Spatiotemporal Dual-Stream Network for Visual Odometry

Chang Xu, Taiping Zeng, Yifan Luo, Fei Song, and Bailu Si

***Abstract*—Visual Odometry (VO) empowers robots with the ability to perform self-localization within unknown environments using visual cues, yet it is faced with challenges in dynamic environments. In this study, we propose a novel monocular visual odometry network called Spatiotemporal Dual-stream Network (STDN-VO) with two parallel streams, i.e. spatial stream and temporal stream, to model spatiotemporal correlation in the image sequences. Technically, the spatial stream is responsible for extracting global context information from an image, while the temporal stream is designed to effectively extract robust temporal context information from consecutive frames. The outputs of the spatial stream and the temporal stream are merged and then fed to a pose head for predicting the relative pose. Experimental results on the KITTI dataset demonstrate competitive pose estimation performance exceeding published deep learning-based methods. These results underscore the effectiveness of the proposed framework for visual odometry.**

***Index Terms*—Monocular visual odometry, dual-stream network, deep learning.**

I. Introduction

VISUAL Simultaneous Localization and Mapping (vSLAM), a technology that employs vision sensors for pose estimation and simultaneous environment mapping, is widely utilized in many fields, such as autonomous driving [1], augmented reality [2], and robotics [3]. As a critical component of vSLAM, Visual Odometry (VO) ensures reliable pose estimation via analyzing sequences of continuous images. In particular, Monocular Visual Odometry (MVO) attracts considerable attention from researchers, attributable to the convenience, cost-efficiency, and adaptability of monocular cameras in diverse environments.

Traditional VO methods, which are provided as geometric models and are typically viewed as an optimization problem, can be divided into two categories: indirect and direct. Indirect VO methods rely on the extraction and matching of feature keypoints, transforming image pairs into corresponding sets of keypoints and deriving the robot's pose by minimizing reprojection error [4]. Conversely, direct VO methods posit that extracting feature keypoints may result in the loss of significant information. Therefore, direct VO methods aim to minimize photometric error by leveraging photometric consistency of raw pixel data [5]. However, the robustness of the aforementioned methods can be compromised in complex environments characterized by dynamic lighting conditions and scene variations, where feature extraction and matching are prone to inaccurate outcomes.

In recent years, deep learning has demonstrated superior competitiveness in a variety of fields [6], [7], [8], attributed to its powerful capacity for feature representation and robustness. VO has also been addressed in the deep learning framework [9], [10], [11], [12], [13], [14], [15], [16], harnessing the proficiency of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) in learning representations and modeling dynamics. For example, DeepVO [9] provides an end-to-end recurrent convolutional neural network model for MVO. In addition to the aforementioned works, transformer-based methods [17], [18] are making strides as well, with TSformer-VO [18] and SWformer-VO [17] being noteworthy illustrations. TSformer-VO, based on the TimeSformer [19], incorporates sequential spatio-temporal attention mechanisms. This design allows the both temporal and spatial self-attention share the same architecture, capturing the spatio-temporal features while reducing the number of model parameters. Additionally, SWformer-VO utilizes the Swin Transformer [20] as its backbone network and innovatively incorporates a novel 'Mixture Embed' module. This module is designed to process the spatial and temporal information by fusing consecutive image pairs into tokens, which are then fed into the backbone network. Consequently, SWformer-VO is capable of estimating the six degrees of freedom (6-DoF) camera pose under monocular camera conditions. These methods have significantly pushed forward the field of MVO. However, these methods do not consider the segregation of temporal and spatial features, jeopardizing the efficiency in

Received 13 January 2025; accepted 31 January 2025. Date of publication 20 February 2025; date of current version 13 March 2025. This article was recommended for publication by Associate Editor Christopher Steven McCool and Editor Abhinav Valada upon evaluation of the reviewers' comments. This work was supported by the National Science and Technology Innovation 2030 Major Program of China under Grant 2022ZD0205005. *(Corresponding author: Bailu Si.)*

Chang Xu is with the School of System Science, Beijing Normal University, Beijing 100875, China (e-mail: changxu@mail.bnu.edu.cn).

Taiping Zeng is with the Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai 200433, China, and also with the Ministry of Education, Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence Fudan University, Shanghai 200433, China (e-mail: zengtaiping@fudan.edu.cn).

Yifan Luo is with the Hangzhou Institute for Advanced Study, UCAS, Hangzhou 310024, China (e-mail: yifanluo930@gmail.com).

Fei Song is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Science, Shenyang 110016, China (e-mail: songfei20160903@gmail.com).

Bailu Si is with the School of System Science, Beijing Normal University, Beijing 100875, China, and also with the Chinese Institute for Brain Research, Beijing 110016, China (e-mail: bailusi@bnu.edu.cn).

Digital Object Identifier 10.1109/LRA.2025.3544521

visual information processing. In contrast, in the human visual cortex, there are two distinct pathways for visual processing: the ventral visual stream (involved in object recognition) [21] and the dorsal visual stream (responsible for encoding motion information) [22]. This theory has been widely recognized as the two-stream hypothesis [23].

