Revisiting the Domain Gap Issue in Non-cooperative Spacecraft Pose Tracking

Kun Liu, Yongjun Yu
Nanjing University of Science and Technology
Nanjing, China

njust_lk@njust.edu.cn, yyj_njust@163.com

Abstract

The deep learning (DL) algorithms have emerged as the foremost approach for close-range navigation of non-cooperative spacecraft. Given the unavailability of in-orbit images, DL models are typically trained on synthetic data. However, when deployed in real-world scenarios, they often encounter a domain gap that leads to performance degradation. To address this, we propose a self-supervised framework based on RANSAC EPnP. Specifically, we first trained a landmark regression network and an object detection network on synthetic data. Utilizing the trained landmark regression network, we then infer keypoints on real-world images. Through RANSAC EPnP, we filter outliers and calculate poses as pseudo-labels. Building on this, the pose estimation network is further trained, optimizing outliers to bridge the domain gap. The proposed method brings a significantly lower training cost compared to adversarial training, the prevailing method for bridging the domain gap, making it suitable for in-orbit training. Moreover, we utilize a Kalman filter to predict the bounding boxes, which circumvents the domain gap's impact on the performance of the object detection network, resulting in more precise bounding boxes. Lastly, we validated the performance of the proposed algorithm on the SPEED+ and SPARK 2024 datasets, achieving the 2nd place in the SPARK 2024 competition.

1. Introduction

Close-range autonomous navigation of non-cooperative objects is one of crucial capabilities for future spacecraft, requiring precise estimation of the target's position and attitude. In this field, non-cooperative objects typically refer to spacecraft or space debris that cannot establish stable communication links and lack known reflectors [1]. Such objects are commonly encountered in missions such as debris removal and in-orbit servicing. Over the past years, image-based sensors have been considered a great source of information for non-cooperative spacecraft pose estimation. Currently, methods of spacecraft pose estimation utilizing deep learning have attracted considerable attention, enabling monocular cameras to perform close-range navigation. Their potential is demonstrated in numerous studies[2–4].

The outstanding performance of DL-based methods in spacecraft pose estimation is closely tied to supervised training on large-scale datasets. Such supervised training assumes that training and testing data are drawn from the same distribution [5]. Yet, unlike terrestrial applications, obtaining large-scale in-orbit data for space missions is almost impossible. As a result, training on synthetic datasets has become the prevalent strategy, unavoidably facing the issue of data distribution discrepancies, known as the Domain Gap. Moreover, DL-based methods face unique challenges in the space environment. Primarily, data acquisition is extremely difficult; images of spacecraft during the close-range rendezvous phase are obtainable solely during the execution of missions. This leads to a scarcity of real in-orbit images for training and validation on the ground. Additionally, the onboard computational resources are limited, making it challenging to support large-scale training tasks. Therefore, it is crucial to fine-tuning using in-orbit training to overcome the domain gap, ensuring such training is computationally efficient to remain feasible.

As in-orbit images do not become available until close-range rendezvous in space, one key strategy is to instead create high-fidelity surrogate images on-ground that can be used to evaluate the robustness of Neural Network models trained on synthetic images across domain gap on in-orbit images [6].This evaluation strategy was first introduced by Tae Ha Park *et al.* [7], who organized the second international Satellite Pose Estimation Competition (SPEC2021) and released the SPEED+ dataset. SPEED+ is the next-generation dataset for spacecraft pose estimation, with a specific emphasis on model robustness across the domain gap. In the results and analyses following the competition, the winning teams explicitly stated that adversarial training with unlabeled test images plays a crucial role in addressing the domain gap [3]. However, Tae Ha Park *et al.* [3] argue that the approach of simultaneously utilizing synthetic training and unlabeled laboratory images is impractical in real mission scenarios because the target spaceborne images are not available until rendezvous. Training on large-scale synthetic datasets is not practicable during this phase. The test set of the SPEED+ dataset contains a diverse distribution of spacecraft images across multiple angles and distances ranging from 3 to 10 meters. It is difficult to acquire such a rich array of spacecraft images in actual rendezvous scenarios. Moreover, the test images of SPEED+ are discretely distributed, lacking temporal information for optimization. Building on this, Djamila Aouada *et al*. [8] from the Interdisciplinary Centre for Security, Reliability, and Trust (SnT) at the University of Luxembourg organized SPARK 2024. Images of the training and validation sets have been synthetically generated, while images of the test set have been acquired in the SnT's ZeroG Laboratory[9]. The released test sets contain time-sequential continuous images, better reflecting the conditions of space rendezvous scenarios where spacecraft attitudes cannot be uniformly acquired.

