SuperEIO: Self-Supervised Event Feature Learning for Event Inertial Odometry

Peiyu Chen, Fuling Lin, *Graduate Student Member, IEEE*, Weipeng Guan, Yi Luo, and Peng Lu, *Member, IEEE*

***Abstract*—Event cameras asynchronously output low-latency event streams, promising for state estimation in complex conditions. The motion-dependent nature of event cameras presents persistent challenges in achieving robust event feature detection and matching. Recent learning-based approaches have demonstrated superior robustness over traditional handcrafted methods, particularly under aggressive motion and HDR scenarios. This article proposes SuperEIO, a novel framework that leverages a learning-based event-only detector and IMU measurements for event-inertial odometry. Our event-only feature detector employs a convolutional neural network on continuous event streams, while a graph neural network achieves event descriptor matching for loop closure. We accelerate network inference with TensorRT, ensuring low-latency, real-time operation on resource-constrained devices. Extensive evaluations on multiple public benchmarks demonstrate its superior accuracy and robustness compared with the advanced event-based methods. Moreover, we conduct a large-scale real-world experiment on an edge handheld platform to demonstrate long-term effectiveness. Our pipeline is open-sourced to facilitate research in the field: https://github.com/arclab-hku/SuperEIO.**

***Index Terms*—Event camera, deep learning, sensor fusion, visual-inertial odometry.**

I. Introduction

EVENT cameras, inspired by biological vision systems, asynchronously capture pixel-level intensity changes rather than producing fixed-rate image frames like conventional cameras [1]. This event-driven design minimizes temporal redundancy, significantly reducing power consumption and bandwidth requirements. With microsecond-level temporal resolution and a 140 dB high dynamic range (HDR), event cameras excel in challenging scenarios such as high-speed motion and HDR illumination. These distinctive properties have significantly advanced event-based vision research, especially for applications demanding low-latency and high efficiency, including high-speed motion estimation [2], underwater darkness localization [3], and dynamic-scene stereo imaging [4].

Numerous traditional handcrafted feature detection methods [5], [6], [7] have been proposed for event streams. However, these detected event features are often constrained by issues such as low distinctiveness, limited repeatability, and high redundancy. Recently, deep learning techniques have been extensively applied in traditional visual tasks such as feature detection [8], descriptor matching [9], and visual odometry (VO) [10]. Although image-based convolutional and graph neural network (CNN/GNN) architectures are well-established, directly applying them to event streams often yields limited performance. This discrepancy stems from the fundamental difference between the asynchronous nature of event streams and the synchronous nature of traditional images. Therefore, most existing event-based odometry systems still rely on traditional methods [11], [12], [13], and how to effectively configure and train learning-based networks for the event domain remains underdeveloped.

Recent research indicates that learning-based odometry can provide more robust and accurate estimations on both frame-based and event-based cameras compared with the nonlearning-based approaches [10], [14]. However, these improvements in accuracy are achieved at the cost of increased resource consumption. To tackle this and enhance the performance of traditional event-inertial odometry (EIO), we integrate deep networks to augment it with a more distinctive event feature detector for tracking and a learning-based descriptor matcher for loop closure in event domains. In addition, we optimize the proposed detector and matcher by leveraging TensorRT to accelerate the network, achieving real-time performance and a deployment-friendly framework.

In this article, we propose SuperEIO, a novel self-supervised learning-driven framework for EIO that handles fundamental challenges in event feature detection and event descriptor matching. Inspired by [8], [9], we develop specialized networks for event domains trained entirely on a synthetically generated event-based dataset. The self-supervised networks and synthetic training strategy eliminate the need for real-world event data collection and manual annotation, significantly improving practicality and scalability. The trained models exhibit strong

Received 19 November 2025; revised 31 January 2026; accepted 13 March 2026. This work was supported in part by the General Research Fund under Grant 17204222, and in part by the Seed Fund for Collaborative Research and General Funding Scheme-HKU-TCL Joint Research Center for Artificial Intelligence. *(Peiyu Chen and Fuling Lin contributed equally to this work.) (Corresponding author: Peng Lu.)*

The authors are with the Adaptive Robotic Controls Lab (ArcLab), Department of Mechanical Engineering, The University of Hong Kong, Hong Kong SAR 999077, China (e-mail: lupeng@hku.hk).

