SPEED leaderboard methods (different test sets; no ranking intended). Leaderboard scores are taken from published challenge results on the official hidden test set.

Method	Params (M)	$E_t \downarrow$	$E_q \downarrow$	$E \downarrow$
UniAdelaide [7]	~49.8	0.00224	0.00716	0.0094
EPFL_cvlab [34]	~59.1	0.00562	0.01588	0.0215
pedro_fairspace [35]	~500	0.01363	0.04347	0.0571
SAPose (Ours) *	9.4	0.0075 ± 0.0004	0.0277 ± 0.0013	0.0352 ± 0.0014

Note: SApOSE is evaluated on the held-out labeled subset and is not directly comparable to the official hidden-test leaderboard entries. $\downarrow$ indicates that a lower value is better.

4.3.4. Qualitative Results

Five typical scenarios are selected: close-range, medium-range, long-range, near-Earth (cloud textures and illumination variation), and far-Earth (weak contrast against high-contrast clouds). These cover the primary challenges of scale variation and background interference (Section 5.1). Figure 12 compares RTMO-S (lightweight keypoint baseline), Bechini & Lavagna (spacecraft-specific competitor), and our SApOSE. All methods share the same 3D landmark model, PnP solver, and nonlinear refinement, thereby isolating differences in front-end keypoint prediction.

In the close-range case (row 1), all methods recover a reasonable wireframe. SApOSE yields the lowest errors ($E_t = 0.0118$, $E_q = 0.0185$) versus RTMO-S (0.0190, 0.0193) and Bechini (0.0172, 0.0200). Baseline wireframes drift along front-facing edges, whereas SApOSE remains tightly aligned.

For the medium-range case (row 2), rotation accuracy becomes the more demanding aspect. RTMO-S ($E_t = 0.0015$, $E_q = 0.0272$) and Bechini ($E_q = 0.0318$) produce visibly tilted

wireframes. SAPose reduces rotation error to 0.0099 rad ($E_t = 0.0026$), closely following the solar-panel edges.

In the challenging long-range case (row 3), the translation errors differ more strongly across methods than the rotation errors. RTMO-S ($E_t = 0.0155$) and Bechini (0.0078) show noticeable scale mismatch, while SAPose achieves $E_t = 0.0021$. The rotation errors remain similar (0.026–0.028 rad).

For the near-Earth case (row 4), the baseline wireframes show larger boresight offsets in the presence of cloud texture ($E_t > 0.024$). SAPose yields lower errors in this representative case ($E_t = 0.0010$, $E_q = 0.0114$).

In the far-Earth extreme case (row 5, P95 tail), none of the methods achieves full alignment under weak boundary contrast. Bechini achieves the lowest E_t (0.0242), while SAPose achieves the lowest E_q (0.0789) but higher E_t (0.0591). The larger translation error is consistent with increased depth sensitivity when the target occupies few pixels, but this single example does not isolate the underlying error source.

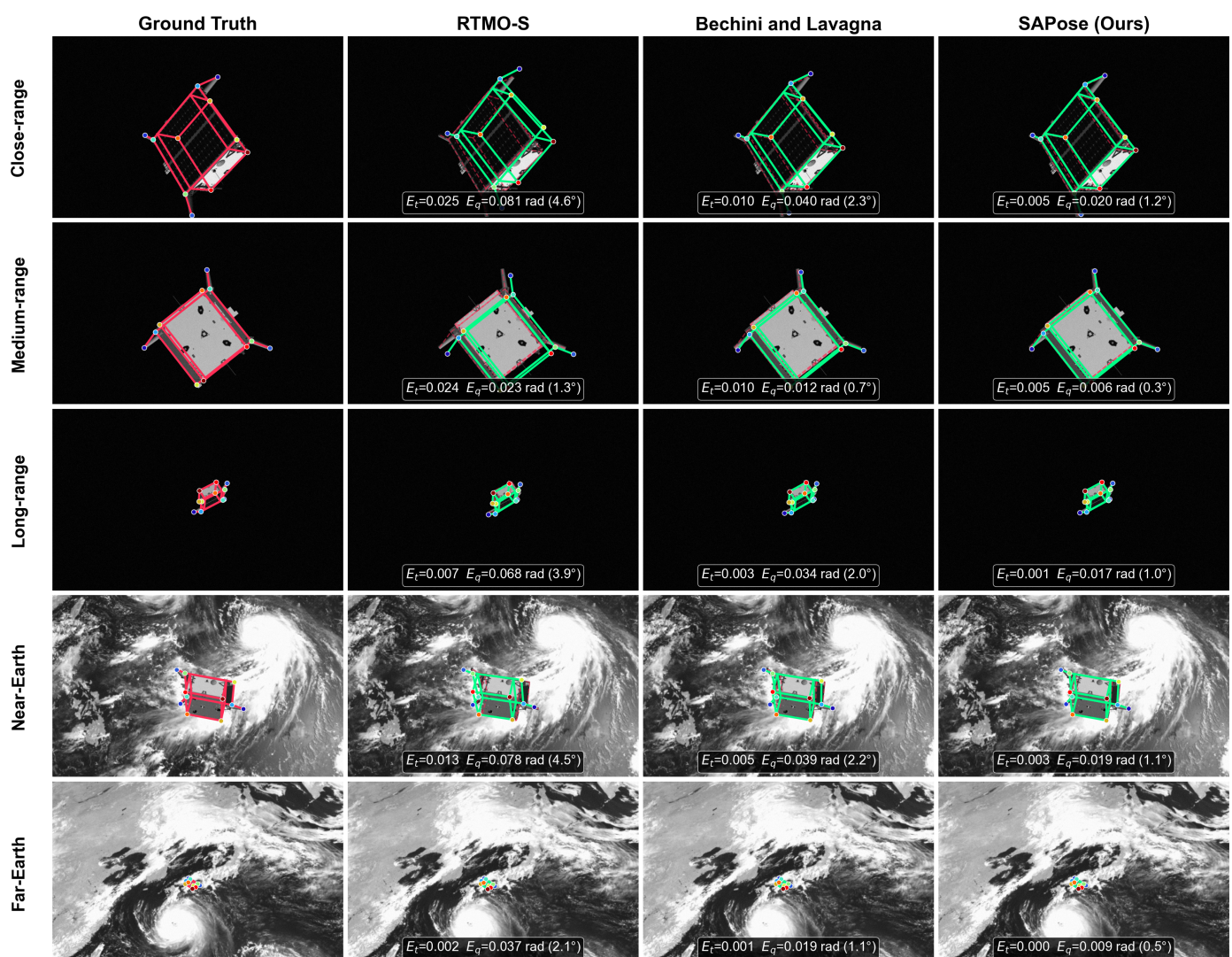


Figure 12. Qualitative comparison on five representative SPEED scenarios. Rows 1–5: close-range, medium-range, long-range, near-Earth, and far-Earth observations. Columns: ground truth, RTMO-S, Bechini & Lavagna, and SAPose. Red and green wireframes denote ground truth and estimated pose, respectively; colored dots mark predicted 2D keypoints. E_t and E_q (rad) are shown below each prediction.

4.4. Cross-Geometry Adaptability

Beyond the SPEED benchmark, the SKD dataset was further used to evaluate the adaptability of the proposed framework to spacecraft with different structural geometries. Unlike SPEED, which focuses on a single Tango spacecraft, SKD contains three targets, namely, Satellite01, Satellite02, and Satellite03, with distinct body shapes, solar-panel layouts, appendage configurations, and semantic keypoint distributions. The SKD imaging conditions are also more challenging, featuring stronger illumination variations, severe self-shadowing from appendages, and wider pose ranges that introduce substantial perspective distortion. Consequently, all methods produce markedly higher absolute pose errors on SKD than on SPEED (e.g., SAPose total error rises from 0.0352 on SPEED to 1.92–3.23 on SKD), making this dataset a demanding testbed for geometric adaptability across diverse target structures and imaging conditions.

For each target, the semantic landmarks were defined according to its own structural characteristics and expressed in the corresponding body frame. The model was trained and evaluated separately on each target using the same network configuration, training strategy, and PnP-based pose recovery pipeline. Since the 3D keypoint definitions are target-specific, this experiment evaluates target-wise adaptability rather than zero-shot cross-target generalization.

