2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Sharing Attention Mechanism in V-SLAM: Relative Pose Estimation with Messenger Tokens on Small Datasets

Dun Dai[1], Quan Quan[1], and Kai-Yuan Cai[1]

***Abstract*— In V-SLAM, the estimation of relative camera pose is crucial to determine the spatial relationship between consecutive camera images, helping to accurately track the movement of the camera in its environment. In small indoor scenes, when the training set is limited, which is very common in robot SLAM, learning-based methods may fail to converge, especially the Transformer architecture, which requires a more substantial dataset to match the performance of the CNN architecture model. This work addresses this problem with the $sharing\ attention\ mechanism$, building on recent improvements in solving visual Transformer architectures on small datasets while incorporating $messenger\ tokens$. Besides, $double\text{-}embedding$ is introduced to capture the spatial of images and order of images. In summary, we introduce an intuitive end-to-end relative pose estimation solution and prove its accuracy on the two smallest sub-datasets of $7Scenes$. The proposed method is tested with a set of comparison experiments conducted across CNN-based, Transformer-based end-to-end relative pose estimation models, and the robust feature-matching non-learning method. Our model outperforms in all comparisons. Furthermore, ablation studies clearly illustrate that these innovations are crucial for the accuracy of relative pose estimation on small datasets.**

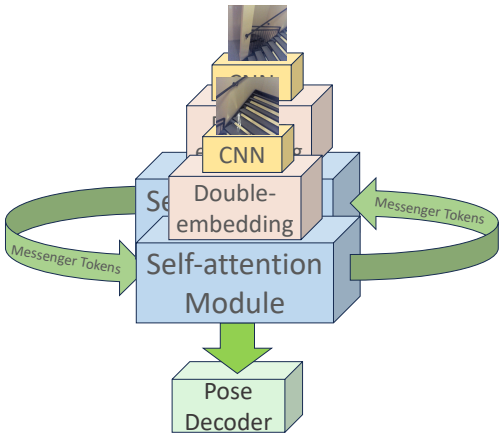


Fig. 1: We propose an end-to-end relative Transformer-based pose estimation solution on small datasets based on the Sharing Attention Mechanism with messenger tokens.

I. Introduction

In recent years, mobile intelligent robots have attracted increasing attention in the field of both academic research and industrial applications [1]. In the field of robotics, accurately estimating a robot's displacement and rotation using sequence images from a single camera is paramount to achieving the requirements of Autonomous, Dependable, and Affordable (ADA) control [2]. This challenge, a cornerstone of visual odometry, is crucial for the advancement of autonomous navigation and mapping.

Historically, the traditional feature-matching scenario of relative estimation has been considered a robust method, especially when the data is limited. This method employs techniques (ex. SURF [3], SIFT [4]) to extract key points and establish the correspondence to recover the pose. However, it also suffers from two main drawbacks. First, the quality of the estimation depends heavily on the correspondence assignment. Second, the traditional method can estimate the translation vector only up to scale [5].

Developments in visual models have increasingly focused on Transformer architectures, such as the Vision Transformer (ViT) [6]. This shift has inspired the creation of specialized Transformer models for camera relative pose estimation, highlighted by studies such as [7], [8]. These models either directly utilize ViT or employ encoders that closely resemble Transformer architecture as their fundamental structure. However, a significant limitation of traditional Transformer models is their reliance on large datasets for effective training. [9] points out that standard ViT models often struggle with local attention in the initial layers.

This requirement for large datasets poses a challenge in camera pose estimation for small indoor scenes, where data availability is often limited. Such environments typically do not offer the diverse range of examples necessary for Transformers to learn effectively, which makes the traditional Transformer-based model unable to effectively converge, and also poses a challenge to learning-based camera relative pose estimation methods. Consequently, this limits the application of Transformers and their extended models in visual SLAM tasks when only a small dataset is available.

This work addresses the specific challenge of relative camera pose estimation within a robotic SLAM framework in small indoor environments. We introduce an enhanced visual Transformer approach designed to efficiently handle smaller datasets. Crucially, we present a pioneering approach powered by the sharing attention mechanism, which facilitates effective information sharing among self-attention modules. Inspired by the feature-matching method, we expect the mechanism to be able to bridge the attention to be high-dimensional feature-matched to further improve the comprehension of limited data.

[1]Dun Dai, Quan Quan and Kai-Yuan Cai are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China d_dai_3@buaa.edu.cn, qq_buaa@buaa.edu.cn, kycai@buaa.edu.cn.