Digital Object Identifier 10.1109/TIE.2026.3677649

generalization in real-world scenarios, enabling robust EIO under challenging conditions. Our contributions are summarized as follows:

1) To address the lack of annotated training data in the event domain, we present a synthetic event-based dataset and a self-supervised framework, enabling effective training of an event feature detector and a descriptor matcher.
2) We propose a complete EIO system that leverages these deep event features for tracking and loop closure, tightly coupled with IMU for robust estimation. Moreover, all networks are optimized with TensorRT, enabling real-time performance on edge devices.
3) Extensive evaluation on public benchmarks demonstrates the robust performance of our SuperEIO under aggressive motion and HDR scenes. We open-source our code to foster future research on event-based odometry.

II. Related Works

A. Event Feature Detection

Early event feature detectors adapted classical frame-based approaches. Clady et al. [15] proposed the first event-based detector, while eHarris [5] and its improved variant luvHarris [16] modified the Harris corner detector for event streams, with the latter optimizing computations through threshold ordinal event-surfaces. Similarly, eFast [6] leveraged the efficient FAST corner detector for event processing. Arc [7] introduced an innovative real-time corner detector and tracker operating directly on event streams. Although traditional event detectors are highly efficient and enable real-time operation, they sacrifice feature distinctiveness and repeatability, leading to high redundancy. Learning-based approaches have emerged to address the limitations of traditional methods. SILC [17] employed random forests on speed-invariant time surfaces, an extension of the time surface (TS) [18], while EventPoint [19] developed a deep learning-based detector using Tencode representation. Learning-based event detectors improve feature quality but often rely on complex representations or pose efficiency challenges for real-time applications. Therefore, we aim to develop a learning-based event detector that avoids this performance-efficiency dilemma, leveraging a self-supervised paradigm and efficient network design to achieve high distinctiveness while maintaining real-time capability.

B. Traditional Event Odometry

Event-based visual odometry (VO) has attracted significant research attention for challenging scenarios. Kim et al. [20] pioneered monocular event-based odometry using probabilistic filters, while EVO [11] established image-to-model tracking with parallel 3-D reconstruction [21]. ESVO [22] established the first stereo event-based odometry pipeline featuring spatio-temporal mapping and direct 3-D-2-D registration, while ESVO2 [23] improved the mapping solution by combining temporal-stereo and static-stereo configurations with fast block-matching. These methods demonstrate the feasibility of event-only odometry, while their reliance on traditional event processing and probabilistic filters makes them inherently vulnerable in extreme scenarios, where the tracking pipeline often fails and leads to a significant loss of estimation accuracy.

Several event-based odometry studies integrate inertial or other visual sensors to enhance robustness and accuracy. Zhu et al. [24] introduced the first event-based visual-inertial odometry (VIO), using an Extended Kalman Filter (EKF) for accurate 6-DoF estimation and scale uncertainty handling. To address event distortion, [25] proposed a feature tracker with motion-compensated synthesized events for robust EIO via nonlinear optimization. Expanding on previous work, Ultimate SLAM [12] unified event streams, image frames, and IMU measurements. Guan [13] presented a feature-based EIO that processes asynchronous event streams with graph optimization, then further enhanced by PL-EVIO [26] through the tight fusion of event-based point/line features, image-based point features, and IMU measurements. To address geometry-based spatial and temporal data associations in consecutive event streams, ESVIO [2] introduced the first stereo event-based visual-inertial odometry. For complete SLAM, EVI-SAM [27] proposed a hybrid tracking pipeline integrating feature-based reprojection with relative pose constraints, enabling dense 3-D reconstruction. C2F-EFIO [28] offered a novel filter-based framework using event line and point features, enhanced by a coarse-to-fine motion compensation scheme. Although these methods enhance robustness through multisensor fusion, this strategy introduces additional system complexity, including the requirement for precise sensor calibration and increased power consumption. More importantly, it creates a dependency on auxiliary sensors, which diverges from the goal of leveraging the single event camera in the most challenging conditions.

