Pose Estimation and Neural Implicit Reconstruction Towards Non-cooperative Spacecraft Without Offline Prior Information

BING HAN
CHENXI WANG
XINYU ZHANG
ZHIBIN ZHAO, Member, IEEE
ZHI ZHAI
JINXIN LIU
Xi'an Jiaotong University, Xi'an 710049, PR China

NAIJIN LIU
China Academy of Space Technology, 100094 Beijing, China

XUEFENG CHEN, Senior Member, IEEE
Xi'an Jiaotong University, Xi'an 710049, PR China

***Abstract*— Non-cooperative spacecraft pose estimation plays a crucial role in on-orbit servicing. However, existing pose estimation methods often assume CAD models of target objects as prior information, used for offline training or online template matching. This limits the generalization of pose estimation methods.**

To explore a generic solution, this work proposes a pose estimation method for unknown spacecraft. Our method is not only independent of prior models or image priors of the target but also synchronously outputs pose parameters and aligned target texture models. Specifically, we employ three modules in parallel: pose tracking, neural object reconstruction, and target reference frame (TRF) estimation. Firstly, leveraging the knowledge of temporal data, we optimize the pose graph to provide stable tracking performance. Then, we use neural implicit representation to reconstruct the target texture model, with pose parameters jointly optimized during the reconstruction process. Finally, we propose TRFE-Net for online estimation of the TRF. The obtained TRF is used to correct the sensor reference frame (SRF), transforming the pose tracking and reconstruction problem from scene-centric to Spacecraft-centric.

Additionally, the PEU dataset was constructed specifically for pose estimation of unknown spacecraft. Comprehensive experiments show that although the proposed method reduces the need for prior information, it still achieves good performance across multiple objects and effectively handles large-scale motions, specular highlights, thin structures, and symmetric structures. Project page: *PEU*.

***Index Terms*— Deep learning, satellite pose estimation, neural implicit reconstruction, model alignment, without prior information**

I. INTRODUCTION

With the rapid advancement of aerospace technology, numerous on-orbit spacecraft missions have emerged, encompassing tasks such as rendezvous and docking, formation flying [1], active debris removal (ADR) [2, 3], and on-orbit servicing (OOS) [4, 5]. Within these missions, the precise determination of the spatial orientation information of space targets is instrumental for deploying automated vision-based systems in orbit. This provision of supplementary information is crucial for close-range relative navigation, playing a pivotal role in ensuring the successful execution of the missions.

Due to the lack of cooperative markers on non-cooperative spacecraft, existing works typically obtain the target's texture model or simplified wireframe model offline and define a reference frame as a prior (Fig. 1). Traditional methods align this reference model or image with the collected data frames during testing to solve for the pose [6–9]. Deep learning-based approaches first generate images or point clouds using this reference model for training and then use the trained neural network to estimate the pose[10–21].

The described approach is suboptimal in several ways. First, the approach cannot be generalized to targets without prior knowledge due to its reliance on available prior information. Second, spaceborne data are scarce, precious, and pose significant privacy concerns. The types and numbers of non-cooperative space objects are extensive, making it impractical and cumbersome to obtain a priori models for each non-cooperative object. Finally, even with prior knowledge of the target's model, due to prolonged exposure to the space environment, certain space objects may undergo physical collisions and surface optical degradation, leading to appearances that differ from expectations[22].

To tackle the above problems, we propose a hybrid pipeline for pose estimation and neural implicit reconstruction targeting non-cooperative Spacecraft without prior information. This pipeline redefines the technical pathway compared to previous methods, as illustrated in Fig. 1.

The framework we proposed consists of three parallel modules: pose tracking, neural object reconstruction, and target reference frame estimation (TRFE). First, we track

Manuscript received XXXXX 00, 0000; revised XXXXX 00, 0000; accepted XXXXX 00, 0000.
Corresponding author: Chenxi Wang (Email:wangchenxi@xjtu.edu.cn)

Bing Han and Xinyu Zhang are with the School of Future Technology, Xi'an Jiaotong University, Xi'an 710049, P.R. China, and National Key Lab of Aerospace Power System and Plasma Technology, Xi'an Jiaotong University, Xi'an 710049, P.R. China (Email:{hbb16,zhangxinyu1026}@stu.xjtu.edu.cn.) Chenxi Wang, Zhibin Zhao, Zhi Zhai, Jinxin Liu and Xuefeng Chenare with the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, P.R.china, and National Key Lab of Aerospace Power System and Plasma Technology, Xi'an Jiaotong University, Xi'an 710049, P.R. China (Email:{wangchenxi,zhaozhibin, zhaizhi, jinxin.liu, chenxf}@xjtu.edu.cn.) Naijin Liu is with the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, P.R.china, and China Academy of Space Technology, Beijing 100094, China (Email:liunaijin@xjtu.edu.cn.)

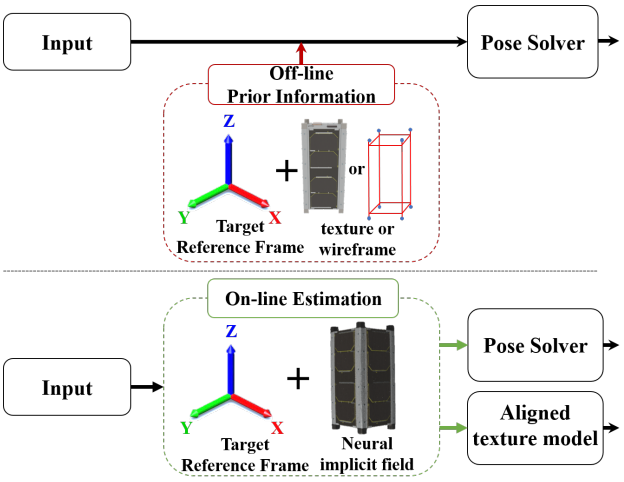


Fig. 1: **Top**: Existing methods require offline acquisition of prior knowledge. **Bottom**: The proposed method does not require prior knowledge of the target and simultaneously outputs the aligned texture model.

the spacecraft's pose in the SRF. Specifically, we optimize the pose graph using temporal context information, significantly reducing tracking drift. Second, we employ neural implicit representation to reconstruct the target texture model. Particularly, this neural reconstruction operates online, with pose parameters being jointly optimized during the reconstruction process. Finally, we propose TRFE-Net for online estimation of the TRF. The obtained TRF is used to correct the SRF, transforming the pose tracking and reconstruction problem from scene-centric to Spacecraft-centric.

The key to our method lies in online prediction of the target reference frame and neural implicit representation, which compensates for the lack of prior information. During execution, it only requires capturing a small number of data frames online for initialization, enabling efficient output of pose parameters and aligned texture 3D models of objects. This approach also effectively avoids mismatches between offline reference models and real-space objects. Experimental results show that our method exhibits robustness in handling symmetrical structures, large-scale displacements, and specular highlights caused by sunlight reflection on spacecraft surfaces. In summary, our contributions could be summarized as follows:

1) We revisit spacecraft pose estimation from the perspective of reference coordinate systems. By leveraging online neural reconstruction and reference frame estimation, we remove the reliance on offline prior information and further explore a general solution for non-cooperative spacecraft pose estimation.
2) We leverage the knowledge of temporal data to optimize pose estimation, resulting in low-drift tracking performance and robust handling of spacecraft symmetrical structures, specular reflections, and large-scale displacements.
3) A two-stage reference frame estimation method is proposed. By decoupling the regression heads and expanding the solution space, the TRFE-Net could effectively predict TRF within the same category.
4) A spacecraft pose tracking dataset named PEU is proposed for evaluating our method. Comprehensive experiments show that although the proposed method reduces the need for prior information, it still achieves good performance across multiple objects. To the best of our knowledge, PEU is the first multimodal dataset that includes masks, depth, point clouds, and multi-object pose trajectories.

II. RELATED WORK

A. Non-cooperative Target Pose Estimation

The estimation of spatial non-cooperative target poses represents a specialized scenario within the broader context of six degrees of freedom (6-DoF) target pose estimation in spatial environments. The primary objective is to infer the three-dimensional translation and rotation of the target relative to its coordinate system from the data frames collected by sensors. Previous research [23] has proposed a pose graph optimization-based SLAM framework for the model reconstruction and pose estimation of spatially unknown non-cooperative rotating targets. This approach utilizes the sensor coordinate system as the reference frame, resulting in sensor-centric pose tracking.

State-of-the-art methods typically necessitate offline training using object CAD models or online template matching [12–16, 19, 24]. Although recent advancements have enhanced the generalization performance of models in new environments through domain adaptation techniques [25–29], these methods remain inadequate for application to new, unknown non-cooperative objects. Some studies [26] leverage temporal information to track the pose of non-cooperative objects. However, these approaches make various assumptions, such as training and testing on the same objects.

The fundamental challenge in estimating the pose of an unknown target lies in the indeterminacy of the reference frame. Our work advances this field by addressing this specific challenge. In our approach, we perform online neural implicit reconstruction of the target and predict its reference frame, thereby eliminating the reliance on model priors. By jointly optimizing pose tracking and reconstruction through neural representation, we achieve not only more robust pose estimation but also more refined 3D shape outputs.

B. Object Reconstruction

The reconstruction of the pose and shape of orbiting objects based on a sequence of images is a crucial component of several conceptual tasks that have emerged in the past decade. It serves as prior knowledge for tasks such as active debris removal, rendezvous, capture (or attach-

ment), and off-nominal trajectory missions. Traditional reconstruction methods employ discrete elements, such as point clouds, meshes, or voxels, to explicitly represent the scene[30–32]. These methods typically generate sparse point cloud structures first, followed by densification of the point cloud, and finally, surface mesh and texture reconstruction. While these methods yield realistic results, they come with high computational costs and often exhibit limited capability in capturing fine details.

Since the introduction of Neural Radiance Fields (NeRF)[33], implicit scene representations have gained attention in spatial object perception tasks[34–36]. Preliminary studies by the European Space Agency (ESA) compared the quality of models reconstructed using NeRF and Generative Radiance Fields(GRAF) [37] on spacecraft. Mahendrakar *et al.* [36] explored the quality of spacecraft model reconstruction accelerated by Instant Neural Graphics Primitives in NeRF. However, they assumed known camera poses or relied on precomputed camera poses obtained through Structure from Motion (SfM)[38] offline. Contrarily, we do not assume known camera poses. Instead, we provide initial pose values through online pose graph optimization, obtaining an accurate 3D model representation.

C. Target Reference Frame Determined

In the task of pose estimation, knowledge of the target's reference frame is essential for initializing the estimator, akin to the initialization of the initial pose in tracking tasks. The estimation of the target's reference coordinate system undoubtedly stands out as the most challenging aspect throughout the entire process. Opromolla *et al.* [22] employed an online template matching algorithm based on three-dimensional Principal Component Analysis (PCA) to estimate the reference frame of the target. Initially, it estimates the relative position vectorusing the centroid method. Subsequently, it confirms the direction of the main axis by identifying the eigenvector corresponding to the maximum eigenvalue of the covariance matrix of the measured point cloud. However, these methods necessitate the offline storage of a template database containing geometric information about the target, which is not assumed in oursetting.

While recent research has explored using neural networks to estimate object upright orientation[39, 40], our approach tackles a more challenging scenario involving space objects. Unlike terrestrial objects with supporting bases influenced by gravity, space objects lack such support, posing greater difficulty. Additionally, existing methods typically focus on estimating vertical orientation only, whereas our method addresses the complete spatial orientation in the $\mathbb{SO}(3)$ space.

III. METHOD

In this section, we first introduce the memory pool design in the pose tracking pipeline and the online pose graph optimization framework (Section 3.A), focusing on how to leverage temporal data knowledge and strategies for handling symmetrical targets. Secondly, we describe the target neural representation method and training details (Section 3.B). Finally, we provide details of the TRFE pipeline (Section 3.C). The overview of our method is shown in Fig. 2.

A. Pose Tracking

1. *Memory Pool*

Considering the continuous and incremental motion of spacecraft in space, we leverage temporal contextual information to perform online pose graph optimization by exploiting the relationships between multiple frames. Inspired by [41, 42], we introduce a memory pool $\mathcal{M}$ that stores historical observation frames with the maximum variance across multiple views. To build the memory pool, the first frame $\mathcal{F}_0$ is automatically added, thus setting the canonical coordinate system for the novel unknown object. For each new frame, its coarse pose ξ_t' is updated to a refined pose ξ_t by aligning it with existing frames in the memory pool. When the viewpoint of the new frame is deemed sufficient to enrich the multi-view diversity in the pool while keeping the pool compact,

To be more precise, we compare the pose ξ_t of the current frame with that of the last keyframe. If the rotational geodesic distance or the relative displacement of the sensor with respect to the previous keyframe exceeds the predefined thresholds, the current frame is selected as a new keyframe. Given the dynamic nature of spatial movement, the relative motion of spatial objects can change rapidly (e.g., during the approach phase) or slowly (e.g., during formation flying and docking phases). Therefore, relative metrics provide higher robustness compared to absolute metrics. Additionally, unlike previous work [41], we do not employ a time-interval-based keyframe insertion method. This is because, in dynamic scenarios, relying on time intervals to insert keyframes is inefficient, often failing to capture enough keyframes while generating redundant ones.