We assess our methodology using the two smallest subsets of the Microsoft $7Scenes$ dataset [10]—specifically, $Heads$ and $Stairs$, which contain 1,000 and 2,000 images, respectively. We employ only RGB images for training, enhancing the model's applicability across diverse scenarios. This evaluation serves as a benchmark to compare our proposed model against established models based on Transformer and Convolutional Neural Network (CNN) architectures [11], including the baseline model, PoseNet [12]. Furthermore, considering the ongoing relevance of robust traditional methods for localization tasks, particularly in scenarios characterized by small-scale scenes, a single environment, and limited training data, we also include a comparison with the non-learning essential matrix using 5-point algorithms combined with SIFT and RANSAC. Furthermore, to test the effect of our proposed methods, the ablation experiment is included.

In summary, our main contributions are as follows:

- Providing a reliable, end-to-end Transformer-based camera pose estimation solution and proving its convergence on small datasets.
- Propose the role of the sharing attention mechanism in relative pose estimation with messenger tokens, an intuitive and efficient mechanism for independent attention modules to share their information.
- Propose the double-embedding, to capture the spatial information and order information of image pairs.
- Conduct comparative experiments and ablation experiments. Experimentally prove that our model outperforms and our proposed mechanism is substantial for relative pose estimation.

II. Related Work

This work introduces an innovative Transformer-based sharing attention end-to-end approach for estimating camera relative pose using small datasets, to latent work similarly to the feature-matching method.

Traditional feature-matching methods commence with key point detection in two images, followed by extracting feature descriptors around these points. Techniques such as brute force matching or Fast Library for Approximate Nearest Neighbors (FLANN) [13] are employed for feature matching. Subsequently, the Random Sampling Consistency (RANSAC) [14] algorithm estimates the fundamental or essential matrices, elucidating the geometric relationship between two views. This facilitates the estimation of the camera's rotation and translation (pose). Although these methods exhibit robustness in scenarios with minor camera movements and favorable scene textures, they falter in environments with repetitive or sparse textures, or significant camera motion between views. To address the issue, some current works focus on improving feature-matching strategies, or novel correspondence [15], [16].

With the advent of learning-based methodologies, several works have proposed enhancements to these feature-matching steps. [17], [18] have incorporated deep learning into feature detection. A pivotal learning-based correspondence technique is introduced by [19]. Alternatively, different from local learning-based improvements, the end-to-end method is that the data is inputted directly to regress the pose. A well-known example is PoseNet [12], a baseline CNN-based lightweight regressor that outputs absolute pose from a single image. Additionally, [20] presents a Long Short-Term Memory [21](LSTM) -based model for absolute pose estimation. [8] devises a model with dual independent Transformer encoders for regressing translation and rotation.

Focusing on end-to-end relative pose estimation, [22] proposes a CNN-based relative pose estimation for first retrieving similar database images and then predicting the relative pose between the query and the database images. [23] designs a trainable framework consisting of learnable modules for detection, feature extraction, matching, and outlier rejection. Moreover, [24] proposes a siamese architecture of two stages of training that is independent of camera parameters. Recently, Transformers and their variants have gained prominence in various fields. To address relative pose estimation, [7] integrates visual and positional features within a ViT framework. [25] develops a siamese convolutional Transformer model for direct regression of relative camera pose. However, a common limitation of the Transformer-based model is its suboptimal performance with small datasets.

We believe that matching features is the key to enabling models to efficiently understand relative poses. Enlighteningly, our learning-based model hypothesizes that a natural mechanism for the relative pose estimation of matching features between two images exists, already embedded within the high-dimensional information. This hypothesis drives our design of a sharing attention mechanism. By the mechanism, we expect the self-attention modules of two images can work closely on high-dimensional matched features by sharing information with messenger tokens, ultimately improving the performance of the proposed method on the small dataset through more efficient information aggregation.

III. Problem Formulation

The objective of this work is to estimate the relative pose between two images captured by the same camera at different time points in the same indoor scene. Specifically, the task is to predict $\begin{bmatrix}\hat{t}\\ \hat{q}\end{bmatrix} \in \mathbb{R}^7$, where:

- t represents the relative translation vector in $\mathbb{R}^3$, defined as $t = [t_x, t_y, t_z]^\top$. This vector captures the camera's displacement along the X, Y, and Z axes between the two time points.
- q denotes the relative rotation in the form of a quaternion, residing in $\mathbb{R}^4$. The quaternion is represented as $q = [q_w, q_x, q_y, q_z]^\top$, providing an efficient representation of 3D rotation.

