This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

FastPose-ViT: A Vision Transformer for Real-Time Spacecraft Pose Estimation

Pierre Ancey[1] Andrew Price[1] Saqib Javed[1] Mathieu Salzmann[1,2]

[1]EPFL [2]Swiss Data Science Center
Lausanne, Switzerland

{firstname.lastname}@epfl.ch

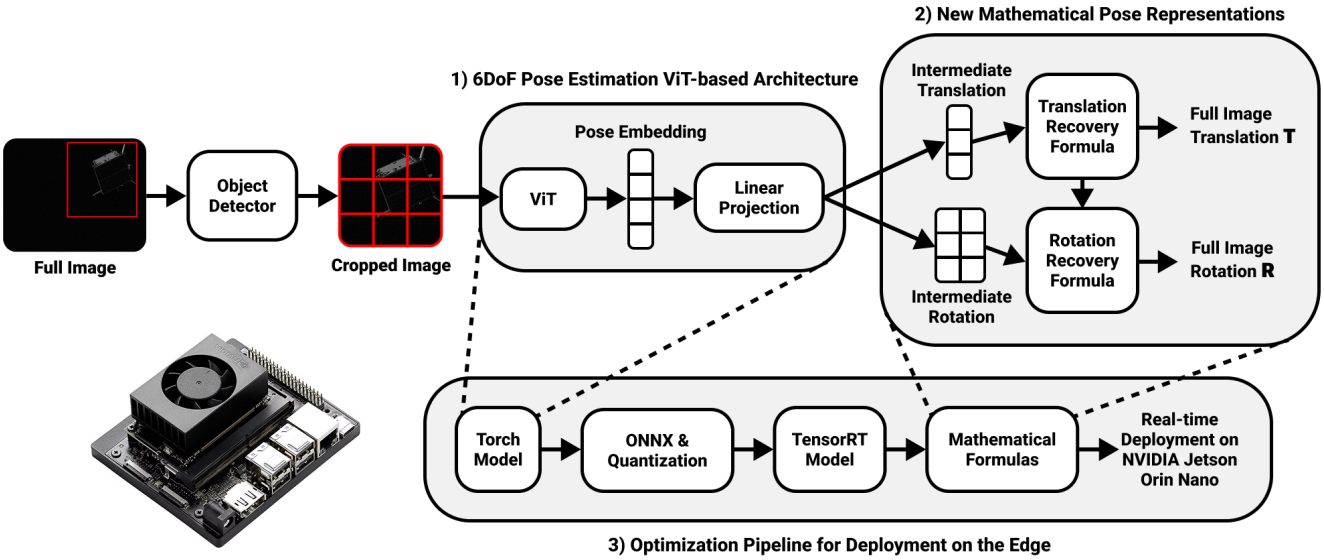


Figure 1. **Overview of the FastPose-ViT pipeline.** A bounding-box detector first extracts the spacecraft from the full image, producing a cropped input for the pose network. Then, (1) a ViT-based architecture regresses intermediate pose parameters from the crop; (2) geometric recovery formulas convert these into the full-frame translation and rotation; (3) the trained model is exported, quantized, and optimized for real-time deployment on a NVIDIA Jetson Orin Nano.

Abstract

Estimating the 6-degrees-of-freedom (6DoF) pose of a spacecraft from a single image is critical for autonomous operations like in-orbit servicing and space debris removal. Existing state-of-the-art methods often rely on iterative Perspective-n-Point (PnP)-based algorithms, which are computationally intensive and ill-suited for real-time deployment on resource-constrained edge devices. To overcome these limitations, we propose FastPose-ViT, a Vision Transformer (ViT)-based architecture that directly regresses the 6DoF pose. Our approach processes cropped images from object bounding boxes and introduces a novel mathematical formalism to map these localized predictions back to the full-image scale. This formalism is derived from the principles of projective geometry and the concept of "apparent rotation", where the model predicts an apparent rotation matrix that is then corrected to find the true orientation. We demonstrate that our method outperforms other non-PnP strategies and achieves performance competitive with state-of-the-art PnP-based techniques on the SPEED dataset. Furthermore, we validate our model's suitability for real-world space missions by quantizing it and deploying it on power-constrained edge hardware. On the NVIDIA Jetson Orin Nano, our end-to-end pipeline achieves a latency of 75 ms per frame under sequential execution, and a non-blocking throughput of up to 33 FPS when stages are scheduled concurrently.

1. Introduction

The increasing complexity of space missions has created a pressing need for autonomous systems capable of performing tasks such as in-orbit servicing, active debris removal, and asteroid mining. A fundamental requirement for these operations is the precise and rapid estimation of a spacecraft's 6-degrees-of-freedom (6DoF) pose –its translation and attitude– from a single monocular image.

While state-of-the-art approaches have achieved high accuracy using hybrid pipelines that combine keypoint detection with iterative Perspective-n-Point (PnP) solvers [10, 13, 21, 25, 35], these methods are inherently slow and poorly suited to real-time deployment on edge computing platforms due to their iterative nature. Direct regression, in which a neural network predicts 6DoF pose in a single feed-forward pass [22, 29], offers a promising alternative due to its computational efficiency. However, existing direct regression methods have typically underperformed in accuracy relative to their hybrid counterparts [16, 26].

To address this limitation, we propose a new architecture based on the Vision Transformer [5] (ViT) that aims to bridge the gap between speed and accuracy in spacecraft pose estimation. Our primary contributions are:

- A ViT [5]-based architecture designed specifically for efficient and direct 6DoF pose regression.
- A reformulation of the regression targets based on geometric principles. Our model predicts an apparent rotation; we then apply a closed-form correction, derived from projective geometry, to recover the true rotation.
- A pipeline that exports the model from PyTorch to TensorRT, enabling FP16 quantization without accuracy loss and real-time inference on edge devices.