Inspired by the two-stream hypothesis, in this work, we introduce a novel model for MVO, namely Spatiotemporal Dual-stream Network (STDN-VO). We design two parallel streams, i.e., spatial stream and temporal stream, to model the segregation of spatiotemporal features of image sequences. Specifically, the spatial stream is responsible for extracting spatial features from an image. It is implemented by using Vision Transformer (ViT) [24] to capture global context information. The process begins by presenting the input as a series of uniformly-sized, non-overlapping patches. To consider the 'order' of these patches, each is combined with a learnable positional embedding. Following this, the multi-head attention mechanism within the ViT enables the modeling of the relationships between different patches, culminating in a comprehensive and detailed understanding of the image. Additionally, the temporal stream is designed to effectively extract robust temporal context information of targets from previous frames. Inspired by RAFT [25], we harness ConvGRU to adeptly capture sequential dependencies between consecutive frames. Upon feeding in a pair of adjacent frames, we calculate the correlation between them, generating a correlation matrix that, when combined with the frame data, forms a new input for the ConvGRU, which adeptly captures the dynamic characteristics within the image sequence. Finally, the outputs of spatial stream and temporal stream are concatenated and then fed to a pose decoder for predicting the relative pose. Extensive experiments on KITTI benchmarks show that the proposed STDN-VO achieves better performance than recent deep learning VO methods.

To summarize, our contributions come in three folds:

- We propose a dual-stream architecture for MVO, i.e. STDN-VO, mimicking the parallel pathway of human visual system.
- The proposed architecture outperforms alternative architectures with only single stream or ViT replaced by a CNN.
- STDN-VO demonstrates improved pose estimation on the KITTI dataset, competitive with recent deep learning VO methods like DeepVO, TSformer-VO, and SWformer-VO.

Additionally, we intend to make the source code of STDN-VO publicly available to facilitate further research and development within this domain.

II. Related Work

Deep learning for pose estimation: Certainly, the realm of learning-based VO has witnessed remarkable advancements. PoseNet [26], in particular, stands out as a pioneering end-to-end VO model that harnesses CNNs, marking a significant milestone in the evolution of this field. However, when dealing with sequential data, relying exclusively on CNNs may not fully capture the complexities of temporal dynamics. Consequently, RNNs, which are more adept at handling sequential data, have been introduced into VO. DeepVO [9], an end-to-end model, harnesses the power of CNNs to extract rich image features, which are then seamlessly channeled into a Long Short-Term Memory (LSTM) module. In this way, the architecture adeptly captures temporal correlations, enabling the model to achieve remarkable accuracy and robust generalization capabilities. As a benchmark of the supervised learning VO method, DeepVO served as a foundational framework for a host of research efforts [10], [11], [12], [13], [14], [15], [16].

In addition to the methods above, advancements in model architecture are being witnessed. For instance, the Transformer [27], renowned for its breakthroughs in natural language processing (NLP), has been effectively applied to the field of VO [17], [18].

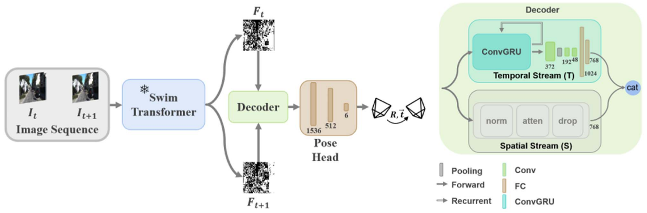
Dual-stream networks: The dual-stream networks draw inspiration from the two-stream hypothesis observed in the human visual cortex [23]. This architecture mimics the two pathways of the human visual system, employing two distinct network branches to process the spatial and temporal information. Consequently, by fusing these distinct streams of information, the dual-stream architecture leverages their combined insights to deliver superior performance.

The pioneering research in the realm of video understanding utilizing a dual-stream architecture is attributed to the efforts of Karen et al. [28] Their approach includes a spatial stream module that takes single-frame image as input and a temporal stream module for handling sequences of frames. After passing through a softmax layer, these two parts are simply fused and then utilized for action recognition. Christoph et al. [29] argued that the feature fusion module in existing dual-stream architectures was overly simplistic. Consequently, they conducted a series of experiments to devise more effective fusion module. Peng et al. [30] introduced a novel dual-stream architecture designed for video understanding. This architecture is distinguished by a spatial-temporal interaction learning module that employs an alternating collaborative attention mechanism to bridge the two streams, thereby enhancing the learning of spatial and temporal feature correlations. The dual-stream architecture is particularly effective for sequential recognition tasks, as it adeptly captures both temporal and spatial features, thereby significantly enhancing the model's performance.

In this study, we leverage a dual-stream architecture for MVO to address the spatial temporal dynamics of visual inputs. Specifically, the temporal stream module employs a ConvGRU, while the spatial stream utilizes ViT. The integration of features extracted from these two distinct streams, facilitated by a specialized pose head, allows us to precisely estimate the 6-DoF relative pose.

III. Method

Given a pair of consecutive monocular images, STDN-VO aims for 6-DoF pose estimation. As depicted in Fig. 1, STDN-VO model is composed of three main modules: a feature extraction module, a decoder module, and a pose head module (Sec. A). The feature extraction module serves to extract effective features from the monocular images. The decoder module is designed



(a)

(b)

Fig. 1. (a) The overall pipeline of the proposed STDN-VO: the model receives two consecutive monocular images $\{I_t, I_{t+1}\}$, which are processed by the pre-trained Swin Transformer. The resultant feature representations are then fed into the decoder block. Finally, the pose head provides an estimation of the 6-DoF relative pose. (b) Details of the decoder block illustrate a dual-stream architecture: the upper stream (T) dedicates to temporal processing and the lower stream (S) focuses on spatial analysis.

to model spatiotemporal correlation in the image sequences. The pose head module, meanwhile, focuses on pose estimation. Furthermore, the loss function, introduced in Sec. B, plays a pivotal role in the model's training phase.