The regression of $\begin{bmatrix}\hat{t}\\ \hat{q}\end{bmatrix} \in \mathbb{R}^7$ from a pair of images effectively captures both the translational and rotational movements of the camera between images. Each data point in the training set consists of RGB image pairs (I_1, I_2) along with their corresponding ground-truth translation and rotation values $\begin{bmatrix}t_0\\ q_0\end{bmatrix} \in \mathbb{R}^7$. In this process, these parameters are

derived by minimizing a supervised learnable loss function $L(\hat{t}, \hat{q}, t_0, q_0)$ over a training dataset. The model will be tested on other independent recording loops from the current scene and the other scene.

IV. Proposed Approach

Our goal is to let two self-attention modules share the information, automatically focusing on the matched high-dimension features to contribute to relative pose estimation. Given the limited size of the dataset, it is impressive that the sharing attention mechanism operates efficiently. Following the success of variant Transformers [26] on small datasets, the main component of the proposed method employs two modified self-attention modules for two images interconnected via a Sharing Dynamic Aggregation Feed-forward (S-DAFF) module based on the sharing attention mechanism. This assembly forms a cohesive sharing attention module, which is then iteratively applied a predetermined number of times to enhance performance.

A. Network Architecture

Our model architecture is as shown in Fig. 2. Central to our model is a customized Transformer, integrated with innovative messenger tokens, which collectively facilitate a sharing attention mechanism.

In Stage 1, a pre-trained EfficientNet-B0 [27] serves as the backbone, to provide global features. The output activation map tensor from the backbone denoted as $\mathbf{A}^{\mathbf{f}}_{\mathbf{output}} \in \mathbb{R}^{N\times H\times W}$, will be flattened into Transformer-compatible inputs $\mathbf{X}^{\mathbf{f}}_{\mathbf{output}} \in \mathbb{R}^{(N+1)\times T}$ as suggested in [8], [28], where $T = H \times W$, representing the token dimension. At the same time, $\mathbf{A}^{\mathbf{f}}_{\mathbf{output}}$ will be embedded separately with a learned encoding to preserve the spatial information of each location. To reduce the number of parameters, two one-dimensional embeddings are separately learned for the X, Y axes, denoted as $\mathbf{E}_{\mathbf{X}} \in \mathbb{R}^{N/2\times H}$ and $\mathbf{E}_{\mathbf{Y}} \in \mathbb{R}^{N/2\times W}$. Then, each position (i, j), $i \in 1, \ldots, H$, $j \in 1, \ldots, W$, of the positional encoding matrix $\mathbf{E}_{\mathbf{P}} \in \mathbb{R}^{N\times T}$ is encoded by the concatenating of the two corresponding embedding vectors $E^j_X \in \mathbb{R}^{N/2}$ and $E^i_Y \in \mathbb{R}^{N/2}$:

$$E^{i,j}_{\mathbf{P}} = \begin{bmatrix} E^j_X \\ E^i_Y \end{bmatrix} \in \mathbb{R}^N. \tag{1}$$

For the intricate task of camera relative pose estimation in SLAM, the sequence of image processing is crucial. We extend this concept further by focusing on the identification of critical information within image patches. To this end, we introduce a novel method, referred to as order encoding, exclusively applied to the second image in the sequence.

We define the order encoding matrix $\mathbf{E}_{\mathbf{O}}$ as follows:

$$\mathbf{E}_{\mathbf{O}} = v_x v_y^\top \in \mathbb{R}^{(N+1)\times T}, \tag{2}$$

where $v_x \in \mathbb{R}^{N+1}$, $v_y \in \mathbb{R}^T$.

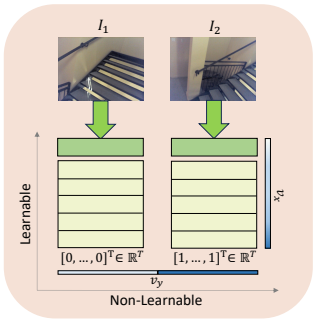


Fig. 3: The order encoding is introduced to improve the processing of time-related information.

As shown in Fig. 3, v_x, a learnable vector, represents the varying importance of each patch (token) in the aspect of time, while v_y is an unlearnable vector filled with ones. $\mathbf{E}_{\mathbf{O}}$ is only applied to the second image.