2. *Online Pose Graph Optimization*

We use the first frame $\mathcal{F}_0$ added to the memory pool as the initial frame for pose tracking, with its pose parameters ξ_0 serving as the initial pose. The initial pose's rotation and translation parameters are set to the identity rotation and the distance from the sensor to the centroid of the acquired point cloud, respectively. In our framework, the coordinate system of this initial pose is defined as the SRF. Subsequently, by executing the TRFE thread, the pose tracking coordinate system is corrected to the TRF. For detailed annotations on SRF and TRF, refer to Fig. 2 and Fig. 5.

The Iterative Closest Point (ICP) algorithm is commonly employed for point cloud registration and pose estimation. In typical scenarios, it iteratively finds an optimal rotation and translation transformation using nonlinear

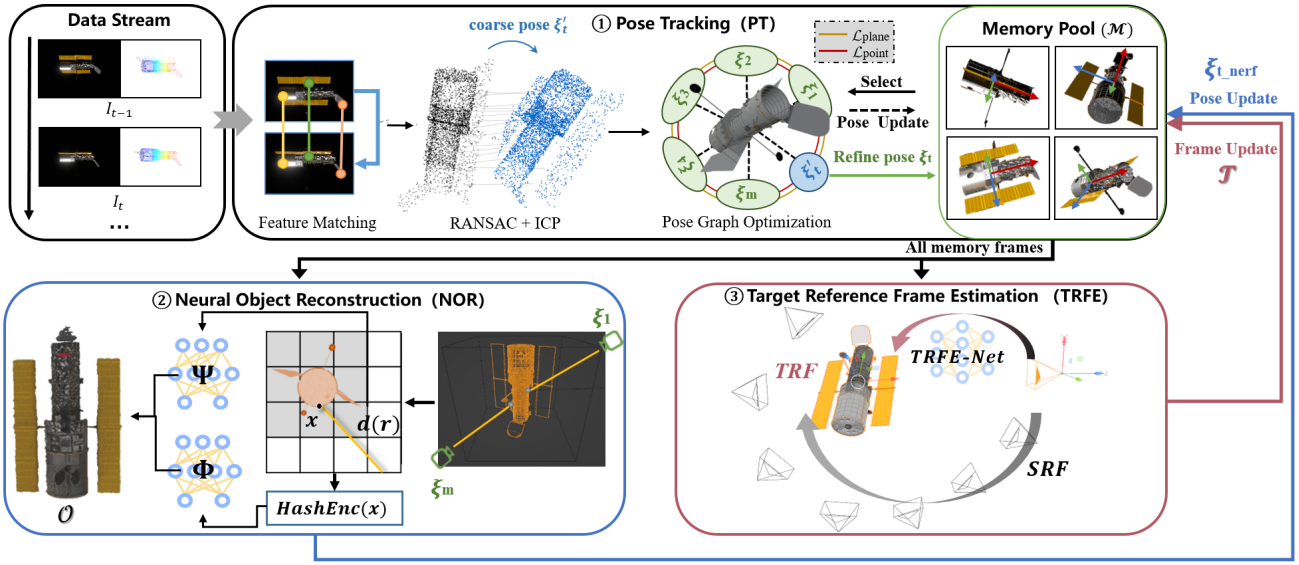


Fig. 2: The pipline of our approach. ① PT serves as the main pipeline and communicates with NOR and TRFE concurrently through the memory pool $\mathcal{M}$. ② NOR utilizes memory frame neural reconstruction and optimizes poses ξ_t. It then updates the optimized poses ξ_{tnerf} to the memory pool and outputs the texture model $\mathcal{O}$. ③ TRFE uses the merged memory frame point cloud to predict the TRF, which is used to correct the reference frame of the PT pipeline.

optimization, aiming to maximize the alignment between the source point cloud $\{p_s\}$ and the target point cloud $\{p_t\}$. The general expression of ICP is represented as:

$$R^*, t^* = \arg\min_{R,t} \frac{1}{I} \sum_{i=1}^{I} \left\| p_t^i - \left(R \cdot p_s^i + t\right) \right\|_2^2 \quad (1)$$

where p_s^i and p_t^i denote the i-th matched points from the source point cloud and the target point cloud. R and t represent the rotation and translation parts of the pose to be optimized.

The correspondence between RGB-D data is established using a Transformer-based feature matching network [43], which has been pre-trained on large-scale internet photo collections and is well-suited for scenarios with weak textures and suboptimal lighting conditions. These identified correspondences undergo filtration using a RANSAC-based pose estimator coupled with least squares. Subsequently, the pose hypothesis maximizing the number of inliers is selected as the coarse pose estimation ξ_t' for the current frame. It is worth noting that in the case of known correspondence, the least squares problem actually admits an analytical solution, thereby obviating the need for iteration.

We limit the number of memory frames involved in pose graph optimization to no more than K for efficiency. In the early stages of tracking, when the size of the memory pool $\leq K$, all frames in the memory pool are utilized. When the memory pool size exceeds K, a selection process is implemented to maximize the multi-view consistency information. Specifically, for each frame $\mathcal{F}_k$ in the memory pool, we compute the dot product between its point normal map and the ray directions in the camera view of the new frame $\mathcal{F}_t$ to test the visibility of the memory frame from the current frame's camera viewpoint. If the visibility of the tracked object in the new frame exceeds a threshold, we compute the rotation geodesic distance between ξ_k and ξ_t to further measure the viewpoint overlap with $\mathcal{F}_t$. Finally, we select the K memory frames with the maximum viewing overlap to participate in the pose graph optimization with $\mathcal{F}_t$. It is worth noting that, in order to avoid tracking failures caused by the inherent symmetry of spacecraft, we use 3D coordinate and normal distance to filter the matching relationships obtained between frames. The experiments in Section IV-D demonstrate that this approach effectively handles mismatches caused by symmetry.

In the pose graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$, the vertices of the graph consist of the poses of the selected K memory frames and the current frame: $\mathcal{V} = \xi_t' \cup \{\xi_{pool}\}$, where $pool \in [1, K]$. The edges of the graph are formed by pairwise matching point pairs between vertices: $\mathcal{E} = \{\mathcal{P}_n^{(ij)}\}$, where $\mathcal{P}$ denotes the matching relationships between keypoints, ij denotes the index of vertex pairs with matching relationships, and n denotes the index of the matching relationship between vertex pairs ij. The objective is to find the optimal pose that minimizes the total loss of the pose graph.

Active tracking of spacecraft is a progressive process from far to near, with a wide range of depth variations[14]. The optimization approach of minimizing reprojection error is suboptimal in handling such cases. Therefore, we employ relative losses between 3D point-to-point and plane-to-plane, rendering the method robust to variations in target distance, As shown in Fig 3. We

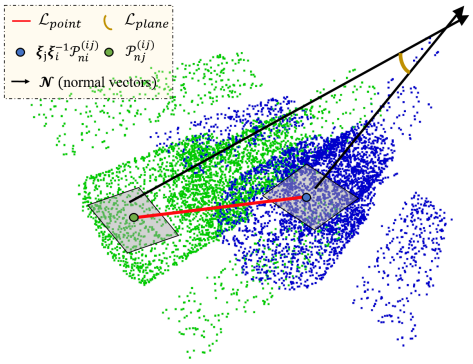


Fig. 3: The objective function for pose graph optimization consists of two parts: distance error $\mathcal{L}_{\text{point}}$ between 3D points and angle error $\mathcal{L}_{\text{plane}}$ between normal vectors.

take our point-to-point loss as:

$$\mathcal{L}_{\text{point}}(i,j) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{H}(\|\xi_j\xi_i^{-1}\mathcal{P}_{ni}^{(ij)} - \mathcal{P}_{nj}^{(ij)}\|_2) \quad (2)$$

where i, j denotes the $i\text{-}th$ and $j\text{-}th$ frames in the vertex set $\mathcal{V}$, and $\mathcal{H}$ represents the Huble robust kernel function. The loss measures the distance of the 3D feature correspondences $\mathcal{P}_{ni}^{(ij)}, \mathcal{P}_{nj}^{(ij)} \in \mathbb{R}^3$. We take our 3D plane-to-plane loss as:

$$\mathcal{L}_{\text{plane}}(i,j) = \frac{1}{N}\sum_{n=1}^{N}\lambda_n\frac{<\mathcal{N}(\mathcal{P}_{ni}^{(ij)}), \mathcal{N}(\mathcal{P}_{nj}^{(ij)})>}{\pi} \quad (3)$$

where $\mathcal{N}$ represents the unit normal vector corresponding to the point, with $<\cdot,\cdot>=<\mathbb{R}^3,\mathbb{R}^3>\in[0,\pi]$ computes the calculation of the angle between two vectors, and coefficient λ_n denotes the scalar factor, which was set to $\|\frac{\mathcal{P}_{ni}^{(ij)}+\mathcal{P}_{nj}^{(ij)}}{2}\|_2$ in the experiments to maintain the same order of magnitude as $\mathcal{L}_{\text{point}}$. The loss measures the angle between the normal vectors of the corresponding points' planes.

From an optimization perspective, the point-to-point loss is already a necessary and sufficient condition for this problem. However, we find that incorporating the plane-to-plane loss always leads to faster convergence of the rotation component $\mathcal{R} \in \mathbb{SO}(3)$ in the optimized pose, and even achieves better results. The reason lies in the fact that after adopting the plane-to-plane loss as an additional term, the gradient of the rotation parameters towards these normal vectors $\mathcal{N}$ becomes larger. Since the sensor data already includes depth information, it can provide a good initial displacement value. In the spatial non-cooperative target pose estimation, the rotation component is more important for the optimization process. Based on this insight, we adopt the plane-to-plane loss as an additional term, ultimately defining the overall graph loss as:

$$\mathcal{L} = \sum_{i\in\mathcal{V}, j\in\mathcal{V}, i\neq j}[\mathcal{L}_{\text{point}}(i,j) + \mathcal{L}_{\text{plane}}(i,j)] \quad (4)$$

where i, j denotes the poses of the ith and jth frames in the vertex set $\mathcal{V}$.

B. Neural Object Reconstruction

1. *Radiance Field Representation*

A key to our approach is the concurrent operation of neural radiance fields, which optimize the pose parameters of the reference frame while learning the shape and appearance of the object, thus mitigating drift in subsequent pose tracking.

Inspired by [42], we adopt the Signed Distance Function (SDF) as the final 3D representation due to its rapid convergence. We adopt the same configuration as [42], using the geometric function Φ and appearance function Ψ to represent the object. The geometric function $\Phi : x \to \sigma$ takes the 3D point coordinates $x \in \mathbb{R}^3$ as input and outputs the signed distance values $\sigma \in \mathbb{R}$. The appearance function $\Psi : (\Phi(x), n, d) \to c$ takes the intermediate feature vector $\Phi(x) \in \mathbb{R}^3$ from the geometry network, a point normal $n \in \mathbb{R}^3$ and a ray direction $d \in \mathbb{R}^3$ as input and outputs the color $c \in \mathbb{R}^3$.

Before forward propagation, we perform multi-resolution hashing encoding [44] on x to improve training speed. Additionally, we embed n and d with a fixed set of low-order spherical harmonic coefficients to prevent potential overfitting that may hinder object pose updates, especially rotations.We employ the hierarchical ray sampling strategy proposed in the classic NeRF and utilize the principles of classic volume rendering to render the color of any ray traversing the scene. We recommend readers refer to [33] and [44] for a more comprehensive survey of neural object reconstruction.

2. *Training*

In addition to the geometry and appearance networks, the multi-resolution hash encoder and the pose refinement of memory frames are also updated during training. At the beginning of each training phase, the Neural Object Field consumes the nearest portion of the memory pool and begins learning. During the learning process, poses are parameterized using Lie Algebra, with initial values set to ξ_t. Upon convergence of the training, the optimized poses are updated to ξ_{t_nerf} to assist subsequent online pose graph optimization. However, pose parameters do not need to be updated in every training phase. For frames that have already been updated to ξ_{t_nerf}, their pose parameters remain fixed and are used as such in subsequent training processes.

Our final training loss is:

$$\mathcal{L} = \mathcal{L}_{\text{s}} + \mathcal{L}_{\text{l}} + \mu\mathcal{L}_{\text{c}} \quad (5)$$

$\mathcal{L}_s$ is defined as a unary loss function used to measure the point-wise distance between the current frame and the neural implicit shape:

$$\mathcal{L}_{\text{s}} = \frac{1}{|\mathcal{F}|}\sum_{\mathcal{P}\in\mathcal{F}}\mathcal{H}(\|\Phi(\xi_t^{-1}(\mathcal{P}))\|_1) \quad (6)$$

where $\mathcal{H}$ represents the Huble robust kernel function and $\mathcal{P}$ denotes the the point cloud data observed in the current frame.

