BRIGHT-VO: Brightness-Guided Hybrid Transformer for Visual Odometry with Multi-modality Refinement Module

Dongzhihan Wang, **Yang Yang**, **Xuyang Chen** and **Liang Xu**[*]
Shanghai University
{wdzhihan, yyangy, sebastianchen, liang-xu}@shu.edu.cn

Abstract

Visual odometry (VO) plays a crucial role in autonomous driving, robotic navigation, and other related tasks by estimating the position and orientation of a camera based on visual input. Significant progress has been made in data-driven VO methods, particularly those leveraging deep learning techniques to extract image features and estimate camera poses. However, these methods often struggle in low-light conditions because of the reduced visibility of features and the increased difficulty of matching keypoints. To address this limitation, we introduce BrightVO, a novel VO model based on Transformer architecture, which not only performs front-end visual feature extraction, but also incorporates a multi-modality refinement module in the back-end that integrates Inertial Measurement Unit (IMU) data. Using pose graph optimization, this module iteratively refines pose estimates to reduce errors and improve both accuracy and robustness. Furthermore, we create a synthetic low-light dataset, KiC4R, which includes a variety of lighting conditions to facilitate the training and evaluation of VO frameworks in challenging environments. Experimental results demonstrate that BrightVO achieves state-of-the-art performance on both the KiC4R dataset and the KITTI benchmarks. Specifically, it provides an average improvement of 20% in pose estimation accuracy in normal outdoor environments and 25% in low-light conditions, outperforming existing methods. This work is open-source at https://github.com/Anastasiawd/BrightVO.

1 Introduction

With the rapid advances in artificial intelligence technologies, the application domains of autonomous vehicles and mobile robots [Liu *et al.*, 2024; Filipenko and Afanasyev, 2018] have broadened, encompassing a more diverse range of scenarios. As these systems increasingly operate in complex and dynamic environments, there is a growing emphasis on ensuring reliable localization, with a strong focus on improving the accuracy of pose estimation. Visual odometry (VO), a critical component in providing accurate and robust localization for autonomous systems, faces even greater challenges in such environments [He *et al.*, 2020; Gui *et al.*, 2015; Aqel *et al.*, 2016].

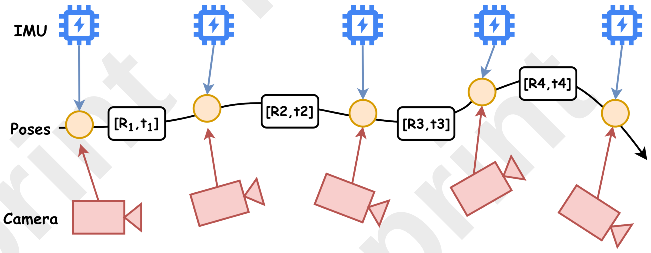


Figure 1: BrightVO estimates the motion of a camera using visual information, while also integrating IMU measurements to achieve more accurate pose estimation.

VO is the task to track a vehicle or robot's pose, i.e., position and orientation over time [Mohamed *et al.*, 2019]. Traditional localization techniques, such as GPS or LIDAR-based systems [Lin and Zhang, 2022; Cai *et al.*, 2019], often face limitations in urban canyons, indoor environments, or GPS-denied areas, where signal obstruction and degradation compromise their accuracy and reliability. In contrast, VO offers a solution to estimate the camera motion and localization, as it relies on visual sensors which are unaffected by such environmental constraints. However, VO methods often struggle in low-light environments [Alismail *et al.*, 2016; Agostinho *et al.*, 2022]. These environments, commonly encountered during nighttime or in areas with limited lighting, present difficulties for VO algorithms. Low-light conditions introduce issues such as poor feature visibility, increased sensor noise, and reduced contrast, all of which significantly affect the accuracy and robustness of traditional VO approaches. As a result, many existing systems are unable to maintain reliable localization or trajectory estimation when operating under such conditions.

To address these challenges, we propose BrightVO, a novel VO framework specifically designed to operate effectively in low-light environments shown in Figure 1. BrightVO uses a Transformer-based architecture [Vaswani, 2017], a cutting-edge approach in deep learning that excels in modeling long-

[*] Corresponding author.

range dependencies and complex feature relationships within images. The key advantage of Transformer models lies in their self-attention mechanism, which allows the model to focus selectively on the most relevant features of an image. Additionally, we introduce a brightness estimation module, which uses convolutional layers to extract brightness features from the image. This allows the Transformer to focus on the illumination information, thereby addressing the limitations of traditional methods that struggle to effectively extract features due to low image quality in low-light conditions. Furthermore, BrightVO's ability to integrate multi-modality data, such as Inertial Measurement Unit (IMU), further enhances its robustness. IMU provides complementary information to visual features, particularly in scenarios where visual input is noisy or sparse [Qin *et al.*, 2018; Huai and Huang, 2022]. By incorporating this measurement into a back-end based on graph optimization, which minimizes the cumulative drift and maintain consistency through iterative corrections, the model refines its pose estimates, improving accuracy over long sequences in challenging light conditions.

The main contributions of the paper can be summarized as follows:

1. We propose a brightness-guided Visual Transformer (ViT) [Dosovitskiy, 2020], which can learn the relative camera pose from multi-modality inputs through end-to-end training.
2. We design a back-end refinement block, using graph optimization with IMU inputs to iteratively improve the pose estimation accuracy. Experiments show that we achieve an average improvement of 20% pose estimation accuracy in normal outdoor scenes and 25% in low-light conditions compared to other methods.
3. We create a low-light scene dataset using the CARLA [Dosovitskiy *et al.*, 2017] simulator, which can be used for training and evaluating various VO frameworks.