We evaluate our method on the SPEED [32] and SPEED+ [24] datasets, where it sets a new state-of-the-art for non-PnP-based methods. Crucially, our approach achieves performance that is competitive with top-tier PnP-based methods on SPEED while also operating in real-time on a small edge device, demonstrating that direct regression is a capable alternative for real-world spacecraft applications.

2. Related Work

Spacecraft Pose Estimation. Spacecraft pose estimation involves unique challenges such as high specularity, extreme dynamic ranges, and non-diffuse lighting. This has been presented in different settings including SwissCube [12], URSO [29], SPARK [30], Synthetic-Minerva [28] and PocketQube [31]. However, the de facto benchmarks for monocular spacecraft 6DoF pose estimation are the European Space Agency's (ESA) Satellite Pose Estimation Challenges [6, 7], which introduced the SPEED and SPEED+ datasets now commonly used in the space community. Against these benchmarks, many 6DoF pose estimation strategies have been tested, and can broadly be categorized as hybrid or direct [26].

Hybrid methods combine deep learning with geometric solvers [2, 10, 12]. Top performers on leaderboards [6, 7], such as UniAdelaide [2] and CVLab [12], detect 2D keypoints on the object and then use a PnP solver with RANSAC [9] to compute the pose. Li et al. [17], predict keypoints along with uncertainty values to guide the PnP algorithm, improving robustness against outliers. While accurate, these methods are computationally intensive. For example, one PnP-based approach [35] reported **150 ms per image (∼7 FPS)** on a high-performance NVIDIA RTX 2080Ti GPU (peak draw 250 Watts) and an i7-7700K CPU, far exceeding the typical power budgets of a spacecraft (∼1–25 Watts). This high latency stems from the multi-stage pipelines and iterative solvers these methods rely on.

Direct methods bypass intermediate steps by regressing pose parameters directly from the image. URSONet [29] uses a CNN to regress translation and classify orientation into discrete bins. E-PoseNet [22] introduces equivariant CNNs to improve robustness to geometric transformations in a lightweight architecture. However, these direct methods still fall short of the accuracy achieved by hybrid approaches. To bridge this gap while preserving efficiency, we extend the direct regression paradigm by introducing geometrically derived regression targets and replacing conventional CNNs with a ViT-based architecture.

Vision Transformers for Pose Estimation. The application of Vision Transformers (ViTs) to direct 6DoF pose estimation has been explored in PViT-6D [34], which employs a Multiscale ViT [8] and was evaluated on the BOP benchmark [11] for terrestrial objects. While conceptually interesting, PViT-6D suffers from practical limitations: the architecture is complex, pretrained weights are unavailable, and the implementation is difficult to reproduce. In contrast, FoundationPose [37] demonstrates that ViTs can achieve high precision, but its multi-stage pipeline is not designed for deployment on edge devices. These limitations motivated us to design a custom ViT-based architecture that is both lightweight and reproducible, providing a practical solution for direct pose estimation in space applications.

Rotation Representations. The representation of rotation is a critical aspect of 6DoF pose estimation. Common representations such as quaternions and Euler angles suffer from ambiguity and discontinuity issues, respectively. To address this, Zhou et al. [38] proposed continuous 6D representations to create a more stable learning target, showing improved performance. While our work also leverages a continuous 6D representation for rotation matrices, our key innovation lies in reformulating the regression target itself.

3. Methodology

Our goal is to predict the translation vector $T \in \mathbb{R}^3$ and rotation matrix $R \in SO(3)$ of a spacecraft relative to the camera frame from a single RGB image. Our method, FastPose-ViT, is built upon a ViT backbone and leverages bounding box priors to improve efficiency and accuracy.

3.1. Vision Transformer Architecture

The standard ViT architecture [5] was designed for image classification on large datasets like ImageNet [4]. It splits an image into non-overlapping patches, linearly projects the patches into embeddings called "tokens," and prepends a learnable [CLS] token to aggregate global information via self-attention for the final classification task.

While our goal is 6DoF regression rather than classification, we can adapt this architecture through transfer learning, leveraging the strong spatial representations acquired during pre-training. For our pose regression task, we modify the architecture as follows:

1. The input image is first cropped using a bounding box obtained from an object detector, focusing the ViT's attention on relevant patches.
2. The original classification Multi-Layer Perceptron (MLP) is discarded.
3. The final '[CLS]' token embedding is passed through a single linear projection layer to predict translation and rotation parameters.

We also experimented with deeper MLPs as regression heads, but found no consistent improvement over the simple linear projection. This suggests that the ViT backbone already encodes sufficient global information alone.

To find the best trade-off between accuracy and computational cost, We evaluate ViT-B backbones, where ViT-B refers to the Base variant of the Vision Transformer architecture introduced by Dosovitskiy et al. [5], at two input resolutions; the models are summarized in Table 1. Additional experiments with slower and less performative ViT backbones are shared in the supplementary material.

3.2. Data Augmentations

To improve model robustness and prevent overfitting, we employ extensive data augmentation strategies at both the pixel and spatial levels.

Backend	Params (M)	GFLOPS	Image Size	Patch
ViT-B-224/16	85.8	17.5	224x224	16x16
ViT-B-384/16	86.1	55.5	384x384	16x16

Table 1. Characteristics of the ViT-B/16 backbones with linear projection layer evaluated in this work.