The $\mathcal{L}_{\mathrm{l}}$ represents the L1 error between the predicted Euclidean distance and the ground-truth distance to the object surface at each query point:

$$\mathcal{L}_{\mathrm{l}} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} (\|\Phi(x) - \tilde{l}\|_1) \tag{7}$$

where $\tilde{l}$ denotes the ground-truth distance labels. To obtain the distance labels, we first merge the point clouds extracted from the *K* memory frames using multi-view stereo reconstruction. We then voxelize the merged point cloud at a resolution of 5mm to discretize the implicit surface representation. The distance value stored at each voxel location corresponds to the actual signed distance to the surface.

The $\mathcal{L}_{\mathrm{c}}$ represents the L2 error between the predicted RGB color and the labeled color at each query point:

$$\mathcal{L}_{\mathrm{c}} = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} (\|\Psi(x, n(x), d(r)) - \tilde{c}\|_2) \tag{8}$$

where $\tilde{c}$ denotes the ground-truth color labels. The color labels are directly obtained from the corresponding image coordinates of the query points projected onto the *K* memory frames.

C. Target Reference Frame Estimation

From a high-level perspective, the main differences between object pose estimation and pose tracking tasks are as follows: the former is object-centric and involves estimating pose parameters in a specific reference frame (typically a predefined coordinate system in prior models). In contrast, the latter is scene-centric and aims to track the pose of the camera or sensor itself (e.g., in applications such as autonomous driving, simultaneous localization and mapping), often using the pose of the first camera frame as the reference pose in the world coordinate system.

Based on the insights outlined above, we propose the Target Reference Frame Estimation Network (TRFE-Net), which employs a data-driven approach to automatically encode the Euclidean transformation $T_{ts} \in \mathbb{R}^{3\times4}$ from the SRF to the TRF.

In this section, the network architecture is introduced in Section III-C.2, the clustering strategy for spatial object shape categories is elaborated in Section III-C.3, and the loss functions are described in Section III-C.4.

1. system overview

The key challenge in determining the body coordinate system of space objects is that each different type of space target possesses unique geometric characteristics. This diversity can introduce ambiguity to the model learning process, leading to poor generalization ability. Therefore, relying solely on a single network encoding for diverse types of spacecraft is fragile.

Based on our observations, the geometric characteristics of spacecraft result from a combination of physical laws, functionality, and design knowledge. This leads to spacecraft of the same category typically sharing common semantic preferences and design structures, such as the number and distribution of solar panels, as well as the orientation of instruments like telescopes, radars, and docking rings, among others.

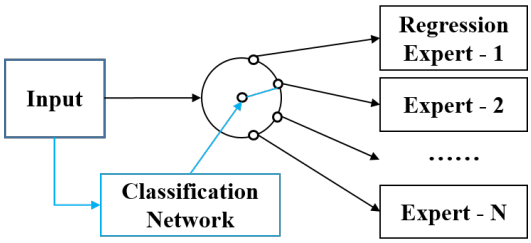


Fig. 4: Divide-and-conquer strategy. The merged point cloud of spacecraft is first classified and then regressed to the corresponding expert model for estimating the TRF.

Following the above assumptions, our system employs a divide-and-conquer approach where the data from each non-cooperative spacecraft is first classified by the network. The data is then input into the corresponding expert direction regression network for the predicted category, as shown in Fig 4. This two-stage pipeline leverages category-specific regression experts for improved pose estimation accuracy.

2. Network Architecture

Experimental findings in [45] demonstrate that both point cloud classification and pose regression tasks benefit from the model's understanding and learning of global features. Therefore, we combine both local and global features for feature encoding of point clouds. To meet these requirements, we utilize the strategy of farthest point sampling and grouping proposed by PointNet++ [46], enhancing the capability of local feature extraction. Then, global features are encoded using stacked attention layers. Different decoder architectures are employed for classification and orientation regression tasks. The architecture of TRFE-Net is depicted in Fig 5.

The TRFE pipeline is driven by a memory pool. When the perspective of the memory pool is deemed sufficient to cover the target object, the input point cloud $\mathcal{C} \in \mathbb{R}^{N\times d}$ is obtained by merging point clouds from all memory frames and uniformly downsampling. It is worth noting that at this stage, the reference frame of $\mathcal{C}$ remains uncorrected sensor reference frame. Specifically, the point cloud $\mathcal{C}$ consists of 2048 points, each point's 6D representation includes its 3D coordinates and 3D normal vector. A MLP is applied to learn the d_e-dimensional embedding features $\mathrm{F_e} \in \mathbb{R}^{\mathrm{N}\times\mathrm{d_e}}$ from the input point features. Specifically, the MLP performs a 1×1 convolution on each point's feature vector, followed by batch normalization [47] and a *ReLU* activation.

$$\mathrm{MLP(x)} = \mathrm{ReLU(BN(conv_{1\times1}(x)))} \tag{9}$$

In the local feature encoding module, we employ the farthest point sampling algorithm to downsample N to

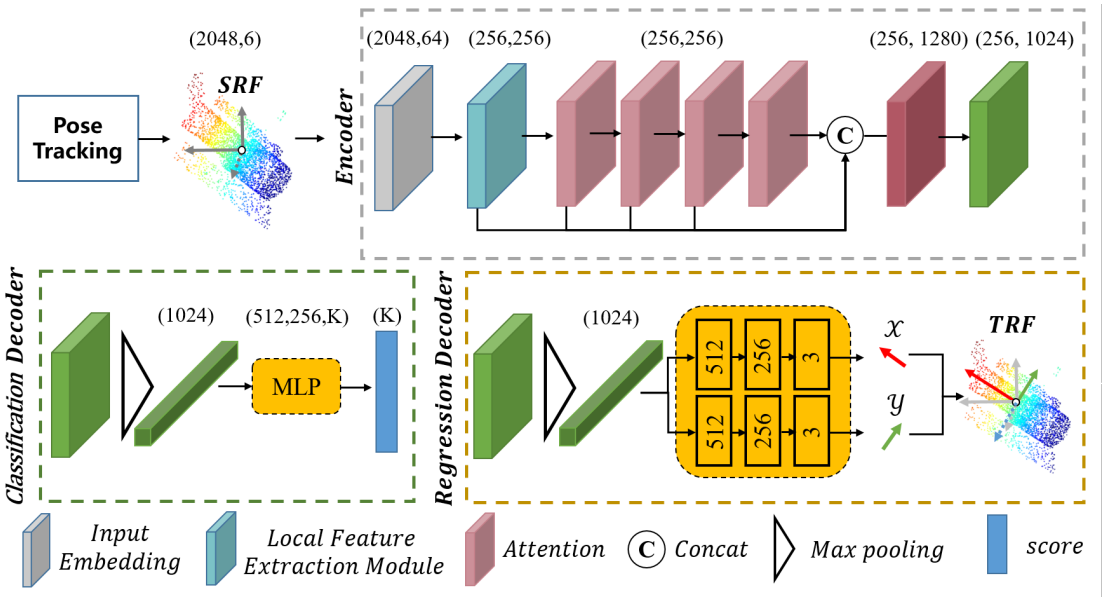


Fig. 5: The overall architecture of TRFE-Net. Our TRFE-Net predicts the target body reference frame based on the geometric features of the spacecraft. The classifier and regressor utilize the same encoder structure and different decoders. The output of the regressor is decoupled into direction vectors $\mathcal{X}$ representing the forward orientation and $\mathcal{Y}$ representing the upright orientation.

N_s. Then, for each sampled point p in $\mathcal{C}_s$, we utilize the KNN algorithm to identify its k nearest neighbors in $\mathcal{C}$, forming a group of k nearest neighbors. Finally, we apply MLP and max pooling operations to obtain the output local feature $\mathrm{F}_{\mathrm{local}} \in \mathbb{R}^{\mathrm{N_s} \times \mathrm{d_o}}$. For point cloud classification and orientations regression, we only need to predict a global class for all points. Therefore, the size of the N_s is reduced to 512 and 256 points in the two local feature extraction layers.

The local features $\mathrm{F}_{\mathrm{local}}$ are then fed into 4 stacked attention modules to learn semantically rich and discriminative representations for each point, followed by a linear layer to generate the output features. Overall, the encoder of TRFE-Net follows nearly the same design principles as the original Transformer, with the exception of positional embeddings being discarded since the coordinates of points already contain this information. More specifically, we first obtain the query matrix Q, the key matrix K, and the value matrix V respectively via three independent MLP Layers as follows:

$$QKV = \mathrm{MLP}(\mathrm{F}) \in \mathbb{R}^{\mathrm{N} \times \mathrm{D}} \tag{10}$$

Specifically, during the first attention operation, $\mathrm{F} = \mathrm{F}_{\mathrm{local}}$, $N = N_s$, $D = d_o$. Then, we compute the dot products of Q and the transposed K, dividing it by $\sqrt{D}$, and apply a Softmax function over the result to obtain a weight matrix W indicating the interdependence of pairwise point features.

$$W(Q, K) = \mathrm{Softmax}(\frac{\mathrm{QK}^{\top}}{\sqrt{\mathrm{D}}}) \in \mathbb{R}^{\mathrm{N} \times \mathrm{N}} \tag{11}$$

With this weight matrix, we compute the weighted feature matrix as:

$$\widetilde{\mathrm{F}} = \mathrm{Linear}(\mathrm{W} \cdot \mathrm{V}) \in \mathbb{R}^{\mathrm{N} \times \mathrm{D}} \tag{12}$$

Finally, the fused feature is computed using the following formula, where [,] denotes concatenation of feature tensors along the channel dimension.

$$\mathrm{Attention}(\mathrm{F}) = \mathrm{MLP}([\mathrm{F}_{\mathrm{loacl}}, \widetilde{\mathrm{F}}_1, \widetilde{\mathrm{F}}_2, \widetilde{\mathrm{F}}_3, \widetilde{\mathrm{F}}_4]) \tag{13}$$

For the classification and orientation regression tasks, we employ different decoder architectures. For the classification head, we use the same configuration as [48]. Specifically, we decouple the output rotation representation into the frontal direction vector $\mathcal{X} \in \mathbb{R}^3$ and the upright direction vector $\mathcal{Y} \in \mathbb{R}^3$, rather than quaternions or rotation matrices. Then, we use Gram-Schmidt orthogonalization as a mapping function to transform this representation into $\mathbb{SO}(3)$.

This approach has two advantages. First, for three-dimensional rotations, all representations in Euclidean spaces of four dimensions or less are discontinuous, they present continuous representations in more suitable 5D and 6D spaces. This was proven in experiments in [49]. Second, directly regressing the precise rotation angles of specific objects is a challenging task prone to overfitting. By decoupling into separate regressions for two directional vectors, it encourages the network to learn features on intra-class directional vectors rather than the entire point cloud. These features contain similar design knowledge and semantic preferences, enabling the model to better induce and generalize.

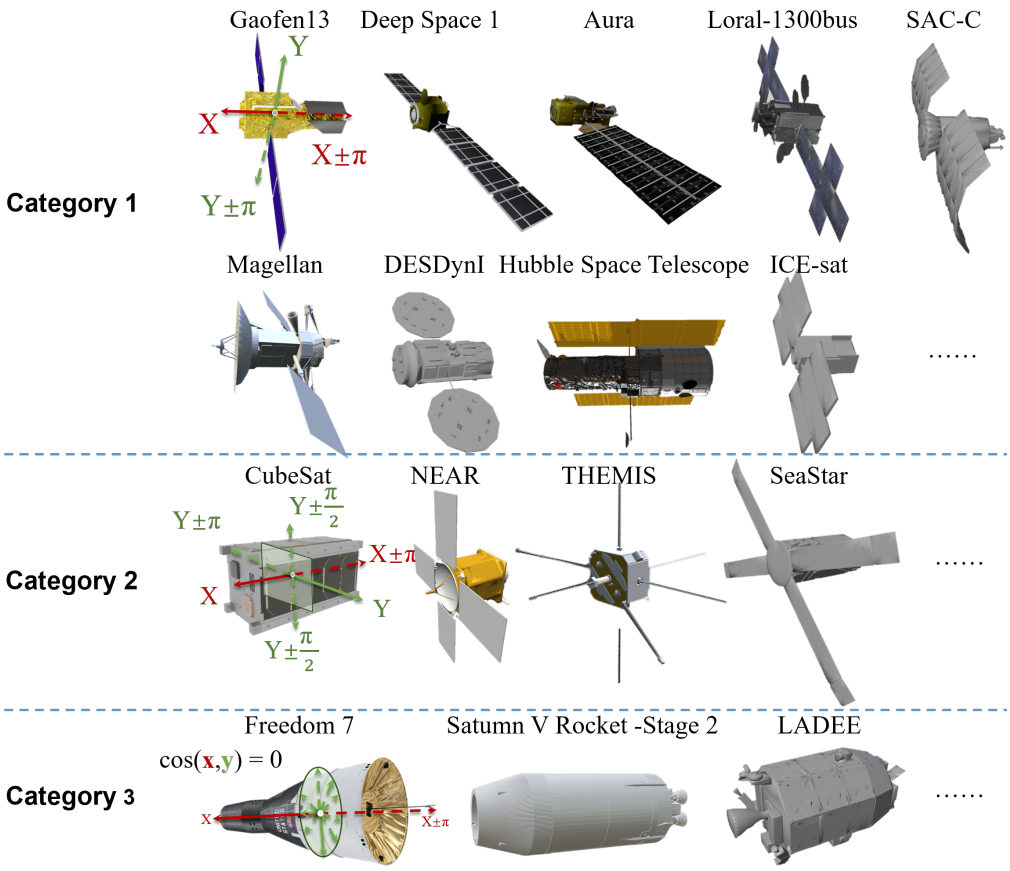


Fig. 6: A subset of shape category clustering. We combine geometric and symmetry properties to cluster spacecraft and define their forward and upright directions. Different categories of spacecraft use different loss functions to regress orientation.