C. Learning-Based Event Odometry

Recently, learning-based event odometry has shown superior robustness and accuracy. Zhu et al. [29] introduced an unsupervised neural network for optical flow, pose estimation, and depth prediction from event streams. DH-PTAM [30] proposed a deep hybrid stereo event-frame SLAM with learning-based feature detection. RAMP-VO [31] developed an end-to-end learning-based VO fusing events and images via recurrent, asynchronous, and parallel encoders. To reduce hardware dependency, DEVO [14] presented the first robust learning-based event-only odometry, using a patch selection mechanism. By combining trainable event-based bundle adjustment with IMU, DEIO [32] established the first learning-based EIO. Despite higher accuracy and robustness, these learning-based methods face significant deployment challenges as they either introduce hardware dependency by fusing with standard cameras or impose heavy GPU resources that hinder online deployment. This limitation motivates our work towards a highly efficient, learning-based, and event-only odometry solution.

III. Methodology

A. System Overview

Our SuperEIO system employs deep neural networks in a modular way, enhancing robustness and precision in

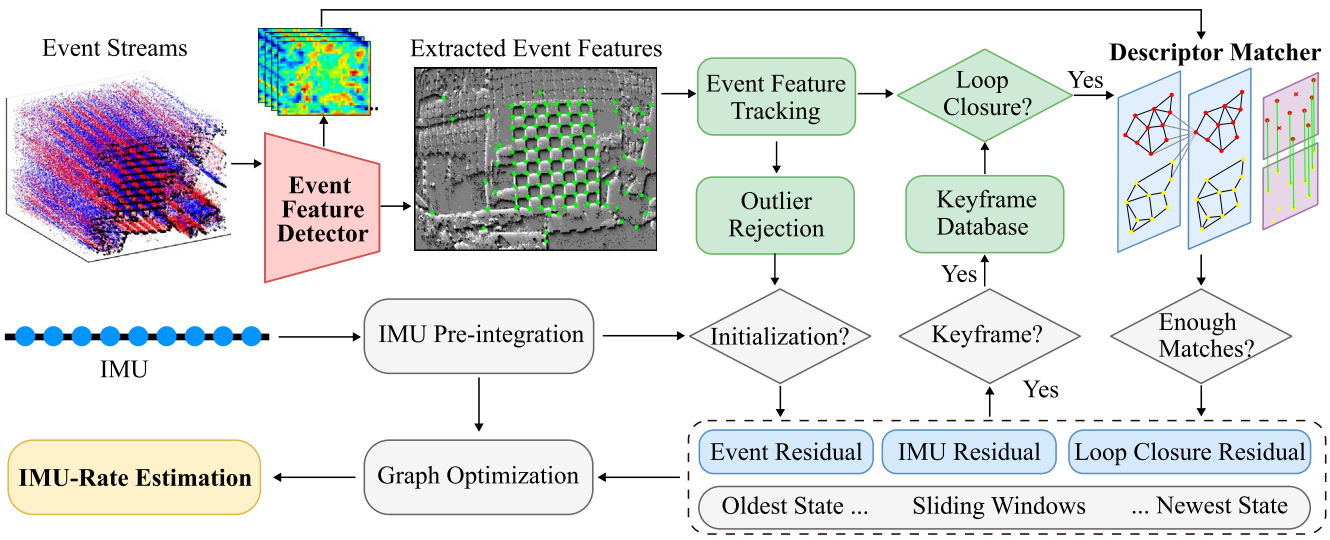

Fig. 1. Overview of our SuperEIO system. We develop a self-supervised event feature detector that extracts features and descriptors from asynchronous events. To enable loop closure, our proposed event descriptor matcher establishes event correspondences. The entire system tightly integrates self-supervised event feature learning with IMU measurements and is optimized with TensorRT, achieving robust and real-time estimation.

challenging environments as illustrated in the framework presented in Fig. 1. It utilizes a self-supervised CNN event feature detector (Section III-B), which is trained on a dedicated synthetic dataset specifically created for event streams. This approach enables the detector to learn robust representations directly from event data, allowing it to detect salient features effectively on the normalized time surface (NTS). These features are then tracked across the NTS using the Lucas-Kanade optical flow with outlier rejection to enhance feature consistency.