We hold the view that adversarial training might not be the most suitable solution in the aerospace sector. It demands access to both simulated and real-world images to bridge the domain gap, requiring substantial data storage for simulated images on servicing satellites. Additionally, the introduction of new targets entails the new simulated images, potentially imposing a significant load on data transmission. Therefore, the solution proposed in this article, aiming to meet the demands of actual space scenarios. The proposed method is divided into three parts: Offline Training, Online Training, and Flight. These correspond to ground training, in-orbit training, and flight missions in practical applications, respectively. During the Offline training phase, a style augmentation strategy is employed to address the overfitting caused by insufficient simulation image textures. In the Online Training phase, RANSAC EPnP is utilized to calculate pseudo-labels for self-supervised training. In the flight phase, UKF (Unscented Kalman Filter) is applied for navigation filtering. The target box is calculated from the prediction value of the filter instead of the target detection network. This method improves target box accuracy and reduces the computational demand of the target detection network. Furthermore, during the Online Training phase, the proposed method may lead to a new catastrophic forgetting issue due to the image distortion, which will be discussed in the article.

2. Related work

In this section, we review advanced methods for monocular pose estimation of spacecraft and discuss the key issues and solutions.

2.1. Monocular spacecraft pose estimation

A critical aspect of spacecraft pose estimation is attitude estimation. Advanced approaches to attitude estimation includes a variety of strategies: classfication for viewpoint [10], soft classification for probabilistic direction estimation [11], landmark vector regression [12, 13] and landmark heatmap regression [14]. Currently, the combination of landmark heatmap regression and EPnP [15] is considered the state-of-the-art.

Furthermore, some studies focuses on the network architectures and training strategies for feature extraction from images. Chen *et al*. [14] argue that maintaining high resolution during the feature extraction process is crucial in landmark regression methods. They recommend using HR-Net to preserve high resolution during feature extraction, thereby generating heatmaps with superior spatial precision. Yinlin Hu *et al*. [13, 14] believe that Wide-Depth-Range variations pose drastic difficulties for the network and have proposed a multi-scale training and inference fusion method based on FPN.

2.2. Domain adaptation in space

The algorithms for bridging the domain gap is broadly defined as domain adaptation, essentially training models on unlabeled data. Adversarial training is the most crucial method of domain adaptation. In the SPEC2021 competition, the two winning teams utilized adversarial training to bridge domain gap. They highlighted the critical role that adversarial training played in their success. [3]. Furthermore, Mohsi Jawaid [3] optimized landmark regression by refining the preliminary object bounding box. This technique proves especially beneficial in processing images under extremely dim lighting conditions, aiding in the retention of crucial visual features that might be compromised by image downsizing or poor visibility. Wang *et al*. [16] implemented a multi-task learning strategy, including landmark regression and semantic segmentation tasks, further augmented by a self-training mechanism. Their work highlights the advantages of integrating multiple learning tasks, enhancing pose estimation accuracy by leveraging the synergistic strengths of landmark detection and semantic segmentation.

However, adversarial training requires access to both simulated and real-world images, necessitating significant data storage for simulated images on service satellites. If there are multiple targets—a likely scenario—switching service targets would necessitate updating a large volume of new target simulation data, imposing substantial data transmission burdens. Therefore, some research directions may hold more practical value. Park *et al*. [17] introduced a multi-task framework named SPNv2 and the Online Domain Refinement (ODR) strategy for in-orbit training. ODR enhances SPNv2 by fine-tuning its normalization layer parameters, guided by the goal of minimizing the Shannon entropy of the segmentation head. Their work does not require simulated data. Though slightly less precise, it offers greater practical application potential. Their subsequent work [18] demonstrates that with the aid of Kalman filters, final accuracy could be elevated to higher levels, proving the efficacy of this approach in practical scenarios.

3. Approach

To achieve spacecraft pose estimation, the first step involves acquiring reliable spacecraft bounding boxes. Then, 2D landmarks are inferred using the proposed landmark heatmap regression network, and finally, position and attitude are calculated through RANSAC EPnP.

This section introduces the key components and essential details of the proposed method. The entire framework is illustrated in Figure 1, primarily divided into three stages: Offline Training (in Section 3.1), Online Training(in Section 3.2), and Flight(in Section 3.3). During the Offline training phase, we trained a Faster R-CNN[19] network for object detection and a landmark heatpmap regression network. The trained Faster R-CNN is then utilized to refer the initial target bounding box in the online training phase. The bounding boxes in online training need further optimization to enhance the Online Training performance (see Section 3.2.3). Subsequently, spacecraft images are cropped according to the bounding boxes for Online Training or inference of landmark regression network. To address the domain gap in landmark regression network, we introduce a self-supervised training method based on RANSAC EPnP and reveal how the method works. Finally, in the flight phase, we suggest using Kalman filters for bounding boxes prediction to replace the target detection network. This method not only reduces computational costs but also predicts more accurate bounding boxes (the target detection network is also affected by domain gap).

3.1. Offline training

3.1.1 Architecture

The proposed landmark heatmap regression architecture includes encoder and decoder as shown in Figure 1. The encoder references the Swin-Transformer's [20], utilizing Swin Transformer Blocks for feature extraction, and Patch Merging for downsampling. The decoder references the swin-Unet [21], utilizing Swin Transformer Blocks for feature extraction, and Patch Expanding to accomplish upsampling.

Moreover, we implemented an improvement in the decoder part. Drawing inspiration from studies [22] and [23], we developed a coarse-to-fine supervision strategy, as shown in Figure 2. Gaussian kernels of different sizes are employed to enhance and optimize the outputs at stages 2 and 3.