A. Model Design

Feature extraction module: We utilize the pre-trained Swin Transformer for feature extraction. As depicted in Fig. 1(a), the input comprises a sequence of two consecutive images, denoted as $\{I_t, I_{t+1}\}$, with $I \in \mathbb{R}^{H\times W\times C}$. Here, C symbolizes the number of channels in each input image, which defaults to 3. The dimensions of the image, represented by $H \times W$, correspond to its height and width, defaulting to 256×256. The inputs are meticulously processed by the pre-trained Swin Transformer, consequently yielding feature representations, $\{F_t, F_{t+1}\}$, with the resolution reduced to a quarter of its original size. These features are subsequently channeled into both the spatial stream module and the temporal stream module, where they are further analyzed and synthesized to capture the complex spatial and temporal dynamics of the visual data.

Spatial stream module: In this letter, we introduce a spatial stream module based on ViT. We utilize patch embeddings by partitioning each of the input features $\{F_t, F_{t+1}\}$ individually into N patches, which are of size $p \times p$ pixels. These patches are then mapped to D dimensions, represented by X_i, via a trainable convolutional layer with a kernel size of $p \times p$ and a stride p matching the patch size. Subsequently, we add a learnable embedding, denoted as X_{class}, to the sequence of embedded patches. X_{class} can be considered as the global features of the input tensor. Furthermore, we add the trainable position embeddings, E_{pos}, to the patch embeddings, capturing the positional information of each patch within the original image.

$$Z_0 = [X_{class}; X_1; X_2; \ldots; X_N] + E_{pos}, \tag{1}$$

$$Z'_\ell = \mathrm{MSA}(\mathrm{LN}(Z_{\ell-1})) + Z_{\ell-1}, \quad \ell = 1\ldots L \tag{2}$$

$$Z_\ell = \mathrm{MLP}(\mathrm{LN}(Z'_\ell)) + Z'_\ell, \qquad \ell = 1\ldots L \tag{3}$$

$$y = \mathrm{LN}(Z_L^0), \tag{4}$$

where L denotes the number of the transformer encoder blocks, Z_0 acts as the input to the initial Transformer encoder block, undergoing layernorm (LN) and multiheaded self-attention (MSA) processes. Ultimately, Z_L^0, representing the state of X_{class} at the final Transformer encoder block, is selected to serve as the global representation y of the input.

Temporal stream module: The temporal stream module borrows the key components of RAFT to adeptly capture the dynamics of temporal sequences [25]. This module takes as input a pair of feature representations $\{F_t, F_{t+1}\}$, which are extracted by the feature extraction module, to construct the correlation volume. The correlation volume, denoted as C, is formulated by taking the dot product between $F_t \in \mathbb{R}^{H\times W\times D}$ and $F_{t+1} \in \mathbb{R}^{H\times W\times D}$:

$$C_{ijkl} = \sum_h (F_t)_{ijh}(F_{t+1})_{klh}. \quad C \in R^{H\times W\times H\times W} \tag{5}$$

Subsequently, we directly derive the context feature from the feature representation F_t. This context feature, endowed with a wealth of image information, significantly enhances the model's capacity for scene understanding. The context feature, along with the correlation volume, is then fed into ConvGRU. Similar to RAFT, we utilize a 1×5 convolution and a 5×1 convolution to replace the 3×3 convolution in the ConvGRU unit, which increases the receptive field without significantly increasing the size of the model. Following ConvGRU, as shown in Fig. 1(b), three convolutional layers are sequentially applied, each with kernel size of 3. These layers are coupled with a ReLU activation function to introduce non-linearity. Subsequently, The output from these convolutional layers is channeled through a fully connected (FC) layer, yielding a 768-dimensional output vector.

Pose head module: The pose head module, as illustrated in Fig. 1(a), is composed of two FC layers, where each is followed by a LeakyReLU activation function. This activation allows the network to learn even when dealing with negative input values, thus mitigating the common issue of 'dead neurons' that can occur with the ReLU activation function. The final output is a 6-dimensional vector, which serves as the estimated pose.

B. Supervision

We supervised our network on the mean squared error (MSE) between the predicted and ground truth pose. The loss is defined as:

$$\mathcal{L} = \frac{1}{3B_s}\sum_{n=1}^{B_s}\sum_{i=1}^{3}\left(t_{i,n} - \hat{t}_{i,n}\right)^2 + k\left(r_{i,n} - \hat{r}_{i,n}\right)^2. \tag{6}$$

Here, B_s denotes the batch size. $t_{i,n}$ and $r_{i,n}$ represent the ground truth translation and rotation respectively. $\hat{t}_{i,n}$ and $\hat{r}_{i,n}$ are the corresponding predictions by the model. $k = 100$ is a positive parameter to weight position and orientation as in DeepVO [9].

IV. Experiments

We first evaluate the effectiveness of the proposed STDN-VO on the KITTI visual odometry benchmark [31] against recent deep learning VO methods. Furthermore, we conduct experiments on the TUM [32] and EuRoC [33] datasets to demonstrate the robustness and generalization ability of STDN-VO. Finally, we design ablation studies to validate the design of our model.