The tracked event features are tightly aligned with IMU pre-integration to achieve accurate initialization. The initialization jointly optimizes initial poses, velocities, and IMU biases by minimizing reprojection errors of the event features alongside IMU pre-integration in the sliding window. This process effectively resolves the scale ambiguity in the monocular vision and yields an accurate initial state for the subsequent sliding-window optimization.

In parallel, the bag-of-words approach searches for potential loop closure candidates within the keyframe database using detected event features. The selection of a new keyframe is determined either by the average parallax of tracked features exceeding a preset threshold, or by their number falling below a minimum requirement. Upon detecting a potential loop closure, a GNN-based event descriptor matcher (Section III-C) establishes precise 2-D-2-D correspondences. If a sufficient number of matches are validated by a RANSAC-based epipolar check, the loop closure module performs global bundle adjustment, effectively reducing the accumulated drift and enhancing long-term trajectory consistency.

For deployment on resource-constrained edge devices, we leverage NVIDIA TensorRT to optimize and deploy the ONNX-converted neural networks, including event feature detector and descriptor matcher, achieving high-throughput and low-latency inference.

For the back-end sliding window optimization, we formulate a unified optimization framework within the Ceres solver that integrates residuals from event features, IMU pre-integration, and loop closure constraints to achieve accurate state estimates. Ultimately, IMU forward propagation achieves high-frequency localization for robust IMU-rate pose estimation.

B. Self-Supervised Event Feature Detector

1) Event Representation: Event camera responds to pixel-level illumination changes and generates asynchronous event streams $E = \{(t_i, x_i, y_i, p_i)\}$ at microsecond precision. Each event i records its timestamp t_i, pixel location (x_i, y_i), and polarity p_i ($+1$ for positive and -1 for negative).

Unlike standard cameras, raw event streams are asynchronous and may be distorted due to rapid ego-motion, which inherently challenges event feature detection. To preserve the spatio-temporal history and ensure compatibility with the event feature detector, we adopt an NTS event representation. This representation effectively encodes recent motion history by emphasizing newly arrived events, which provides a computationally efficient input for real-time odometry, as follows:

$$S_{\text{norm}}(x, y, t) = \frac{\sum_{(x_i, y_i)} p_i \cdot \exp\left(-\frac{t_i - t_{\text{last}}}{\tau}\right) - S_{\min}}{S_{\max} - S_{\min}} \tag{1}$$

where t_{last} denotes the timestamp of the last event at each pixel coordinate. Δt represents the accumulated event intervals and $t_i - t_{\text{last}} < \Delta t$. τ is the time constant that controls the rate of time decay. We use $\Delta t = 16.7\,\text{ms}$ and set $\tau = 20\,\text{ms}$, so that events at the beginning of each slice retain sufficient weight, balancing surface sparsity and temporal smearing. Exponential decay function $\exp(\cdot)$ models the diminishing influence of past events over time. $S_{\min}, S_{\max}$ represent respectively the minimum and maximum values in $S(x, y, t) = \sum p_i e^{-(t_i - t_{\text{last}})/\tau}$. $S_{\text{norm}}(x, y, t)$ is the NTS, which falls within the range $[0, 1]$.

2) Network Architecture: To effectively leverage the unique spatio-temporal characteristics of event streams, we adopt a CNN-based detector that directly operates on the NTS representation to detect interest points and extract descriptors jointly. Our event feature detector is shown in Fig. 1, which includes a backbone and two heads.

The backbone consists of an input layer and three blocks. The input layer applies two stacked Conv-BN-ReLU layers, while each block includes a 2×2 max pooling layer followed by two Conv-BN-ReLU layers. Thus, the backbone encodes the NTS $S_{\text{norm}} \in \mathbb{R}^{H \times W}$ derived from raw event streams to produce a feature map of size $\mathbb{R}^{128 \times H/8 \times W/8}$.

The interest point decoder head processes the feature map from the backbone through two convolutional blocks, including a 3×3 convolution with 256 channels followed by batch normalization and ReLU activation, and a 1×1 convolution that reduces the channels to 65 with batch normalization. This produces a 65-channel output tensor $\mathbb{R}^{65 \times H/8 \times W/8}$, where each position encodes scores for 64 potential keypoints in an 8×8 local grid and an additional class indicating the absence of keypoints. The tensor is then passed through a channel-wise softmax function to compute the probability distributions and resized to the original resolution $\mathbb{R}^{H \times W}$ through an 8×8 grid cell decoding. The full-resolution probability map is then refined with nonmaximum suppression (NMS), producing the dense interest point map with size $\mathbb{R}^{H \times W}$.