Table 1. List of Data augmentation used in offline training phase. p represents the activation probability

Augmentation	Effect	p
Style Aug.[25]	Texture enchancement	0.5
Sunflare	Exposure enhancement	0.5
Blur	Noise enhancement	0.5
Contrast	Contrast enhancement	0.5

3.1.2 Loss function

To attain coarse-to-fine supervision, the loss function is determined as follows,

$$L_h = \sum_i \left(\text{JS}\left(\hat{H}_i^{(s=3)} \middle| \bar{H}_{(i,\sigma=1.5)} \right) + \text{JS}\left(\hat{H}_i^{(s=2)} \middle| \bar{H}_{(i,\sigma=3)} \right) \right). \tag{1}$$

Where $\hat{H}_i^{(s=3)}$ represents the i-th landmark heatmap output from stage 3. The heatmap is normalized by softmax. JS$(\cdot)$ represents the Jensen-Shannon divergence [24], which is used to calculate the distance between the estimated value $\hat{H}$ and the true value $\bar{H}$. Gaussian kernels σ of size 3 and 1.5 are employed for stage 2 and stage 3 respectively.

3.1.3 Data augmentation

In scenarios involving an unknown target domain, data augmentation is often the sole strategy for domain generalization. Kisantal *et al.* [2] mentions that there are some discrepancies between synthetic and real-world image properties, such as the spacecraft's texture and illumination. Therefore, the specific data augmentation techniques utilized in this study are detailed in Table 1, notably Style Augmentation [25] and Sunflare, which significantly contribute to domain generalization. Data augmentation excluding Style Augmentation is implemented by Albumentations[26].

3.2. Online training

As shown in Figure 1, Online Training generates pseudo labels according to the prediction from stage3, and the trains the network by calculating the loss of stage2.

3.2.1 Pseudo labels generation

The first step in generating pseudo-labels is to regress to 2D landmarks from the output of stage 3. The output that normalized by softmax is present as a probability distribution heatmap. Following the differentiable spatial to numerical transform(DSNT) [27], the probability distribution will be multiplied with the preset $X_{i,j} = (2j - (n+1))/n$ and $Y_{(i,j)} = (2i - (m+1)/m$.

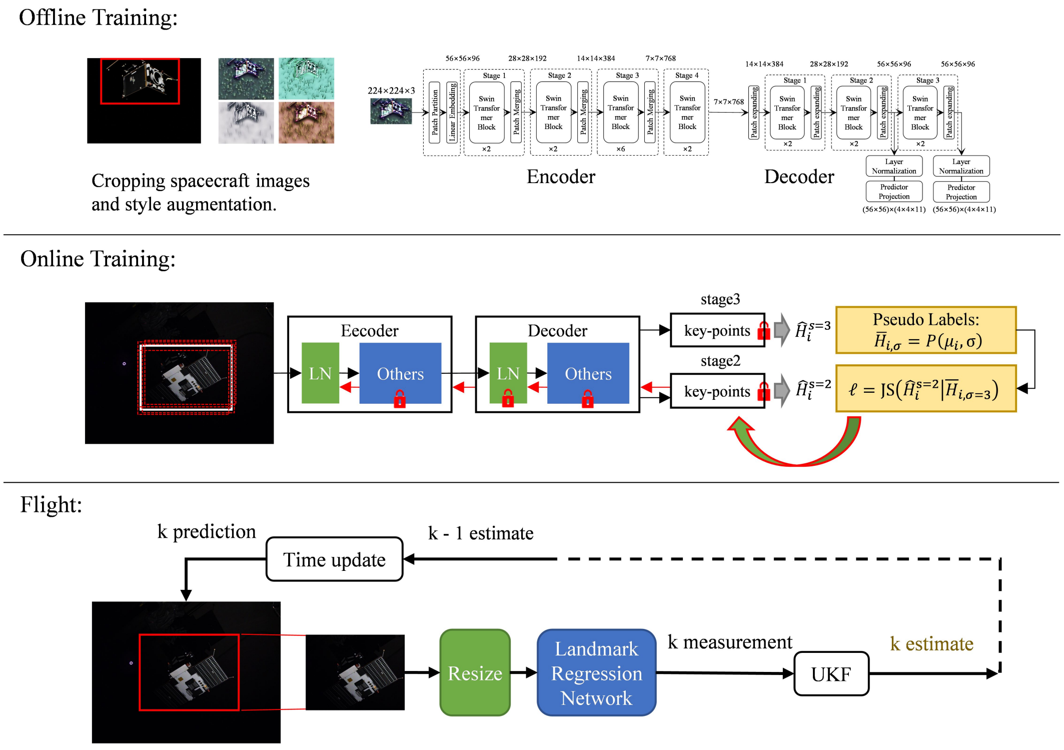


Figure 1. The whole framework of the proposed method.