2 Related Work

2.1 Monocular Visual Odometry

Monocular visual odometry is a technique used to estimate the motion trajectory of a camera in a 3D space from a sequence of images captured by a single camera. It relies on computer vision algorithms to calculate the camera's pose by analyzing changes between consecutive image frames. Currently, there are two main approaches to monocular visual odometry: traditional geometry-based methods and deep learning-based methods.

Geometry-based methods such as ORB-SLAM3 [Campos *et al.*, 2021] extract key feature points from images and perform feature matching between consecutive frames, subsequently estimating camera motion based on geometric relationships. In contrast, methods such as DSO [Wang *et al.*, 2017a] and LSD-SLAM [Engel *et al.*, 2014] do not rely on feature points; instead, they directly utilize pixel intensity differences to estimate camera motion. These approaches optimize camera poses by minimizing the photometric error between adjacent frames. However, due to the presence of noise and errors, the accumulated drift may increase over time, potentially leading to progressively inaccurate pose estimations.

In recent years, deep learning-based methods such as DeepVO [Wang *et al.*, 2017b] and TartanVO [Wang *et al.*, 2021] have utilized deep neural networks to extract more robust features. These methods typically employ an end-to-end approach, training neural network models to directly estimate the relative pose of the camera from input images or image pairs. However they also exhibit greater adaptability to challenges such as challenging lighting conditions.

2.2 VO Under Low-Light Condition

Challenging lighting conditions present significant challenges to VO. Existing methods for these conditions adopt image enhancement algorithms prior to improve image rightness and enrich image detail features, thereby enhancing the accuracy of VO. Light-SLAM [Zhao *et al.*, 2024] replaces traditional handcrafted features with LightGlue [Sarlin *et al.*, 2020] network to improve feature extraction in dark environments [Burri *et al.*, 2016]. [Zhang *et al.*, 2018] combines VO with 3D point cloud technology to enhance scene understanding under low-light conditions. However, these approaches still rely heavily on local feature matching, which may not fully capture long-range dependencies and complex contextual relationships in images. Also, Light-SLAM has not been open-sourced, limiting reproducibility and preventing us from directly evaluating its performance.

2.3 Transformers-Based VO

In recent years, Transformer models have shown exceptional performance in both natural language processing and computer vision fields. Consequently, end-to-end VO methods can leverage the self-attention mechanism of Transformers to effectively capture global features in images, enhancing the temporal information processing in complex scenes. [Han *et al.*, 2020; Wu *et al.*, 2020].

TSformer-VO [Françani and Maximo, 2023] proposes an end-to-end Transformer-based architecture for estimating 6 degrees of freedom (DoF) camera poses. Based on the TimeSformer [Bertasius *et al.*, 2021] model, it extracts features from image sequences through both spatial and temporal self-attention mechanisms. However, this method achieves reduced accuracy on the KITTI [Geiger *et al.*, 2012] dataset. [Memmel *et al.*, 2023] propose a Transformer model which is a modality-agnostic. Experimental results indicate the model achieves robust performance in indoor navigation tasks.

These models successfully integrate Transformer technology into VO and demonstrate superior results on common datasets compared to traditional methods. However, these models were not specifically designed to address the challenges posed by low-light conditions. Additionally, existing approaches cannot incorporate traditional mathematical optimization methods [Martínez-Otzeta *et al.*, 2022; Carlone *et al.*, 2015], potentially resulting in noticeable scale drift and a lack of effective correction in extended sequences.