The descriptor decoder head applies two convolutional layers with batch normalization to the feature map from the backbone, aiming to characterize the event features detected from the interest point branch. The resulting descriptors, which have a shape of $\mathbb{R}^{256 \times H/8 \times W/8}$, are sampled at interest point locations using bilinear interpolation and are then L2-normalized along the channel dimension to unit length. Each sampled descriptor is a 256-dimensional vector, resulting in a shape of $\mathbb{R}^{256 \times N}$, where N is the number of interest points.

3) Training Detail: Most current event-based datasets are collected for localization [33], [34], with a lack of real-world datasets focused on event feature detection. Therefore, we generate simulated event streams based on COCO-2014 [35], which provides large-scale and diverse natural images with rich textures and geometric structures that are beneficial for learning generic event-based features. Fig. 2 shows how we generate corresponding pair images by applying random crop, translation, rotation, and zoom to the original images

$$I_2 = s\left((w_c + \Delta w_c), (h_c + \Delta h_c)\right) \cdot e^{i\theta}(I_1) \tag{2}$$

where I_1 is the cropped image from the original image in the dataset, and I_2 is the transformed image. s represents the scaling parameters. w_c, h_c represent the center of the cropped image I_1, respectively, and $\Delta w_c, \Delta h_c$ denote the offset of the image center. θ is the rotation angle applied to image I_1. Then, we utilize ESIM [36] to generate simulated event streams for each image pair (I_1, I_2). We adopt the same base detector in [8] and random homographic adaptation to generate pseudo ground truth labels on the NTS for training, which further augments geometric diversity and improves generalization when evaluating on real event datasets.

To enhance interest point repeatability and descriptor similarity for event-based matching, we generate NTS pairs using a random homographic matrix $\mathcal{H}$. Each NTS pair is assigned pseudo ground truth correspondences through homographic transformations. Thus the total loss L_{total} combines interest point losses L_{i}, L'_{i} on the original and warped NTS, and the descriptor loss L_{d}, enabling joint training

$$L_{\text{total}} = L_{\text{i}} + L'_{\text{i}} + \beta L_{\text{d}}(D, D', \mathcal{H}) \tag{3}$$

where β is a weighting factor that balances the descriptor loss to the overall loss. D, D' represent a set of descriptors for all keypoints in the original and homography-transformed NTS.

For interest point loss, we use the positions of pseudo ground truth points to compute the binary cross-entropy loss

$$L_{\text{i}} = -\sum_{p=1}^{H \times W} \sum_{n=1}^{N} t_{p,n} \cdot \log\left(\frac{\exp(s_{p,n})}{\sum_{m=1}^{N} \exp(s_{p,m})}\right) \tag{4}$$

where P_h, P_w represent the height and width of the predicted feature map, respectively. N represents the total number of possible categories that the model can predict. $t_{p,n}$ represents the binary ground truth for position p. $s_{p,n}$ represents the logit score of the model at position p for class n.

For descriptors, we classify the descriptor pairs as positive or negative samples based on the NTS pairs generated using the homography matrix $\mathcal{H}$. The descriptor pair relationships between the original NTS and the homography-transformed NTS are defined as follows:

$$t_{p,q} = \tau\left(\frac{\delta - ||\mathcal{H} \cdot p_p - p'_q||}{\delta}\right) \tag{5}$$

where $t_{p,q}$ represents matching relations between pair descriptors. p_p, p'_q denote pixel positions of pair descriptors. δ is the distance threshold parameter. $\tau(\cdot)$ is the step function. Then, we employ hinge loss for descriptor training as follows:

$$L_{\text{d}}(D, D', \mathcal{H}) = \frac{1}{HW} \sum_{p=1}^{H \times W} \left[\frac{1}{HW} \sum_{q=1}^{H \times W} l_{\text{d}}(d_p, d'_q; t_{p,q})\right] \tag{6}$$