This innovative double-embedding approach, $\mathbf{E}_{\mathbf{P}}$ and $\mathbf{E}_{\mathbf{O}}$, enriches our model's ability to discern and learn from the spatial-temporal variations in the data, essential for accurate relative pose estimation in SLAM.

Innovatively, different from the general Transformer architecture, we concatenate the new learnable messenger token $X_M \in \mathbb{R}^{1\times T}$ to extracted feature patches $\mathbf{X}^{\mathbf{f}}_{\mathbf{output}}$ and embed:

$$\mathbf{X}^{\mathbf{s}}_{\mathbf{input}} = \begin{bmatrix} X_M \\ \mathbf{X}^{\mathbf{f}}_{\mathbf{output}} \end{bmatrix} + \mathbf{E}_{\mathbf{P}} + \mathbf{E}_{\mathbf{O}} \in \mathbb{R}^{(N+1)\times T}. \tag{3}$$

This is a simple but crucial improvement to the structure of the visual Transformer, playing the role of messenger to share the information between two networks, which we believe is the latent mechanism of relative camera pose estimation.

At the beginning of Stage 2, following the recommendations of [26] for improved performance on small datasets, we propose an additional computation for the Head Interaction. This process involves several transformation steps, enhancing the representational capacity of the model. The head tokens, $\mathbf{X}_{\mathbf{H}}$, are computed as follows:

First, we reshape the input tokens $\mathbf{X}^{\mathbf{s}}_{\mathbf{input}}$ and compute their average:

$$\mathbf{X}_{\mathbf{avg}} = \mathrm{Avg}(\mathrm{Reshape}(\mathbf{X}^{\mathbf{s}}_{\mathbf{input}})) \in \mathbb{R}^{h\times(T/h)}, \tag{4}$$

where h represents the number of heads. Afterward, we apply a linear projection followed by a Gaussian Error Linear Units (GELU) [29] activation function:

$$\mathbf{X}_{\mathbf{transformed}} = \mathrm{GELU}(\mathrm{Linear}(\mathbf{X}_{\mathbf{avg}})) \in \mathbb{R}^{h\times T}. \tag{5}$$

To finish the head token, we add the head embedding $\mathbf{E}_{\mathrm{H}}$ to form the head tokens:

$$\mathbf{X}_{\mathbf{H}} = \mathbf{X}_{\mathbf{transformed}} + \mathbf{E}_{\mathrm{H}} \in \mathbb{R}^{h\times T}. \tag{6}$$

These transformations effectively integrate the information from the input image with the head embedding, resulting in the formation of enriched head tokens $\mathbf{X}_{\mathbf{H}}$. These tokens are

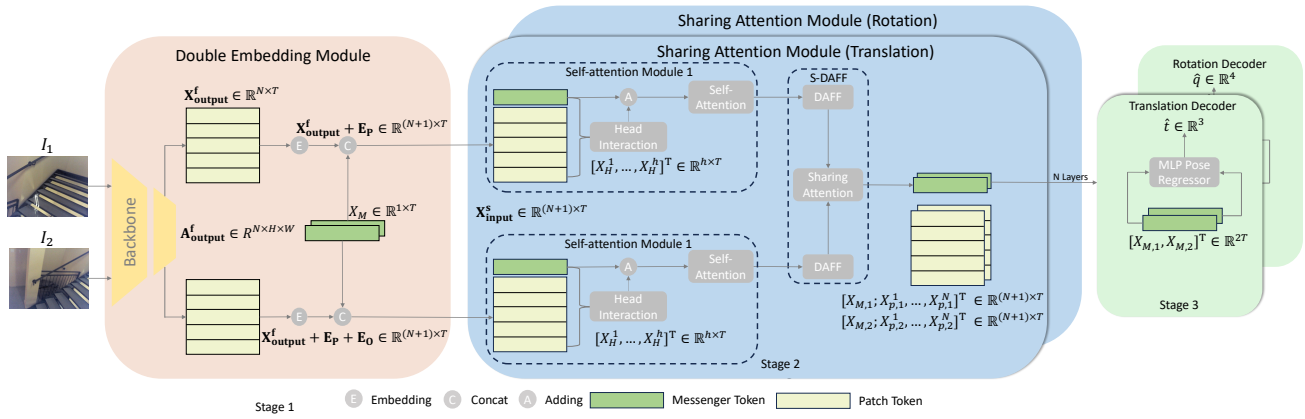


Fig. 2: The architecture of our proposed model.

crucial for capturing the essence of the image, particularly in scenarios with limited data availability.