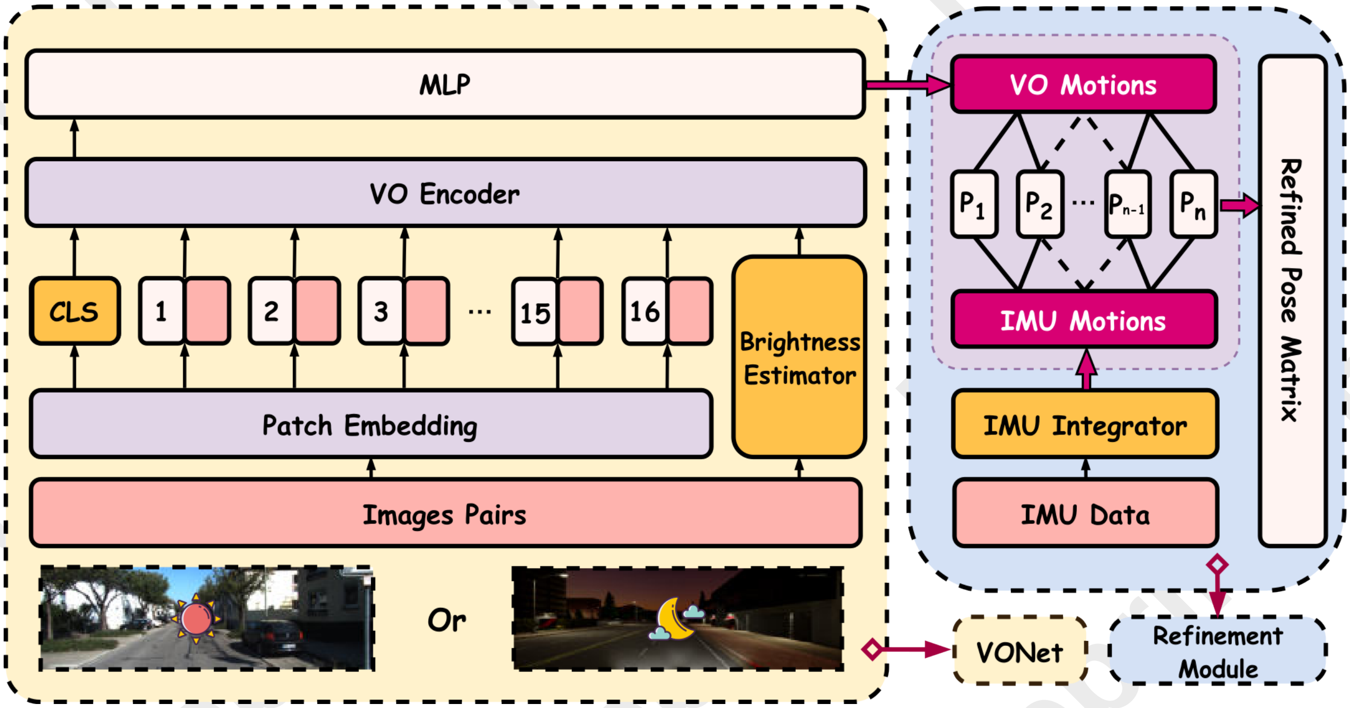


Figure 2: The pipeline of BrightVO, which begins with the input of consecutive image frames along with other modality information. In the VONet, images are transformed into vectors and passed into the Transformer Encoder after embedding. The global features are stored in the `cls_token`, which is then mapped to a 6-dimension vector via an MLP. This vector represents the output initial VO motion. In the Refinement Module, IMU measurements are integrated to motions in an integrator. The VO motion, along with IMU motion, is fed into the PGO module, where iterative optimization occurs, yielding a high-precision pose estimate.

3 Approaches

3.1 Preliminaries

The core task of visual odometry (VO) is to estimate the camera's position and pose changes. From an End-to-End (E2E) perspective, VO is data-driven and aims to automate the entire process of motion estimation from image input to output using neural networks. The input of Visual Odometry (VO) consists of a sequence of RGB images I_t and I_{t+1} captured by a monocular camera, along with other modality information such as IMU sensor data and depth maps. The output is a 6-DOF vector that includes both the translational vector $\mathbf{t}$ and the rotational matrix $\mathbf{R}$ information. The transformation from time t to time $t+1$ can be expressed as:

$$\mathbf{T}_{t\to t+1} = [\mathbf{R}_t \quad \mathbf{t}_t]\,, \tag{1}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix representing the rotation change of the camera from time t to $t+1$, $\mathbf{t}_t \in \mathbb{R}^3$ is the translation vector representing the translation change of the camera from time t to $t+1$. Thus, the entire process can be mathematically represented as:

$$f(I_t, I_{t+1}) \to \mathbf{T}_{t\to t+1} = [\mathbf{R}_t \quad \mathbf{t}_t]\,. \tag{2}$$

The goal of our model is to extract deep features from the input multi-modality information, solve for the camera's motion, and achieve superior performance in pose estimation under low-light conditions.

3.2 Model Architecture

Our pipeline consists of a front-end VOnet and a back-end refinement module, as shown in the Figure 2. The front-end VOnet is designed based on a basic Encoder-Decoder architecture. The encoder uses an enhanced ViT model to encode multi-modality input information and we propose a brightness-guided strategy in this part. The decoder is relatively simple, consisting of a multi-layer perceptron (MLP) that progressively extracts features and generates the final 6-DOF pose vector. The back-end refinement block uses Pose Graph Optimization to iteratively refine the pose estimation for more accurate results.

Transformer Encoder: The encoder is responsible for extracting features from the input image sequence and passing these features to the decoder. First, the input images are resized to 224 × 224 pixels, a size consistent with the pretrained model. Our experiments demonstrate that this resizing does not significantly impact the model's performance. The images are then divided into several 16 × 16 patches and passed through a convolutional layer, which transforms them into fixed-dimensional vectors. To preserve spatial information, the encoder adds positional encoding to each patch.

Next, a brightness estimator is used to extract brightness features from the image as illustrated in Figure 3, which are then combined with the image features and input into the Transformer model. This module is particularly effective at handling lighting variations. A similar module is also used

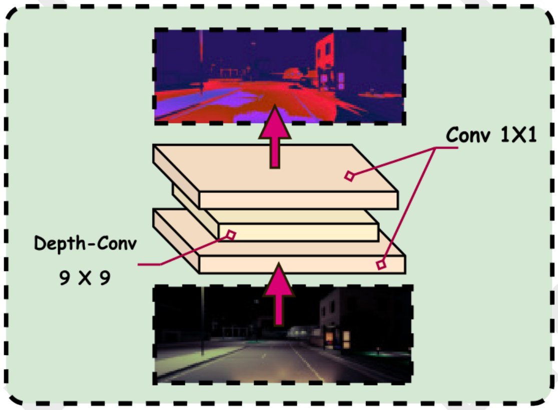


Figure 3: The overview of Brightness Estimator. The image at the bottom is the input. After passing through a network composed of two 1x1 convolutions and a 9x9 depth-wise convolution, we obtain the brightness feature map shown in the top image, where different RGB values represent different illumination intensities.