3. The setting of the solution space

We primarily define categories and the frontal direction of spatial objects based on their geometric symmetry and similarity, as illustrated in Fig 6. In addition to solar panels, the main structure of satellites is typically rigid and equipped with effective payloads such as radar and docking rings. Therefore, the orientation can be estimated based on the arrangement of effective payloads on the main structure. Solar panels are usually positioned on the side of the main structure, which can be used to define the vertical direction.

Although different shape categories exhibit distinct features, some categories may be handled by similar strategy to find their frontal orientations and upright orientations. Those categories can be clustered together and processed with the same regression network. As a result, redundant networks can be removed and then the memory cost could be reduced.

We cluster categories by increasing the freedom of direction labels. Specifically, following the principle of symmetry, we allow the network to predict vertical or completely opposite directions as candidate answers. The insight here is that for pose estimation tasks, regardless of whether the positive direction of the target body reference frame points towards the positive or negative direction of effective payloads, the geometric properties and semantic information it encompasses remain significant, thus making it an effective pose reference frame. Moreover,by expanding the solution space, ambiguities in the network learning process are reduced, resulting in faster convergence of the model.

4. Loss Function

For the classification task, the standard multi-class cross-entropy[50] is used as the loss function. We elaborate on the design of the loss function for the orientation regression task. To maintain gradient stability and robustness to outliers, we utilize the angle error (in radians) between vectors as the loss value. The loss functions for 2-dof and 4-dof could be expressed uniformly as follows:

$$\mathcal{L}_{self} = 2 \cdot \arccos(< v \cdot \widetilde{v} >) \tag{14}$$

where v and $\widetilde{v}$ respectively denote the network output direction vector and the ground truth. Note that for different degrees of freedom in the solution space, a segmented training loss strategy is employed, using the minimum error value among all solutions as the loss.

For cases where the solution space manifests as a plane, such as category 3 in Fig 6, the solution set for the vertical direction is always constrained within a plane perpendicular to the positive direction. Therefore, we employ sine loss for supervised training:

$$\mathcal{L}_{cross}(v_y, \widetilde{v_x}) = 1 - \sin(v_y, \widetilde{v_x}) \tag{15}$$

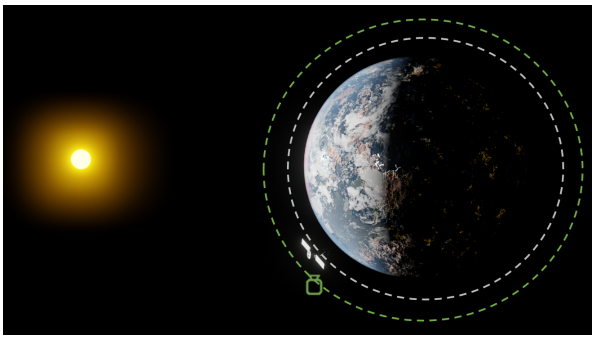
The final regression loss for each category is defined as follows:

$$\mathcal{L}_{reg} = \begin{cases} \mathcal{L}_{self}(\mathcal{X}, \widetilde{\mathcal{X}}) + \mathcal{L}_{self}(\mathcal{Y}, \widetilde{\mathcal{Y}}) & \text{category} = 1 \\ \mathcal{L}_{self}(\mathcal{X}, \widetilde{\mathcal{X}}) + \mathcal{L}_{self}(\mathcal{Y}, \widetilde{\mathcal{Y}}) & \text{category} = 2 \\ \mathcal{L}_{self}(\mathcal{X}, \widetilde{\mathcal{X}}) + w\mathcal{L}_{cross}(\mathcal{Y}, \widetilde{\mathcal{X}}) & \text{category} = 3 \end{cases} \tag{16}$$

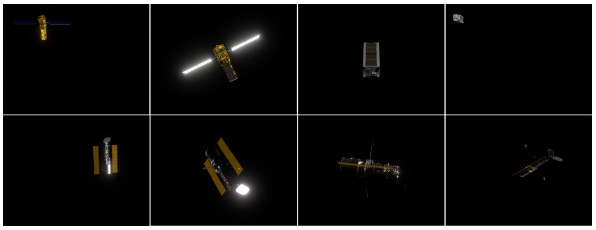
where w is hyper-parameter, seting as 0.1 empirically.

IV. EXPERIMENTS

In this section, we first introduced the PEU dataset and evaluation metrics. Next, we evaluated the performance of our framework on unknown space satellite without prior information using the PEU dataset. Both quantitative and qualitative results demonstrate that despite not requiring CAD models or images from the target, our method achieves comparable or even higher performance compared to methods that rely on priors.



(a) Settings for physical rendering of PEU-track.



(b) A subset of PEU-track.

Fig. 7: PEU-track.

A. PEU Dataset

We introduce a dataset named PEU (Pose Estimation for unknown spacecraft) with the aim to foster research on both the unknown target and temporal information processing for spacecraft pose estimation. The PEU dataset includes the pose tracking dataset PEU-track and the large-scale point cloud dataset PEU-cloud. The dataset and source files are publicly available at *Projectpage*.

1. *PEU-track*

To construct PEU-track, we utilized realistic satellite models, accounting for the influence of celestial backgrounds, the Sun, and global illumination. To create the 3D models of the target satellites, we conducted detailed modeling of each mechanical component based on the original CAD files available on the NASA website. This included components such as solar panels, antennas, screws, and labels. The material parameters for each component were assigned according to the visual characteristics of real satellite metal materials.

The satellite models were placed at their actual orbital altitudes, and their motion was modeled using real orbital velocities. In the experiments, the HST orbits approximately 340 miles above the Earth's surface, moving at a velocity of 5 miles per second, with random spin set to simulate failure conditions. We simulated relative distances according to the scenarios of space rendezvous, with the distance between the tracking satellite and the target satellite restricted to within 100 meters. The chaser was placed in a slightly higher orbit than the target satellite and transferred to the target orbit through two deceleration maneuvers, The scene setup is shown in Fig. 7(a). A virtual camera in the Blender engine was used to capture image data, with camera intrinsics calibrated using a standard checkerboard pattern. For close-range scenarios (<10m), we set up two virtual cameras to simulate stereo vision for acquiring 3D point cloud data. For long-range scenarios (>10m), we directly output the depth values from the images as point cloud data, simulating the data produced by a Lidar system that has been jointly calibrated and fused with the camera.

It is important to note that the focus of these simulated scenarios is not on generating more precise and realistic sensor data, but rather on how to utilize the acquired images and point cloud data for pose estimation and reconstruction of unknown space targets without model priors. This includes handling various target objects and different conditions such as large displacements, symmetry, specular highlights, and incomplete observations.

The dataset consists of three objects: Hubble Space Telescope, Gaofen-13, and CubeSat. For each target satellite, 20 scenes were generated, each comprising a sequence of 300 frames, resulting in a total of 18K RGB images and depth maps. The resolution of each image is 1440x1080, some of which are shown in Fig 7(b). To the best of our knowledge, this is the first spacecraft trajectory dataset with depth information and multiple targets. Table I summarizes the differences between PEU-track and other datasets. Given that our pose estimation framework targets unknown Spacecraft without prior information, these images will be exclusively used for testing purposes.

TABLE I: Overview of existing SSA datasets.

	SPEED[11]	SPEED+[51]	URSO[16]	SwissCube[14]	Cube-CDT[26]	SPARK2022[52]	**PEU-track**
Images	15.3k	70k	15k	50k	22k	32.4k	18k
Object Classes	1	1	2	1	1	1	3
CAD Model	×	×	×	No Texture	×	×	Texture
Trajectories	×	×	×	✓	✓	✓	✓
Image Resolution	1920 ×1200	1920 ×1200	1080×960	1024×1024	1440×1080	1440×1080	1440×1080
Color	×	×	✓	✓	✓	✓	✓
Depth	×	×	×	×	×	×	✓
Mask	×	×	×	✓	×	×	✓

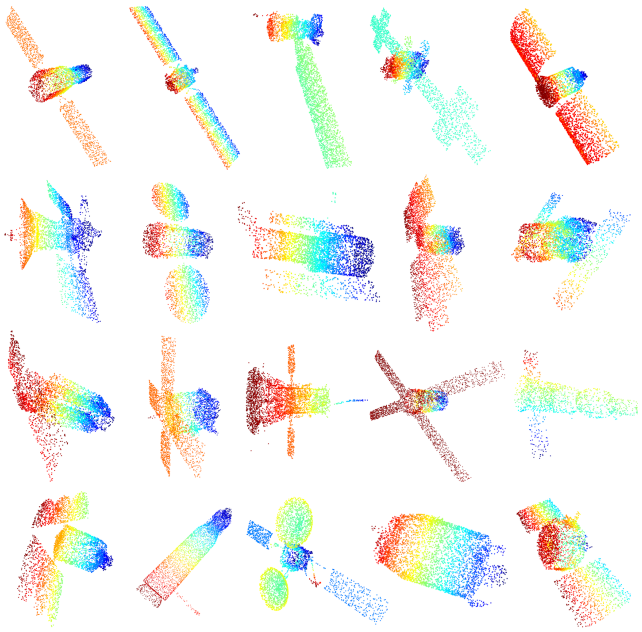


Fig. 8: A subset of PEU-cloud.

2. *PEU-cloud*

One of the key aspects of our approach is to leverage historical spacecraft knowledge to predict future unknown spacecraft. Therefore, we constructed a point cloud dataset named PEU-cloud to train TRFE-Net, enabling it to learn a universal geometric representation of spacecraft from large-scale spacecraft data. PEU-cloud comprises 100 aligned point cloud models of spacecraft. Specifically, these models were obtained from the NASA public website, manually aligned using Blender, and finally, 4000 points were uniformly sampled on the surfaces to generate point clouds. Inspired by point cloud self-supervised learning, each object is randomly rotated K times (set at 2000) in $\mathbb{SO}(3)$ space to generate new data. These rotated point clouds are then split in a ratio of 7:2:1 for training, testing, and validation sets, respectively. This results in 140,000 point clouds for training, 40,000 for testing, and 20,000 for validation. A subset of the PEU-cloud data is illustrated in Fig 8.