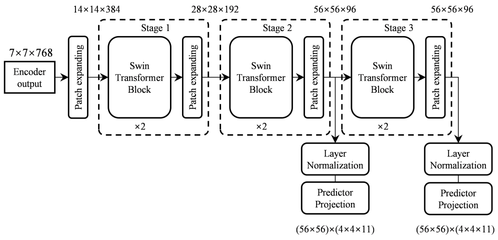


Figure 2. The proposed decoder architecture.

$$\mu_i = \mathrm{DSNT}(\widehat{H}) = \left[\left\langle \widehat{H}_i^{(s=3)}, X\right\rangle_{\mathrm{F}} \quad \left\langle \widehat{H}_i^{(s=3)}, Y\right\rangle_{\mathrm{F}}\right] \tag{2}$$

where $\widehat{H}_i^{(s=3)}$ is the normalized output from stage 3, and $\langle \cdot, \cdot \rangle_{\mathrm{F}}$ denotes the Frobenius inner product, which is equivalent to taking the scalar dot product of vectorized matrices. μ_i is the estimated coordinates of the i-th landmark.

After determining the estimated value of the 2D projection, utilize PnP algorithm to resolve the nonlinear equation.

$$\mu_i = K \begin{bmatrix} R & t \end{bmatrix} X_i \tag{3}$$

where the camera internal parameter K and the 3D coordinates X_i of the landmarks in the body coordinate system are known quantities. RANSAC EPnP is used to solve the equation to obtain the rotation matrix R and the translation matrix t which is treated as pseudo labels. Subsequently, these 3D coordinates are reprojected onto the image plane, leading to the generation of pseudo-Gaussian heatmaps.

$$\bar{H}_{(i,\sigma)} = P(\bar{\mu}_i, \sigma) \tag{4}$$

where, $P(\cdot)$ represents the mass function, which generates a heatmap at the 2D coordinate $\bar{\mu}_i$ with a Gaussian kernel of size σ.

3.2.2 Outlier filter

RANSAC EPnP is used for outlier landmarks elimination, with reprojection error serving as the criterion to classify landmarks as inliers or outliers. A landmark is considered an inlier if its reprojection error falls below the threshold; otherwise, it is classified as an outlier.

$$E_i(R,t) = \left\| \mu_i - K \begin{bmatrix} R & t \end{bmatrix} X_i \right\| \tag{5}$$

where, μ_i denotes the estimated coordinates and $E_i(R,t)$ represents the reprojection error of the i-th landmark.

3.2.3 Loss function

Online training encourages outliers to learn the correct 2D coordinates and align across domains by minimizing the JS distance of the heatmaps. Mathematically, let $\theta_{\mathcal{G}}$ denote learning parameters in the selected part of the network $\mathcal{G}$. Let $\theta_{\mathcal{G}}^{\text{norm}} \subset \theta_{\mathcal{G}}$ denote the normalization layers of $\theta_{\mathcal{G}}$. Online training amounts to solving,

$$\min_{\theta_{\mathcal{G}}^{\text{norm}}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(x_i; \theta_{\mathcal{G}}\right) \tag{6}$$

where, x_i represents the unlabeled target domain image. The loss function determines as follow.

$$l(x_i;\theta_G) = \begin{cases} \text{loss1} = & \sum_i \text{JS}(\hat{H}_i^{(s=2)} \| \bar{H}_{(i,\sigma=3)}) \\ \text{loss2} = & \sum_i (\text{JS}(\hat{H}_i^{(s=2)} \| \bar{H}_{(i,\sigma=3)}) \\ & +\text{JS}(\hat{H}_i^{(s=3)} \| \bar{H}_{(i,\sigma=1.5)})) \end{cases} \tag{7}$$

Pseudo labels are generated from high-precision heatmaps, and the inliers landmarks is likely to be overfitted if trained the high-precision heatmaps directly. With this consideration, it is proposed to minimize the loss of the middle layer. The comparison between loss1 and loss2 will be examined in the subsequent section.

3.2.4 LayerNorm layer parameter updates

The feature extraction layers have been trained to extract correct features from synthetic images with various styles and textures during the offline training phase. Updating all the parameters may destroy the trained feature extractor due to unreliable pseudo-labels. Therefore, only the parameters of LayerNorm(LN) layers are updated.

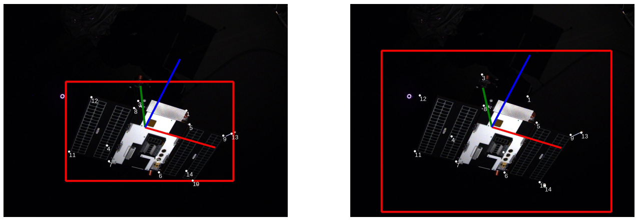
LN normalizes the input features along channel direction. x_i is input features, $\hat{x}_i$ represents normalized output features.

$$\mu = \frac{1}{d}\sum_{i=1}^{d} x_i \quad \sigma^2 = \frac{1}{d}\sum_{i=1}^{d}(x_i - \mu)^2 \quad \hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{8}$$

LN scales and aligns features through two learnable parameters γ and β.