A. Dataset

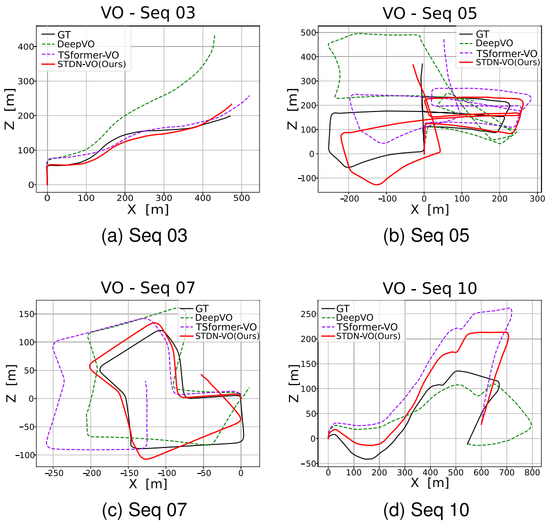
We utilize the KITTI dataset, recognized as the primary benchmark for evaluating VO models, to assess the effectiveness of STDN-VO. Spanning 39.2 kilometers of visual odometry sequences, the dataset is divided into categories including ‘Road’, ‘City’, ‘Residential’, ‘Campus’, and ‘Person’. It comprises a total of 22 stereo sequences, with sequences $00-10$ offering ground truth trajectories for training, sequences $11-21$, however, are without ground truth. For the purpose of quantitative evaluation, we choose to train and validate our model using the sequences from 00 to 10. Given our focus on monocular VO, we only use the left camera view.

B. Implementation Details

STDN-VO is implemented by PyTorch [34] and trained on a single NVIDIA RTX 4090 GPU with a batch size of 4. During the preprocessing phase, the input images were uniformly resized to dimensions of 256×256 pixels, and their pixel values were normalized to ensure consistency. STDN-VO consists of three principal modules: feature extraction, decoder, and pose head. The parameters of the feature extraction module are fixed, thus exempting them from further training iterations. The remaining two modules are actively engaged in the training process. We implemented the AdamW optimizer [35], incorporating a weight decay of $1e-4$, and employed the OneCycleLR scheduler [36]. The initial learning rates were set to $1e-4$. The loss weighting parameter k was set to 100. We trained our model for 200 epochs in total.

C. Evaluation of Visual Odometry

1) Evaluation on the KITTI dataset: The performance of the trained STDN-VO is assessed based on the standard evaluation metrics, including average translation errors T_{err} (in percentage), rotation errors R_{err} (in degrees per 100 meters), absolute trajectory error ATE (in meters), relative pose error (RPE) for rotation (in degrees) and translation (in meters), average translational RMSE drift T_{rel} (in percentage), and average rotational RMSE drift R_{rel} (in degrees per 100 meters). T_{err} and T_{rel} are considered for all possible subsequences within a test sequence of lengths $(100, \ldots, 800)$ meters. Monocular methods suffer from scale ambiguity when attempting to restore the real-world scale. Prior works have utilized 7-DoF optimization [37], [38] to address this issue by applying a scaling factor to align the predicted poses to the ground truths. Following these works [17], [18], [37], [38], we also applied a 7-DoF optimization in validation. The final metrics were calculated using the Python KITTI evaluation toolbox, which was employed in TSformer-VO [18]. In this study, we conduct two training strategies.



(a) Seq 03 (b) Seq 05

(c) Seq 07 (d) Seq 10

Fig. 2. Qualitative trajectory results on KITTI Odometry Sequences 03, 05, 07, and 10. EST: Estimation, GT: Ground Truth.

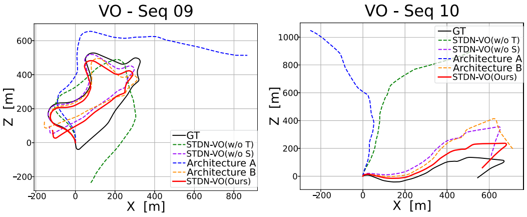
Strategy 1: We employ the KITTI odometry sequences 00, 02, 08, and 09 for training and compare our method with recent deep learning VO methods on sequences 01, 03, 04, 05, 06, 07, and 10. Table I offers a comprehensive comparison. The best values for each sequence among the deep learning methods are highlighted in bold, while the second-best values are underscored. The average errors of the experiments are derived from the mean of the scores across all sequences utilized for testing. Compared with ORB-SLAM2 [38], a traditional VO method, STDN-VO showed advantage in $T_{err}(\%)$ on sequences 01, 05 and 07. Compared with deep learning VO methods, STDN-VO achieved superior performance in terms of $T_{err}(\%)$ across all sequences except for sequence 06. Evaluated with respect to $R_{err}(^\circ/100\text{ m})$, STDN-VO surpassed these deep learning methods on sequences 01, 03, 05, 07, and 10. In terms of ATE(m), STDN-VO is ranked the best on sequences 04, 05, 07, and 10. Similarly, STDN-VO also showed a marked advantage in the RPE(m) and $T_{erl}(\%)$ metrics, leading in performance across the majority of test sequences. Nonetheless, when it comes to the RPE$(^\circ)$ and $R_{erl}(^\circ/m)$ metrics, ORB-SLAM2 continues to hold a substantial lead over all other methods. Overall, the average error achieved by STDN-VO is superior to all the deep learning methods listed. The results demonstrated the effectiveness of our model for VO estimation. Fig. 2 showcases the estimated trajectories.