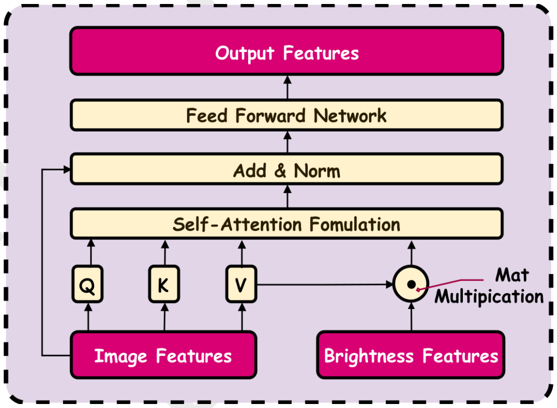


Figure 4: The overview pf Transformer Layer in the VO Encoder. After patch embedding, image features are used to generate Q (queries), K (keys), and V (values). These, combined with brightness information, are employed to compute attention scores. The self-attention mechanism is then applied, and the resulting hidden states are passed through a feed-forward network to produce the final output features.

in Retinexformer [Cai *et al.*, 2023] to recover a normally exposed image from a low-exposure image using a UNet-based architecture [Ronneberger *et al.*, 2015]. However, we believe that it is unnecessary to up-sample the image to a normally exposed version in the VO task as perceived by the human eye. Therefore, we simplify the module as follows to enhance the performance of pose estimation.

$$(\mathbf{I}_{\text{br}}, \mathbf{F}_{\text{br}}) = \mathcal{E}(\mathbf{I}, \mathbf{L}_p), \tag{3}$$

where $\mathcal{E}$ denotes the brightness estimator and $\mathbf{I}$ represents the resized images. $\mathcal{E}$ takes $\mathbf{I}$ and its brightness prior map $\mathbf{L}_p \in \mathbb{R}^{H \times W}$ as inputs. $\mathbf{L}_p = \text{mean}_c(\mathbf{I})$, mean_c indicates the operation that calculates the mean values for each pixel along the channel dimension. $\mathcal{E}$ outputs the brightness-enhanced image $\mathbf{I}_{\text{br}}$ and the brightness feature $\mathbf{F}_{\text{br}} \in \mathbb{R}^{H \times W \times C}$.

The encoder consists of several Transformer layers, as shown in the Figure 4, each containing a self-attention mechanism and a feed-forward network to capture complex features in the image. The final output of the encoder is a hidden state that contains both image and brightness features, which is used for subsequent pose estimation.

We also reshape $\mathbf{F}_{\text{br}}$ into $\mathcal{V} \in \mathbb{R}^{HW \times C}$ and then the self-attention is formulated as:

$$Atten(Q, K, V, \mathcal{V}) = (V \odot \mathcal{V})\ softmax(\frac{K^T Q}{\alpha}), \tag{4}$$

where $\alpha \in \mathbb{R}^1$ is a learnable parameter that adaptively scales the matrix multiplication. $Atten$ represents the self-attention mechanism.

As value (V) contains the actual feature information and represents the real content or information of each position in the feature space, this self-attention mechanism allows the model to weigh the importance of different parts of the input sequence, effectively capturing the relationships between patches and encoding complex image features.

Transformer Decoder: The decoder is designed to process the features extracted by the encoder and produce the final pose estimation. It follows a simple structure leveraging a multi-layer perceptron (MLP) and normalization techniques to refine the feature representation. The mathematical formulation of the decoder can be summarized as follows:

$$x' = DropPath(MLP(LayerNorm(x))), \tag{5}$$

$$\hat{T} = W_{out} \cdot x' + b_{out}, \tag{6}$$

where $W_{out} \in \mathbb{R}^{768 \times 6}$ is the weight matrix of the output layer, which transforms the feature vector x' into the final output pose vector. $b_{out} \in \mathbb{R}^6$ is the bias vector of the output layer, added after the matrix multiplication to provide additional flexibility. $\hat{T} \in \mathbb{R}^6$ is the final output, which is a 6-DOF pose vector including rotational and translational components.

Back-end refinement module: The refinement module in our approach is based on Pose Graph Optimization (PGO) [Fu *et al.*, 2024], where the goal is to optimize the trajectory estimates by minimizing errors across multiple sensor modalities. This back-end module enables two modalities with different error characteristics VO and IMU to mutually verify and correct each other through graph constraints, leading to a more precise and robust trajectory estimation.

In our approach, we reshape the pose estimated by VOnet into the SE(3) form to adapt to the optimization mechanism of PGO. The motion of the IMU is integrated from raw sensor data using an IMU integrator. Specifically, the IMU's motion is calculated by integrating the acceleration measurements, as well as by considering the positional estimates from Global Navigation Satellite System (GNSS) when available. The equations governing the motion of the IMU are as follows:

$$\Delta \mathbf{R}_{ik+1} = \Delta \mathbf{R}_{ik} \exp(\mathbf{w}_k \Delta t), \tag{7}$$

$$\Delta \mathbf{v}_{ik+1} = \Delta \mathbf{v}_{ik} + \Delta \mathbf{R}_{ik} \mathbf{a}_k \Delta t, \tag{8}$$

$$\Delta \mathbf{p}_{ik+1} = \Delta \mathbf{p}_{ik} + \Delta \mathbf{v}_{ik} \Delta t + \frac{1}{2} \Delta \mathbf{R}_{ik} \mathbf{a}_k \Delta t^2, \tag{9}$$

where $\Delta\mathbf{R}_{ik}$ is the pre-integrated rotation between the i-th and k-th time steps. $\Delta\mathbf{v}_{ik}$ is the pre-integrated velocity between the i-th and k-th time steps. $\Delta\mathbf{p}_{ik}$ is the pre-integrated position between the i-th and k-th time steps. $\mathbf{a}_k$ is the linear acceleration at the k-th time step. $\mathbf{w}_k$ is the angular velocity at the k-th time step. Δt is the time interval between the k-th and $(k+1)$-th time steps.