The quantitative results are reported in Table 8. Among the lightweight baselines with comparable parameter counts (~ 9 – 12 M), SAPose obtains the lowest total pose error on all three targets, indicating that its structure-aware design performs consistently across the separately trained spacecraft geometries.

GKNet [36] achieves the best overall performance across all targets (total errors of 1.3099, 2.1798, and 2.4631). Its graph convolutional decoder encodes the topological structure among keypoints as a learnable graph, enforcing geometric consistency constraints, such as fixed inter-keypoint distances and bilateral symmetry of the bus structure, that are difficult to capture with independent per-keypoint regression. These structural priors are especially beneficial when individual keypoint predictions become ambiguous under occlusion or illumination extremes. However, this comes at a substantial computational cost: GKNet uses $5.3\times$ more parameters (50 M vs. 9.4 M) and $3.3\times$ more FLOPs (71.6 vs. 21.6 GFLOPs) than SAPose, making it less attractive under tight onboard resource constraints.

Table 8. Controlled comparison on the three SKD spacecraft targets. SAPose is compared with two lightweight methods of comparable size and GKNet as a high-capacity reference.

Target	Method	Params (M)	GFLOPs	$E_t \downarrow$	$E_q \downarrow$	$E \downarrow$
Satellite01	GKNet	50.0	71.6	0.7629	0.5470	1.3099
	RTMO-S	9.9	24.1	1.1243	1.0613	2.1856
	Bechini & Lavagna	11.5	29.5	1.0426	1.0092	2.0518
	SAPose (Ours)	9.4	21.6	0.9914	0.9326	1.9240
Satellite02	GKNet	50.0	71.6	1.1634	1.0164	2.1798
	RTMO-S	9.9	24.1	2.0631	1.5643	3.6274
	Bechini & Lavagna	11.5	29.5	1.9534	1.5192	3.4726
	SAPose (Ours)	9.4	21.6	1.8742	1.3589	3.2331
Satellite03	GKNet	50.0	71.6	0.9782	1.4849	2.4631
	RTMO-S	9.9	24.1	1.3052	2.1531	3.4583
	Bechini & Lavagna	11.5	29.5	1.2647	2.0845	3.3492
	SAPose (Ours)	9.4	21.6	1.1936	1.9675	3.1611

Note: Bold values indicate the best result for each target. $\downarrow$ indicates that a lower value is better.

Representative keypoint detection results are shown in Figure 13. Despite the substantial geometric differences among the three satellites and the challenging imaging conditions, SAPose consistently localizes the predefined semantic landmarks, indicating its adaptability to different spacecraft structural configurations.

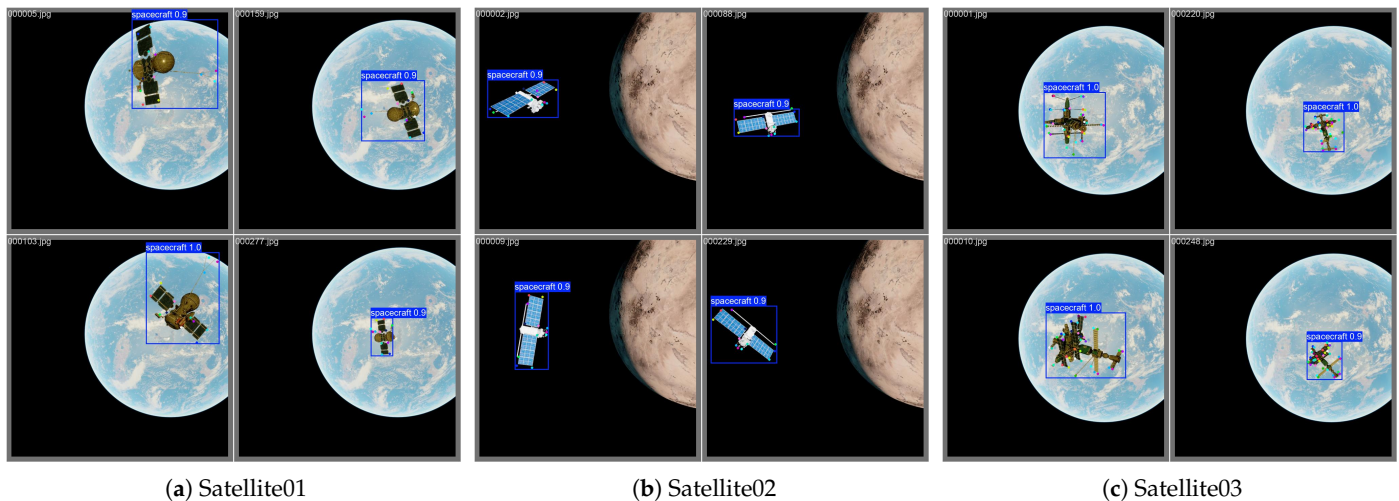


Figure 13. SAPose keypoint predictions on the three SKD spacecraft targets. Colored dots denote the predicted 2D semantic keypoints.

4.5. Cross-Domain Evaluation on Real SPEED Images

Both the SPEED training subset and SKD primarily consist of synthetic imagery. Because benchmark performance on synthetic imagery does not directly measure real-image transfer, we evaluate cross-domain performance separately on the available real SPEED images. Differences between synthetic and real images may arise from imperfect material and reflectance modeling, illumination and shadow statistics, specular reflections, sensor noise, exposure variation, image blur, lens effects, and background appearance. To provide an initial assessment of synthetic-to-real transfer, the SApOse model trained exclusively on the synthetic SPEED training subset is directly evaluated on the five real images distributed with the SPEED benchmark. None of these real images is used for network training, fine-tuning, landmark reconstruction, threshold selection, or model selection.

The five real SPEED images span target ranges from 3.28 m to 4.57 m and are captured under the hardware-in-the-loop experimental setting provided with the benchmark. Their Vicon-calibrated 6-DoF poses are used as reference poses for quantitative evaluation. Figure 14 presents the per-image qualitative and quantitative comparison among RTMO-S, Bechini & Lavagna, and SApOse. Across the five real images, SApOse obtains the lowest translation and rotation errors for every sample. In particular, SApOse achieves a median normalized translation error of $E_t = 0.012$ and a median rotation error of $E_q = 0.039$ rad, compared with 0.051/0.121 for RTMO-S and 0.029/0.077 for Bechini & Lavagna, respectively. The corresponding mean errors are $E_t = 0.011$ and $E_q = 0.032$ rad for SApOse, which are also lower than those of the two comparison methods, as summarized in Table 9.

These results are consistent with the relative performance observed on the synthetic SPEED test subset and show that the complete SApOse pipeline retains pose-estimation capability when transferred directly from synthetic training data to the available real SPEED images. Because the evaluation contains only five real images with pose annotations, it provides initial cross-domain evidence; broader real-image and on-orbit validation remains necessary.