$$l_{\text{d}}(d, d'; t) = \begin{cases} \lambda_i \cdot \max(0, \mu_+ - d^T d'), & \text{if } t = 1, \\ \max(0, d^T d' - \mu_-), & \text{if } t = 0 \end{cases} \tag{7}$$

where l_{d} denotes the pairwise hinge loss, which measures the similarity between a pair of descriptors d_p, d'_q based on their matching status $t_{p,q}$. λ_i is a scaling factor for the loss contribution of positive matches. μ_+, μ_- represent margin parameters for positive and negative pairs, which ensure pair descriptors are sufficiently similar and dissimilar, respectively. $d^T d'$ is the similarity score between two descriptors, typically computed as the dot product of their vectors.

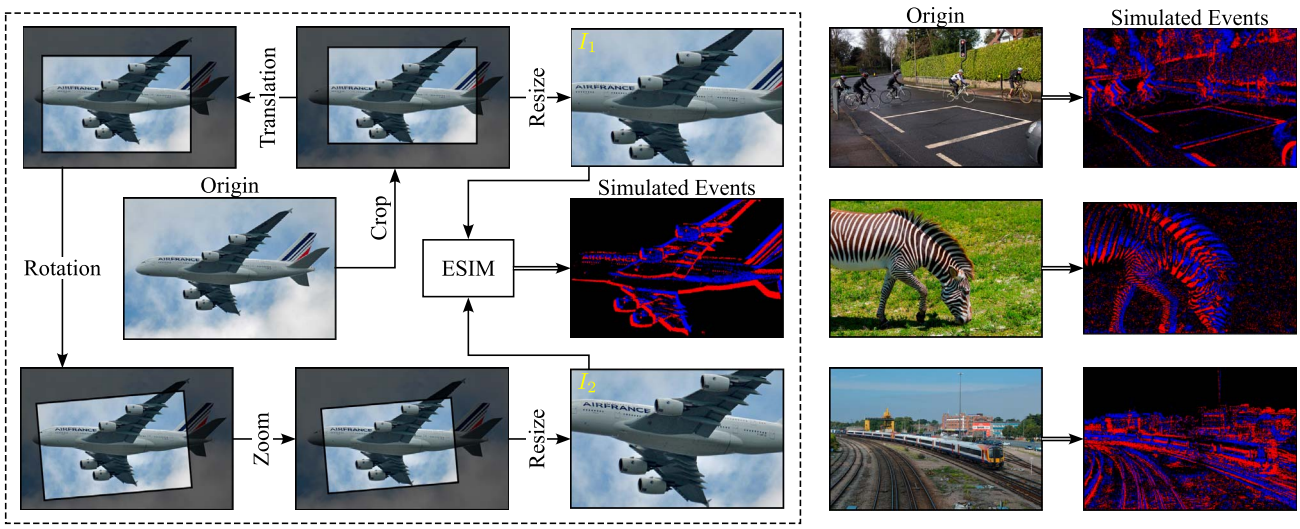

Fig. 2. Generation of simulated events from a single image. The left side illustrates the specific process: (1) selecting a random crop region; (2) translating; (3) rotating; (4) zooming the region; and (5) generating simulated events from the resized image pair (I_1, I_2) using ESIM. Additional generated samples are presented on the right side.

C. Self-Supervised Event Descriptor Matcher

1) Network Architecture: Matching event descriptors poses unique challenges since these features encode historical brightness changes rather than static visual appearance. As shown in Fig. 1, our event descriptor matcher employs a GNN-based attention mechanism for event descriptor matching in loop closure. We use the detector described in Section III-B to detect feature locations $p^A \in \mathbb{R}^{R\times 2}, p^B \in \mathbb{R}^{C\times 2}$ and descriptors $d^A, d^B \in \mathbb{R}^{C\times 256}$, where R, C denote the numbers of detected features in NTS pairs (S^A, S^B), respectively. Subsequently, the detected feature locations and descriptors are used as input to the multiplex graph neural network, which consists of a keypoint encoder and an attentional aggregation module.