Pixel-Level Augmentations. To improve robustness to the challenging imaging conditions encountered in orbit, we apply a diverse set of pixel-level transformations using the Albumentations library [1]. These include variations in illumination, lens and optical effects, sensor noise, motion blur, and compression artifacts, each applied stochastically with probability 0.5. All images are then normalized to ImageNet statistics and converted to tensors. This pipeline replicates the lighting, optical, and sensor degradations typical of spacecraft imagery, encouraging the model to learn invariant representations.

Bounding-Box Augmentations. To avoid overfitting to perfectly framed crops, which is unrealistic in practice, we deliberately perturb the bounding boxes used to generate input crops. This forces the model to rely on actual pose cues rather than memorizing canonical crop sizes and positions.

To obtain the initial bounding boxes, we hand-annotated all images in the SPEED training set and trained an LW-DETR [3] detector on these annotations, modifying its output head to predict a single spacecraft class instead of COCO's 80 classes [18]. We then applied this detector to produce bounding boxes for the SPEED evaluation set and for all splits of SPEED+. These automatically generated boxes serve as the baseline from which augmentations are applied.

During training of the pose network, each box $B = (x_{\min}, y_{\min}, x_{\max}, y_{\max})$ is randomly expanded or contracted along each side by up to 10% of its width or height. The resulting box is clipped to image boundaries and discarded if it becomes invalid (e.g., degenerate or too small). This strategy increases robustness to the noisy detections commonly encountered in real settings.

Spatial-Level Augmentations. We apply in-plane rotations to the images to augment the dataset with varied viewpoints. A key challenge is that rotating the image also changes the ground-truth pose. For an in-plane rotation of angle θ around the image center, the new rotation matrix R_{new} and translation vector T_{new} can be computed from the original pose (R, T) and the intrinsic matrix K as:

$$R_{new} = K^{-1}M(\theta)KR, \quad T_{new} = K^{-1}M(\theta)KT, \quad (1)$$

where $M(\theta)$ is the 2D rotation matrix in homogeneous coordinates. The term $K^{-1}M(\theta)K$ simplifies to a pure 3D rotation matrix $R_z(\theta)$ around the camera's Z-axis (the axis orthogonal to the image plane). For completeness, the spatial augmentation derivation is shared in the supplementary material. The pose is updated as

$$R_{new} = R_z(\theta)R, \quad T_{new} = R_z(\theta)T. \quad (2)$$

This process allows for the accurate relabeling of our spatially augmented data. During training, a spatial augmentation is applied with a 50% probability, where one of two rotation ranges is selected with equal likelihood: Either a minor rotation within $[-20°, 20°]$ or a major rotation within $[160°, 200°]$. These specific ranges were chosen because they guarantee that, for the extreme majority of SPEED and SPEED+ samples, at least part of the satellite remains visible in the augmented crop, ensuring that a valid bounding box can still be defined.

3.3. Geometric Reformulation of Pose Regression for Cropped Images

A key innovation of our work is reformulating the pose regression targets to align with the localized visual information available in a cropped image. While using crops is highly efficient for ViTs, it presents a challenge; the full image absolute pose must be recovered from the cropped image apparent pose.

Our solution is to train the network to predict a set of intermediate, normalized parameters that are directly observable in the crop, rather than the absolute pose itself. We then use a geometric post-processing step to deterministically recover the full-frame pose from these intermediate values. We address the translation and rotation components separately.

3.3.1. Translation via Normalized Coordinates

Step 1: Formulating a Normalized Depth Target (U_z). The act of cropping and resizing an image patch is analogous to a digital zoom. We relate the object's true depth Z to a normalized "crop-depth proxy", U_z, using the scaling factors s_x and s_y derived from the crop and full-image dimensions as

$$s_x := \frac{W}{\alpha\, w}, \qquad s_y := \frac{H}{h}, \tag{3}$$

where (W, H) and (w, h) are the full-image and crop dimensions, respectively, and α is a dataset-specific aspect-ratio factor (1.6 for SPEED/SPEED+) which corrects the non-square aspect ratio of the full image. Since the zoom can be estimated along either the x or y-axes, we define U_z as the least-squares solution satisfying $U_z \approx Z/s_x$ and $U_z \approx Z/s_y$, which yields

$$U_z = \frac{1}{2}\left(\frac{1}{s_x} + \frac{1}{s_y}\right) Z. \tag{4}$$

This gives us our first regression target, U_z. During inference, we can invert this equation to recover the absolute depth Z.

Step 2: Formulating Normalized Lateral Targets (U_x, U_y). Next, we define the targets for the lateral translation components (X, Y). We start with the standard pinhole camera model, which maps a 3D point (X, Y, Z) to its 2D pixel projection (x, y) as

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \qquad K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

A key issue is that the satellite's projected 3D center on the image (x, y) does not necessarily coincide with the center of its bounding box (b_x, b_y). To account for this, we define our regression targets based on the offsets $(\Delta x, \Delta y) = (x - b_x, y - b_y)$. To make these targets scale-invariant and independent of the crop size, we normalize them by the crop width w and height h. This yields

$$\begin{cases} U_x := \dfrac{\Delta x}{w} = \dfrac{1}{w}\left(\dfrac{f_x}{Z} X + c_x - b_x\right), \\ U_y := \dfrac{\Delta y}{h} = \dfrac{1}{h}\left(\dfrac{f_y}{Z} Y + c_y - b_y\right). \end{cases} \tag{6}$$

The network is thus trained to regress the normalized translation vector $U = (U_x, U_y, U_z)$. At inference, once we predict U and recover Z from Eq. (4), the lateral components (X, Y) can be directly computed by inverting these relations, i.e.