B. Metrics

We adopt the evaluation criteria proposed in the Kelvin's Pose Estimation Challenge (KPEC) [11] as the pose evaluation metric. The pose error E_P is based on the combination of the translation error E_T and rotation error E_R. Let t_{gt} and t denote the ground truth and estimated translation vectors of an image. The translation error of image j is defined as the Euclidean distance between the estimated translation vector and the ground truth translation vector, normalized by the magnitude of the ground truth position vector:

$$E_T^j = \frac{\|t - t_{gt}\|_2}{\|t_{gt}\|_2} \tag{17}$$

Let q_{gt} and q denote the rotation quaternion ground truth of an image and its estimation. The rotation error E_R^j of image j is defined as the angle (in radians) required to align the estimated quaternion orientation with the ground truth quaternion orientation:

$$E_R^j = 2 \cdot \arccos\left(| < q, q_{gt} > |\right) \tag{18}$$

The total error E_P is the average of the sum of rotation and translation errors for the $\mathcal{J}$ images in the test set:

$$E_P = \frac{1}{\mathcal{J}} \sum_{j=1}^{\mathcal{J}} (E_R^j + E_T^j) \tag{19}$$

For 3D shape reconstruction, we use the chamfer distance between the reconstructed mesh and the ground truth mesh, The variable x represents sampling in the model $\mathcal{O}$ at 5mm intervals.

$$\begin{aligned} E_M = &\frac{1}{2\mathcal{O}_1} \sum_{x_1 \in \mathcal{O}_1} \min_{x_2 \in \mathcal{O}_2} \|x_1 - x_2\|_2 + \\ &\frac{1}{2\mathcal{O}_2} \sum_{x_2 \in \mathcal{O}_2} \min_{x_1 \in \mathcal{O}_1} \|x_1 - x_2\|_2 \end{aligned} \tag{20}$$

C. Effect of TRFE

1. *Implementation and Training Details*

We adopt a transfer learning strategy to enable the model's encoder to learn generic features from a large-scale dataset. Fortunately, the ModelNet40 dataset [53] provides objects already aligned in a canonical coordinate system, where the *+x-axis* points toward the positive orientation and the *+y-axis* points toward the upright

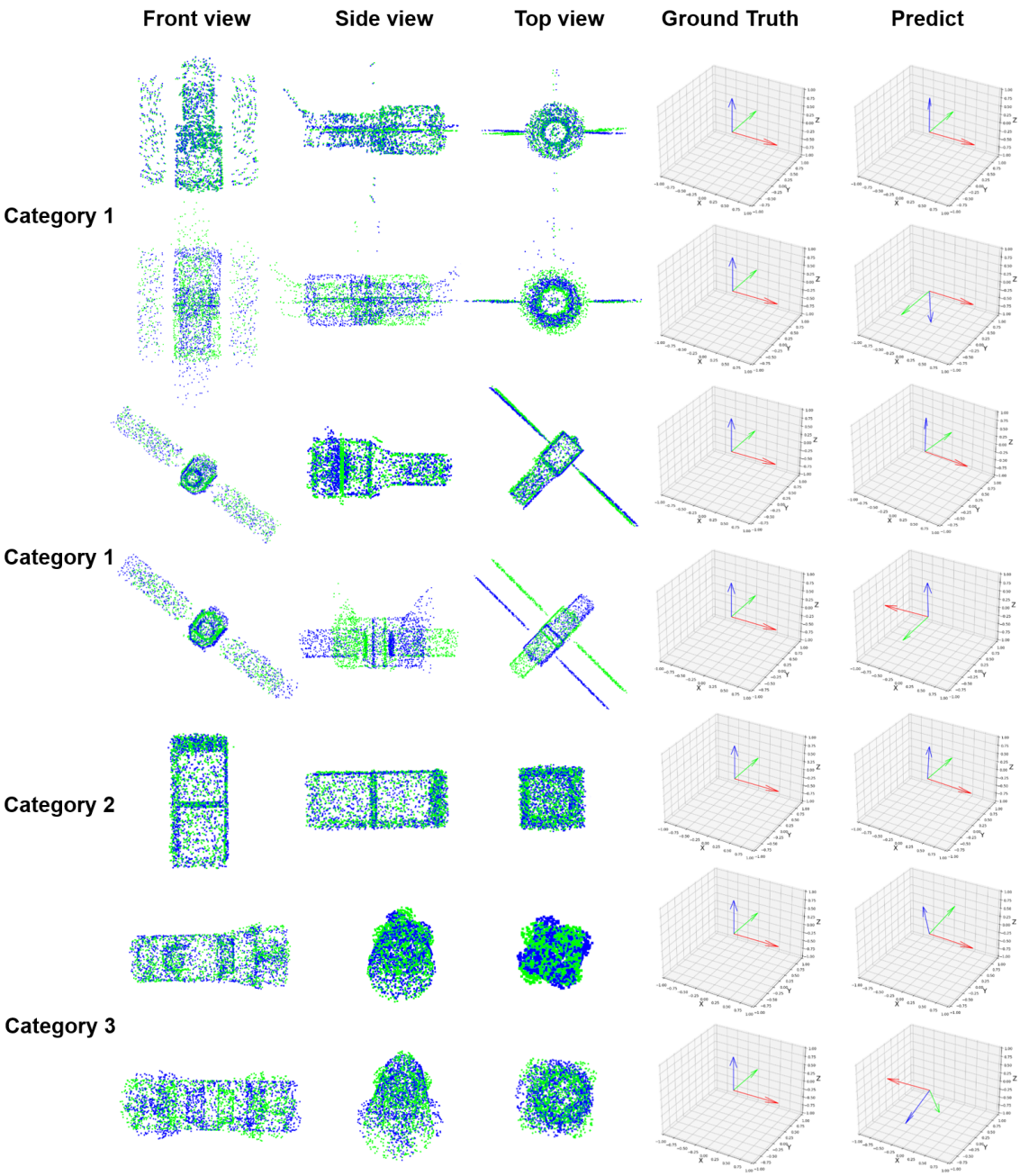


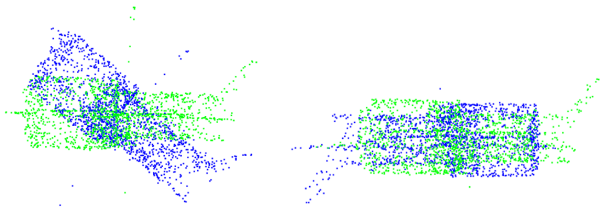
Fig. 9: The qualitative results of TRFE-Net. **Left**: Point cloud visualization. The green color represents the ground truth of the TRF, while the blue color represents the estimate value. **Right**: Coordinate system Visualization. The red and green arrows represent the positive X and Y directions, respectively, while the blue arrow represents the positive Z direction generated through the right-hand coordinate system.

TABLE II: Ablation study of our design choices on TRFE-Net.

Metrics	w/o Two-Stage	w/o Decoupled Regression	w/o Expand the solution space	**TRFE-Net**
$E_R(^\circ)$	7.469	5.998	5.613	**5.172**

direction. The ModelNet40 dataset comprises 40 object categories totaling 12,311 shapes. To obtain a dataset covering various potential poses, each model in ModelNet40 was randomly rotated 100 times at uniformly sampled angles, generating a new orientation.

We employ classification as an auxiliary task to pretrain the TRFE-Net model on the ModelNet40 dataset . Subsequently, we utilize the pre-trained encoder and discard the decoder for fine-tuning. Specifically, TRFE-Net is fine-tuned on our proposed PEU-Cloud dataset, where the classifier learns from all data while the regressors only



(a) Results of the single label. (b) Results of the multiple labels.

Fig. 10: The green color represents the ground truth of the TRF, while the blue color represents the estimate from TRFE-Net. In some angles where the learning is not sufficient, the approach of expanding the solution space can enhance the robustness of TRFE-Net.

learn from corresponding category data. Ultimately, we obtain a classifier and N regressors for our divide-and-conquer approach.

Pre-training and [54] with a learning rate of 10^{-3}, weight decay regularization of 0.0001, a batch size of 512, and set the pre-training and fine-tuning epochs to 400 and 100, respectively. These hyperparameters were selected based on validation performance.

To enhance robustness during training, we apply the following data augmentation techniques to the input point clouds. Gaussian noise N (0, 0.02) is added to perturb each point's position . Random translations in the range [-0.2, 0.2], anisotropic scaling in the range [0.67, 1.5], and random point dropout with probability [0, 0.875] are also applied. Furthermore, to simulate articulated motions of space targets and structural damage caused by collisions, we performed random articulation rotations and block cropping on the solar panels in the training dataset. This mimics incomplete observations that may occur during long-term missions.

2. *Result on PEU-cloud*

The qualitative results of TRFE-Net are shown in Fig 9. We investigated the effectiveness of the design choices in TRFE-Net, including the two-stage strategy of classification before regression, decoupled design of regression heads, and expanding solution space. Tab II records the results of our ablation experiments. The term "w/o Two-Stage" denotes the strategy of eliminating the two-stage process of classification followed by regression. In this comparative experiment, we solely employ a single regression network to predict the TRF, instead of using multiple expert regression models.

As described in Section III-C, we decouple the output rotation representation into the frontal direction vector $\mathcal{X} \in \mathbb{R}^3$ and the upright direction vector $\mathcal{Y} \in \mathbb{R}^3$. Then, we use Gram-Schmidt orthogonalization as a mapping function to transform this representation into $\mathbb{SO}(3)$. The term "W/o Decoupled Regression" refers to the removal of the decoupled regression setting, which led to a decrease in performance by directly outputting quaternion representation for rotation using the regressor.

We attribute this to two main reasons: 1. Quaternion representation of rotation is discrete, whereas our regression of a 6D vector is continuous in space, which is more suitable for regression tasks. 2. Directly regressing the rotation in the model tends to be more oriented towards a task specific to a particular object, which leads to a decrease in performance when dealing with multiple objects. By decoupling the regression, we reduce the learning complexity from directly regressing the entire rotation to only learning the significant directional vectors of the model, thus reducing the complexity of model learning.

The term "w/o Expand the solution space" denotes using only a single direction of solution as the training label, without considering the object's symmetric properties. This setup reduces the accuracy of TRFE-Net, possibly due to the ambiguity in learning caused by the similarity of symmetric structures.

As illustrated in the second, fourth, and seventh rows of Fig. 9, TRFE yielded completely opposite results on Hubble, Gaofen 13, and Saturn V Rocket-Stage 2. This outcome aligns with our explanation in Section III-C.3. We adhere to the principle of symmetry, allowing the network to predict either the vertical direction or its exact opposite as candidate answers to avoid learning ambiguities caused by symmetry. For pose estimation tasks, regardless of whether the positive direction of the target body's reference frame points towards the positive or negative direction of the effective payloads, the geometric properties and semantic information it encompasses remain significant, thus it is still considered an effective pose reference frame. Moreover, by expanding the solution space, the ambiguities in the network learning process are reduced, resulting in faster convergence of the model.

Apart from that, due to the discrete nature of random rotations used during training relative to the continuous $\mathbb{SO}(3)$ space, model performance can falter at certain angles not well-represented in the training data. As shown in Fig 10. Our strategy of setting multiple candidate correct directions provides more robust pose estimation across various angles.

D. Effect of PT and NOR

In this section, we evaluate the effectiveness of the pose tracking and Neural Object Reconstruction pipeline. First of all, we selected a small subset of samples from the PEU dataset characterized by large-scale displacements, symmetrical structures, and specular highlights (*Hubble_0003, Hubble_0004, Cubesat_0001, Cubesat_0002, Gaofen_0001, Gaofen_0006*) for qualitative and quantitative visualization analysis. The qualitative results of PT and NOR are shown in Fig 11 and Fig 12, respectively. Finally, we evaluate each condition on the test set of the PEU dataset and report both pose estimation accuracy and reconstruction quality metrics in Tab III. Furthermore, the prediction results of TRFE-Net are used to rectify the SRF

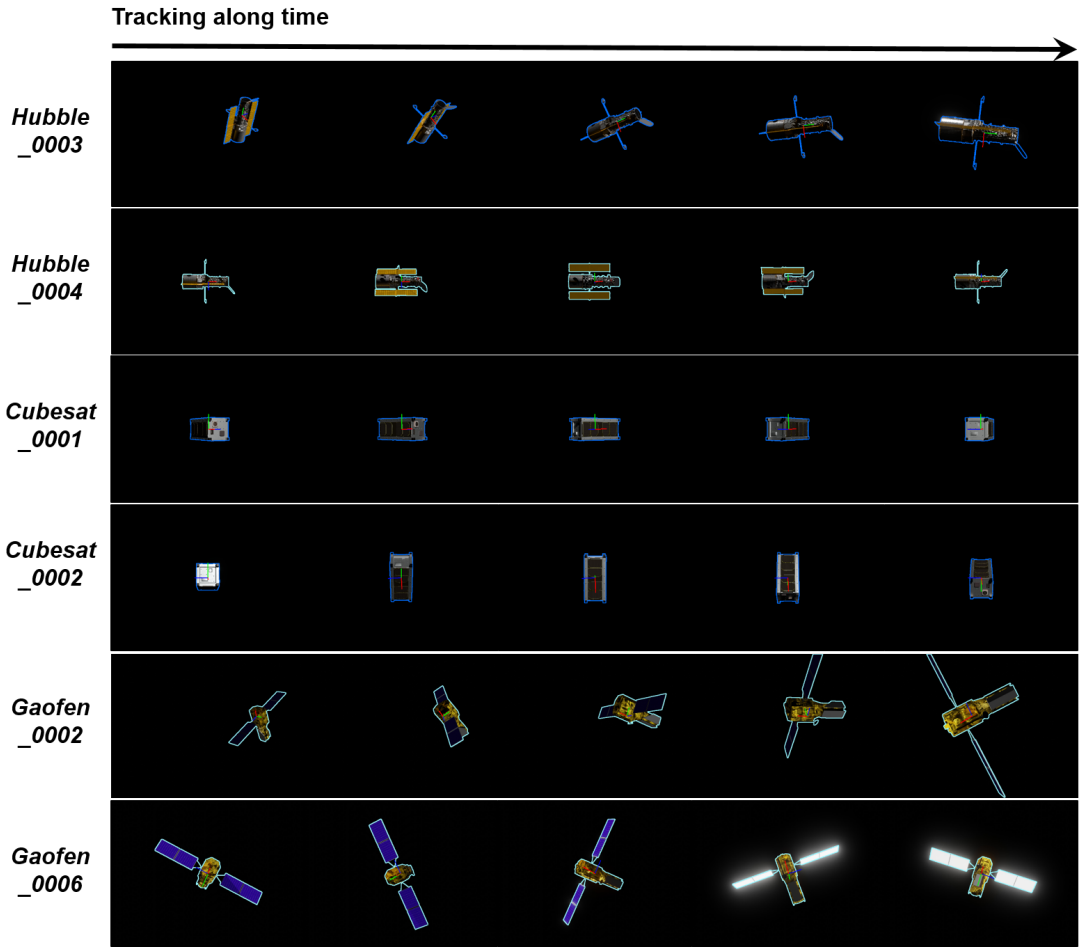


Fig. 11: 6-DoF pose tracking visualization. Our method demonstrates robust pose tracking performance across different targets. Particularly, qualitative results confirm the effectiveness of our method in handling symmetrical objects, large-scale displacements, and specular highlights.