3.2.5 Bounding boxes adjustment

Since domain adaptation is not applied to the target detection network, the predicted bounding boxes tend to be either too large or too small. As depicted in Figure 3, larger bounding boxes increase the susceptibility of landmark predictions to background interference, whereas smaller boxes may result in missing crucial landmarks. To mitigate this issue, we propose recalculating and updating the bounding box based on pose prediction results. Additionally, we implemented data augmentation on the bounding box for online training. We adjusted the four parameters of the bounding box by adding a mean error, calculated as ten percent of the box's length or width. This adjustment was made because networks trained with a static bounding box perform excessive sensitivity to the target box's position during self-supervised training. It's important to note that, for the same image, even minor bounding box modifications result in substantial pose estimation inaccuracies, which is undesirable.



(a) An example of a too-small bounding box.

(b) An example of a too-large bounding box.

Figure 3. The visualization of the bounding box's impact. (a) The predictions for points 13 and 14 are constrained by the boundaries of the bounding box. (b) The prediction of point 12 is disrupted by light spots in the background.

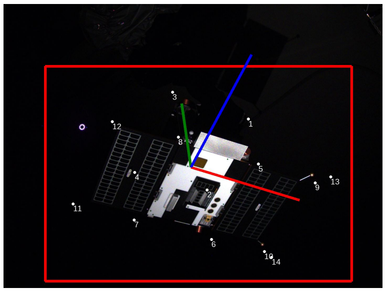


Figure 4. The forgetting issue in Online Training. The predicted values of landmarks have deviated from the set values, possibly due to image distortion.

Additionally, when training on SPARK2024, we encountered a distinct type of catastrophic forgetting, leading to significant deviations in the predicted landmarks from their initial setups, as shown in Figure 4. This deviation, not present in the SPEED+ test set, is suspected to be caused by camera distortion. The actual distortion parameters of the camera, not revealed by SPARK2024, likely intensified this issue. Accurate distortion parameters are crucial for effective Online Training. Unfortunately, we currently lack a robust method to model errors in landmarks post Online Training. To mitigate distortion-related issues, we used target boxes that precisely fit the spacecraft, excluding any landmarks that extended beyond these boxes from training.

3.3. Flight

During the flight phase, we use an Unscented Kalman Filter(UKF) for close-range navigation around a non-cooperative target. The absence of precise information about orbit, rotational inertia, and other relevant factors makes establishing an accurate dynamical model unfeasible. Therefore, it is not meaningful to discuss our Kalman filter design in depth here. However, as shown in Figure 1, we use the Kalman filter to predict the spacecraft's pose for the next moment. This prediction is used to calculate the bounding box, resulting in more precise target boxes than those predicted by the object detection network. The significance of this work lies in the following: In methods where target detection and pose estimation operate independently, the precision of target boxes is crucial for ensuring accurate pose estimation; For integrated approaches that combine target detection and pose estimation within the same feature extraction framework, challenges such as Wide-Deep-Range necessitate larger inputs to maintain resolution. Adopting this method allows for the selection of smaller networks, thereby achieving computationally efficient pose estimation.

4. Experiments

4.1. Metric

The performance of the proposed algorithm is evaluated by translation and attitude errors. The translation error is defined as follows, where t_{gt} represents the true relative distance, and t_{est} is the estimated value.

$$E_{\mathrm{t}} = \begin{cases} 0 & \text{if } E_t / \left|t_{gt}\right|_2 < 2.173\mathrm{mm/m} \\ \left|t_{\mathrm{gt}} - t_{\mathrm{est}}\right|_2 & \text{otherwise} \end{cases} \tag{9}$$

The attitude error is defined as follows, where q_{gt} represents the true relative attitude and q_{est} represents the estimated value.

$$\begin{aligned} E_{\mathrm{R}} &= 2 * \arccos\left(\left|z_{\mathrm{s}}\right|\right), \text{where} \\ z &= \left[\begin{array}{ll} z_{\mathrm{s}} & z_v \end{array}\right] = q_{\mathrm{gt}} * \operatorname{conj}\left(q_{\mathrm{est}}\right) \end{aligned} \tag{10}$$

The final pose score is defined as follows.

$$E_{\text{pose}} = E_t / \left|t_{gt}\right|_2 + E_R \tag{11}$$

4.2. Implementation details

During offline training phase, the proposed network is trained on GPU NVIDIA GTX3090 with AdamW[28] optimizer. Batchsize is set to 32. The training lasts 25 epochs with a learning rate warmup at first epoch from 0 to 0.0005. The learning rate is reduced to 0 according to cosine annealing during the rest epochs.

During Online Training phase, the proposed network is trained with the same optimizer with a learning rate of 0.0002. The batchsize is set to 1.

Table 2. The operational speed of Online Training on different devices.

Hardware	Operating time
i9-12900K@3.2/5.0(GHz)	0.25s
GTX3090@35.6(TFLOPS)	0.05s

In addition, Table 2 shows the online operating efficiency on different device. Online Training runs at about 4 it/s on the CPU, including inference, pseudo-label generation and gradient backpropagation. The running speed on GTX3090 at 35.6(TFLOPS) is about 20 it/s.