$$\begin{cases} X = \dfrac{1}{f_x}\Big(b_x + U_x w - c_x\Big) Z, \\ Y = \dfrac{1}{f_y}\Big(b_y + U_y h - c_y\Big) Z. \end{cases} \tag{7}$$

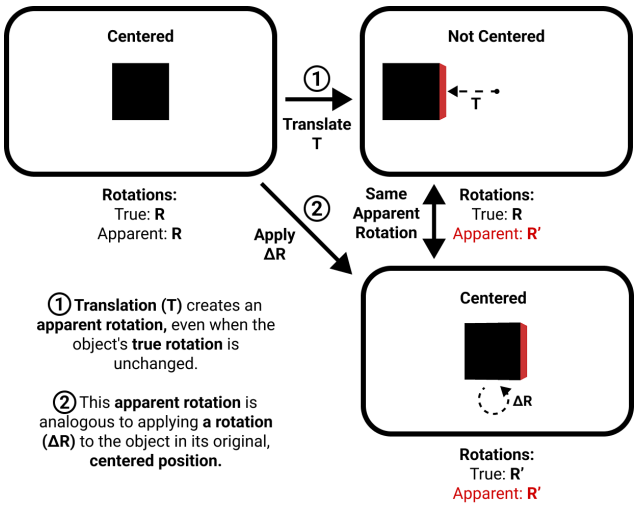


Figure 2. **Apparent Rotation (R').** An object's perceived orientation changes with its position in the image due to the camera perspective. The apparent rotation R' is the true orientation a *centered* object must have to best represent the off-center object.

3.3.2. Rotation Target: True vs. Apparent Orientation

A challenge in pose regression is that perspective projection distorts an object's perceived orientation. As an object moves away from the camera's principal axis, it appears rotated, even if its true orientation R remains constant.

To address this, we introduce the notion of *apparent orientation* R'. Instead of forcing the network to learn the translation-dependent true rotation, we task the network with predicting the directly observable apparent orientation. As presented in Figure 2, we define R' as the orientation a *centered* object would need to have to visually match the appearance of the translated object. Formally, we seek a corrective rotation ΔR that maps the true orientation to the apparent one,

$$R' = \Delta R\, R. \tag{8}$$

Given the camera's optical axis $e_z = (0, 0, 1)^\top$ and the estimated object translation T we obtain ΔR by computing the rotation axis u and rotation angle θ as

$$\vec{\boldsymbol{T}} = \frac{T}{\|T\|}, \qquad u = \frac{\vec{\boldsymbol{T}} \times e_z}{\|\vec{\boldsymbol{T}} \times e_z\|}, \qquad \theta = \arccos(\vec{\boldsymbol{T}}^\top e_z).$$

The corrective rotation ΔR is then obtained via Rodrigues' rotation formula as

$$\Delta R = I + \sin\theta [u]_\times + (1 - \cos\theta)[u]_\times^2, \tag{9}$$

where $[u]_\times$ is the skew-symmetric matrix associated with the vector u. We then recover the true orientation by inverting the corrective rotation as

$$R_{\text{pred}} = \Delta R^\top R'_{\text{pred}}. \tag{10}$$

The network is supervised to predict only the first two columns of the apparent rotation matrix, $\hat{r}_1, \hat{r}_2 \in \mathbb{R}^3$. We construct an orthonormal basis via the Gram–Schmidt procedure, which yields

$$r_1 = \frac{\hat{r}_1}{\|\hat{r}_1\|}, \qquad r_2 = \frac{\hat{r}_2 - (r_1^\top \hat{r}_2) r_1}{\|\hat{r}_2 - (r_1^\top \hat{r}_2) r_1\|}, \qquad r_3 = r_1 \times r_2. \tag{11}$$

The predicted apparent rotation is simply assembled as

$$R'_{\text{pred}} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}. \tag{12}$$

This formulation decouples the perspective effects from the rotation prediction task, allowing the network to learn a simpler mapping from image appearance to orientation.

4. Experiments and Results

4.1. Experimental Setup

Data. We train and evaluate our models on the SPEED and SPEED+ datasets. Models trained on SPEED are evaluated on SPEED, and models trained on SPEED+ are evaluated on SPEED+, treating each dataset independently.

Evaluation Metrics. We use the official metrics from the ESA Pose Estimation Challenge. The translation error, E_T, is the Euclidean distance between the predicted and ground-truth translation vectors (T_{pred} and T_{gt}), i.e.,

$$E_T = ||T_{pred} - T_{gt}||_2. \tag{13}$$

The rotation error, E_R, is the geodesic distance between the predicted and ground-truth orientations. For this metric, the rotations are represented as quaternions (q), and the error is calculated as

$$E_R = 2 \cdot \arccos(|\langle q_{pred}, q_{gt} \rangle|). \tag{14}$$

Implementation Details. We initialize all ViT backbones with weights from the Facebook SWAG paper [33] pretrained on ImageNet-1K.

For optimization, we employ the Muon optimizer [15], using its default parameters but without weight decay , as we observed that enabling weight decay consistently degraded performance in our pose-regression setting. While Muon has been primarily studied in the context of large language models [19], we find that it consistently outperforms AdamW in this ViT-based pose regression setting. To further improve convergence, we adopt a cosine annealing learning rate scheduler with warm restarts [20], performing 5 restart periods with $T_0 = 10$ epochs and $T_{mult} = 2$, for a total of 310 epochs on SPEED and 4 restarts periods for a total of 150 epochs on SPEED+. It is worth noting that our extensive data augmentations help ensure the network does not overfit. The learning rate is cycled between $1 \cdot 10^{-4}$ and $1 \cdot 10^{-6}$, which helps the model escape sharp local minima. We use a batch size of 8.