The last term, we also use GNSS data at each time step as correction. This correction ensures the position stays globally aligned with the real-world coordinates. Therefore, the final error of the VO motions $\mathbf{T}_{ij}$ and IMU poses $\mathbf{p}_{IMU}$ can be defined as the weighted summation of the two constraints:

$$\mathcal{L} = \sum_{(i,j)\in\mathcal{E}} \|\mathbf{p}_{ij} - \mathbf{T}_{ij}\|^2_{\Sigma_{ij}} + \lambda \sum_{(i,j)\in\mathcal{E}} \|\mathbf{p}_{ij} - \mathbf{p}_{IMU}\|^2_{\Sigma_{ij}}, \quad (10)$$

where $\mathcal{E}$ represents all image frames and $\mathbf{p}_{ij}$ is the relative pose between frames i and j, which is defined as a parameter and iteratively optimized during the PGO.

We then employ a Levenberg-Marquardt (LM) algorithm in PyPose [Wang *et al.*, 2023] to solve the PGO process, which leads to more accurate and consistent trajectory estimation and ensures robustness across a variety of scenarios, especially when GNSS data is available to correct for long-term drift.

4 Experiments

4.1 Datasets Preparation

KiC4R: To validate the performance of our proposed model under low-light conditions, we need a sufficiently large dataset that includes various lighting conditions for training and testing. However, to date, we have not been able to find a suitable dataset that meets these criteria. The low-light scenes in the TUM dataset [Keimel *et al.*, 2012] are limited to indoor environments, and TartanAir [Wang *et al.*, 2020] only provides the ”abandonedfactory_night” sequence as a low-light sequence, which is insufficient for training purposes. Many previous works [Rashed *et al.*, 2019] have addressed the lack of such datasets by simulating nighttime scenes using networks or capturing real-world data. However, we believe these methods are not the most efficient or comprehensive ways to obtain the necessary data.

To address this issue, we created a dataset called KiC4R using the CARLA simulator. The KiC4R dataset includes seven sequences i.e., 00-06 and features four types of low-light conditions: dusk, night, midnight, and extreme weather shown in Figure 5. We model the passage of time by adjusting the sun's direct angle in the simulator. When the angle is negative, the scene transitions to nighttime, with the time progressively changing, eventually reaching midnight. Additionally, vehicle headlights and streetlights are activated to simulate realistic nighttime street scenes. To emulate low-light conditions induced by extreme weather, we generated two sequences i.e., 00 and 03 by manipulating the cumulus cloud cover, precipitation, and road water accumulation, thereby simulating low-light scenarios under heavy rainfall.

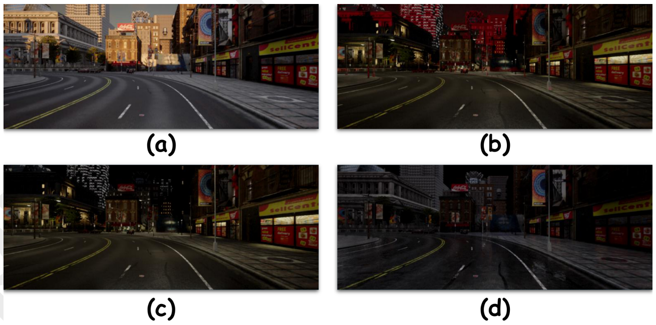


Figure 5: The overview of 4 light conditions in KiC4R. (a) Dusk. (b) Nighttime. (c) Mid-night. (d) Extreme weather.

SHIFT [Sun *et al.*, 2022] also provides a large-scale, multi-scenario, and multi-task dataset using CARLA . However, due to differences in data annotation formats, this dataset cannot be directly used for VO task training and evaluation, which posed some challenges for experimentation. For ease of comparison with other prominent works, KiC4R consists of both RGB and IMU sensor data and adopts the same annotation format as KITTI.

We ran the CARLA simulator on a Windows PC equipped with an NVIDIA RTX 4090 GPU. Sensor data was collected using CarlaScenes and the official CARLA Python API. To generate sensor data from the simulator, we selected maps 01-07 and 10 for urban scenes similar to those found in datasets such as *Virtual KITTI* [Cabon *et al.*, 2020].

We ultimately collected over 37,000 RGB images with a resolution of 512 × 1392 using the approaches described above. The corresponding maps, lighting conditions, and sequence lengths for these sequences are shown in the table 1.

Sequence	Map	Light Condition	Lengths
00	Town_01	E	5000
01	Town_01	D	3200
02	Town_02	M	4000
03	Town_03	E	8000
04	Town_06	D	7446
05	Town_07	N	8000
06	Town_10	N	2000

Table 1: Details of KiC4R sequences, where D, N, M, E represents of dusk, nighttime, mid-night and extreme weather and Lengths represent how many frames in the sequence.

4.2 Experiment Setup

To evaluate the effectiveness of our proposed method, we conducted experiments on the KITTI and KiC4R datasets. We compared our approach with several state-of-the-art frameworks, including ORB-SLAM2 [Mur-Artal and Tardós, 2017], ORB-SLAM3, DeepVO , TartanVO, DPVO [Teed *et al.*, 2024], and DPV-SLAM [Lipson *et al.*, 2025]. In our experiments, only the front-end VONet was trained. Specifically, we used sequences 01, 03, 07, and 08 from the KITTI dataset and sequences 04-06 from the KiC4R dataset as the training set. The inputs contain N monocular image pairs with resolution 512×1392. The outputs are N poses in SE(3).