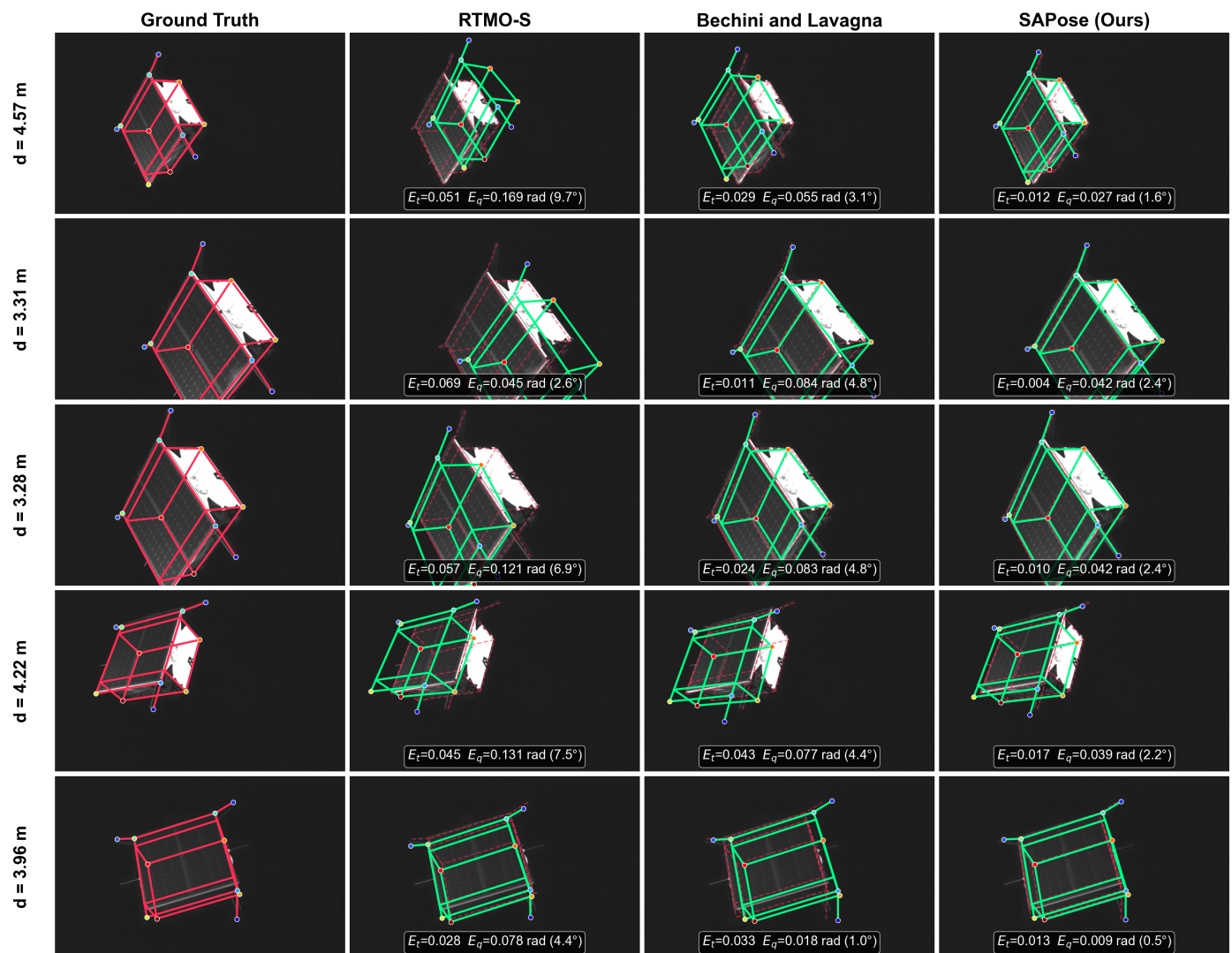


Figure 14. Synthetic-to-real evaluation on five real SPEED images. All methods are trained on synthetic SPEED data and evaluated without real-image fine-tuning or adaptation. Rows correspond to the five images; columns show ground truth, RTMO-S, Bechini & Lavagna, and SAPose. Red and green wireframes denote Vicon ground truth and estimated pose, respectively; E_t and E_q are shown below each prediction.

Table 9. Summary of pose-estimation errors on the five real SPEED images. All methods are evaluated without real-image fine-tuning or adaptation. E_t denotes the normalized translation error and E_q denotes the rotation error in radians.

Method	Median E_t	Median E_q	Mean E_t	Mean E_q
RTMO-S	0.051	0.121	0.050	0.109
Bechini & Lavagna	0.029	0.077	0.028	0.064
SAPose (Ours)	0.012	0.039	0.011	0.032

Note: Bold values indicate the best result in each column.

4.6. Ablation Study

To systematically evaluate the contribution and robustness of the proposed design, the ablation study is organized into four parts: (1) the individual and combined effects of the stage-aware backbone, SAFFPN feature fusion, confidence-ranked keypoint selection, and nonlinear refinement; (2) the internal enhancement allocation of the stage-aware backbone;

(3) the effectiveness of the adaptive coarse-to-fine inference strategy; and (4) sensitivity to landmark configuration, partial landmark occlusion, and uncertainty in the reconstructed 3D landmark prior. Unless otherwise specified, all experiments use the same SPEED data split, input resolution, training schedule, loss configuration, and pose recovery protocol. Trainable architectural variants are evaluated using five independent training seeds, whereas inference-only variants are applied to the same trained checkpoints. These secondary analyses are summarized using the mean $\pm$ SD and effect magnitude. The coarse-to-fine operating threshold is fixed at $\tau = 0.010$, as determined in Section 4.2.

4.6.1. Contribution of Major Components

The quantitative results of the major-component ablation are summarized in Table 10. We denote the stage-aware backbone as *A*, the SAFPN feature-fusion module as *B*, confidence-ranked keypoint selection as *C*, and nonlinear object-space refinement as *D*. The baseline adopts a plain C3k2 backbone and the original top-down feature-fusion structure, while all predicted valid keypoints are directly provided to EPnP without confidence-based correspondence selection or nonlinear refinement. To avoid confounding the architectural comparison with target-scale processing, the same coarse-to-fine inference policy with $\tau = 0.01$ is used for all variants in this table.

Components *A* and *B* modify the trainable visual front-end and therefore require independent training. In contrast, *C* and *D* operate only during geometric pose recovery and are evaluated using the same *A* + *B* checkpoints. Consequently, *C* and *D* do not alter the raw 2D keypoint predictions and therefore do not change PCK@0.05. The configurations {0, *A*, *B*, *A* + *B*} are used to examine the contribution and interaction of the two trainable components, whereas {*A* + *B*, *A* + *B* + *C*, *A* + *B* + *D*, *A* + *B* + *C* + *D*} isolates the effects of correspondence selection and nonlinear refinement.

Table 10. Ablation of the major SApPose components on SPEED. *A*: stage-aware backbone; *B*: SAFPN; *C*: confidence-ranked keypoint selection; *D*: nonlinear refinement. Results are reported as the mean $\pm$ SD over five runs. For *C* and *D*, the same trained *A* + *B* checkpoints are used because these components operate only during pose recovery.

Variant	A	B	C	D	Params (M)	PCK@0.05 $\uparrow$	E_t $\downarrow$	E_q $\downarrow$	E $\downarrow$
Baseline	–	–	–	–	8.7	97.36 $\pm$ 0.18	0.0098 $\pm$ 0.0009	0.0346 $\pm$ 0.0028	0.0444 $\pm$ 0.0034
+ <i>A</i>	✓	–	–	–	9.0	98.02 $\pm$ 0.15	0.0086 $\pm$ 0.0008	0.0315 $\pm$ 0.0024	0.0401 $\pm$ 0.0029
+ <i>B</i>	–	✓	–	–	9.0	98.05 $\pm$ 0.16	0.0083 $\pm$ 0.0007	0.0307 $\pm$ 0.0023	0.0390 $\pm$ 0.0027
<i>A</i> + <i>B</i>	✓	✓	–	–	9.4	98.58 $\pm$ 0.13	0.0078 $\pm$ 0.0007	0.0290 $\pm$ 0.0021	0.0368 $\pm$ 0.0025
<i>A</i> + <i>B</i> + <i>C</i>	✓	✓	✓	–	9.4	98.58 $\pm$ 0.13	0.0076 $\pm$ 0.0007	0.0285 $\pm$ 0.0020	0.0361 $\pm$ 0.0024
<i>A</i> + <i>B</i> + <i>D</i>	✓	✓	–	✓	9.4	98.58 $\pm$ 0.13	0.0075 $\pm$ 0.0007	0.0282 $\pm$ 0.0020	0.0357 $\pm$ 0.0024
<i>A</i> + <i>B</i> + <i>C</i> + <i>D</i>	✓	✓	✓	✓	9.4	98.58 $\pm$ 0.13	0.0075 $\pm$ 0.0004	0.0277 $\pm$ 0.0013	0.0352 $\pm$ 0.0014

Note: ✓ indicates that the corresponding component is enabled. Bold values indicate the complete SApPose configuration. $\uparrow$ indicates that a higher value is better, whereas $\downarrow$ indicates that a lower value is better.

Both trainable components improve the visual front-end independently. Relative to the baseline, introducing the stage-aware backbone increases PCK@0.05 by 0.66 percentage points and reduces the overall pose error from 0.0444 to 0.0401. SAFPN alone provides a similar improvement, increasing PCK by 0.69 percentage points and reducing *E* to 0.0390. Combining the two components further improves PCK to 98.58% and reduces *E* to 0.0368, corresponding to a 17.1% reduction relative to the baseline.

The combined improvement is slightly smaller than the arithmetic sum of the two individual gains, indicating a mildly sub-additive interaction. The *A* + *B* configuration nevertheless outperforms either *A* or *B* alone, showing that the two components are complementary in the tested configuration. A plausible explanation is partial overlap in the

multi-scale structural information enhanced by the two modules; this mechanism remains a hypothesis because the ablation varies modules rather than internal feature responses.