For training the LW-DETR-based object detector, we use a cosine annealing learning rate scheduler with warm restarts, cycling between $5 \cdot 10^{-4}$ and $5 \cdot 10^{-5}$. We set the batch size to 16 and initialize the model with pretrained LW-DETR weights from the COCO dataset.

Training Losses. Our model is trained with two relative losses, one for rotation and one for translation, both designed to be invariant to scale and perspective.

The rotation loss is defined as a relative Frobenius error, i.e.,

$$\mathcal{L}_{\text{rot}} = \left\| R'_{\text{pred},1:2} - R'_{\text{gt},1:2} \right\|_F^2, \tag{15}$$

where $R'_{1:2}$ denotes the first two columns of the apparent rotation matrix.

For translation, we supervise the normalized crop-space targets $U = (U_x, U_y, U_z)$ defined in Eq. (6)–(7). The relative translation loss is given by

$$\mathcal{L}_{\text{trans}} = \|U_{\text{pred}} - U_{\text{gt}}\|_2^2. \tag{16}$$

The total loss is the simple sum of these two terms,

$$\mathcal{L} = \mathcal{L}_{\text{rot}} + \mathcal{L}_{\text{trans}}. \tag{17}$$

These formulations yield stable gradients and significantly faster convergence compared to direct L_2 supervision on absolute pose parameters (R, T).

4.2. Comparison with the State of the Art

We compare our models against prior work on SPEED and SPEED+. On SPEED, our direct regression approach clearly surpasses all other non-PnP methods and matches state-of-the-art PnP pipelines, while maintaining real-time inference on edge hardware (Table 2), indicating that PnP solvers are not essential when training and testing domains align.

On SPEED+ (Tables 3, 4), the training data is synthetic as in SPEED, but the test data consists of real laboratory images with a large domain gap (see Fig. 4). In this setting, our method lags behind hybrid pipelines. We attribute this to two key factors:

(1) Leading competition teams aggressively reduced the domain gap using heavy pixel-level and style-transfer augmentations, in some cases tuned with prior knowledge of the hidden test domains. This strategy was later criticized by the organizers [25], as it compromises competition fairness and limits reproducibility.

(2) Several teams reconstructed the unreleased 3D spacecraft model to synthesize large custom training datasets. While effective, this approach addresses a specific benchmark rather than the broader problem of generalizable spacecraft pose estimation.

In contrast, our pipeline relies solely on released datasets and generic augmentations, without exploiting hidden domain cues or private 3D assets. This makes our approach simple, reproducible, and generally applicable to realistic deployment scenarios, though it highlights the open challenge of achieving domain-robust direct regression.

Table 2. Comparison with state-of-the-art methods on SPEED.

Model	PnP?	E_T [m] ↓	E_R [deg] ↓
Chen et al. [2]	Yes	0.0320	**0.4100**
ViT-B-384/16 (Ours)	No	**0.0221**	0.4183
ViT-B-224/16 (Ours)	No	0.0231	0.4469
Gerard et al. [10]	Yes	0.0730	0.9100
E-PoseNet [22]	No	0.1806	2.3073
URSONet [29]	No	0.1450	2.4900

4.3. Ablation Studies and Analysis

We conduct a series of comprehensive ablation studies on the SPEED dataset to validate our key design choices. Unless otherwise specified, all experiments use our final proposed configuration on the 'ViT-B-224/16' model to ensure clear and concise comparisons.

Table 3. Comparison with state-of-the-art methods on the **SPEED+ Lightbox** subset. "?" indicates unreleased information.

Method	PnP?	E_T [m] ↓	E_R [deg] ↓
TangoUnchained	Yes	0.105	3.187
VPU [27]	Yes	0.131	4.577
SPNv2 [23]	Yes	0.150	5.577
lava1302 [36]	Yes	0.302	6.665
haoranhuang_njust	Yes	0.180	8.131
u3s_lab	Yes	0.333	9.694
ViT-B-384/16 (Ours)	No	0.270	12.261
ViT-B-224/16 (Ours)	No	0.305	14.668
chushunhao	?	0.205	16.378
for graduate	?	0.402	23.666
Pivot SDA AI & al.	?	0.409	23.918
bbnc	?	0.687	24.889
ItTakesTwoToTango	?	0.485	31.094

Table 4. Comparison with state-of-the-art methods on the **SPEED+ Sunlamp** subset. "?" indicates unreleased information.

Method	PnP?	E_T [m] ↓	E_R [deg] ↓
lava1302 [36]	Yes	0.065	2.728
VPU [27]	Yes	0.076	2.828
TangoUnchained	Yes	0.086	4.299
u3s_lab	Yes	0.181	6.241
haoranhuang_njust	Yes	0.163	8.406
SPNv2 [23]	Yes	0.161	9.788
ViT-B-384/16 (Ours)	No	0.400	18.277
bbnc	?	0.542	21.955
for graduate	?	0.455	22.970
ViT-B-224/16 (Ours)	No	0.460	28.017
Pivot SDA AI & al.	?	0.854	36.445
ItTakesTwoToTango	?	0.473	39.660
chushunhao	?	0.338	43.356

4.3.1. Impact of Transfer Learning

As shown in Table 5, training backbones from scratch results in an order-of-magnitude drop in performance across all metrics, highlighting the importance of pre-training. This is especially true for attention-based architectures, which require large-scale supervision to learn spatial priors. Although the source task is image classification rather than pose estimation, the pre-trained weights provide a strong spatial understanding that fine-tuning can adapt to the downstream task. Without such initialization, the ViT struggles to organize attention patterns in data-limited scenarios like spacecraft pose estimation.