4.3. Evaluation on the SPEED+ dataset

SPEED+ is a pioneering dataset designed for vision-only spacecraft pose estimation and relative navigation, focusing on the domain gap. It consists of three different domains, synthetic, lightbox and sunlamp. The synthetic domain comprises 59,960 images of the Tango spacecraft rendered with OpenGL. The lightbox and sunlamp domains contain 6,740 and 2,791 images of a model of the same spacecraft captured in a robotic simulation environment. The satellite in the lightbox domain is illuminated by several lightboxes to approximate the diffuse light of Earth, while the same object in the sunlamp domain is exposed to an arc lamp to simulate the direct sunlight.

Table 3 shows the performance of the Online Training and Offline Training parts in sunlamp and lightbox domains, incrementally revealing the test results following the application of style transfer, sunflare, and Online Training, based on a baseline model. The result without any augmentation constitutes the baseline. The results indicates that offline training conduct certain improvements, yet the outcomes are not entirely satisfactory.

4.3.1 Loss function

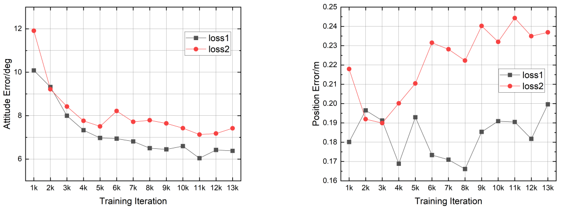
Figure 5 illustrates the performance of the loss function of Online Training mentioned in Section 3.2.3. Loss1 that minimizes the loss of the middle layer achieves better results. Obviously, the results are consistent with the previous inferences.

4.3.2 Learnable parameter

Each set of parameters is trained 10 times with a 13,000 training iteration, and the mean and variance are counted to verify the performance of the algorithm. $\mathcal{G}_{\mathrm{E}}$ and $\mathcal{G}_{\mathrm{D}}$ are defined to represent the Encoder and Decoder parts of the

Table 3. The performance of the Online Training and Offline Training parts on SPEED+.

Config.	sunlamp			lightbox		
	$E_R[°]$	$E_t[\text{m}]$	$E_{\text{pose}}[\text{-}]$	$E_R[°]$	$E_t[\text{m}]$	$E_{\text{pose}}[\text{-}]$
Our Baseline	65.72	0.74	1.27	57.71	1.07	1.17
+Style Aug.	33.36	0.35	0.64	24.17	0.31	0.48
+Sunflare	19.71	0.24	0.39	21.27	0.30	0.42
+Online Training	5.51	0.22	0.13	6.68	0.17	0.15



(a) Attitude Error

(b) Position Error

Figure 5. Comparisons of the proposed loss functions.

network, $\theta^{\text{norm}}_{\mathcal{G}_{\text{E}}}$ denotes that only LN parameters of the Encoder part are learnable. As shown in Table 4, Online Training works when only the normalization layers are learnable. Comparing $\theta^{\text{norm}}_{\mathcal{G}_{\text{E}}}$ and $\theta^{\text{norm}}_{\mathcal{G}_{\text{D}}}$, the encoder part plays a key role in domain adaptation. Furthermore, only training the normalization layers of the encoder part make Online Training more stable.

Table 4. Ablation study for learnable parameters on SPEED+

Params.	$E_R[°]$	$\delta_R[°]$	$E_t[\text{m}]$	$\delta_t[\text{m}]$
$\theta^{\text{all}}_{\mathcal{G}_{\text{all}}}$ @epoch1	127.1	N/A	6e4	N/A
$\theta^{\text{norm}}_{\mathcal{G}_{\text{E}}}$	6.36	0.153	0.180	9.28e-3
$\theta^{\text{norm}}_{\mathcal{G}_{\text{D}}}$	17.86	0.268	0.240	7.02e-3
$\theta^{\text{norm}}_{\mathcal{G}_{\text{all}}}$	6.54	0.431	0.198	13.4e-3

4.4. Evaluation on the SPARK2024 dataset

The test images of SPARK2024[8, 9] originate from the 'SnT Zero-G Lab' of the Interdisciplinary Center of Security, Reliability, and Trust (SnT) at the University of Luxembourg. SPARK2024 comprises two streams: Spacecraft Semantic Segmentation and Spacecraft Trajectory Estimation. The performance of our proposed algorithm was validated within the dataset of the SPARK 2024 Spacecraft Trajectory Estimation. The training set includes 100 groups of various trajectories, each with 300 labeled synthetic images. The test set consists of four groups of trajectory obtained from the SnT Zero-G Lab, named RT001, RT002, RT003, and RT004, containing 681, 424, 678, and 340 time-sequential images, respectively. These images cover relative distances ranging from 2m to 6m.

Table 5. The performance of the three phases on SPARK2024.