Sequence	00	01	02	04	05	06	07	08	09	10	Avg
ORB-SLAM2	**1.3**	10.4	5.7	**0.2**	**0.8**	**0.8**	**0.5**	3.6	3.2	**1.0**	2.75
ORB-SLAM3	6.77	-	30.50	0.93	5.54	16.61	9.70	60.69	7.90	8.65	-
DeepVO	95.92	68.26	150.56	5.65	54.86	88.47	7.96	68.19	30.70	22.76	59.33
DPVO	111.97	12.69	123.40	0.68	58.96	54.78	19.26	115.90	75.10	13.63	58.64
DPV-SLAM	8.30	11.86	39.64	0.78	5.74	11.6	1.52	110.9	76.7	13.7	28.09
TSFormer-VO	46.52	160.55	55.24	3.06	61.38	88.31	31.49	26.46	23.68	22.70	51.93
Ours	2.12	**2.36**	**2.52**	0.44	2.31	2.7	2.04	**2.92**	**2.26**	2.11	**2.18**

Table 2: Absolute Trajectory Error (ATE) on 10 sequences on *KITTI* dataset, given in meters. Due to the absence of raw IMU data in sequence 3, we had to discard this sequence. ORB-SLAM 2, 3 are feature-based methods; DeepVO, TartanVO, DPVO and DPV-SLAM are traditional learning-based methods; TSFormer-VO and ours are transformer-based methods.

Sequence	06		07		09		10		Avg	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
TartanVO	4.72	2.95	4.32	3.41	6.00	3.11	6.89	2.73	5.48	3.05
DPV-SLAM	4.95	**0.16**	1.29	**0.24**	17.69	**0.23**	6.32	**0.23**	7.56	**0.22**
Ours	**1.35**	0.98	**1.00**	1.22	**1.31**	0.76	**1.27**	0.99	**1.23**	0.99

Table 3: Sence TartanVO only reports results for sequences 06,07,09,10 in relative pose error (RPE), here we compared our method with two advanced approaches using the t_{rel}/r_{rel} metrics.

Sequence	00		01		02		03		Avg	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
ORB-SLAM3	1.11	1.15	1.08	0.99	1.31	1.21	1.44	1.20	1.24	1.14
TartanVO	5.62	2.44	5.32	3.21	4.33	3.72	3.78	1.88	4.76	2.81
DPVO	1.29	0.97	1.32	0.98	1.28	0.89	1.31	**0.75**	1.30	0.90
Ours	**0.74**	**0.93**	**0.62**	**0.85**	**0.13**	**0.13**	**0.93**	0.94	**0.61**	**0.71**

Table 4: RPE on 4 sequences on KiC4R. Here we compared our method with ORB-SLAM3, TartanVO and DPVO using the t_{rel}/r_{rel} metrics.

For evaluation, we adpoted two widely used metrics: Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) [Prokhorov *et al.*, 2019]. Shown as follows:

$$ATE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i - \hat{\mathbf{p}}_i\|^2}, \tag{11}$$

$$RPE = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N-1} \|(\mathbf{T}_i^{-1}\mathbf{T}_{i+1}) - (\hat{\mathbf{T}}_i^{-1}\hat{\mathbf{T}}_{i+1})\|^2}, \tag{12}$$

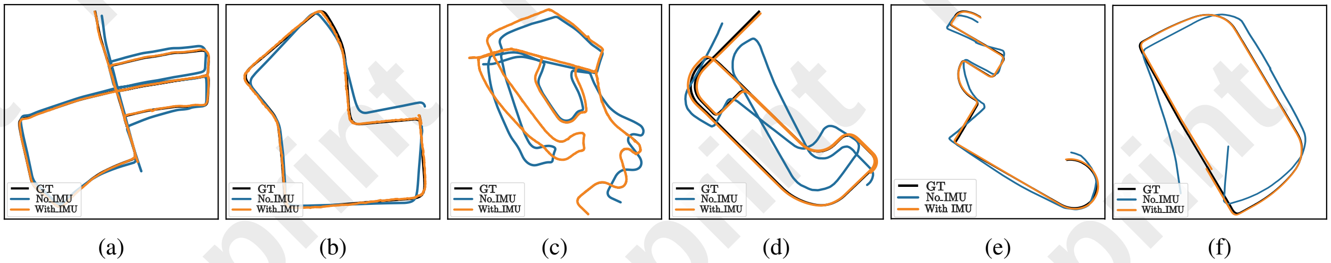
where $\mathbf{p}$ is the camera pose of each frame, i is the number of frame, $\mathbf{T}$ and $\hat{\mathbf{T}}$ represent the estimated and ground-truth translation vector.

To deal with the large data requirements of ViTs, we use the pre-trained model "vit-base-patch16-224" made publicly available by [Dosovitskiy, 2020]. We then trained and finetuned our model for 250 epochs on a single NVIDIA RTX 4090 GPU with a batch size of 12. The weights of the model are updated using AdamW optimizor with a learning rate of $1 * 10^{-4}$.

4.3 Experiment Results

We first conducted experiments on the KITTI dataset to validate the performance of our VO network under normal lighting conditions. The model was trained for 40 hours using sequences 01, 03, 07, and 08. Following this, we evaluated the model across all sequences, utilizing the evo tool to align the results, recover the scale, and compute the ATE and RPE. These results were then compared against several state-of-the-art methods, including ORB-SLAM2, ORB-SLAM3, TartanVO, DPVO, DPV-SLAM, and TSFormer-VO. Notably, DPVO represents an improvement over DROID-SLAM [Teed and Deng, 2021], which is widely recognized as a state-of-the-art method. The results are presented in Table 2 and 3.

The experiment results demonstrate that while ORB-SLAM2 achieves minimal ATE in certain sequences (e.g., 04-07), BrightVO consistently outperforms it, achieving a 20% improvement in average ATE over all 10 sequences. In comparison to methods lacking backend optimization, such as TartanVO, DeepVO, and TSFormer-VO, BrightVO demonstrates a substantial reduction in error nearly 96%. Furthermore, in comparison to DPV-SLAM, which also includes mapping process, BrightVO achieves superior performance, benefiting from its multi-modalaity refinement approach. These results collectively underscore that our model



(a) (b) (c) (d) (e) (f)

Figure 6: The overview of trajectory plots with/without IMU measurements. (a)-(c) represent sequence 02, 05, 07 on KITTI dataset. (d)-(f) represent sequence 00-02 on KiC4R dataset.

not only functions effectively under normal lighting conditions but also delivers state-of-the-art pose estimation accuracy.