Starting from the common $A + B$ visual front-end, confidence-ranked correspondence selection reduces E from 0.0368 to 0.0361, whereas nonlinear object-space refinement reduces it to 0.0357. Using both components produces the lowest overall error of 0.0352. The approximately additive improvement is consistent with the two geometric components acting at different stages of pose recovery: C filters unreliable 2D–3D correspondences before pose initialization, whereas D refines geometric consistency after initialization.

4.6.2. Ablation of the Stage-Aware Backbone

The SA-C3k2 backbone assigns different enhancement mechanisms to different stages according to spatial resolution and semantic depth. To isolate this allocation choice, SAFPN, confidence-ranked keypoint selection, nonlinear refinement, and the CTF policy with $\tau = 0.010$ are kept enabled and fixed for all variants in this experiment; only the stage-wise enhancement mechanism is changed. The Plain C3k2 variant removes all stage-specific enhancement branches. All-Lite, All-ECA, and All-CA apply the same enhancement mechanism to all four stages, whereas the proposed stage-aware design adopts Lite in Stage 1, ECA in Stages 2–3, and CA in Stage 4.

The results in Table 11 show that the stage-aware allocation achieves the best overall performance while remaining more compact than the All-CA configuration. Uniformly applying Lite enhancement provides only a modest gain over Plain C3k2, whereas ECA produces a more noticeable improvement in both keypoint localization and pose accuracy. All-CA further improves the pose errors but requires additional parameters.

Table 11. Ablation of the stage-aware enhancement strategy. SAFPN, confidence-ranked keypoint selection, nonlinear refinement, and the CTF inference policy are enabled and fixed for all variants; only the stage-wise enhancement allocation is changed. Results are the mean $\pm$ SD over five independent runs.

Variant	Stage 1	Stage 2	Stage 3	Stage 4	Params (M)	PCK@0.05	E_t/E_q
Plain C3k2	None	None	None	None	9.0	98.05 $\pm$ 0.16	0.0080 $\pm$ 0.0007/0.0301 $\pm$ 0.0022
All-Lite	Lite	Lite	Lite	Lite	9.1	98.12 $\pm$ 0.15	0.0081 $\pm$ 0.0007 / 0.0299 $\pm$ 0.0021
All-ECA	ECA	ECA	ECA	ECA	9.3	98.41 $\pm$ 0.13	0.0078 $\pm$ 0.0006/0.0289 $\pm$ 0.0018
All-CA	CA	CA	CA	CA	9.8	98.49 $\pm$ 0.13	0.0077 $\pm$ 0.0005/0.0285 $\pm$ 0.0016
Stage-aware	Lite	ECA	ECA	CA	9.4	98.58 $\pm$ 0.13	0.0075 $\pm$ 0.0004/0.0277 $\pm$ 0.0013

Note: Bold values indicate the best result in each column.

The stage-aware configuration achieves the highest PCK@0.05 of 98.58% and reduces E_t/E_q to 0.0075/0.0277, while using 0.4 M fewer parameters than All-CA. Among the tested variants, it therefore provides the best accuracy–parameter trade-off. A plausible mechanism is that high-resolution shallow features benefit more from preserving local spatial detail, whereas deeper low-resolution features can make greater use of channel and coordinate-context modeling. This mechanism remains an interpretation because the ablation evaluates stage assignments rather than internal feature responses.

4.6.3. Effectiveness of the Adaptive Coarse-to-Fine Inference Strategy

The preceding experiments use the same conditional coarse-to-fine inference policy so that the effects of network and geometric components can be compared under a common operating condition. This subsection specifically isolates the contribution of the adaptive inference strategy using the same trained SAPose checkpoints.

Three inference modes are considered. SAPose-SI performs a single full-image inference. SAPose-AR performs ROI-based secondary refinement for every test sample. SAPose-CTF performs the second pass only when the predicted bounding-box occupancy

satisfies $\eta < \tau$. Therefore, the three configurations differ only in their test-time refinement policy and require no additional training.

For scale-wise analysis, the SPEED test samples are divided into four equal-frequency groups according to the normalized linear target scale

$$s_i = \sqrt{w_i h_i}, \quad (33)$$

where w_i and h_i denote the normalized width and height of the ground-truth bounding box. The groups are defined as

$$\begin{cases} s \leq \tau_1, & \text{Tiny,} \\ \tau_1 < s \leq \tau_2, & \text{Small,} \\ \tau_2 < s \leq \tau_3, & \text{Medium,} \\ s > \tau_3, & \text{Large,} \end{cases} \quad (34)$$

where τ_1 , τ_2 , and τ_3 correspond to the quartiles of the test-set target-scale distribution. These ground-truth scale groups are used only for post hoc performance analysis. The actual SApOse-CTF trigger is always determined from the predicted bounding-box occupancy ratio and does not use ground-truth target size.

Figure 15 compares the three inference policies under different target scales. For Tiny targets, single-pass inference shows the most pronounced degradation, with PCK@0.05 decreasing to 86.0%. Conditional ROI refinement increases PCK to 97.6% and reduces E_t and E_q by approximately 70.3% and 64.1%, respectively. The observed benefit becomes smaller as target size increases. For the Medium and Large groups, always-on refinement slightly degrades some of the measured aggregate metrics. A plausible explanation is that cropping and resizing an already well-resolved target introduces localization variation without a compensating resolution gain. The current experiment measures the performance trend but does not isolate this mechanism. SApOse-CTF retains the measured refinement benefit for small targets while avoiding unnecessary second-pass processing for most well-resolved samples.

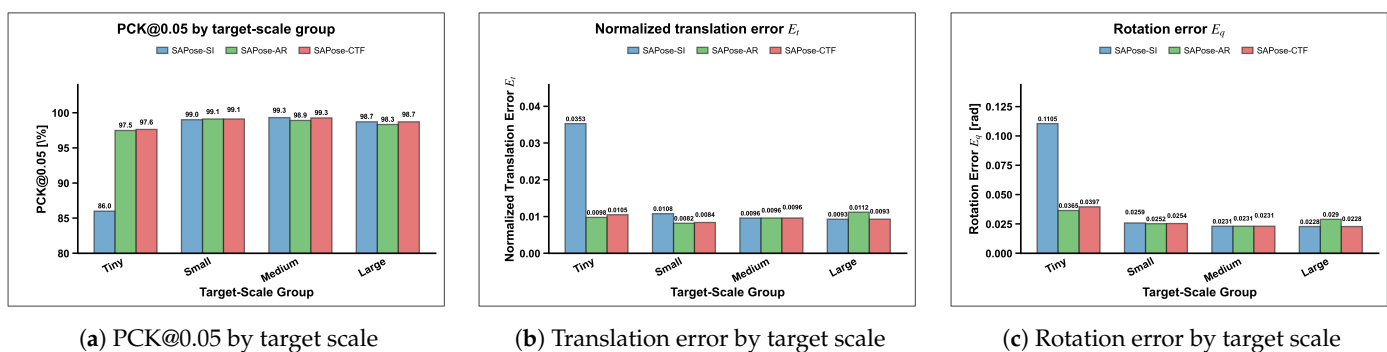


Figure 15. Performance comparison among SApOse-SI, SApOse-AR, and SApOse-CTF across different target-scale groups.

Table 12 summarizes the overall accuracy–efficiency trade-off. Relative to SApOse-SI, conditional refinement reduces the overall pose error from 0.0619 to 0.0352, corresponding to a 43.1% reduction, while increasing mean latency from 9.78 ms to 12.55 ms. Always-on refinement achieves a similar accuracy level but increases latency to 18.31 ms. SApOse-CTF therefore preserves most of the small-target benefit of ROI refinement while avoiding the cost of applying a second forward pass to every sample. It also slightly outperforms

SAPose-AR in the aggregate metrics, which is consistent with the scale-wise observation that secondary refinement can occasionally perturb already well-resolved predictions.

Table 12. Accuracy comparison of different SAPose inference configurations on the SPEED test subset. All three modes use the same five trained SAPose checkpoints and differ only in test-time inference.