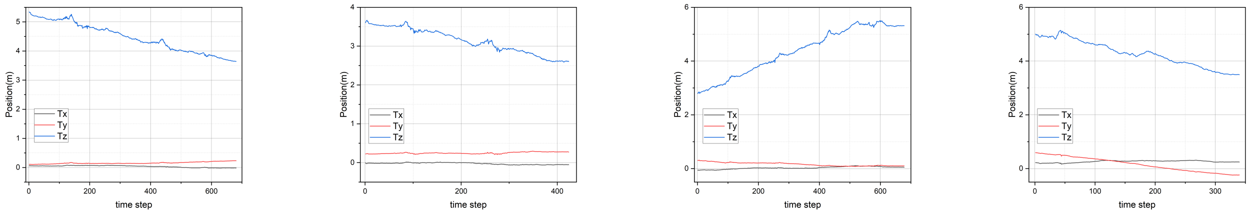
Phase	SPARK2024		
	$E_t[\text{m}]$	$E_R[\text{rad}]$	$E_{\text{pose}}[\text{-}]$
Offline Training	0.0492	0.1843	0.1971
Online Training	0.0275	0.0509	0.0574
Flight	0.0243	0.0448	0.0508

Table 5 summarizes our testing results at each phase. It is obvious that Online Training is crucial, especially for pose estimation. The improvements during Flight phase are primarily attributed to the increased accuracy of bounding boxes. Furthermore, data augmentation for bounding boxes is equally important, mentioned in Section 3.2.5. Without this augmentation, applying precise bounding boxes from UKF in Flight phase would have led to inferior pose prediction accuracy. The final leaderboard scores for SPARK2024 are presented in Table 6.

Table 6. Comparison of the results from different phases.

Config.	SPARK2024		
	$E_t[\text{m}]$	$E_R[\text{rad}]$	$E_{\text{pose}}[\text{-}]$
csu_nuaa_pang	0.0252	0.0187	0.0252
lucca(ours)	0.0243	0.0448	0.0508
juanqilai	0.0335	0.0843	0.0934
yyyyy	0.0335	0.0918	0.1009
shashasha	0.0335	0.0939	0.1030

Figure 6 shows our position prediction curve. Throughout the entire trajectory prediction process, there were no instances of sudden failure. However, we observed significant periodic jitters, which seem to be related to the satellite's attitude. Additionally, we present samples of our attitude prediction in Figure 7. We select steps 50, 100, 150, and 200 from each of the four trajectory groups for demonstration. It's also worth noting that target box prediction



(a) RT001.

(b) RT002.

(c) RT003.

(d) RT004.

Figure 6. Position prediction results with time step. Periodic jitters can be found in each curve.

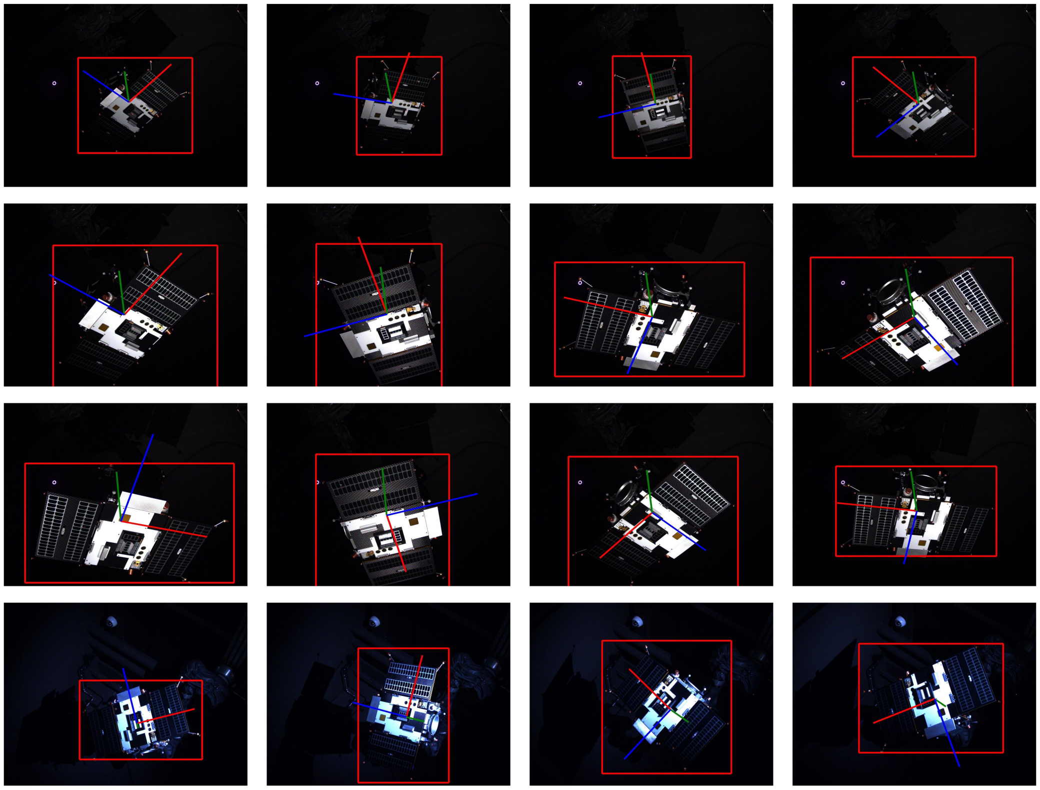


Figure 7. Visualization of pose prediction results on the SPARK2024 dataset. Each row represents a different trajectory. They are RT001, RT002, RT003 and RT004 from top to bottom.

using the Unscented Kalman Filter proves to be remarkably stable and accurate.