Next, we concatenate these head tokens to the original patches and apply Self-Attention [30] along with the messenger token and head tokens:

$$\text{Self-Attention}(\left[X_M; X_p^1, \ldots, X_p^N; X_H^1, \ldots, X_H^h\right]^\top), \quad (7)$$

where $X_p^i \in \mathbb{R}^{1\times T}, i = 1, \ldots, N$, is the patch token, and h is the number of heads specified. Subsequently, we average the head tokens, and add the average to the messenger token:

$$\left[X_M + \text{Avg}(X_H^1, \ldots, X_H^h); X_p^1, \ldots, X_p^N\right]^\top, \quad (8)$$

forming back to the inputting dimension of $(N+1)\times T$.

In the realm of Transformer architectures, small dataset performance is a notable concern. Addressing this, the Dynamic Aggregation Feed-forward (DAFF) suggested by the work of [26], was originally introduced to replace the conventional feed-forward network in the standard ViT to learn stronger feature representation under insufficient data. In our model, we propose a modified Sharing DAFF (S-DAFF), aiming to integrate the information into the messenger token for the next step of sharing attention, ultimately enhancing the model's adaptability and efficiency when dealing with limited data.

The S-DAFF network is composed of two main components: the DAFF mechanism and a sharing attention component.

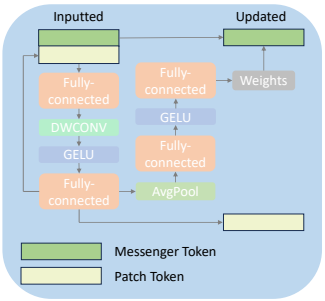


Fig. 4: The pipeline of DAFF in sharing attention Module.

In the DAFF part as shown in Fig. 4, depth-wise convolution [31] (DWCONV) is integrated into the feed-forward network, akin to CNNs in that it leverages convolution for feature representation. Specifically, the messenger token is partitioned prior to entering the projection layers. Following this, patch tokens are directed through a depth-wise integrated MLP, which incorporates a shortcut internally. The resulting patch tokens are then averaged to produce a weight vector. After a squeeze-excitation process, this weight vector is multiplied channel-wise with the messenger token. The newly updated messenger token is subsequently concatenated with the output patch tokens to reconstruct the token sequence.

Most importantly, we consider the interaction between the two self-attention modules by messenger tokens, $X_{M,1}$ and $X_{M,2}$. Since DAFF effectively aggregates the information of current patch tokens, the messenger token with the sharing attention mechanism plays a role in building a bridge after packing. The key idea is facilitated by aggregating the information from the other self-attention module to the current module, ultimately influencing the next self-attention computations due to its global nature. Accordingly, the messenger token updating strategy is as follows:

$$X_{M,1} = X_{M,1} + \text{Linear}(\text{GELU}(\text{Linear}(X_{M,2}))) \quad (9)$$

$$X_{M,2} = X_{M,2} + \text{Linear}(\text{GELU}(\text{Linear}(X_{M,1}))). \quad (10)$$

Afterward, the aggregated messenger tokens will be concatenated with patch tokens for the next sharing attention module layer or pose regression decoders.

In Stage 3, only the ultimate pair of messenger tokens are propagated forward to the pose decoders. Pose decoders are MLPs. Each MLP head consists of two hidden layers with GELU activation, tasked with regressing the translation prediction $\hat{t} \in \mathbb{R}^3$ or quaternion prediction $\hat{q} \in \mathbb{R}^4$.

B. Relative Camera Pose Loss

Assuming that our predicted pose is $\begin{bmatrix}\hat{t}\\ \hat{q}\end{bmatrix}$ and the true pose is $\begin{bmatrix}t_0\\ q_0\end{bmatrix}$, we define l_t and l_q as follows:

$$l_t = \|\hat{t} - t_0\|_2 \quad (11)$$

$$l_q = \|\hat{q} - \frac{q_0}{\|q_0\|_2}\|_2. \quad (12)$$

Here, the translational error l_t and the rotational errorl_q are defined using the Euclidean norm. The normalization of

q_0 ensures that it is a unit quaternion to guarantee a valid orientation encoding.