Strategy 2: In order to show that the performance of STDN-VO is insensitive to training set, we employ a different training set, i.e. sequences $00-08$ in the KITTI dataset, for training and compare STDN-VO with recent deep learning VO methods on sequences 09, and 10. As illustrated in Table II, STDN-VO achieved state-of-the-art performance among these methods in terms of both average translation errors and average rotation errors. Specifically, STDN-VO reduced T_{err} (average translation errors) from 9.880, as reported by Wang et al. [42], to 6.784 on sequence 09. Similarly, on sequence 10, STDN-VO lowered T_{err} from 8.927, as achieved by SWformer-VO, to 7.730. In terms of R_{err} (average rotation errors), STDN-VO improved the result from 3.340, as seen in Bian et al. [43], to 2.491 on sequence 09. Furthermore, STDN-VO reduced average rotation

TABLE I
QUANTITATIVE RESULTS ON KITTI ODOMETRY

Evaluation Index	Method	Sequence 01	03	04	05	06	07	10	Avg. Error
T_{err}(%)↓	ORB-SLAM2	107.565	1.554	1.554	9.671	18.899	10.195	3.638	21.868
	DeepVO	156.389	73.552	10.803	56.184	64.397	71.790	128.732	80.263
	TSformer-VO	28.860	25.587	4.852	14.746	13.365	12.892	16.387	16.669
	SWformer-VO	29.755	12.752	5.312	7.651	**12.978**	10.586	9.672	12.672
	STDN-VO(ours)	**18.774**	**10.984**	**4.158**	**5.716**	20.242	**6.780**	**7.125**	**10.540**
R_{err}(°/100m)↓	ORB-SLAM2	0.888	0.182	0.267	0.243	0.234	0.327	0.322	0.351
	DeepVO	10.036	15.671	3.849	29.898	31.395	50.508	21.465	23.260
	TSformer-VO	6.508	15.484	2.947	5.713	4.409	8.611	5.072	6.963
	SWformer-VO	6.769	8.600	**1.978**	3.322	**4.030**	8.795	4.677	5.453
	STDN-VO(ours)	**3.695**	**7.756**	2.208	**2.175**	6.779	**5.906**	**2.480**	**4.428**
ATE(m)↓	ORB-SLAM2	502.201	1.752	1.296	33.196	55.025	16.557	7.735	88.251
	DeepVO	**19.981**	**11.744**	3.850	123.298	107.995	22.831	57.901	49.657
	TSformer-VO	121.760	24.118	3.487	59.480	**33.047**	29.824	25.045	42.388
	SWformer-VO	130.065	19.005	4.340	37.518	39.062	27.084	18.293	39.338
	STDN-VO(ours)	74.904	16.571	**3.403**	**19.130**	71.347	**16.358**	**11.630**	**30.478**
RPE(m)↓	ORB-SLAM2	2.970	0.033	0.078	0.147	0.300	0.112	0.055	0.527
	DeepVO	3.577	0.553	0.261	0.808	1.152	0.741	1.135	1.175
	TSformer-VO	0.688	0.124	0.105	0.129	**0.167**	0.143	0.159	0.216
	SWformer-VO	0.679	0.109	0.110	0.109	0.173	0.112	0.118	0.201
	STDN-VO(ours)	**0.503**	**0.079**	**0.090**	**0.091**	0.261	**0.109**	**0.110**	**0.178**
RPE(°)↓	ORB-SLAM2	0.098	0.053	0.079	0.057	0.057	0.049	0.067	0.065
	DeepVO	0.440	0.438	0.137	0.535	0.476	0.703	0.580	0.472
	TSformer-VO	0.294	0.246	0.144	0.216	0.205	0.237	0.274	0.230
	SWformer-VO	0.332	0.231	0.129	0.204	0.203	0.230	0.250	0.225
	STDN-VO(ours)	**0.196**	**0.119**	**0.063**	**0.099**	**0.129**	**0.111**	**0.137**	**0.122**
T_{rel}(%)↓	ORB-SLAM2	96.058	90.621	98.124	61.974	66.043	60.086	83.719	79.518
	DeepVO	161.347	92.761	14.743	90.068	93.389	109.142	144.200	100.807
	TSformer-VO	67.694	30.163	29.427	17.639	19.181	14.608	21.883	28.656
	SWformer-VO	66.860	17.117	14.344	8.953	**16.988**	11.421	16.512	21.742
	STDN-VO(ours)	**55.094**	**14.031**	**13.867**	**6.778**	17.304	**6.881**	**10.093**	**17.721**
R_{rel}(°/m)↓	ORB-SLAM2	0.007	0.002	0.003	0.003	0.003	0.003	0.004	0.004
	DeepVO	0.204	0.216	0.039	0.410	0.475	0.597	0.296	0.320
	TSformer-VO	0.103	0.159	0.031	0.069	0.053	0.101	0.061	0.082
	SWformer-VO	0.101	0.099	**0.020**	0.038	**0.052**	0.097	0.052	0.066
	STDN-VO(ours)	**0.036**	**0.077**	0.022	**0.021**	0.067	**0.059**	**0.024**	**0.044**

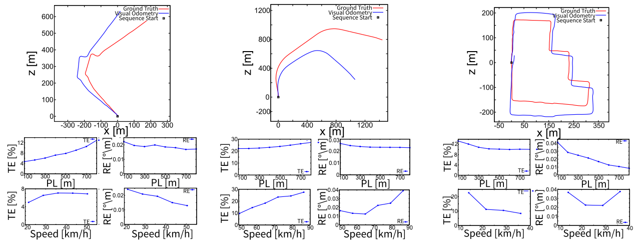
We present a comparison with representative VO methods. The first column enumerates the evaluation metrics, and the second column presents the names of VO methods, including traditional and deep learning methods. The subsequent columns showcase the performance across various test sequences, along with the computed average error. Bold: best result among the deep learning methods, underscore: second best result among the deep learning methods.