Method	Inference Mode	PCK@0.05	E_t	E_q	E
SAPose-SI	Single inference	95.80 $\pm$ 0.32	0.0163 $\pm$ 0.0015	0.0456 $\pm$ 0.0037	0.0619 $\pm$ 0.0050
SAPose-AR	Full refinement	98.50 $\pm$ 0.16	0.0086 $\pm$ 0.0009	0.0283 $\pm$ 0.0024	0.0369 $\pm$ 0.0029
SAPose-CTF	Conditional refinement	98.58 $\pm$ 0.13	0.0075 $\pm$ 0.0004	0.0277 $\pm$ 0.0013	0.0352 $\pm$ 0.0014

Note: Bold values indicate the best result in each column.

To quantify the deployment-related cost of the conditional second pass without changing the evaluation protocol, Table 13 reports the trigger ratio, end-to-end latency, latency overhead, and model-memory footprint under the same 512×512 setting with a batch size of 1 used in Table 12.

For memory, we report the static FP32 weight footprint because it is directly determined by the model size and is independent of framework-specific allocator caching. With 9.4 M parameters, SAPose requires approximately 35.9 MiB for FP32 weights. The second stage reuses the same network instance sequentially, so conditional refinement does not duplicate model parameters or introduce additional static weight memory.

Table 13. Computational profile and second-stage overhead of the three SAPose inference modes on the SPEED test subset. FPS is reported together with the measured mean latency, and FP32 weight memory is derived from the 9.4 M-parameter model size.

Inference Mode	Trigger Ratio r	Mean Latency (ms)	FPS	Latency Overhead	FP32 Weight Memory
SAPose-SI	0%	9.78 $\pm$ 0.29	102.2 $\pm$ 3.1	0	35.9 MiB
SAPose-AR	100%	18.31 $\pm$ 0.51	54.6 $\pm$ 1.5	+8.53 ms (+87.2%)	35.9 MiB
SAPose-CTF	33.4%	12.55 $\pm$ 0.36	79.7 $\pm$ 2.3	+2.77 ms (+28.3%)	35.9 MiB

At the selected operating point, SAPose-CTF activates the second pass for only 33.4% of the SPEED test images. The measured mean latency increases from 9.78 to 12.55 ms, corresponding to an additional 2.77 ms or 28.3% relative to single-pass inference. In contrast, always-on refinement increases latency to 18.31 ms. Conditional refinement therefore reduces latency by 31.5% relative to always-on refinement while preserving the same 9.4 M model and 35.9 MiB FP32 weight footprint.

4.6.4. Landmark Configuration and Robustness Analysis

The robustness of the sparse geometric representation is evaluated from three complementary perspectives. First, the effect of landmark configuration is examined by comparing the eight body-corner landmarks with the complete 11-landmark set. Second, local image evidence around selected landmarks is synthetically removed at test time to evaluate robustness to partial landmark occlusion. Third, the manually annotated reference-image coordinates used for 3D triangulation are perturbed to assess sensitivity to inaccuracies in the reconstructed geometric prior.

Effect of landmark configuration. Two fixed landmark configurations are considered. Body-8 contains the eight body-corner landmarks, whereas Full-11 additionally includes the three antenna-tip landmarks. Since the eight body-corner landmarks are shared by both configurations, this comparison directly evaluates the effect of incorporating the additional antenna-tip landmarks into the sparse geometric representation. Each configuration is independently trained using the same five-seed protocol. Because the two configurations

contain different numbers of keypoints, PCK@0.05 is reported as a complementary localization metric, while the final pose errors are used as the primary basis for comparison. The comparison results are summarized in Table 14.

Table 14. Comparison of different semantic-landmark configurations on SPEED. Results are reported as the mean $\pm$ SD over five independent training runs.

Configuration	PCK@0.05	E_t	E_q	E
Body-8	98.05 $\pm$ 0.18	0.0084 $\pm$ 0.0008	0.0311 $\pm$ 0.0024	0.0395 $\pm$ 0.0030
Full-11	98.58 $\pm$ 0.13	0.0075 $\pm$ 0.0004	0.0277 $\pm$ 0.0013	0.0352 $\pm$ 0.0014

Compared with Body-8, Full-11 increases PCK@0.05 from 98.05% to 98.58% and reduces the overall pose error from 0.0395 to 0.0352, corresponding to a 10.9% reduction. Improvements are also observed in both translation and rotation estimation, with E_t decreasing from 0.0084 to 0.0075 and E_q decreasing from 0.0311 to 0.0277.

Because the eight body-corner landmarks are retained in both configurations, the observed improvement is consistent with the additional geometric information provided by the three antenna-tip landmarks. Their asymmetric and spatially separated locations extend the spatial coverage of the correspondence set and provide additional constraints for pose recovery. At the same time, Body-8 still maintains relatively stable pose-estimation performance, indicating that the framework does not depend exclusively on the antenna-tip landmarks.

Robustness to partial landmark occlusion. To evaluate the behavior of the complete Full-11 configuration when local landmark evidence is partially unavailable, local image regions around selected in-frame landmarks are synthetically masked at test time. No retraining is performed, so this experiment isolates the sensitivity of the trained inference pipeline to progressively reduced landmark visibility. The quantitative results under different occlusion levels are reported in Table 15.

Table 15. Robustness to partial landmark occlusion. Local 32×32 pixel regions centered on randomly selected in-frame landmarks are masked at test time. No retraining is performed.

Occluded Landmarks	PCK@0.05	E_t	E_q	E
0	98.58 $\pm$ 0.13	0.0075 $\pm$ 0.0004	0.0277 $\pm$ 0.0013	0.0352 $\pm$ 0.0014
1	98.09 $\pm$ 0.17	0.0082 $\pm$ 0.0008	0.0306 $\pm$ 0.0023	0.0388 $\pm$ 0.0029
2	97.18 $\pm$ 0.23	0.0098 $\pm$ 0.0010	0.0344 $\pm$ 0.0028	0.0442 $\pm$ 0.0036
3	95.61 $\pm$ 0.34	0.0121 $\pm$ 0.0013	0.0398 $\pm$ 0.0035	0.0519 $\pm$ 0.0045

For each test sample, local 32×32 pixel regions centered on randomly selected in-frame landmarks are replaced by a constant background-intensity patch. A fixed random seed is used to generate the masking assignments, and the same assignments are applied to all five checkpoints so that the comparison across checkpoints is not affected by different occlusion patterns.

Masking a single landmark decreases PCK@0.05 from 98.58% to 98.09% and increases the overall pose error from 0.0352 to 0.0388, indicating only a moderate loss in accuracy under isolated local occlusion. When two landmarks are masked, E increases to 0.0442, while masking three landmarks further increases it to 0.0519. The corresponding translation and rotation errors exhibit the same progressive degradation.

These results show that the complete pipeline retains useful pose-estimation capability when a limited amount of local landmark evidence is removed, but the degradation becomes increasingly pronounced as more landmarks are corrupted. The observed robustness under mild occlusion is consistent with the redundancy of the remaining correspondences

and the confidence-ranked correspondence-selection strategy. However, when several landmarks simultaneously lose reliable visual support, the remaining geometric constraints become less informative, and confidence-based selection alone cannot fully compensate for the loss of landmark evidence.

Sensitivity to landmark-initialization uncertainty. The previous experiment modifies the image evidence available to the network while keeping the 3D landmark prior unchanged. A complementary experiment is therefore conducted to evaluate sensitivity to uncertainty in the manually initialized geometric prior itself. Gaussian perturbations are introduced into the manually annotated reference-image coordinates before 3D triangulation, while the trained network and its predicted 2D keypoints are kept fixed. The results of this perturbation analysis are presented in Table 16.

Table 16. Sensitivity to perturbations of the manually initialized landmark annotations. Gaussian perturbations are applied to the annotated reference-image coordinates before 3D triangulation. The trained network and predicted 2D keypoints are kept fixed.

Annotation Noise σ (px)	3D Landmark RMSE (mm)	Mean Reproj. Error (px)	E_t	E_q	E
0	0.00	0.00	0.0075 ± 0.0004	0.0277 ± 0.0013	0.0352 ± 0.0014
1	0.34	0.82	0.0077 ± 0.0007	0.0291 ± 0.0021	0.0368 ± 0.0026
2	0.71	1.55	0.0085 ± 0.0008	0.0310 ± 0.0024	0.0395 ± 0.0030
3	1.18	2.31	0.0096 ± 0.0010	0.0342 ± 0.0028	0.0438 ± 0.0035

4.7. Failure Case Analysis

Although SAPose achieves low average pose error on the SPEED test set, a small fraction of samples remains in the high-error tail. To characterize these cases, Figure 16 presents eight representative examples covering four challenging imaging conditions: extreme illumination, severe occlusion, background clutter, and very small target scale. Two high-error examples are shown for each condition, together with their translation error E_t , rotation error E_q , and overall pose error $E = E_t + E_q$.