To combine the two parts of losses, following the approach suggested by [32], the robust pose loss function L can be formulated with exponential components and two learnable parameters s_t and s_q:

$$L = l_t \exp(-s_t) + s_t + l_q \exp(-s_q) + s_q \tag{13}$$

- The exponential components $\exp(-s_t)$ and $\exp(-s_q)$ act as adaptive weights for the translation and rotation loss components, respectively. The negative sign in the exponentials ensures that as the values of s_t or s_q increase, the influence of the corresponding loss components (l_t or l_q) diminishes. This allows the model to adjust the emphasis between translational and rotational errors dynamically during training.
- The addition of s_t and s_q directly to the loss function serves to balance the influence of translation and rotation errors. This balanced approach allows the network to learn the relative importance of minimizing translational errors versus rotational errors, which may vary depending on the specific application or dataset.

V. Data Generation & Experiment Settings

This section details the camera relative pose data generation procedure with the two smallest subsets of $Heads$ and $Stairs$ from $7Scenes$ with 1000 and 2000 images total with corresponding pose annotations, respectively. Each scene is small-scale in an indoor environment. Furthermore, the experiment implementation details of the network are provided.

A. Generation of Relative Camera Pose Dataset

To generate a dataset for estimating relative camera poses, we pair random images and compute their relative poses. For any two images I_i and I_j, with their respective camera poses: $\begin{bmatrix} t_i \\ q_i \end{bmatrix}$, and $\begin{bmatrix} t_j \\ q_j \end{bmatrix}$, where t_i and t_j are the translation vectors, and q_i and q_j are the quaternions representing orientation, the relative rotation matrix $R_{i,j}$ and relative translation vector $t_{i,j}$ can be computed as:

$$R_{i,j} = R(q_j)R(q_i)^\top,$$
$$t_{i,j} = t_j - R_{i,j}t_i.$$

Here, $R(q)$ converts a quaternion q into its corresponding rotation matrix. The relative pose is then represented as a 7-dimensional vector, combining the quaternion representation of $R_{i,j}$, i.e. $q_{i,j}$, with $t_{i,j}$.

This structured approach efficiently generates a diverse dataset of image pairs, each annotated with their corresponding relative poses, essential for training models to interpret spatial relationships between camera viewpoints accurately.

B. Training Procedure

The training pipeline is meticulously designed to ensure robust learning and generalization of the model. The following steps delineate the procedure:

1) **Data Preparation:** Initially, each image in the dataset of a scene is separated by sequences. Each sequence represents a camera moving loop and can help strictly split the train/validation set and test set. Subsequently, images from the same scene are paired randomly within train/validation/test sequences.
2) **Network Training Paradigm:** The model is trained using Adam [33] optimizer with $\beta = (0.9, 0.999)$, $\epsilon = 10^{-10}$, and an initial learning rate of 0.001 computed on one Nvidia V100 GPU and Intel Xeon Gold 6130 CPU @ 2.10GHz, which gracefully decays to 0.000001 according to a "CosineAnnealingLR" schedule. A batch size of 16 is employed to balance computational efficiency and gradient accuracy. To mitigate overfitting, dropout, and drop path rates are set within the regression network.

These training methodologies are carefully chosen to optimize performance on the task of camera relative pose estimation, ensuring that the model is not only accurate but also resilient to variations within the data.

C. Experiment Design

We propose three comparison experiment scenarios for evaluating camera relative pose estimation, encompassing three major paths:

1) **Traditional Feature-Matching Method:**
   - Assesses the proposed model against traditional feature-matching methods.
   - Adjusts the step size of image overlap by sampling the dataset without shuffling, due to the necessity of overlapping areas for feature matching.
2) **Learning-Based Method:**
   - Compares the proposed model with other end-to-end relative pose estimation models, incorporating both Transformer and CNN structures.
3) **Model Generalization:**
   - Tests the generalization ability of the model trained on one scene type when applied to a different scene type.
4) **Ablation Study:**
   - Test the single effect of each core component of our model trained by replacing it respectively.

For each experiment, we compute the rotation error (defined as the rotation geodesic to the ground truth) and translation error (defined as the usual Euclidean distance to the ground truth) and report the summary statistics of the $mean$ and the $median$.

VI. Experiment Results

This section provides examples of the estimation results and comparisons that we propose in Section V-C. To clarify, our model and each learning-based model are trained with a single scene for the specified comparison of the scene. Table I reports the size of the images for each scene scenario.

TABLE I: The size of train/validation/test set for each scenario.