4.3.2. Impact of Geometric Formulation and Cropping

We study the effects of cropped inputs and our apparent rotation formulation. Vision Transformers operate at a fixed

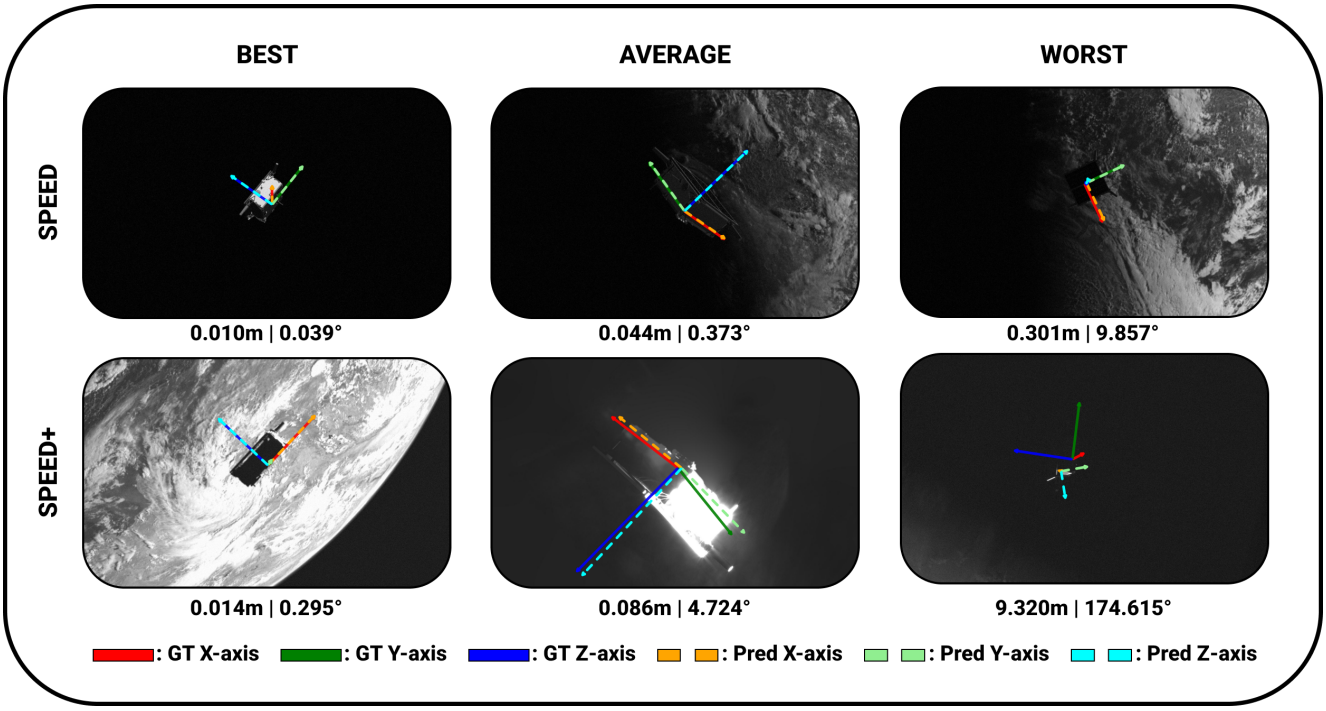


Figure 3. **Qualitative Results on SPEED and SPEED+.** We visualize our best model's performance on the test sets, showing examples of the model's best, average, and worst predictions. Ground-truth (GT) axes are in solid primary colors, while predicted (Pred) axes are shown as lighter, dashed lines. Errors below each image are reported as Translation Error [m] | Rotation Error [deg].



Figure 4. **Impact of bounding box quality on pose estimation.** Left: a poor bounding box prediction (red) compared to the ground truth (green) leading to erroneous pose regression. Right: a typical bounding box prediction (red) closely aligned with the ground truth (green) resulting in stable pose estimates.

Table 5. Effect of pre-training on performance.

Training	E_T [m] ↓	E_R [deg] ↓
From Scratch	0.2391	5.1280
Pre-trained	**0.0231**	**0.4469**

input size (224×224), so directly resizing the original SPEED/SPEED+ images (1920×1200) implies a 5–10× downscaling, causing severe loss of fine details. Cropping around the satellite alleviates this issue by ensuring that patches capture informative object regions rather than mostly background.

Within this cropped setting, our apparent rotation formulation further compensates for perspective distortions introduced when the satellite is off-center. As shown in Table 6, both components yield cumulative improvements, with their combination achieving the best performance.

Table 6. Ablation of our apparent rotation formulation on both full and cropped image inputs.

Image size	Rotation	E_T [m] ↓	E_R [deg] ↓
Full	Absolute	0.2938	7.2880
Full	Relative	0.2906	7.3396
Cropped	Absolute	0.0337	2.3778
Cropped	**Relative**	**0.0231**	**0.4469**

4.3.3. Impact of Data Augmentations

We analyze the effect of our augmentation strategy in two stages. As shown in Table 7, spatial augmentations are crucial for SPEED, while pixel augmentations provide little benefit and can even be detrimental. This can be explained by the fact that both the training and test images are primarily synthetic, so pixel-level distortions do not reflect the target distribution and instead hinder learning. In contrast, bounding box perturbations (Table 8) consistently improve robustness, with a perturbation scale of 10% yielding the best generalization.

On the other hand, for SPEED+, pixel augmentations play an important role due to the domain gap between syn-

Table 7. Impact of different data augmentation on SPEED.