We then conduct experiments on the KiC4R dataset to assess whether our method enhances VO performance in low-light scenarios. The test set comprises four sequences i.e., 00-03 from the KiC4R dataset, containing a total of 20,200 images after excluding frames affected by camera shake. For evaluation, we computed the RPE on every 100 meters for each sequence and compared our results with TartanVO, ORB-SLAM3, and DPVO. Our experimental results demonstrate that BrightVO consistently achieves smaller relative errors across all four sequences, outperforming existing approaches as shown in Table 4. Our method decreases the average relative error by approximately 50% compared with state-of-the-art methods. We attribute this significant improvement to the use of longer sequences and more extreme lighting conditions, where BrightVO benefits from the robust feature extraction capabilities of the Brightness-Guided ViT for long sequences, coupled with back-end optimization that minimizes drift during extended operations.

4.4 Ablation Study

Modality independent: In real-world conditions, IMU and GNSS measurments may fail when GNSS signals are obstructed or completely lost, which prevents GNSS from providing accurate positional corrections. Similarly, IMU data can be compromised due to factors like high vibrations, rapid accelerations, or sensor malfunctions, leading to inaccuracies or even complete data loss. Since the KiC4R dataset does not include GNSS data for correction, the experiments conducted in the previous section have already demonstrated that BrightVO can still achieve robust estimation accuracy in the absence of GNSS corrections. To further assess BrightVO's performance without IMU data, we removed the refinement module while keeping all other settings unchanged, and conducted experiments on the KITTI and KiC4R datasets. The results in Figure 6 show that, in the absence of the back-end refinement module, BrightVO experienced significant drift in both normal and low-light conditions. This highlights the critical role of the refinement module in ensuring the stability and accuracy of BrightVO. In addition, as shown in Figure 7, in shorter sequences, such as 03, 04, and 07, even when the back-end refinement module is removed, BrightVO demonstrates minimal estimation errors. This suggests that our model is capable of maintaining reliable estimation accuracy, even in short-term absence of IMU measurements.

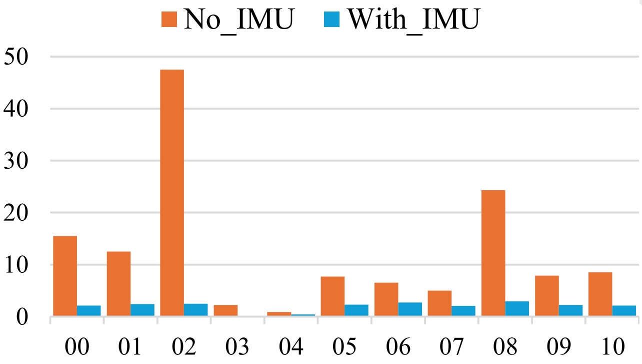


Figure 7: The illustration of ATE on KITTI sequences with/without IMU measurements. Due to the absence of IMU raw data on sequence 03, the figure only contains result without IMU inputs.

5 Conclusion

In this paper, we propose BrightVO, a model designed to enhance accuracy in VO under low-light conditions. Our model consists of a ViT-based VO network and an optimization module that integrates multi-modality information. By combining visual information with IMU data, BrightVO significantly improves pose estimation accuracy under extreme lighting conditions.

We conducted extensive experiments on the KiC4R and KITTI datasets and the experimental results demonstrate that BrightVO outperforms existing VO methods, achieving state-of-the-art performance in various environments. We also designed several ablation experiments which confirmed that BrightVO still maintain good estimation accuracy even with short-term IMU data loss.

Ultimately, BrightVO not only excels in low-light scenarios but also operates stably under normal environmental conditions, demonstrating broad application potential. Our research provides a new perspective for data-driven VO tasks and offers strong support for future applications in autonomous driving, robotics, and other fields.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grant 62373239. Specially, we would like to thank Dr. Chen Wang for their technical support.

References

[Agostinho *et al.*, 2022] Lucas R Agostinho, Nuno M Ricardo, Maria I Pereira, Antoine Hiolle, and Andry M Pinto. A practical survey on visual odometry for autonomous driving in challenging scenarios and conditions. *IEEE Access*, 10:72182–72205, 2022.

[Alismail *et al.*, 2016] Hatem Alismail, Michael Kaess, Brett Browning, and Simon Lucey. Direct visual odometry in low light using binary descriptors. *IEEE Robotics and Automation Letters*, 2(2):444–451, 2016.

[Aqel *et al.*, 2016] Mohammad OA Aqel, Mohammad H Marhaban, M Iqbal Saripan, and Napsiah Bt Ismail. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, 5:1–26, 2016.

[Bertasius *et al.*, 2021] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[Burri *et al.*, 2016] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[Cabon *et al.*, 2020] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020.

[Cai *et al.*, 2019] Guo-Sheng Cai, Huei-Yung Lin, and Shih-Fen Kao. Mobile robot localization using gps, imu and visual odometry. In *2019 International Automatic Control Conference (CACS)*, pages 1–6. IEEE, 2019.

[Cai *et al.*, 2023] Yuanhao Cai, Hao Bian, Jing Lin, Haoqian Wang, Radu Timofte, and Yulun Zhang. Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12504–12513, 2023.