(a) Seq 09

(b) Seq 10

Fig. 3. Qualitative trajectory results on KITTI Odometry Sequences 09 and 10.



(a) Seq 11

(b) Seq 12

(c) Seq 15

Fig. 4. Qualitative trajectory results on KITTI Odometry Sequences 11-15. TE: Translation Error, RE: Rotation Error, PL: Path Length.

errors from 3.460, reported by Depth-VO-Feat [44], to 1.903 on sequence 10. Fig. 3 showcases the estimated trajectories.

Strategy 3: To conduct a thorough assessment the performance of STDN-VO, we trained the model on sequences $00-10$ and subsequently tested it on sequences $11-21$. The results were submitted to the KITTI official website for testing. On sequences 11 to 21, the model achieved an average translation error of 11.10 (%) and a rotation error of 0.023 (°/m). Fig. 4 illustrates the trajectories and corresponding error plots for

TABLE II
Quantitative Results on KITTI Odometry Sequences 09 and 10

Evaluation Index	Method	Sequence 09	Sequence 10	Avg. Error
t_{err}(%)↓	SFMlearner [39]	17.840	37.910	27.875
	Depth-VO-Feat	11.930	12.450	12.190
	GeoNet [40]	41.470	32.740	37.105
	Bian et al	11.200	10.100	10.650
	Masked GANs [41]	12.830	13.580	13.205
	Wang et al	9.880	12.240	11.060
	SWformer-VO	12.114	8.927	10.520
	STDN-VO(ours)	**6.784**	**7.730**	**7.257**
	STDN-VO(w/o T)	42.049	37.979	40.014
	STDN-VO(w/o S)	7.066	8.917	7.992
	Architecture A	53.829	46.095	49.962
	Architecture B	9.094	13.148	11.121
r_{err}(°/100m)↓	SFMlearner	6.780	17.780	12.280
	Depth-VO-Feat	3.910	3.460	3.685
	GeoNet	13.140	13.120	13.130
	Bian et al	3.340	4.960	4.150
	Masked GANs	3.870	4.330	4.100
	Wang et al	3.400	5.200	4.300
	SWformer-VO	4.596	5.190	4.893
	STDN-VO(ours)	**2.491**	**1.903**	**2.197**
	STDN-VO(w/o T)	16.223	17.097	16.660
	STDN-VO(w/o S)	2.663	3.598	3.131
	Architecture A	17.365	24.050	20.708
	Architecture B	3.834	7.041	5.438

Bold: best, underscore: second best.

TABLE III
Comparison of Tnference Times for Advanced Methods on the KITTI Dataset

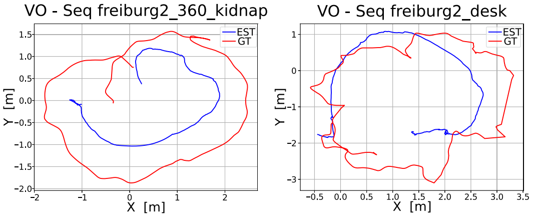
Methods	Avg. Time(s)
DeepVO	0.0587
TSformer-VO	0.0140
SWformer-VO	0.0186
STDN-VO(ours)	0.0389

sequences 11, 12, and 15, which are exclusively shown on the KITTI website. The outcomes for each sequence consist of:

1) A trajectory graph that contrasts the ground truth trajectory (marked by a red line) with the predicted trajectory (marked by a blue line).
2) Errors are measured in percent (for translation) and in degrees per meter (for rotation) across different trajectory lengths and driving speeds.

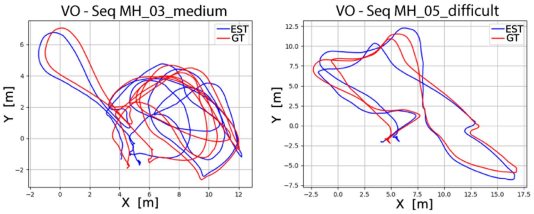
For sequences 11 and 12, the translation error tends to escalate with longer path lengths and higher speeds. In contrast, sequence 15 exhibits a reduction in translation error as both path length and speed increase. Meanwhile, the rotation error generally demonstrates a decreasing trend with increased path length across these sequences.

Inference time: In our comparative analysis of the real-time performance in visual odometry, we focused on the average inference times of DeepVO, TSformer-VO, SWformer-VO, and STDN-VO methods during pose estimation on the KITTI dataset. The results, as presented in Table III, reveal that although STDN-VO does not match the inference speed of TSformer-VO and SWformer-VO, it outperforms DeepVO. Notably, STDN-VO achieves highest accuracy among these methods, despite a trade-off in terms of inference speed.



(a) fr2/360 kidnap (b) fr2/desk

Fig. 5. Qualitative trajectory results on TUM Sequences ‘fr2/360 kidnap’ and ‘fr2/desk’.



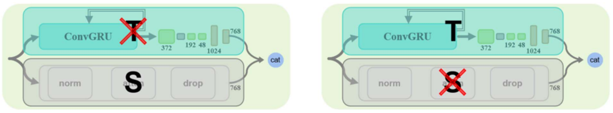
(a) MH_03_medium (b) MH_05_difficult

Fig. 6. Qualitative trajectory results on EuRoC Sequences ‘MH_03_medium’ and ‘MH_05_difficult’.