In each example, the yellow dashed wireframe represents the spacecraft geometry projected using the ground-truth pose, whereas the green solid wireframe visualizes the configuration of the 11 semantic keypoints predicted by SAPose. The additional vertices used to complete the reference wireframe are included only for visualization and are not network-predicted semantic landmarks. Compared with typical low-error samples, these cases exhibit larger discrepancies between the predicted keypoint configuration and the reference spacecraft geometry.

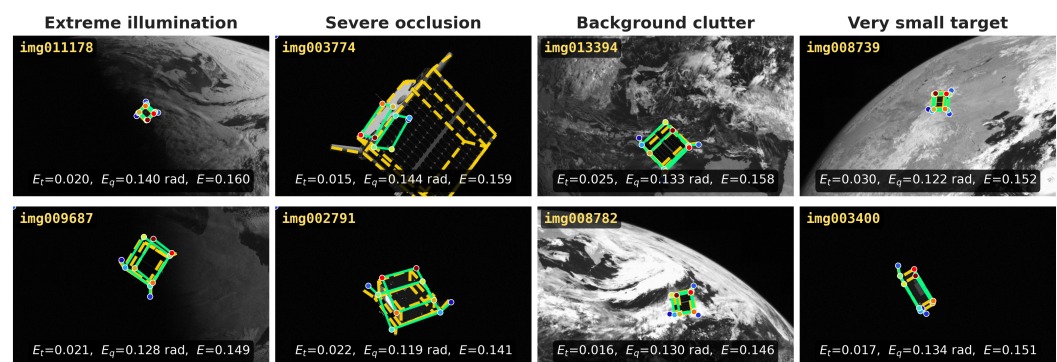


Figure 16. Representative SAPose failure cases on SPEED under extreme illumination, severe occlusion, background clutter, and very small target scale (columns). Yellow dashed and green solid wireframes denote the ground-truth projection and the predicted 11-keypoint configuration, respectively; E_t , E_q , and $E = E_t + E_q$ are shown below each image.

Across the eight representative failure cases, the overall pose error E ranges from 0.141 to 0.160. The rotation component is the dominant contributor, with E_q ranging from 0.119 to 0.144 rad, whereas E_t remains between 0.015 and 0.030.

These cases show that large pose errors recur under several challenging imaging conditions rather than a single failure regime. The observed associations are consistent with four mechanisms: illumination variation can weaken local structural cues, occlusion removes landmark evidence, background clutter increases local visual ambiguity, and very small targets reduce the pixel support available for separating neighboring semantic landmarks. These mechanisms motivate the targeted improvements discussed in Section 5.

5. Discussion

The preceding experiments characterize the accuracy, robustness, and computational cost of SAPose. This section analyzes the remaining error distribution, its association with background appearance, the overhead of conditional refinement, and the main limitations, and then it links these observations to targeted improvement directions.

5.1. Error Distribution and Tail Analysis

The comparison experiments show that SAPose achieves low average pose error on the SPEED test set. However, mean values alone do not indicate whether the remaining error is broadly distributed across the dataset or concentrated in a small number of difficult samples. We therefore examine both the component-wise error distributions and the empirical high-error tail.

Figure 17 shows the translation-error components in the camera coordinate system and the corresponding attitude-error components. Most components are concentrated near zero over the nominal region, with no pronounced systematic offset for the majority of test samples. Among the translation components, the boresight direction shows a broader spread and several larger deviations. This pattern is consistent with greater monocular depth uncertainty, but the distribution alone does not separate depth ambiguity from other error sources.

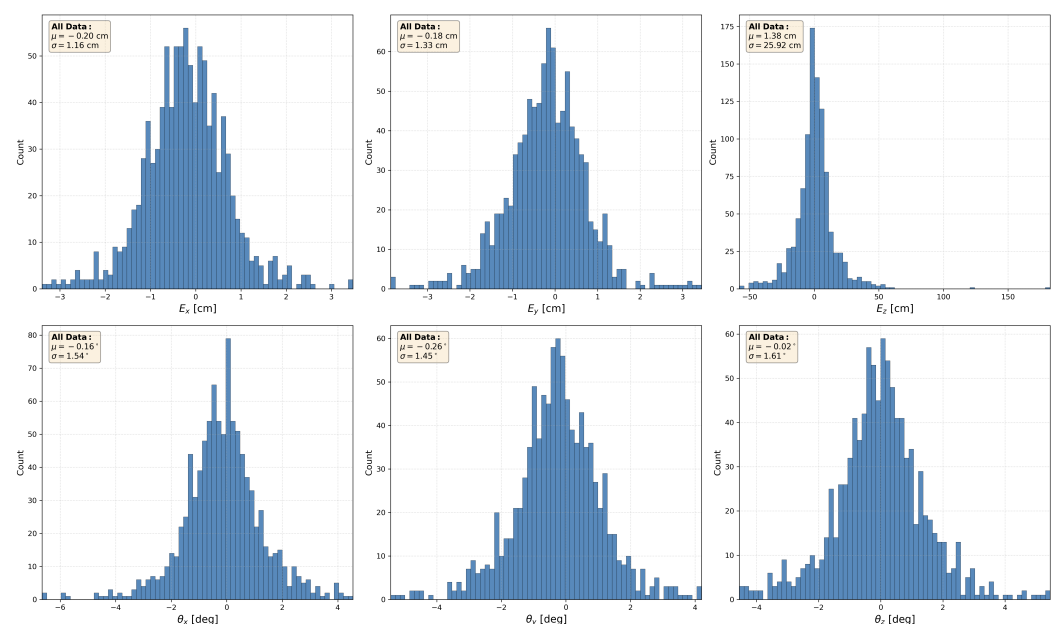


Figure 17. Component-wise SAPose error distributions on SPEED. The upper row shows translation errors along the camera-frame axes; the lower row shows Euler-angle attitude errors.

To characterize the high-error region more directly, we further compute the empirical complementary cumulative distribution function (CCDF) of the normalized translation error E_t and rotation error E_q . The CCDF reports the fraction of test samples whose error is greater than or equal to a given threshold and therefore complements the mean-error statistics by explicitly showing the tail behavior.

As shown in Figure 18, the survival probabilities of both E_t and E_q decrease rapidly with increasing error threshold. The top 5% of samples account for approximately 20.1% of the accumulated translation error and 14.7% of the accumulated rotation error. Thus, a relatively small subset of difficult samples contributes disproportionately to the remaining error, whereas most SPEED test samples remain in the low-error region.

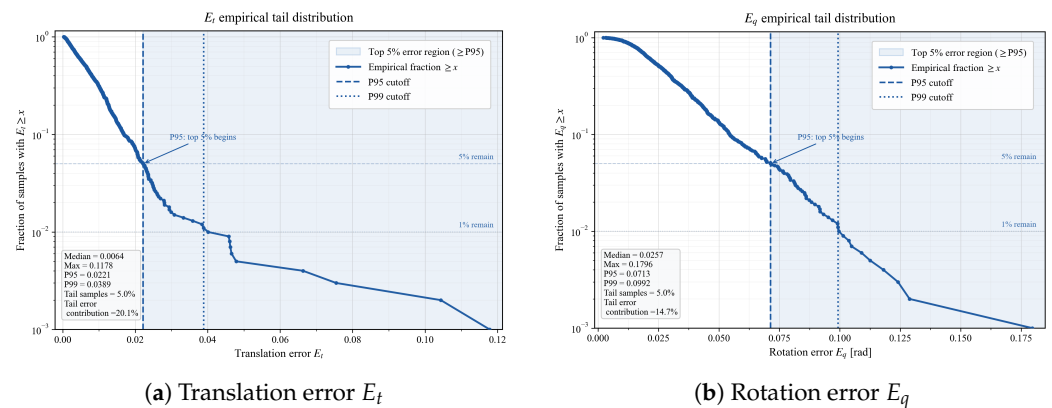


Figure 18. Empirical SApOse error-tail distributions on SPEED. The panels show the fraction of samples with E_t or E_q greater than or equal to threshold x ; shaded regions mark the top 5% beyond the corresponding P95 thresholds.