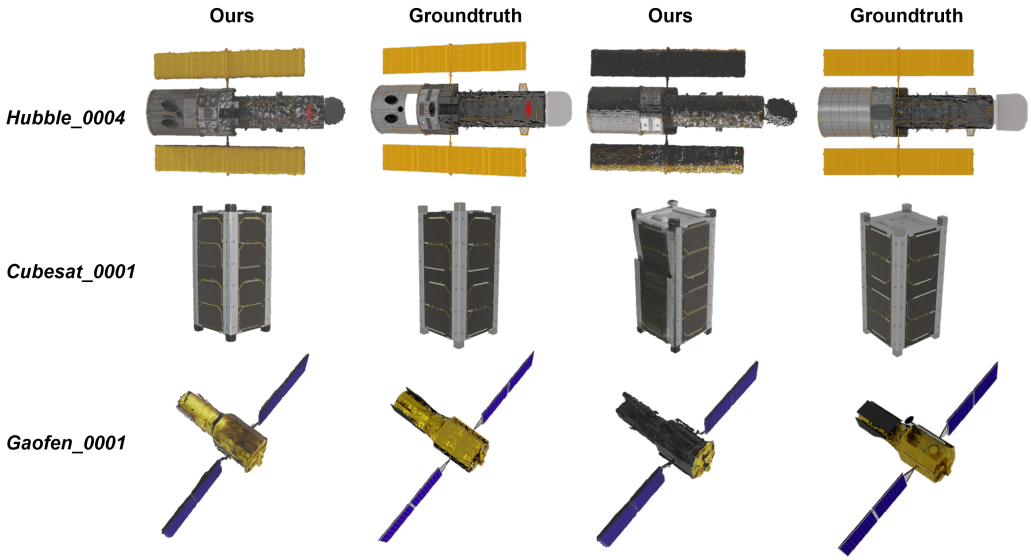


Fig. 12: Qualitative results of online neural reconstruction for sample trajectories. Note that the viewpoints of trajectories *Hubble_0004* and *Gaofen_0001* do not cover the entire spacecraft, resulting in unsampled and unreconstructed gaps.

TABLE III: Ablation study of our design choices on Pose Trcking. The E_P metric includes the rotation error (E_R) and translation error (R_T). The E_M metric is used to measure the loss of 3D reconstruction.

Object	Metrics	w/o Temporal Context	w/o 3D Feature Match	w/o NeRF Optimization	**Pose Tracking**
Hubble	E_P (-) ↓	0.1657	0.0892	0.0685	**0.0486**
	E_R (°) ↓	5.2972	3.2345	2.7189	**2.2892**
	E_M(cm) ↓	-	2.081	0.619	**0.594**
GaoFen	E_P (-) ↓	0.2324	0.1812	0.1549	**0.1456**
	E_R (°) ↓	6.7296	5.7584	5.0107	**4.7242**
	E_M(cm) ↓	-	1.979	0.513	**0.485**
CubeSat	E_P (-) ↓	0.1953	0.2189	0.0910	**0.0708**
	E_R (°) ↓	6.1566	6.9845	3.5497	**3.0054**
	E_M(cm) ↓	-	2.122	0.766	**0.722**
Average	E_P (-) ↓	0.1978	0.1631	0.1048	**0.0883**
	E_R (°) ↓	6.0611	5.3258	3.7598	**3.3396**
	E_M(cm) ↓	-	2.061	0.633	**0.603**

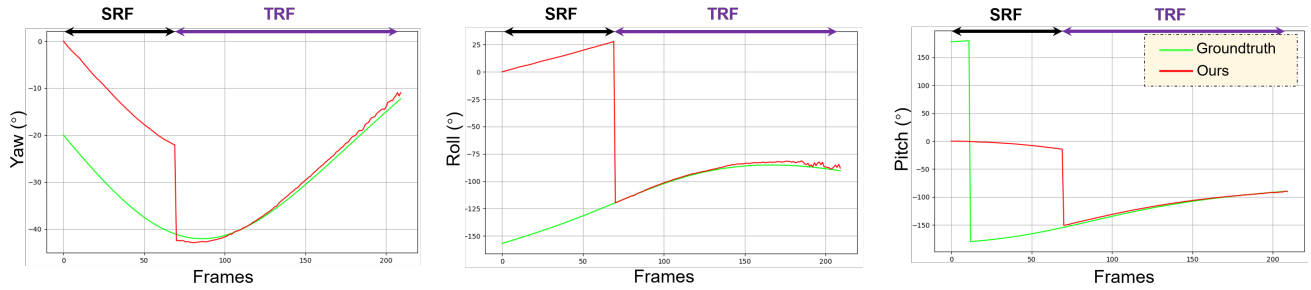


Fig. 13: Quantitative results of the trajectory *Hubble-0001* from the PEU-track dataset. The initial pose is with respect to the SRF. After correcting the reference frame using TRFE-Net, the model transitions from scene-centric pose tracking to object-centric pose estimation.

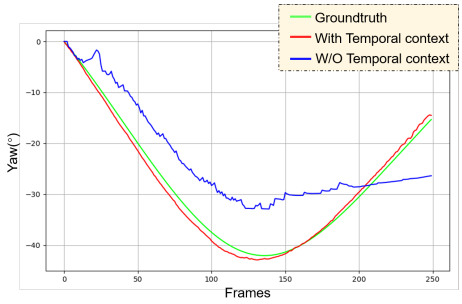


Fig. 14: Comparison of tracking performance on yaw angle. The data is derived from the trajectory of *Hubble_0003* from PEU-track.

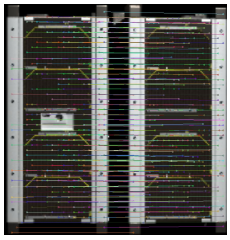


Fig. 15: Instances of failed matching. Relying solely on RGB-based matching leads to misalignment in symmetric objects, causing tracking drift and failure.

to the TRF. We combine the PT and TRFE processes to quantitatively analyze the angle tracking errors of the pose before and after coordinate system correction, as shown in Fig 13.

w/o Temporal Context removes the use of temporal context information by omitting the pose graph optimization step. Coarse pose estimation is performed only between consecutive image frames without global refinement. We demonstrate the importance of utilizing temporal information for more accurate predictions. As shown in Fig 14, leveraging temporal information from longer trajectories results in a smoother and more coherent pose trajectory, alleviating the issue of tracking drift. This is something that cannot be achieved with single-frame estimation alone.

As described in Section III-A, we employ the distance between the coordinates and normals of 3D points to filter the correspondences obtained between frames for handling symmetric objects. The setting w/o 3D feature match disables this configuration, relying solely on RGB feature matching. This results in an interesting scenario: when an object rotates to the opposite symmetric plane within the field of view, erroneous feature matching causes the system to believe it has returned to the original position, leading to the estimation of an opposite pose and further impacting subsequent tracking. Fig D illustrates the effectiveness of using 3D data for filtering matching relations when dealing with symmetrical objects.

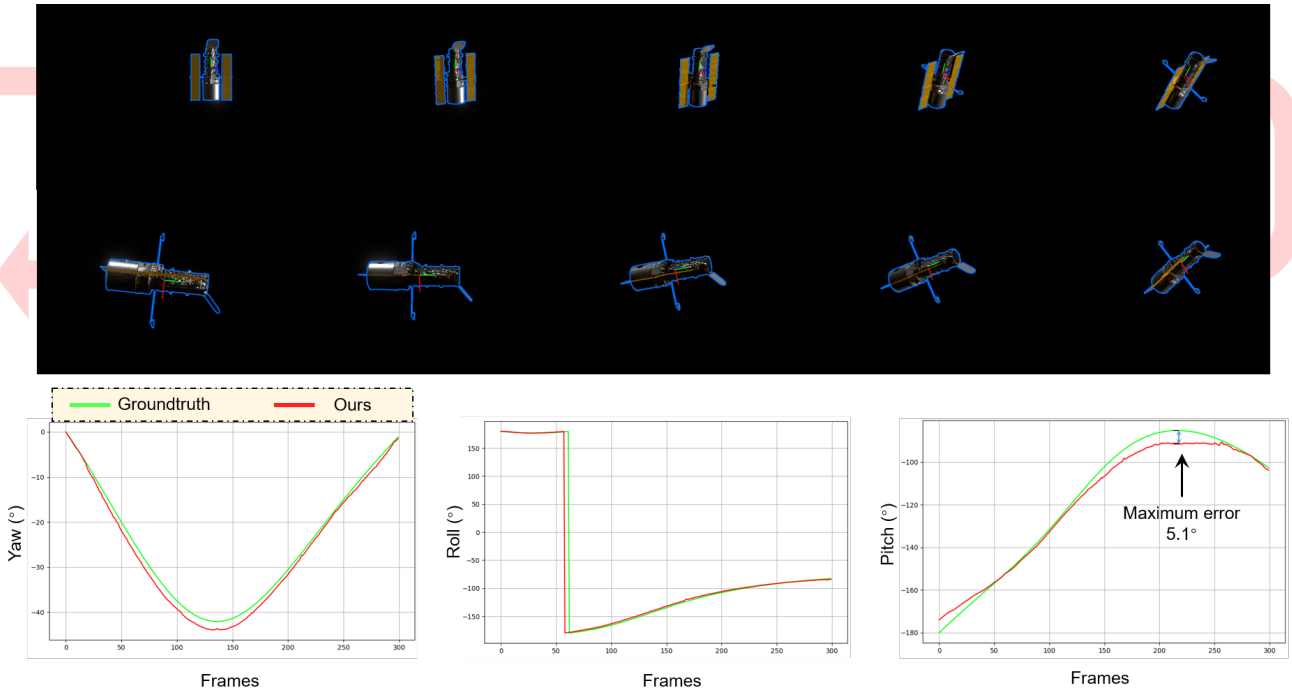


Fig. 16: Quantitative and qualitative results of the trajectory *Hubble_0003* from PEU-track. This trajectory simulates a scenario with large-scale displacements.

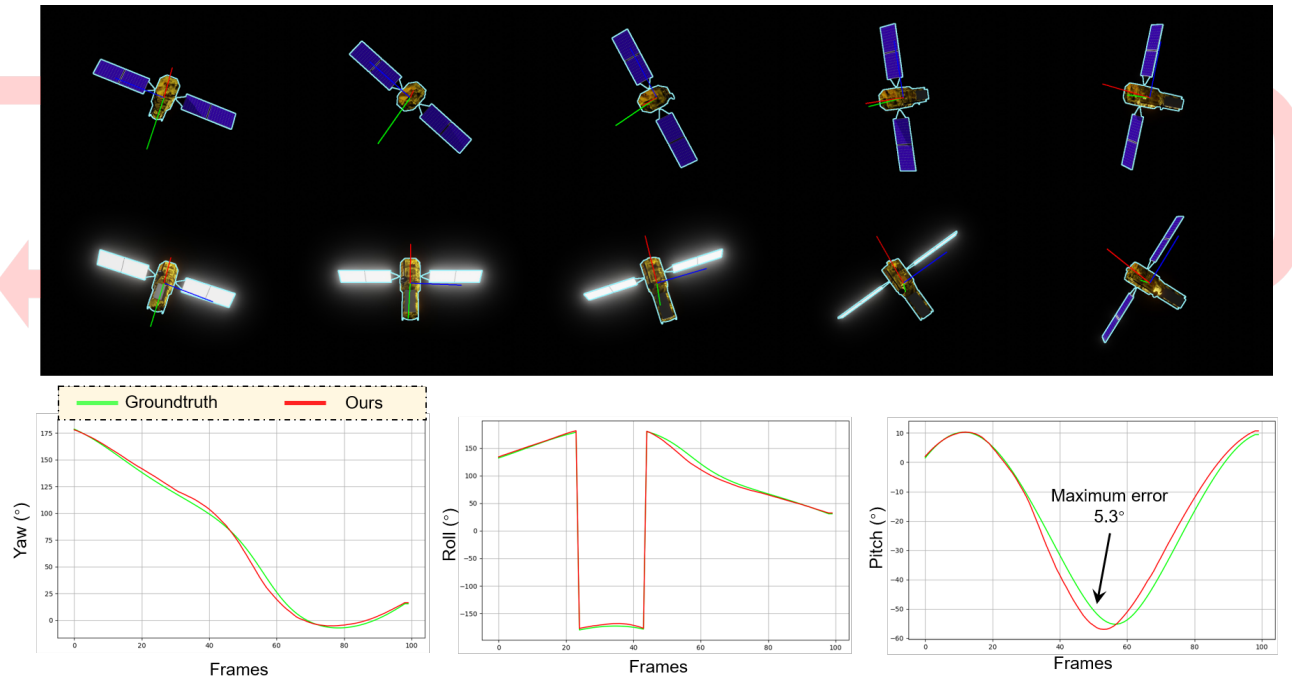


Fig. 17: Quantitative and qualitative results of the trajectory *Gaofen_0006* from PEU-track. This trajectory simulates a scenario with specular highlights.