2) Evaluation on the TUM and EuRoC datasets: In order to further assess the generalization ability of STDN-VO, we tested its performance on the TUM and EuRoC datasets. The TUM dataset was collected by hand-held cameras, capturing data under challenging conditions in indoor environments. Additionally, the EuRoC dataset contains 11 sequences captured by a MAV in an indoor environment. These sequences are divided into three levels of difficulty, with each level being characterized by motion patterns and lighting conditions. We evaluate the performance of STDN-VO in terms of ATE (in meters). For the TUM dataset, we trained STDN-VO on the same train split as in Xue et al [45]. Subsequently, we tested STDN-VO on sequence ‘fr2/360 kidnap’ and ‘fr2/desk’. The ATE values for these test sequences are $0.745\ m$ and $1.031\ m$, respectively. Fig. 5 illustrates the estimated trajectories. As to the EuRoC dataset, we conducted tests on the ‘MH_03_medium’ and ‘MH_05_difficult’ sequences, utilizing the remaining sequences for training. The corresponding ATE values for these test sequences are 1.115 m and 1.310 m, respectively. Fig. 6 presents the estimated trajectories.

D. Ablation Study

Both streams are indispensable in STDN-VO: To explore the influence of each stream on the performance of STDN-VO, we respectively remove one stream module from the dual-stream architecture, while keeping the remaining stream untouched (Fig. 7): (1) STDN-VO(w/o S), denoting STDN-VO without the spatial stream, endures a slight decline in performance (Table II).



(a) Removing the temporal stream module

(b) Removing the spatial stream module

Fig. 7. Single stream archetectures.

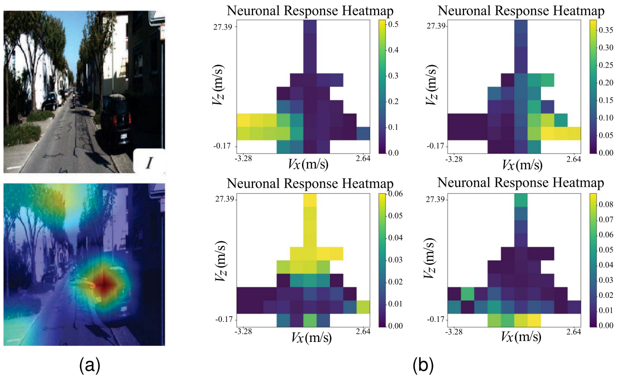
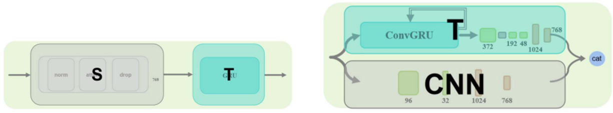


Fig. 8. **Visualization and analysis.** (a) The learned spatial attention in ViT. The more red, the higher the computed attention. (b) The responses of four example neurons at the output of the ConvGRU in velocity space. V_x and V_z: the velocity in the left-right wand forward-backward directions. The more light, the higher the response of the neuron.

To delve deeper into the spatial stream's contribution to performance, we visualize the spatial attention in ViT (Fig. 8(a)), utilizing the Attention Rollout introduced in [46]. Here, the learned attention focuses on the car and other critical spatial areas, while ignoring the less relevant details. These results demonstrate the importance of effective spatial information in enhancing the performance of visual odometry estimation. (2) STDN-VO(w/o T), referring to STDN-VO without the temporal stream, suffers from a considerable decline in performance (Table II). To further investigate the temporal stream's contribution to performance, we visualize the typical response patterns of the output neurons of ConvGRU as functions of velocities in space (Fig. 8(b)). The activities of the output neurons of ConvGRU are selective to particular movement velocities. Given that velocity is the cause of the changes between consecutive frames, Fig. 8(b) highlights the capacity of the temporal stream to capture sequential dependencies between images. These results demonstrate that VO significantly relies on the sequential dependencies between consecutive frames. Therefore, the performance of VO is influenced by effective spatial and temporal information, with the synergy between these two streams being a pivotal characteristic of the STDN-VO model.

Comparison of dual-stream and sequential integration architecture: To assess the influence of different integration strategies of the two steams on VO, we conducted experiments to compare with sequential integration architecture (Fig. 9(a)). STDN-VO adopts parallel integration architecture (Fig. 1(b)), processing both temporal and spatial information at the same time. The sequential integration architecture (architecture A in Fig. 9(a)) provides an alternative solution, in which the output generated by the Transformer is seamlessly channeled into the GRU as its input. As illustrated in Table II, the parallel architecture of STDN-VO markedly outperforms architecture A. We calculated the average Structural Similarity Index (SSIM) for the features of the two consecutive frames fed into the GRU module. In Architecture A, the average SSIM values were 0.990 and 0.989 for sequences 09 and 10, respectively. In contrast, STDN-VO exhibited SSIM values of 0.868 and 0.873 for the same sequences. The GRU in Architecture A is not sufficiently sensitive to the subtle differences between highly similar features, which could be leading to its poor performance. The results highlight the effectiveness of the dual-stream architecture, which empowers the model to capture both temporal and spatial information simultaneously, achieving a more robust integration of information.



(a) Architecture A, sequential integration

(b) Architecture B, replacing ViT with a CNN

Fig. 9. Archetectures with sequential processing (a) or with degraded spatial stream (b).