This distribution-level observation is consistent with the representative failure cases examined in Section 4.7. The selected high-error examples include strong illumination variation, severe occlusion, complex backgrounds, and very small target scales. These examples identify several recurring visual situations associated with the high-error tail, but the case-based analysis does not separate their individual effects. A plausible common pathway is degraded 2D correspondence quality under limited resolution, reduced landmark visibility, or locally ambiguous structure, which then propagates into geometric pose recovery.

The scale dependence is also consistent with the coarse-to-fine analysis in Section 4.6.3, where small targets exhibit substantially larger errors under single-pass inference and benefit most from secondary ROI refinement. We next examine whether background appearance shows a similar association with the remaining high-error samples.

5.2. Robustness Under Different Background Conditions

Background appearance represents another potential source of visual ambiguity for monocular spacecraft pose estimation. SPEED contains images in which the spacecraft appears against relatively simple dark backgrounds as well as images containing Earth-background structures. The latter may include cloud texture, Earth-limb boundaries, and stronger illumination variation around the target. To examine whether the residual pose error is associated with these different background conditions, we compare their empirical error distributions rather than relying only on overall mean values.

For each background category, the samples are sorted according to E_t and E_q , respectively, and the resulting distributions are shown in Figure 19. Over most of the error range, the black-background and Earth-background distributions remain close to one another.

This indicates that no large background-dependent difference is apparent for the majority of samples under the present evaluation conditions.

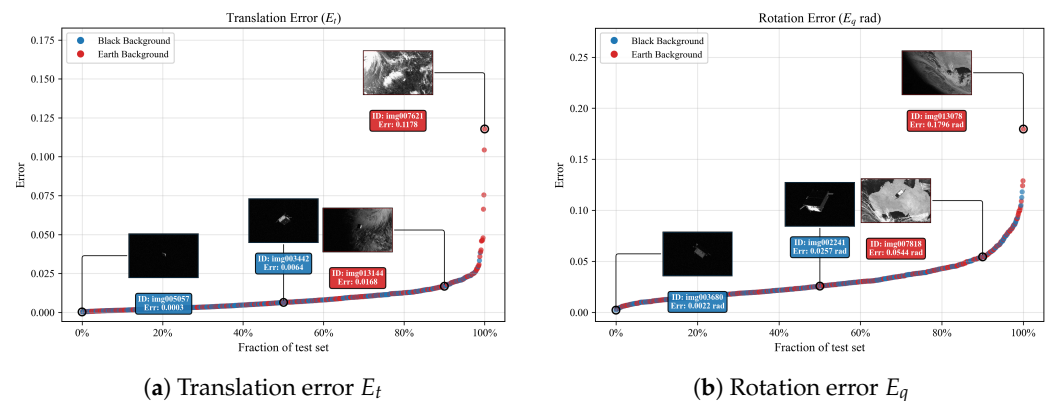


Figure 19. Sorted SAPI pose-error distributions under black- and Earth-background conditions on SPEED. The panels show normalized translation error E_t and rotation error E_q .

A difference becomes more visible in the upper tail, where the Earth-background distribution reaches higher error values for several tail samples, particularly for translation error. The same tendency is present but less pronounced for rotation error. The distributions show no large background-dependent difference for most SPEED samples, while several Earth-background samples occupy the higher-error tail. Because target scale, illumination, and viewing geometry can co-vary with background category, this analysis establishes an association rather than a causal background effect. Together with the representative cluttered-background cases in Section 4.7, the result motivates targeted robustness tests under controlled background variation.

5.3. Computational Efficiency and Conditional Refinement Overhead

For a consistent deployment-oriented comparison, the computational profile is reported using the same model, 512×512 input resolution, batch size 1, and RTX 4060 Laptop GPU protocol as the main SPEED efficiency experiment. This avoids mixing latency values obtained from different input resolutions or hardware platforms. SAPI contains 9.4 M parameters, corresponding to an FP32 static weight footprint of approximately 35.9 MiB.

The conditional second stage does not introduce a second network or additional trainable parameters; it reuses SAPI only for samples satisfying $\eta < \tau$. At the selected $\tau = 0.010$, the measured test-set trigger ratio is 33.4%. As summarized in Table 13, end-to-end latency increases from 9.78 ms for single inference to 12.55 ms for conditional refinement, i.e., an additional 2.77 ms or 28.3%. By comparison, always applying the second pass requires 18.31 ms. Thus, the conditional policy retains the accuracy benefit of ROI refinement for resolution-limited targets while avoiding most unnecessary second-pass computations, and it does so without increasing the static model-memory footprint.

5.4. Limitations

The current evaluation has three main limitations.

First, the large-scale quantitative evaluations in this work are primarily based on synthetic or simulation-derived imagery. The five real SPEED images provide a useful initial synthetic-to-real check, but they are insufficient to represent the diversity of real on-orbit sensors, illumination geometries, surface materials, backgrounds, optical effects, and image degradations. Accordingly, the real-image experiment provides initial cross-domain evidence, while comprehensive validation requires a broader range of real and on-orbit imagery.

Second, the failure analysis shows that the remaining high-error cases are associated with several difficult visual conditions, including strong illumination variation, severe occlusion, background clutter, and very small target scales. The case analysis does not decompose the relative contributions of keypoint localization, PnP initialization, and nonlinear refinement. Promising directions therefore include visibility-aware landmark reasoning for occlusion, geometric-consistency criteria alongside point-wise confidence, stronger photometric and background augmentation, and further refinement of the resolution-adaptive inference policy for extremely small targets. These directions are hypotheses for improvement and require controlled evaluation.

Third, the efficiency measurements characterize relative inference cost under the stated 512×512 RTX 4060 Laptop GPU protocol rather than real space-grade onboard execution. Power consumption, numerical precision, operator support, and embedded-hardware scheduling are platform dependent and remain to be evaluated on representative flight-oriented hardware.

Future work will consequently focus on broader real-image evaluation, sensor-aware domain adaptation, improved visibility and geometric reasoning for semantic landmarks, and validation on representative embedded or space-grade computing platforms under a unified input and runtime protocol.

6. Conclusions

In this paper, we propose SAPose, a lightweight multi-task framework for monocular pose estimation of space targets. The proposed network jointly performs spacecraft detection and keypoint localization, and it introduces a stage-aware feature extraction structure and a scale-adaptive feature fusion module to improve the representation of geometric details across different target scales. To improve accuracy for long-range small-target observations, an adaptive coarse-to-fine inference strategy selectively activates secondary ROI-based refinement for small-scale targets. In addition, a confidence-ranked keypoint selection strategy and object-space nonlinear refinement are adopted to improve the stability of the final geometric pose estimate.

Experiments on SPEED and SKD show that SAPose provides accurate keypoint localization and pose recovery with a compact model. Under the common SPEED protocol, the final configuration reaches 98.58% PCK@0.05 and an overall pose error of $E = 0.0352$. The conditional coarse-to-fine policy reduces the mean pose error relative to single inference while requiring substantially less second-pass computation than always-on refinement, with the selected threshold $\tau = 0.010$ triggering secondary inference on 33.4% of the SPEED test images. Experiments on the three SKD spacecraft models further show that the same framework can be retrained for targets with different structural geometries.

The robustness analyses show gradual degradation under landmark occlusion and perturbation of the reconstructed 3D prior. Evaluation on five real SPEED images without real-image fine-tuning provides initial evidence of direct synthetic-to-real transfer; broader real-image and on-orbit validation remains necessary.

Future work will focus on visibility-aware structural reasoning, sensor- and background-aware domain adaptation, temporal cues for ambiguous observations, and validation on representative embedded hardware and broader real-image datasets.

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

Funding: This work was supported by the National Natural Science Foundation of China (grant number 62503492).

Data Availability Statement: The SPEED dataset is publicly available at <https://kelvins.esa.int/satellite-pose-estimation-challenge/home/> (accessed on 15 September 2025). The SKD dataset is publicly available at <https://github.com/Dongzhou-1996/GKNet> (accessed on 10 April 2026).