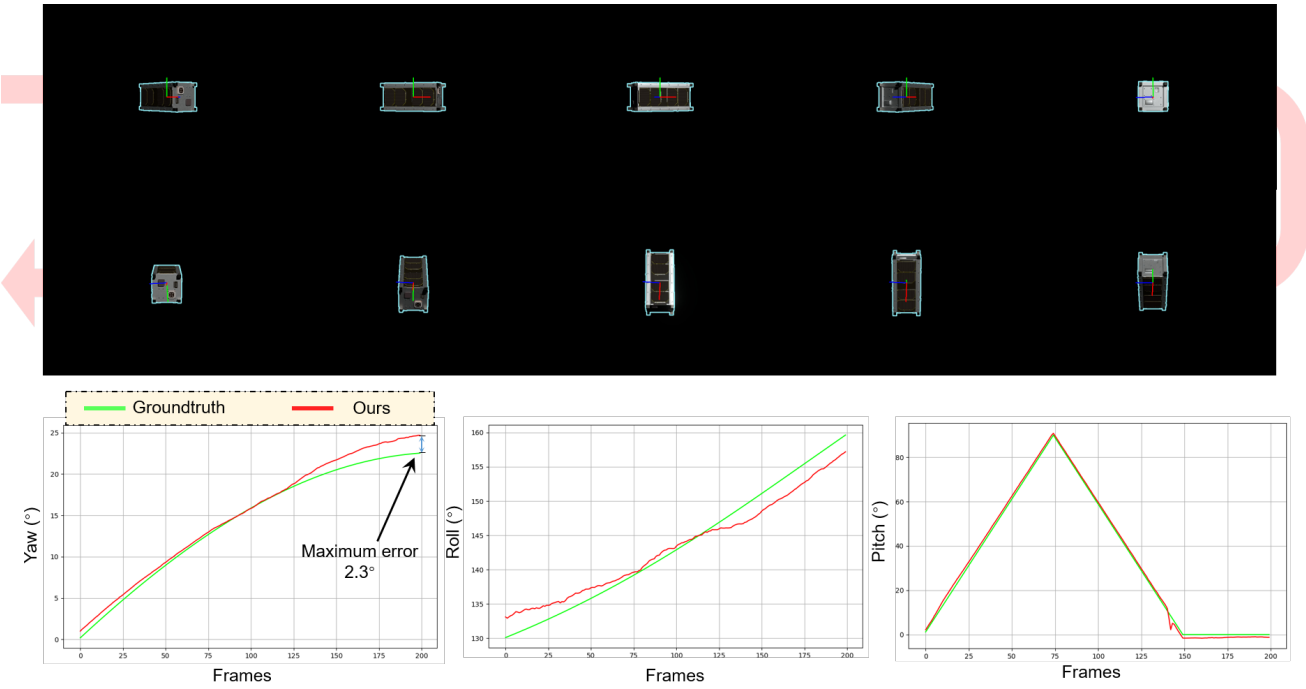


Fig. 18: Quantitative and qualitative results of the trajectory *Cubesat_0002* from PEU-track. This trajectory simulates a scenario with symmetrical structures.

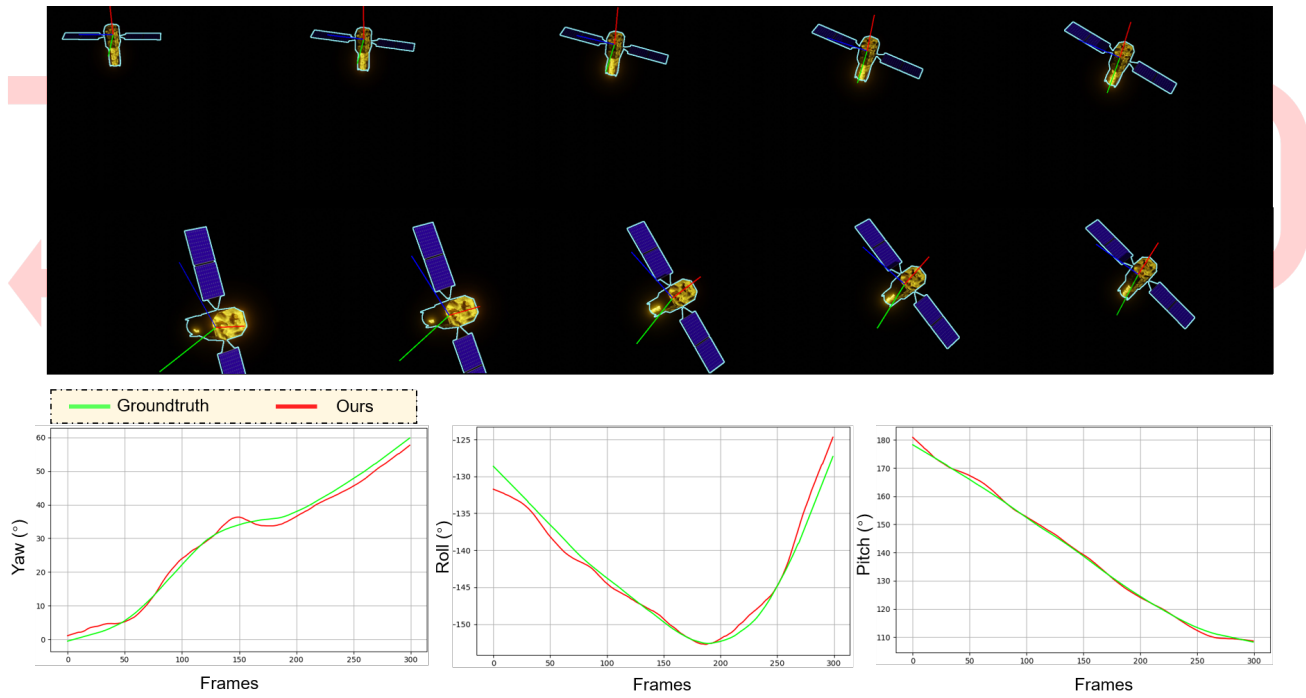


Fig. 19: Quantitative and qualitative results of the trajectory *Gaofen_0001* from PEU-track. This trajectory simulates a scenario with incomplete observations.

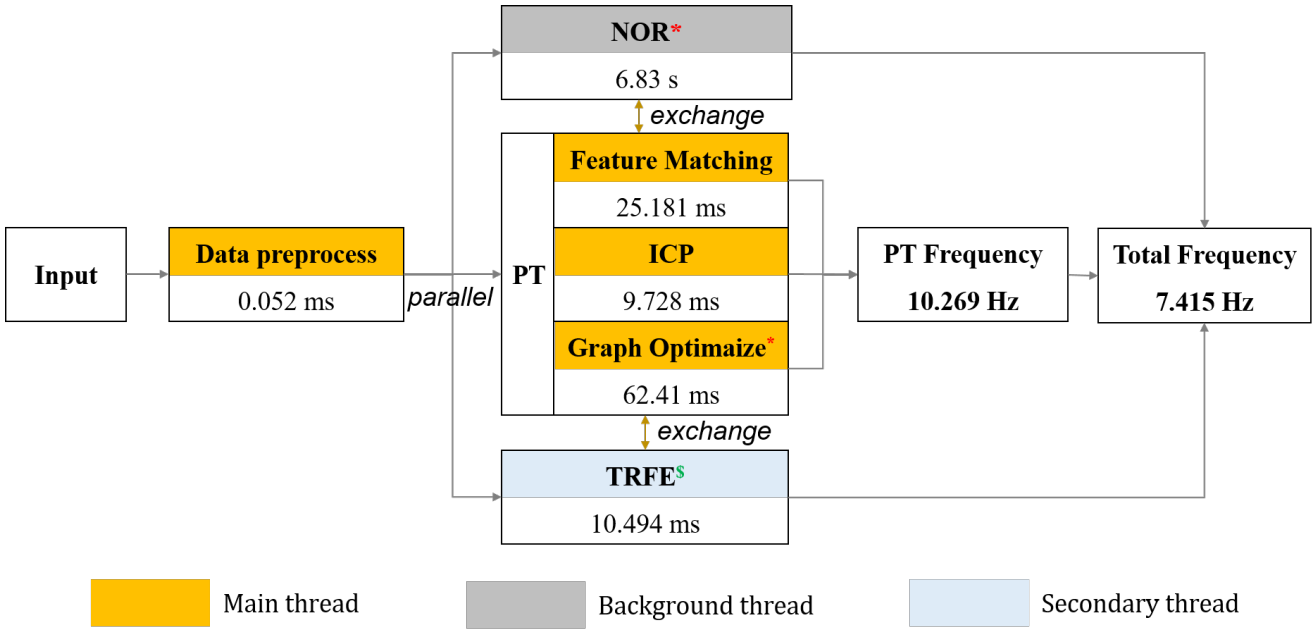


Fig. 20: The computation time required for pose estimation. (*) indicates that the process is executed once every certain number of frames. ($) indicates that the process is executed only once throughout the entire duration.

The setting w/o 3D feature match struggles to handle symmetric objects, resulting in inaccurate input pose parameters for neural reconstruction, which further impacts the overall reconstruction performance. w/o NeRF Optimization: In this condition, the NeRF model is solely used for view synthesis and reconstruction purposes. The process of jointly optimizing poses through NeRF's differentiable rendering is ablated. Noted that when NeRF optimization is removed, the pose error increases from 0.0883 to 0.1048. This result validates that utilizing NeRF's differentiable rendering for synchronized pose refinement can enhance the accuracy of our pose tracking pipeline.

E. Robustness testing

In space operations, several challenging conditions could occur: Firstly, due to the rapid movement and changes in the relative position of the spacecraft, large-scale displacements can occur. This poses a challenge for long-term pose tracking, as it may lead to tracking drift and catastrophic forgetting. Secondly, specular highlights, caused by reflections from the spacecraft's surface, especially from metallic or reflective materials, complicate visual tracking and pose estimation. Additionally, symmetrical structures can create ambiguities in determining the spacecraft's orientation and pose, as similar features may appear identical from different angles. This is particularly detrimental to pose estimation methods relying on single frames. Finally, occlusions occur when parts of the spacecraft are obscured by other objects, or when the camera's field of view fails to fully cover the target, making it difficult to obtain a complete view for accurate pose estimation and reconstruction.

To address these challenges, we specifically simulated these scenarios in the PEU-Track dataset. For instance, we simulated large-scale displacements by introducing significant movements and rotations in the trajectory of *Hubble_0003*. Specular highlights were modeled by incorporating reflective materials on the surface of the *Gaofen_13* spacecraft in rendering engine. Symmetrical structures were simulated by selecting spacecraft with inherently symmetrical designs, such as *CubeSat*. Lastly, we simulated partial occlusions by capturing incomplete camera views in the trajectory of *Gaofen_0001*. Furthermore, to increase the difficulty of pose estimation, we introduced sudden changes in rotational acceleration to simulate the natural evolution of an uncontrolled satellite.

We quantitatively analyzed the performance of our method under these challenging scenarios. The rotational components of the pose parameters were decoupled into yaw, pitch, and roll Euler angles and visualized separately. The experimental results demonstrate the effectiveness of our method under these challenging conditions. As shown in Fig. 16, our pose tracking and estimation algorithms maintained robust performance under large-scale displacements, without significant tracking drift or catastrophic forgetting. In Fig. 17, our method accurately estimated poses despite the presence of specular highlights, unaffected by reflections. In Fig. 18, our method correctly handled symmetrical structures by leveraging temporal context information, thus avoiding pose ambiguities. Finally, as shown in Fig. 19, under conditions of incomplete observations, our system maintains stable tracking by

effectively utilizing partial observations, avoiding tracking loss and significant oscillations.