Degrading of the spatial stream: ViT, empowered by its self-attention mechanism, excels at capturing long-range dependencies and spatial information. We explored the influence of degraded spatial stream by replacing ViT with a CNN (Fig. 9(b)), which is good at capturing local information in object level. The CNN module comprises two convolutional layers, each with a kernel size of 3, followed by two FC layers. All of these layers are accompanied by a ReLU activation function. Experimental results, as detailed in Table II, reveal that the ViT module surpasses the performance of the CNN. This finding underscores the advantages of the ViT in acquiring spatial information over CNNs in VO tasks.

V. Conclusion

In this letter, we harness a dual stream architecture, mimicking the human visual system, to tackle monocular visual odometry. The proposed model, STDN-VO, employs Swim Transformer as the feature extractor and two parallel streams, i.e. spatial stream and temporal stream, to capture spatiotemporal correlation in image sequences. Subsequently, the outputs of the spatial stream and the temporal stream are concatenated and fed to a pose decoder to predict the 6-DoF relative pose. Experiments conducted on the standard KITTI, TUM and EuRoC datasets confirm the effectiveness of STDN-VO, showing robustness and generalization ability. Compared with other recent deep learning VO methods, STDN-VO achieved superior performance. In addition, ablation studies demonstrated the important role played by the dual stream architecture in STDN-VO. In future research endeavors, a focused effort will be dedicated to optimizing inference times and simultaneously improving the accuracy of the model.

REFERENCES

[1] J. Cheng, L. Zhang, Q. Chen, X. Hu, and J. Cai, "A review of visual SLAM methods for autonomous driving vehicles," *Eng. Appl. Artif. Intell.*, vol. 114, 2022, Art. no. 104992.

[2] P.-E. Sarlin et al., "LaMAR: Benchmarking localization and mapping for augmented reality," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 686–704.

[3] S. Hong, A. Bangunharcana, J.-M. Park, M. Choi, and H.-S. Shin, "Visual SLAM-based robotic mapping method for planetary construction," *Sensors*, vol. 21, no. 22, 2021, Art. no. 7715.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[5] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[6] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[7] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning–based text classification: A comprehensive review," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–40, 2021.

[8] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, 2019.

[9] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 2043–2050.

[10] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen, "VidLoc: A deep spatio-temporal model for 6-DoF video-clip relocalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6856–6864.

[11] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 627–637.

[12] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "VINet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1.

[13] J. Tang, J. Folkesson, and P. Jensfelt, "Geometric correspondence network for camera motion estimation," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1010–1017, Apr. 2018.

[14] M. Turan, Y. Almalioglu, H. Araujo, E. Konukoglu, and M. Sitti, "Deep EndoVO: A recurrent convolutional neural network (RCNN) based visual odometry approach for endoscopic capsule robots," *Neurocomputing*, vol. 275, pp. 1861–1870, 2018.

[15] G. Costante and M. Mancini, "Uncertainty estimation for data-driven visual odometry," *IEEE Trans. Robot.*, vol. 36, no. 6, pp. 1738–1757, Dec. 2020.

[16] N. Kaygusuz, O. Mendez, and R. Bowden, "MDN-VO: Estimating visual odometry with confidence," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 3528–3533.

[17] Z. Wu and Y. Zhu, "SWformer-VO: A monocular visual odometry model based on swin transformer," *IEEE Robot. Automat. Lett.*, vol. 9, no. 5, pp. 4766–4773, May 2024.

[18] A. O. Françani and M. R. Maximo, "Transformer-based model for monocular visual odometry: A video understanding approach," *IEEE Access*, vol. 13, pp. 13959–13971, 2025.

[19] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?," in *Proc. Int. Conf. Mach. Learn.*, 2021, vol. 2, no. 3, pp. 813–824.

[20] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[21] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How does the brain solve visual object recognition?," *Neuron*, vol. 73, no. 3, pp. 415–434, 2012.

[22] C. Galletti and P. Fattori, "The dorsal visual stream revisited: Stable circuits or dynamic pathways?," *Cortex*, vol. 98, pp. 203–217, 2018.

[23] M. A. Goodale and A. D. Milner, "Separate visual pathways for perception and action," *Trends Neurosci.*, vol. 15, no. 1, pp. 20–25, 1992.

[24] A. Dosovitskiy, "An image is worth 16 x 16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[25] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. 16th Eur. Conf. Comput. Vis.*,2020, pp. 402–419.

[26] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2938–2946.

[27] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.

[28] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, vol. 27.

[29] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1933–1941.

[30] Y. Peng, Y. Zhao, and J. Zhang, "Two-stream collaborative learning with spatial-temporal attention for video classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 773–786, Mar. 2019.

[31] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[32] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[33] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[34] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32.

[35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[36] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," *Proc. SPIE*, vol. 11006, pp. 369–386, 2019.

[37] H. Zhan, C. S. Weerasekera, J.-W. Bian, and I. Reid, "Visual odometry revisited: What should be learnt?," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4203–4210.

[38] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[39] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1851–1858.

[40] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1983–1992.

[41] C. Zhao, G. G. Yen, Q. Sun, C. Zhang, and Y. Tang, "Masked GAN for unsupervised depth and pose prediction with scale consistency," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5392–5403, Dec. 2021.

[42] R. Wang, S. M. Pizer, and J.-M. Frahm, "Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5555–5564.

[43] J. Bian et al., "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32.

[44] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 340–349.

[45] F. Xue, X. Wang, S. Li, Q. Wang, J. Wang, and H. Zha, "Beyond tracking: Selecting memory and refining poses for deep visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8575–8583.

[46] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2020, pp. 4190–4197, doi: 10.18653/v1/2020.acl-main.385.