	Train	Val.	Test
Heads	800	200	1000
Stairs	1200	300	1000

A. Comparison with Traditional Feature-Matching Method

To accommodate the intrinsic requirements of traditional feature-matching methods, such as pose recovery using the essential matrix approach with SIFT extracting sparse keypoints, establishing 2D correspondences between keypoints and RANSAC to reject outliers robustly (denoted as $S+R$), which necessitate overlapping areas for extracting and matching feature points, we have refined our testing strategy. We designate each image in the test sequence as the primary frame and choose the subsequent frame based on a predefined step size. Specifically, the index of the second frame is determined by adding the step size to the index of the first frame, denoted as $Index_2 = Index_1 + step\ size$. Regarding the choice of step size, we consider values of $[10, 15, 20, 30]$ to represent varying scales of motion across different scenes, while ensuring sufficient overlap to facilitate the feasibility of the feature-matching approach. Given that the $S+R$ method can only recover the translation vector to an unknown scale, our comparison focuses exclusively on the accuracy of rotation prediction, measured in degrees.

TABLE II: Rotation error comparison between methods for two scenes and various step sizes.

Step	Scene 1: Heads				Scene 2: Stairs			
	Proposed		S+R		Proposed		S+R	
	Med.↓	Mean↓	Med.↓	Mean↓	Med.↓	Mean.↓	Med.↓	Mean↓
10	**2.548**	**2.871**	5.212	6.029	**2.195**	**2.718**	2.498	3.384
15	**3.701**	**4.151**	7.705	9.812	**2.709**	**3.332**	3.338	4.913
20	**4.657**	**5.371**	10.672	13.680	**2.984**	**3.721**	4.128	6.225
30	**6.879**	**7.589**	15.706	21.410	**3.600**	**4.279**	5.449	8.917

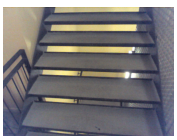
B. Comparison with Learning-Based Method

In this subsection, we present a comparison study comparing our method with other learning-based methods. Fig. 5 and Fig. 6 show randomly selected sample pairs from $Heads$ and $Stairs$ respectively, and the corresponding translation and rotation errors predicted by our proposed method.

(a) Heads sample 1

(b) Heads sample 2

Fig. 5: Trans. Error: 0.196m; Rot. Error: 4.816°



(a) Stairs sample 1

(b) Stairs sample 2

Fig. 6: Trans. Error: 0.313m; Rot. Error: 4.373°

Numerically, Table III reports the comparison across three end-to-end learning-based methods, including CNN-based and Transformer-based models.

TABLE III: Compare our proposed method with other end-to-end Learning-based methods on a small dataset.

Method	Scene	Translation (m)		Rotation (degrees)	
		Med.↓	Mean↓	Med.↓	Mean↓
PoseNet [12]	Heads	0.595	0.620	13.091	15.131
CNN-based [22]		0.397	0.427	9.171	**9.339**
Transformer-based [25]		0.368	0.396	9.464	9.808
Proposed		**0.301**	**0.344**	**8.794**	9.901
PoseNet [12]	Stairs	0.796	0.936	11.043	12.862
CNN-based [22]		0.668	0.698	6.733	8.650
Transformer-based [25]		0.538	0.632	6.829	7.188
Proposed		**0.504**	**0.522**	**5.904**	**6.690**

C. Model Generalization

In this subsection, we will test Model Generalization. Table IV reports rotation error and translation error of the current scene-trained model predicting the other scene, i.e. the scene of $Stairs$ with $Heads$ trained model, and the scene of $Heads$ with $Stairs$ trained model.

TABLE IV: Compare the Generalization ability of our proposed method with other learning-based methods.

Method	Scene/Train	Translation (m)		Rotation (degrees)	
		Med.↓	Mean↓	Med.↓	Mean↓
PoseNet [12]	Heads/Stairs	0.607	0.660	14.426	16.020
CNN-based [22]		0.589	**0.608**	15.608	18.752
Transformer-based [25]		**0.580**	0.621	15.289	17.362
Proposed		0.659	0.699	**13.521**	**14.858**
PoseNet [12]	Stairs/Heads	0.892	**0.935**	11.252	13.090
CNN-based [22]		0.905	0.941	10.851	12.794
Transformer-based [25]		0.941	0.967	9.177	12.870
Proposed		**0.872**	1.014	**9.041**	**12.149**

D. Ablation Study

We examine three distinct scenarios in Table V: 1) The substitution of the $messenger\ token$ with the $class\ token$ in a standard Transformer for regression purposes. 2) The substitution of the $S\text{-}DAFF$ with a regular $feed-forward\ network$ of a standard Transformer. 3) The adoption of $positional\ embeddings$ of ViT instead of employing a $double-embedding$.