[Campos *et al.*, 2021] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[Carlone *et al.*, 2015] Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. Initialization techniques for 3d slam: A survey on rotation estimation and its use in pose graph optimization. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4597–4604. IEEE, 2015.

[Dosovitskiy *et al.*, 2017] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[Dosovitskiy, 2020] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Engel *et al.*, 2014] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[Filipenko and Afanasyev, 2018] Maksim Filipenko and Ilya Afanasyev. Comparison of various slam systems for mobile robot in an indoor environment. In *2018 International Conference on Intelligent Systems (IS)*, pages 400–407. IEEE, 2018.

[Françani and Maximo, 2023] André O Françani and Marcos ROA Maximo. Transformer-based model for monocular visual odometry: a video understanding approach. *arXiv preprint arXiv:2305.06121*, 2023.

[Fu *et al.*, 2024] Taimeng Fu, Shaoshu Su, Yiren Lu, and Chen Wang. islam: Imperative slam. *IEEE Robotics and Automation Letters*, 2024.

[Geiger *et al.*, 2012] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[Gui *et al.*, 2015] Jianjun Gui, Dongbing Gu, Sen Wang, and Huosheng Hu. A review of visual inertial odometry from filtering and optimisation perspectives. *Advanced Robotics*, 29(20):1289–1301, 2015.

[Han *et al.*, 2020] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on visual transformer. *arXiv preprint arXiv:2012.12556*, 2020.

[He *et al.*, 2020] Ming He, Chaozheng Zhu, Qian Huang, Baosen Ren, and Jintao Liu. A review of monocular visual odometry. *The Visual Computer*, 36(5):1053–1065, 2020.

[Huai and Huang, 2022] Zheng Huai and Guoquan Huang. Robocentric visual–inertial odometry. *The International Journal of Robotics Research*, 41(7):667–689, 2022.

[Keimel *et al.*, 2012] Christian Keimel, Arne Redl, and Klaus Diepold. The tum high definition video datasets. In *2012 Fourth international workshop on quality of multimedia experience*, pages 97–102. IEEE, 2012.

[Lin and Zhang, 2022] Jiarong Lin and Fu Zhang. R 3 live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10672–10678. IEEE, 2022.

[Lipson *et al.*, 2025] Lahav Lipson, Zachary Teed, and Jia Deng. Deep patch visual slam. In *European Conference on Computer Vision*, pages 424–440. Springer, 2025.

[Liu *et al.*, 2024] Xiaorui Liu, Zijie Li, Weihua Zong, Hang Su, Peng Liu, and Shuzhi Sam Ge. Graph representation learning and optimization for spherical emission source microscopy system. *IEEE Transactions on Automation Science and Engineering*, pages 1–14, 2024.

[Martínez-Otzeta *et al.*, 2022] José María Martínez-Otzeta, Itsaso Rodríguez-Moreno, Iñigo Mendialdua, and Basilio Sierra. Ransac for robotic applications: A survey. *Sensors*, 23(1):327, 2022.

[Memmel *et al.*, 2023] Marius Memmel, Roman Bachmann, and Amir Zamir. Modality-invariant visual odometry for embodied vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21549–21559, 2023.

[Mohamed *et al.*, 2019] Sherif AS Mohamed, Mohammad-Hashem Haghbayan, Tomi Westerlund, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. A survey on odometry for autonomous navigation systems. *IEEE access*, 7:97466–97486, 2019.

[Mur-Artal and Tardós, 2017] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[Prokhorov *et al.*, 2019] David Prokhorov, Dmitry Zhukov, Olga Barinova, Konushin Anton, and Anna Vorontsova. Measuring robustness of visual slam. In *2019 16th International conference on machine vision applications (MVA)*, pages 1–6. IEEE, 2019.

[Qin *et al.*, 2018] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE transactions on robotics*, 34(4):1004–1020, 2018.

[Rashed *et al.*, 2019] Hazem Rashed, Mohamed Ramzy, Victor Vaquero, Ahmad El Sallab, Ganesh Sistu, and Senthil Yogamani. Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[Sarlin *et al.*, 2020] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[Sun *et al.*, 2022] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. Shift: a synthetic driving dataset for continuous multi-task domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21371–21382, 2022.

[Teed and Deng, 2021] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

[Teed *et al.*, 2024] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *Advances in Neural Information Processing Systems*, 36, 2024.

[Vaswani, 2017] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[Wang *et al.*, 2017a] Rui Wang, Martin Schworer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *Proceedings of the IEEE international conference on computer vision*, pages 3903–3911, 2017.

[Wang *et al.*, 2017b] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2043–2050. IEEE, 2017.

[Wang *et al.*, 2020] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020.

[Wang *et al.*, 2021] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. Tartanvo: A generalizable learning-based vo. In *Conference on Robot Learning*, pages 1761–1772. PMLR, 2021.

[Wang *et al.*, 2023] Chen Wang, Dasong Gao, Kuan Xu, Junyi Geng, Yaoyu Hu, Yuheng Qiu, Bowen Li, Fan Yang, Brady Moon, Abhinav Pandey, et al. Pypose: A library for robot learning with physics-based optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22024–22034, 2023.

[Wu *et al.*, 2020] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020.

[Zhang *et al.*, 2018] Hongmou Zhang, Ines Ernst, Sergey Zuev, Anko Börner, Martin Knoche, and Reinhard Klette. Visual odometry and 3d point clouds under low-light conditions. In *2018 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2018.

[Zhao *et al.*, 2024] Zhiqi Zhao, Chang Wu, Xiaotong Kong, Zejie Lv, Xiaoqi Du, and Qiyan Li. Light-slam: A robust deep-learning visual slam system based on lightglue under challenging lighting conditions. *arXiv preprint arXiv:2407.02382*, 2024.