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

References

1. Poozhivil, M.; Nair, M.H.; Rai, M.C.; Hall, A.; Meringolo, C.; Shilton, M.; Kay, S.; Forte, D.; Sweeting, M.; Antoniou, N.; et al. Active Debris Removal: A Review and Case Study on LEO-PARD Phase 0-A Mission. *Adv. Space Res.* **2023**, *72*, 3386–3413. [CrossRef]
2. Xiao, Y.; Jin, Y.; Yan, H.; Dou, Y.; Cui, G.; Li, F.; Meng, T. Review and Prospect of On-Orbit Servicing Technologies. *Adv. Astronaut.* **2025**, *8*, 107–127. [CrossRef]
3. Phisannupawong, T.; Kamsing, P.; Torteeka, P.; Channumsin, S.; Sawangwit, U.; Hematulin, W.; Jarawan, T.; Somjit, T.; Yooyen, S.; Delahaye, D.; et al. Vision-Based Spacecraft Pose Estimation via a Deep Convolutional Neural Network for Noncooperative Docking Operations. *Aerospace* **2020**, *7*, 126. [CrossRef]
4. Pauly, L.; Rharbaoui, W.; Shneider, C.; Rathinam, A.; Gaudillière, V.; Aouada, D. A Survey on Deep Learning-Based Monocular Spacecraft Pose Estimation: Current State, Limitations and Prospects. *Acta Astronaut.* **2023**, *212*, 339–360. [CrossRef]
5. Sharma, S.; D’Amico, S. Neural Network-Based Pose Estimation for Noncooperative Spacecraft Rendezvous. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 4638–4658. [CrossRef]
6. Fei, J.; Deng, Z. Rotation Invariance and Equivariance in 3D Deep Learning: A Survey. *Artif. Intell. Rev.* **2024**, *57*, 168. [CrossRef]
7. Chen, B.; Cao, J.; Parra, Á.; Chin, T.J. Satellite Pose Estimation with Deep Landmark Regression and Nonlinear Pose Refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*; IEEE: New York, NY, USA, 2019; pp. 2816–2824. [CrossRef]
8. Piazza, M.; Maestrini, M.; Di Lizia, P. Monocular Relative Pose Estimation Pipeline for Uncooperative Resident Space Objects. *J. Aerosp. Inf. Syst.* **2022**, *19*, 613–632. [CrossRef]
9. Zhong, L.; Chen, S.; Jin, Z.; Guo, P.; Yang, X. Keypoints Filtrating Nonlinear Refinement in Spatial Target Pose Estimation with Deep Learning. *IEEE Trans. Ind. Inform.* **2024**, *20*, 12523–12532. [CrossRef]
10. Chen, S.; Yang, W.; Wang, W.; Mai, J.; Liang, J.; Zhang, X. Spacecraft Homography Pose Estimation with Single-Stage Deep Convolutional Neural Network. *Sensors* **2024**, *24*, 1828. [CrossRef] [PubMed]
11. Liu, F.; Zhang, Z.; Li, S. DTSE-SpaceNet: Deformable-Transformer-Based Single-Stage End-to-End Network for 6-D Pose Estimation in Space. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 2555–2571. [CrossRef]
12. Hu, Y.; Fua, P.; Wang, W.; Salzmann, M. Single-Stage 6D Object Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2020; pp. 2927–2936. [CrossRef]
13. He, Z.; Zhan, F.; Chen, J.; Xi, R. A Monocular Vision-Based Algorithm for Spacecraft Pose Estimation. *IFAC-PapersOnLine* **2025**, *59*, 250–255. [CrossRef]
14. Park, T.H.; D’Amico, S. Adaptive Neural-Network-Based Unscented Kalman Filter for Robust Pose Tracking of Noncooperative Spacecraft. *J. Guid. Control Dyn.* **2023**, *46*, 1671–1687. [CrossRef]
15. Gui, Y.; Qi, Y.; Xiao, X.; Lin, B.; Cui, H.; Huang, X. SPTN: Transformer-based spacecraft pose estimation network for space objects tracking. *Astrodynamics* **2025**, *9*, 713–725. [CrossRef]
16. Bi, Z.; Chen, M.; Ding, G.; Yan, H.; Han, S.; Li, Z.; Ma, R. Balance-URSONet: A Real-Time Efficient Pose Spacecraft Estimation Network. *Aerospace* **2025**, *12*, 840. [CrossRef]
17. Bechini, M.; Gu, G.; Lunghi, P.; Lavagna, M. Robust spacecraft relative pose estimation via CNN-aided line segments detection in monocular images. *Acta Astronaut.* **2024**, *215*, 20–43. [CrossRef]
18. Bechini, M.; Lavagna, M. Robust and efficient single-CNN-based spacecraft relative pose estimation from monocular images. *Acta Astronaut.* **2025**, *233*, 198–217. [CrossRef]
19. Park, T.H.; D’Amico, S. Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap. *Adv. Space Res.* **2024**, *73*, 5726–5740. [CrossRef]
20. Gu, X.; Yang, X.; Liu, H.; Yang, D. Adaptive Granularity-Fused Keypoint Detection for 6D Pose Estimation of Space Targets. *Remote Sens.* **2024**, *16*, 4138. [CrossRef]
21. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12993–13000. [CrossRef]
22. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.

23. Lin, T.Y.; Maire, M.; Belongie, S.J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Proceedings of the Computer Vision—ECCV 2014*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755. [\[CrossRef\]](#)
24. Lu, C.P.; Hager, G.D.; Mjolsness, E. Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 610–622. [\[CrossRef\]](#)
25. Kisantal, M.; Sharma, S.; Park, T.H.; Izzo, D.; Mörtens, M.; D’Amico, S. Satellite Pose Estimation Challenge: Dataset, Competition Design, and Results. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 4083–4098. [\[CrossRef\]](#)
26. Maji, D.; Nagori, S.; Mathew, M.; Poddar, D. YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*; IEEE: New York, NY, USA, 2022; pp. 2637–2646.
27. McNally, W.; Vats, K.; Wong, A.; McPhee, J. Rethinking Keypoint Representations: Modeling Keypoints and Poses as Objects for Multi-Person Human Pose Estimation. In *Computer Vision—ECCV 2022, Proceedings, Part VI*; Springer: Cham, Switzerland, 2022; pp. 37–54. [\[CrossRef\]](#)
28. Xu, Y.; Zhang, J.; ZHANG, Q.; Tao, D. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. In *Proceedings of the Advances in Neural Information Processing Systems*; Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 38571–38584.
29. Lu, P.; Jiang, T.; Li, Y.; Li, X.; Chen, K.; Yang, W. RTMO: Towards High-Performance One-Stage Real-Time Multi-Person Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 1491–1500.
30. Purkrabek, M.; Matas, J. ProbPose: A Probabilistic Approach to 2D Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 27124–27133.
31. Song, Q.; Liu, K.; Li, S.; Wang, M.; Wang, J. A DSP-FPGA Heterogeneous Accelerator for On-Board Pose Estimation of Non-Cooperative Targets. *Aerospace* **2025**, *12*, 641. [\[CrossRef\]](#)
32. Ye, R.; Ren, Y.; Zhu, X.; Wang, Y.; Liu, M.; Wang, L. An Efficient Pose Estimation Algorithm for Non-Cooperative Space Objects Based on Dual-Channel Transformer. *Remote Sens.* **2023**, *15*, 5278. [\[CrossRef\]](#)
33. Mu, Y.; Zhi, X.; Gong, J.; Jiang, S.; Hu, J.; Wang, X.; Bao, G. StructGCN: A Structure-Aware Graph Convolutional Network for Spacecraft Pose Estimation. *Chin. J. Aeronaut.* **2026**, 104204. [\[CrossRef\]](#)
34. Hu, Y.; Hugonot, J.; Fua, P.; Salzmann, M. Segmentation-Driven 6D Object Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2019; pp. 3385–3394. [\[CrossRef\]](#)
35. Proença, P.F.; Gao, Y. Deep Learning for Spacecraft Pose Estimation from Photorealistic Rendering. In *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2020; pp. 6007–6013. [\[CrossRef\]](#)
36. Ma, W.; Zhou, D.; Hu, Y.; He, Z. GKNet: Graph-based Keypoints Network for Monocular Pose Estimation of Non-cooperative Spacecraft. *IFAC-PapersOnLine* **2025**, *59*, 1297–1302. [\[CrossRef\]](#)

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.