Augmentation Strategy	E_T [m] ↓	E_R [deg] ↓
No Augmentations	0.0616	1.2032
Pixel Augs Only	0.0987	1.4782
Spatial Augs Only	**0.0231**	**0.4469**
All Augs	0.0469	1.0371

Table 8. Impact of bounding box augmentations.

BBox Perturbation (%)	E_T [m] ↓	E_R [deg] ↓
0%	0.0317	0.5787
10%	**0.0231**	**0.4469**
20%	0.0242	0.4584
30%	0.0255	0.4870

Table 9. Impact of data augmentation strategies on SPEED+ averaged over lightbox and sunlamp.

Augmentation Strategy	E_T [m] ↓	E_R [deg] ↓
No Augmentations	1.5054	59.7137
Pixel Augs	0.4027	20.8900
Pixel Augs + Style	**0.3506**	**18.5638**

thetic training data and real evaluation images (Table 9). In fact, further enriching the pixel space with style-based augmentations has been shown to be highly beneficial, as demonstrated by state-of-the-art methods on SPEED+ that leverage style transfer techniques [14].

4.4. Model Compression and Deployment

We deploy our models on a NVIDIA Jetson Orin Nano configured in 25W mode, running CUDA 12.6, TensorRT 10.7, and Python 3.10. Our focus is on the *end-to-end pipeline* for spacecraft pose estimation rather than isolated inference speed. We assume that an input image is already available in memory (e.g., from an onboard camera), and process it through the following stages:

1. **Pre-processing:** Resizing with bilinear interpolation and normalization using ImageNet statistics;
2. **Object detection:** Running a lightweight detector to predict a single bounding box for the spacecraft;
3. **Cropping:** Cropping the detected region, resizing it to the network input resolution, and re-normalizing;
4. **Pose estimation:** Running the ViT-based pose regressor on the cropped input;
5. **Post-processing:** Recovering the absolute translation and rotation from the intermediate outputs.

We measure the latency of each stage on the Jetson Orin Nano for different backbones (ViT-B/16 at 224×224 and 384×384 resolution) and for different inference modes (unquantized FP32 vs. FP16 TensorRT engines). The ViT and overhead latencies are shared in Tables 10 and 11.

Table 10. Pose inference latency (ms) for different ViT backbones and precisions on the NVIDIA Jetson Orin Nano. Shared pre/post-processing overhead is 57.2 ± 2.8 ms (Table 11).

Model @ Precision	ViT Only [ms]	End-to-End [ms]
ViT-B-224/16 @ FP32	17.0 ± 4.2	74.2 ± 5.0
ViT-B-224/16 @ FP16	19.9 ± 3.6	77.1 ± 4.6
ViT-B-384/16 @ FP32	49.2 ± 7.5	106.4 ± 8.0
ViT-B-384/16 @ FP16	40.7 ± 5.9	97.9 ± 6.5

Table 11. Best averaged latency (ms) of processing steps independent of the ViT backbone. Values are mean ± std.

Pre-proc	Detector	Bridge	Post-processing
13.23 ± 0.31	30.5 ± 2.4	3.7 ± 1.3	9.75 ± 0.49

On the Jetson Orin Nano, the end-to-end latency for ViT-B-224/16 is about 75 ms per image (∼13 FPS) under strictly sequential execution. With non-blocking scheduling, different stages run concurrently (e.g., the detector processes image $i+1$ while the pose network handles image i), so throughput is limited by the slowest stage; this yields up to ∼33 FPS with batch size $= 1$. A key factor is detector quantization: its runtime drops from 71.5 ± 3.1 ms in FP32 to 30.5 ± 2.4 ms in FP16, a more than $2\times$ speedup with negligible accuracy loss.

For the ViT backbones, FP16 shows mixed results. ViT-B-224/16 runs slightly faster in FP32 (17.0 ± 4.2 ms) than FP16 (19.9 ± 3.6 ms), as the conversion overheads outweigh the GPU acceleration for this small model. Larger backbones such as ViT-B-384/16 do benefit, improving from 49.2 ± 7.5 ms (FP32) to 40.7 ± 5.9 ms (FP16). Beyond latency, FP16 halves model size and improves hardware efficiency, making it preferable in most cases, except for ViT-B-224/16, where the overhead dominates.

5. Conclusion

We have presented FastPose-ViT, a Vision Transformer–based architecture for direct 6DoF spacecraft pose estimation. By reformulating pose regression with the concept of apparent rotation, we achieve state-of-the-art performance among non-PnP methods while maintaining a lightweight and reproducible design. A key contribution is demonstrating, for the first time, real-time ViT-based pose estimation on embedded edge hardware (up to 33 FPS on a Jetson Orin Nano), meeting the stringent requirements of spaceborne deployment. Our ablations validate the importance of pre-training, cropping, and geometric targets in this setting. While challenges remain under large domain gaps and bounding-box sensitivity, training on datasets that better reflect real-world spacecraft imagery could significantly reduce this gap and improve generalization.

References

[1] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020.

[2] Bo Chen, Jiewei Cao, Alvaro Parra, and Tat-Jun Chin. Satellite pose estimation with deep landmark regression and nonlinear pose refinement. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019.

[3] Qiang Chen, Xiangbo Su, Xinyu Zhang, Jian Wang, Jiahui Chen, Yunpeng Shen, Chuchu Han, Ziliang Chen, Weixiang Xu, Fanrong Li, et al. Lw-detr: A transformer replacement to yolo for real-time detection, 2024.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: transformers for image recognition at scale, 2020.

[6] European Space Agency (ESA). Satellite pose estimation challenge: Post-mortem leaderboard. `https://kelvins.esa.int/satellite-pose-estimation-challenge/leaderboard/leaderboard`, 2019. Accessed: December 5, 2025.