TABLE V: Compare our proposed method with other different architectures. Models are trained and tested on a single scene.

Substitution	Scene	Translation (m)		Rotation (degrees)	
		Med.↓	Mean↓	Med.↓	Mean↓
Standard Class Token	Heads	0.766	0.920	11.084	13.620
Standard Feed-forward		0.793	0.950	12.171	13.769
ViT's Positional Embedding		0.905	1.048	9.757	12.296
Proposed		**0.301**	**0.344**	**8.794**	**9.901**
Standard Class Token	Stairs	0.902	0.996	7.128	8.431
Standard Feed-forward		0.949	1.072	7.411	8.886
ViT's Positional Embedding		0.704	0.786	6.872	7.380
Proposed		**0.504**	**0.522**	**5.904**	**6.690**

Notably, the substitution of messenger tokens and the S-DAFF module significantly deteriorates the model's performance, emphatically highlighting the pivotal role of the proposed sharing attention mechanism in enhancing efficiency.

VII. Conclusion

In this work, we present an innovative approach to relative pose estimation through the development of a Transformer-

based, end-to-end methodology. Our approach introduces the novel concept of sharing attention mechanism with messenger tokens, alongside a double-embedding, specifically designed to tackle the inherent challenges of relative pose tasks. By leveraging advancements in the standard Transformer architecture, our method effectively addresses the issue of data scarcity, making strategic enhancements that significantly boost performance. Notably, the sharing attention mechanism emerges as a pivotal innovation of our approach, promising to be a focal point for future research when extending to various scales and larger data sizes.

REFERENCES

[1] Q. Quan, *Introduction to Multicopter Design and Control*. Springer Singapore, 2017.

[2] K. Y. Cai, K. S. Trivedi, and B. Yin, "S-ada: Software as an autonomous, dependable and affordable system," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S)*, 2021, pp. 17–18.

[3] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.

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

[5] S. En, A. Lechervy, and F. Jurie, "Rpnet: An end-to-end network for relative camera pose estimation," in *Computer Vision – ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2019, p. 738–745.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[7] C. Rockwell, J. Johnson, and D. F. Fouhey, "The 8-point algorithm as an inductive bias for relative pose prediction by vits," in *2022 International Conference on 3D Vision (3DV)*, 2022, pp. 1–11.

[8] Y. Shavit, R. Ferens, and Y. Keller, "Learning multi-scene absolute pose regression with transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, oct 2021, pp. 2713–2722.

[9] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.

[10] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2930–2937.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[12] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, dec 2015, pp. 2938–2946.

[13] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration." in *VISAPP (1)*, A. Ranchordas and H. Araújo, Eds. INSTICC Press, 2009, pp. 331–340.

[14] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[15] Y. Zhang, Y. Zhang, B. Hu, Y. Yin, W. Chen, X. Liu, and Q. Yu, "An efficient and accurate solution to camera pose estimation problem from point and line correspondences based on null space analysis," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3762–3769.

[16] M. Zins, G. Simon, and M.-O. Berger, "Level set-based camera pose estimation from multiple 2d/3d ellipse-ellipsoid correspondences," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 939–946.

[17] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[18] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint detection and description of local features." *CoRR*, vol. abs/1905.03561, 2019.

[19] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2018.

[20] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 627–637.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017.

[23] Y.-Y. Jau, R. Zhu, H. Su, and M. Chandraker, "Deep keypoint-based camera pose estimation with geometric constraints," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4950–4957.

[24] P. Rajendran, S. Mishra, L. F. Vecchietti, and D. Har, "Relmobnet: End-to-end relative camera pose estimation using a robust two-stage training," in *ECCV Workshops*, 2022.

[25] K. Leng, C. Yang, W. Sui, J. Liu, and Z. Li, "Sitpose: A siamese convolutional transformer for relative camera pose estimation," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*. Los Alamitos, CA, USA: IEEE Computer Society, jul 2023, pp. 1871–1876.

[26] Z. Lu, H. Xie, C. Liu, and Y. Zhang, "Bridging the gap between vision transformers and convolutional neural networks on small datasets," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14 663–14 677, 2022.

[27] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[28] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2020, pp. 213–229.

[29] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*. Long Beach, CA, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[32] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jul 2017, pp. 6555–6564.

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