F. Real-time analysis

Due to the utilization of GPU acceleration in the feature matching, pose graph optimization, and NOR stages of our method, our model was deployed on a platform equipped with an Intel i5 10400F CPU and an NVIDIA RTX 4070 GPU to measure the computation time required for each module.

As shown in Fig. 20, the primary processes include data preprocessing, PT, TRFE, and NOR threads. Data preprocessing involves point cloud denoising and image mask computation based on point cloud data. It is important to note that the NOR thread always runs in the background, while TRFE is executed only once globally. The PT thread processes frames at 10.269 Hz, while NOR runs in the background, requiring an average of 6.83 seconds per training iteration. Ultimately, the entire model operates at near real-time with a frequency of approximately 7.415 Hz.

V. CONCLUSION

This paper discusses the non-cooperative satellite pose estimation without prior information. The proposed method effectively bridges the gap of missing reference models through online target reference frame estimation and neural reconstruction. Meanwhile, the online pose graph optimization leverages temporal data knowledge to achieve smoother and more robust pose tracking performance. Experimental results demonstrate that our approach exhibits good performance on different spacecraft and shows strong robustness in handling challenging cases such as specular highlights, large-scale motions, thin structures, and symmetrical configurations.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant U22B2013 and 52105480.

REFERENCES

[1] Y. Xing, X. Cao, S. Zhang, H. Guo, and F. Wang, "Relative position and attitude estimation for satellite formation with coupled translational and rotational dynamics," *Acta Astronautica*, vol. 67, no. 3-4, pp. 455–467, 2010.

[2] W. De Jongh, H. Jordaan, and C. Van Daalen, "Experiment for pose estimation of uncooperative space debris using stereo vision," *Acta Astronautica*, vol. 168, pp. 164–173, 2020.

[3] J. L. Forshaw, G. S. Aglietti, N. Navarathinam, H. Kadhem, T. Salmon, A. Pisseloup, E. Joffre, T. Chabot, I. Retat, R. Axthelm *et al.*, "Removedebris: An in-orbit active debris removal demonstration mission," *Acta Astronautica*, vol. 127, pp. 448–463, 2016.

[4] Y. Li and A. Zhang, "Observability analysis and autonomous navigation for two satellites with relative position measurements," *Acta Astronautica*, vol. 163, pp. 77–86, 2019.

[5] D. Pinard, S. Reynaud, P. Delpy, and S. E. Strandmoe, "Accurate and autonomous navigation for the atv," *Aerospace Science and Technology*, vol. 11, no. 6, pp. 490–498, 2007.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[7] C. Harris, M. Stephens *et al.*, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[8] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 4, pp. 722–732, 2008.

[9] H. Rhody, "Lecture 10: Hough circle transform," *Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology*, 2005.

[10] S. Sharma, C. Beierle, and S. D'Amico, "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," in *2018 IEEE Aerospace Conference*. IEEE, 2018, pp. 1–12.

[11] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Märtens, and S. D'Amico, "Satellite pose estimation challenge: Dataset, competition design, and results," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 5, pp. 4083–4098, 2020.

[12] S. Sharma and S. D'Amico, "Neural network-based pose estimation for noncooperative spacecraft rendezvous," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4638–4658, 2020.

[13] A. Garcia, M. A. Musallam, V. Gaudilliere, E. Ghorbel, K. Al Ismaeil, M. Perez, and D. Aouada, "Lspnet: A 2d localization-oriented spacecraft pose estimation neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2048–2056.

[14] Y. Hu, S. Speierer, W. Jakob, P. Fua, and M. Salzmann, "Wide-depth-range 6d object pose estimation in space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 870–15 879.

[15] S. Zhang, W. Hu, and W. Guo, "6-dof pose estimation of uncooperative space object using deep learning with point cloud," in *2022 IEEE Aerospace Conference (AERO)*. IEEE, 2022, pp. 1–7.

[16] P. F. Proença and Y. Gao, "Deep learning for spacecraft pose estimation from photorealistic rendering," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6007–6013.

[17] S. Qiao, H. Zhang, F. Xie, and Z. Jiang, "Deep-learning-based direct attitude estimation for uncooperative known space objects," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.

[18] B. Chen, J. Cao, A. Parra, and T.-J. Chin, "Satellite pose estimation with deep landmark regression and nonlinear pose refinement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[19] A. Legrand, R. Detry, and C. De Vleeschouwer, "End-to-end neural estimation of spacecraft pose with intermediate detection of keypoints," in *European Conference on Computer Vision*. Springer, 2022, pp. 154–169.

[20] Z. Wang, Z. Zhang, X. Sun, Z. Li, and Q. Yu, "Revisiting monocular satellite pose estimation with transformer," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 5, pp. 4279–4294, 2022.

[21] X. Liu, H. Wang, X. Chen, W. Chen, and Z. Xie, "Position awareness network for noncooperative spacecraft pose estimation based on point cloud," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 1, pp. 507–518, 2022.

[22] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "Pose estimation for spacecraft relative navigation using model-based algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 431–447, 2017.

[23] Y. Wu, "Model reconstruction and pose optimization of non-cooperative rotating space target," *Opt. Precision Eng*, vol. 27, no. 8, pp. 1854–1862, 2019.

[24] V. Capuano, K. Kim, A. Harvard, and S.-J. Chung, "Monocular-based pose determination of uncooperative space objects," *Acta Astronautica*, vol. 166, pp. 493–506, 2020.

[25] J. I. B. Pérez-Villar, Á. García-Martín, and J. Bescós, "Spacecraft pose estimation based on unsupervised domain adaptation and on a 3d-guided loss combination," in *European Conference on Computer Vision*. Springer, 2022, pp. 37–52.

[26] M. A. Musallam, A. Rathinam, V. Gaudillière, M. O. d. Castillo, and D. Aouada, "Cubesat-cdt: A cross-domain dataset for 6-dof trajectory estimation of a symmetric spacecraft," in *European Conference on Computer Vision*. Springer, 2022, pp. 112–126.

[27] Z. Wang, M. Chen, Y. Guo, Z. Li, and Q. Yu, "Bridging the domain gap in satellite pose estimation: a self-training approach based on geometrical constraints," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.

[28] M. Jawaid, E. Elms, Y. Latif, and T.-J. Chin, "Towards bridging the space domain gap for satellite pose estimation using event sensing," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 866–11 873.

[29] T. H. Park and S. D'Amico, "Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap," *Advances in Space Research*, 2023.

[30] A. Stumpf, J.-P. Malet, P. Allemand, and P. Ulrich, "Surface reconstruction and landslide displacement measurements with pléiades satellite images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 95, pp. 1–12, 2014.

[31] G. Facciolo, C. De Franchis, and E. Meinhardt-Llopis, "Automatic 3d reconstruction from multi-date satellite images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 57–66.

[32] A. Alvera-Azcárate, A. Barth, J.-M. Beckers, and R. H. Weisberg, "Multivariate reconstruction of missing data in sea surface temperature, chlorophyll, and wind satellite fields," *Journal of Geophysical Research: Oceans*, vol. 112, no. C3, 2007.

[33] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[34] A. Mergy, G. Lecuyer, D. Derksen, and D. Izzo, "Vision-based neural scene representations for spacecraft," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2002–2011.

[35] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar, "Neural fields in visual computing and beyond," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 641–676.

[36] T. Mahendrakar, B. Caruso, V. M. Nguyen, R. T. White, and T. Steffen, "3d reconstruction of non-cooperative resident space objects using instant ngp-accelerated nerf and d-nerf," *arXiv preprint arXiv:2301.09060*, 2023.

[37] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "Graf: Generative radiance fields for 3d-aware image synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 154–20 166, 2020.

[38] S. Ullman, "The interpretation of structure from motion," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 203, no. 1153, pp. 405–426, 1979.

[39] X. Pang, F. Li, N. Ding, and X. Zhong, "Upright-net: Learning upright orientation for 3d point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 911–14 919.

[40] S. Noh, R. Kang, T. Kim, S. Back, S. Bak, and K. Lee, "Learning to place unseen objects stably using a large-scale simulation," *IEEE Robotics and Automation Letters*, 2024.

[41] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[42] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield, "Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 606–617.

[43] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8922–8931.

[44] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[45] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim, "Self-supervised learning of point clouds via orientation estimation," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 1018–1028.

[46] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," 2017.

[47] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[48] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, pp. 187–199, 2021.

[49] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5745–5753.

[50] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, pp. 19–67, 2005.

[51] T. H. Park, M. Märtens, G. Lecuyer, D. Izzo, and S. D'Amico, "Speed+: Next-generation dataset for spacecraft pose estimation across domain gap," in *2022 IEEE Aerospace Conference (AERO)*. IEEE, 2022, pp. 1–15.

[52] L. Pauly, M. L. Jamrozik, M. O. Del Castillo, O. Borgue, I. P. Singh, M. R. Makhdoomi, O.-O. Christidi-Loumpasefski, V. Gaudillière, C. Martinez, A. Rathinam *et al.*, "Lessons from a space lab: An image acquisition perspective," *International Journal of Aerospace Engineering*, vol. 2023, 2022.

[53] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Bing Han received the B.S. degree (Hons.) from Wuhan University of Technology, Hubei, China, in 2022. He is currently pursuing a Ph.D. in Mechanical Engineering at the School of Future Technology, Xi'an Jiaotong University, China.

His research interests include spacecraft pose estimation and deep learning.

Chenxi Wang received the Ph.D. degree in mechanical engineering from Xian Jiaotong University, Xi'an, China, in 2020. He is currently an Assistant Professor with the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China.

His research interests include rocket engine status monitoring and fault diagnosis, in orbit spacecraft space threat identification and trajectory prediction, satellite formation dynamics modeling and orbit attitude control, antenna in orbit assembly and manufacturing, space 3D printing, helicopter system identification and active vibration control, additive manufacturing process parameter optimization, process quality monitoring, and intelligent feedback control.

Xinyu Zhang is currently a graduate student majoring in Mechanical Engineering with the School of Future Technology, Xian Jiaotong University, Xi'an, China.

Her research interests include spacecraft identification and intelligent perception for non-cooperative targets

Zhibin Zhao (Member, IEEE) received the B.S. degree (Hons.) and the Ph.D. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2015 and 2020, respectively. He was also a Visiting Ph.D. Student in AI for healthcare at the University of Manchester, Manchester, U.K., from 2019 to 2020. He is now a Lecturer in mechanical engineering at the Depart ment of Mechanical Engineering, Xi'an Jiaotong University.

His current research interests include sparse signal processing and machine learning algorithms for machinery health monitoring and healthcare.

Dr. Zhao is an Associate Editor of IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

Zhi Zhai received the B.S. degree in mechanical manufacturing and automation and the Ph.D. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2007 and 2014, respectively. She is currently an Assistant Researcher with the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China.

Her research interests include remaining useful life (RUL) prediction of satellite battery,

and fault diagnosis of liquid rocket engine. She has a background in prognosis and health monitoring of spacecraft.

Jinxin Liu received the B.S.degree in mechanical engineering from the School of Electrical Engineering,Xi'an Jiaotong University,Xi'an.China,in 2011,and the Ph.D. degreein mechanical engineering from the School ofMechanical Engineering,Xian Jiaotong University, in 2016. He was a Visiting Scholar withthe Lawrence Berkeley National Laboratory,Berkeley,CA, USA, during 2014 and 2015.

He is currently a Professor with the Schoolof Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China.He is working on active safety control of machinery and equipment.including active vibration control, fault tolerant control, and precisioncontrol of large-scale,complex,and dynamical systems.

Naijin Liu received the Ph.D. degree from the China University of Science and Technology, New Taipei, Taiwan. He is currently the deputy Director of Qian Xuesen Space Technology Laboratory, China Academy of Space Technology, Beijing, China. He is also a Professor with the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China.

His research interests include satellite communication and space intelligent information networking. He is a council Member of the Chinese Society of Astronautics.

Xuefeng Chen (Senior Member, IEEE) received the Ph.D.degree from the School olMechanical Engineering,Xi'an Jiaotong University,Xi'an, China, in 2004.

He is currently a Full Professor with theSchool of Mechanical Engineering, Xi'an Jiaotong University. His research interests includefinite-element method, mechanical signal processing, and fault diagnosis.

Dr. Chen is also a member of the American Society of Mechanical Engineers (ASME). He received the National Excellent Doctoral Thesis Award in 2007, the First Technological Invention Award of Ministry of Education in 2008, the Second National Technological Invention Award in 2009, the First Provincial Teaching Achievement Award in 2013, and the First Technological Invention Award of Ministry of Education in 2015. He received the Science and Technology Award for Chinese Youth in 2013. He is the Chair of the IEEE Xi'an and Chengdu Joint Section Instrumentation and Measurement Society Chapter. He works as the Executive Director of the Fault Diagnosis Branch, China Mechanical Engineering Society. In addition, he hosted a National Key 973 Research Program of China as a Principal Scientist in 2015.