[7] European Space Agency (ESA). Pose estimation 2021: Sunlamp post-mortem leaderboard. `https://kelvins.esa.int/pose-estimation-2021/leaderboard/sunlamp-final-result`, 2021. Accessed: December 5, 2025.

[8] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021.

[9] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[10] Kyle Gerard, Mathieu Salzmann, and Yinlin Hu. Segmentation-driven satellite pose estimation. `https://indico.esa.int/event/319/attachments/3561/4754/pose_gerard_segmentation.pdf`, 2019. Accessed: December 5, 2025.

[11] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[12] Yinlin Hu, Sebastien Speierer, Wenzel Jakob, Pascal Fua, and Mathieu Salzmann. Wide-depth-range 6d object pose estimation in space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15870–15879, 2021.

[13] Yurong Huo, Zhi Li, and Feng Zhang. Fast and accurate spacecraft pose estimation from single shot space imagery using box reliability and keypoints existence judgments. *IEEE Access*, 8:216283–216297, 2020.

[14] Philip TG Jackson, Amir Atapour Abarghouei, Stephen Bonner, Toby P Breckon, and Boguslaw Obara. Style augmentation: data augmentation via style randomization. In *CVPR workshops*, pages 10–11, 2019.

[15] Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024.

[16] Mate Kisantal, Sumant Sharma, Tae Ha Park, Dario Izzo, Marcus Märtens, and Simone D'Amico. Satellite pose estimation challenge: Dataset, competition design, and results. *IEEE Transactions on Aerospace and Electronic Systems*, 56 (5):4083–4098, 2020.

[17] Kecen Li, Haopeng Zhang, and Chenyu Hu. Learning-based pose estimation of non-cooperative spacecrafts with uncertainty prediction. *Aerospace*, 9(10), 2022.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[19] Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training, 2025.

[20] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[21] Alessandro Lotti, Dario Modenini, Paolo Tortora, Massimiliano Saponara, and Maria A. Perino. Deep learning for real-time satellite pose estimation on tensor processing units. *Journal of Spacecraft and Rockets*, 60(3):1034–1038, 2023.

[22] Mohamed Adel Musallam, Vincent Gaudilliere, Miguel Ortiz Del Castillo, Kassem Al Ismaeil, and Djamila Aouada. Leveraging equivariant features for absolute pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6876–6886, 2022.

[23] Tae Ha Park and Simone D'Amico. Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap. *Advances in Space Research*, 73(11): 5726–5740, 2024. Recent Advances in Satellite Constellations and Formation Flying.

[24] Tae Ha Park, Marcus Märtens, Gurvan Lecuyer, Dario Izzo, and Simone D'Amico. Speed+: Next-generation dataset for spacecraft pose estimation across domain gap. In *2022 IEEE aerospace conference (AERO)*, pages 1–15. IEEE, 2022.

[25] Tae Ha Park, Marcus Märtens, Mohsi Jawaid, Zi Wang, Bo Chen, Tat-Jun Chin, Dario Izzo, and Simone D'Amico. Satellite pose estimation competition 2021: Results and analyses. *Acta Astronautica*, 204:640–665, 2023.

[26] Leo Pauly, Wassim Rharbaoui, Carl Shneider, Arunkumar Rathinam, Vincent Gaudillière, and Djamila Aouada. A survey on deep learning-based monocular spacecraft pose esti-

mation: Current state, limitations and prospects. *Acta Astronautica*, 212:339–360, 2023.

[27] Juan Ignacio Bravo Pérez-Villar, Álvaro García-Martín, and Jesús Bescós. Spacecraft pose estimation based on unsupervised domain adaptation and on a 3d-guided loss combination. In *European Conference on Computer Vision*, pages 37–52. Springer, 2022.

[28] Andrew Price and Kazuya Yoshida. A monocular pose estimation case study: The hayabusa2 minerva-ii2 deployment. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1992–2001, 2021.

[29] Pedro F. Proença and Yang Gao. Deep learning for spacecraft pose estimation from photorealistic rendering. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6007–6013, 2020.

[30] Arunkumar Rathinam, Vincent Gaudilliere, Mohamed Adel Mohamed Ali, Miguel Ortiz Del Castillo, Leo Pauly, and Djamila Aouada. Spark 2022 dataset : Spacecraft detection and trajectory estimation, 2022.

[31] Niki Sajjad, Andrew Price, Mehran Mirshams, and Mathieu Salzmann. Tiny satellites, big challenges: A feasibility study of machine vision pose estimation for pocketqubes during conjunctions. *Acta Astronautica*, 228:1023–1041, 2025.

[32] Sumant Sharma and Simone D'Amico. Pose estimation for non-cooperative rendezvous using neural networks, 2019.

[33] Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens Van Der Maaten. Revisiting weakly supervised pre-training of visual perception models, 2022.

[34] Sebastian Stapf, Tobias Bauernfeind, and Marco Riboldi. Pvit-6d: Overclocking vision transformers for 6d pose estimation with confidence-level prediction and pose tokens, 2023.

[35] Zi Wang, Zhuo Zhang, Xiaoliang Sun, Zhang Li, and Qifeng Yu. Revisiting monocular satellite pose estimation with transformer. *IEEE Transactions on Aerospace and Electronic Systems*, 58(5):4279–4294, 2022.

[36] Zi Wang, Minglin Chen, Yulan Guo, Zhang Li, and Qifeng Yu. Bridging the domain gap in satellite pose estimation: A self-training approach based on geometrical constraints. *IEEE transactions on aerospace and electronic systems*, 60 (3):2500–2514, 2023.

[37] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024.

[38] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.