5. Conclusion

This paper divides the spacecraft pose estimation strategy into three phases: Offline Training, Online Training, and Flight. Offline Training employs style augmentation to improve the network's generalizability. Online Training introduces a self-supervised learning method for domain adaptation. Flight uses UKF for more accurate bounding boxes to enhance pose estimation accuracy. The algorithm's effectiveness was validated on the SPEED+ and SPARK2024 datasets. However, given that Online Training is an independent process within mission execution, it is crucial to consider fewer training iterations to shorten the time spent on this stage and achieve rapid domain adaptation.

Furthermore, each spacecraft currently requires its own dataset for pose estimation, in both offline and online scenarios. Developing more generalized algorithms for pose estimation, particularly for unseen objects, could be the most promising area of future research in this field.

References

[1] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations," *Progress in Aerospace Sciences*, vol. 93, pp. 53–72, Aug. 2017. 1

[2] 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, Oct. 2020. 1, 3

[3] T. H. Park, M. Märtens, M. Jawaid, Z. Wang, B. Chen, T.-J. Chin, D. Izzo, and S. D'Amico, "Satellite Pose Estimation Competition 2021: Results and Analyses," *Acta Astronautica*, vol. 204, pp. 640–665, Mar. 2023. 1, 2

[4] L. Pauly, W. Rharbaoui, C. Shneider, A. Rathinam, V. Gaudillière, and D. Aouada, "A survey on deep learning-based monocular spacecraft pose estimation: Current state, limitations and prospects," *Acta Astronautica*, p. S0094576523003995, Aug. 2023. 1

[5] G. Wilson and D. J. Cook, "A Survey of Unsupervised Deep Domain Adaptation," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 5, pp. 1–46, Oct. 2020. 1

[6] T. H. Park and S. D'Amico, "Online Supervised Training of Spaceborne Vision during Proximity Operations using Adaptive Kalman Filtering," Sep. 2023. 1

[7] 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)*, Mar. 2022, pp. 1–15. 1

[8] A. Rathinam, M. A. Mohamed Ali, V. Gaudilliere, and D. Aouada, "SPARK 2024: Datasets for Spacecraft Semantic Segmentation and Spacecraft Trajectory Estimation," Feb. 2024. 2, 7

[9] 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, A. Hein, M. Olivares-Mendez, and D. Aouada, "Lessons from a Space Lab: An Image Acquisition Perspective," *International Journal of Aerospace Engineering*, vol. 2023, pp. 1–16, Sep. 2023. 2, 7

[10] 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, Dec. 2020. 2

[11] P. F. Proenca and Y. Gao, "Deep Learning for Spacecraft Pose Estimation from Photorealistic Rendering," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, May 2020, pp. 6007–6013. 2

[12] T. H. Park, S. Sharma, and S. D'Amico, "Towards Robust Learning-Based Pose Estimation of Noncooperative Spacecraft," Sep. 2019. 2

[13] Y. Hu, S. Speierer, W. Jakob, P. Fua, and M. Salzmann, "Wide-Depth-Range 6D Object Pose Estimation in Space," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, Jun. 2021, pp. 15 865–15 874. 2

[14] B. Chen, J. Cao, A. Parra, and T.-J. Chin, "Satellite Pose Estimation with Deep Landmark Regression and Nonlinear Pose Refinement," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 2816–2824. 2

[15] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, Feb. 2009. 2

[16] 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*, pp. 1–14, 2023. 2

[17] 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*, p. S0273117723002284, Mar. 2023. 2

[18] ——, "Adaptive Neural Network-based Unscented Kalman Filter for Spacecraft Pose Tracking at Rendezvous," Jun. 2022. 3

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc., 2015. 3

[20] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9992–10 002. 3

[21] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation," in *Computer Vision – ECCV 2022 Workshops*, ser. Lecture Notes in Computer Science, L. Karlinsky, T. Michaeli, and K. Nishino, Eds. Cham: Springer Nature Switzerland, 2023, pp. 205–218. 3

[22] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun, "Rethinking on Multi-Stage Networks for Human Pose Estimation," May 2019. 3

[23] B. Chen, T.-J. Chin, and M. Klimavicius, "Occlusion-Robust Object Pose Estimation With Holistic Representation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2929–2939. 3

[24] B. Fuglede and F. Topsoe, *Jensen-Shannon Divergence and Hilbert Space Embedding*, Jan. 2004. 3

[25] P. T. Jackson, A. Atapour-Abarghouei, S. Bonner, T. Breckon, and B. Obara, "Style Augmentation: Data Augmentation via Style Randomization," Apr. 2019. 3

[26] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," *Information*, vol. 11, no. 2, p. 125, Feb. 2020. 3

[27] A. Nibali, Z. He, S. Morgan, and L. Prendergast, "Numerical Coordinate Regression with Convolutional Neural Networks," May 2018. 3

[28] I. Loshchilov and F. Hutter, “Fixing Weight Decay Regularization in Adam,” Feb. 2018. 6