For the keypoint encoder, we use the descriptor d_i of each keypoint i and its location p_i transformed through a multilayer perceptron (MLP) to form the initial representation $z_i^{(0)} = d_i + \mathrm{MLP}_{\mathrm{enc}}(p_i)$ for the attention aggregation module. Each node in the graph first updates its representation via self-edges (SE) E_{self}, then propagates messages through cross-edges (CE) E_{cross}. The propagation of information in each layer ℓ is shown as follows:

$$z_i^{(\ell+1)} = z_i^{(\ell)} + \mathrm{MLP}\left(\mathrm{Concat}(z_i^{(\ell)}, m_{E\to i})\right) \tag{8}$$

where Concat() represents the concatenation operation. $m_{E\to i} = \sum_{j:(i,j)\in E} \alpha_{ij} v_j$ represents the attention messages passed to node i from all neighboring nodes in the graph, where $E \in \{E_{\mathrm{self}}, E_{\mathrm{cross}}\}$. α_{ij} denotes the corresponding attention weights, and v_j represents the value of neighboring nodes. The SE and CE attention alternating process is repeated for L layers, after which the final node representation $z_i^{(L)}$ is linearly transformed to produce the matching descriptors f_i^A.

For the optimal matching layer, we use the similarity matrix $M_{ij} = \langle f_i^A, f_j^B \rangle$ to represent the matching scores between f_i^A and f_j^B matching descriptors, where $\langle\cdot\rangle$ is the inner product operation. We extend $M \in \mathbb{R}^{R\times C}$ to $M' \in \mathbb{R}^{(R+1)\times(C+1)}$ via dustbin strategy to handle unmatched keypoints. The partial assignment constraints $P \in [0,1]^{(R+1)\times(C+1)}$ can be expressed as

$$\begin{aligned} \sum_{j=1}^{C+1} P_{i,j} &= 1, \quad \forall i \in \{1,\ldots,R+1\}, \\ \sum_{i=1}^{R+1} P_{i,j} &= 1, \quad \forall j \in \{1,\ldots,C+1\} \end{aligned} \tag{9}$$

where $P_{i,j}$ represents the probability of a match between the i-th ($i \le R$) keypoint in S_A and the j-th ($j \le C$) keypoint in S_B. $P_{i,C+1}, P_{R+1,j}$ represents the probability of the i-th, j-th keypoint being assigned to no matching.

Finally, we employ the Sinkhorn algorithm [37] to solve the above optimization problem on the extended similarity matrix M'. Leveraging the entropy regularization, the neural network efficiently produces the desired matching results.

2) Training Detail: For descriptor matcher training, we use the same dataset in Section III-B. First, the event streams are converted into the NTS. A random homography matrix is then applied to this event representation to generate NTS pairs and obtain pixel-level correspondences. Next, the trained event feature detector (Section III-B) is used to detect event features on NTS pairs. Based on these event features and ground truth matching labels $\mathcal{Y}$, we optimize the matcher by minimizing the negative log-likelihood of the assignment P as follows:

$$L_{\mathrm{m}} = -\sum_{(i,j)\in M} \log P_{i,j} - \sum_{i\in I} \log P_{i,C+1} - \sum_{j\in J} \log P_{R+1,j} \tag{10}$$

where I, J represent the set of unmatched keypoints in S_A, S_B respectively.

D. Graph Optimization Construction

Our SuperEIO system leverages graph optimization to construct and optimize residual constraints from event features and IMU measurements, achieving real-time pose estimation. We define the state vector $\boldsymbol{\chi}$ in the sliding window as follows:

$$\boldsymbol{\chi} = \{\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_n, \boldsymbol{\lambda}\} \tag{11}$$

where each state $\mathbf{s}_k$ (for $k = 0, 1, \ldots, n$) contains the position, orientation, velocity, and IMU bias terms at timestamp k. The set $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_m]$ represents the inverse depths of event features observed within the sliding window, which are used to model the 3-D spatial structure of the environment.

The joint nonlinear optimization problem $J(\boldsymbol{\chi})$ combines IMU residuals $\mathcal{R}_{\text{imu}}(k, k+1)$, event measurement residuals $\mathcal{R}_{\text{evt}}(m)$, and loop closure terms $\mathcal{R}_{\text{loop}}(t, v)$, which is formulated as follows:

$$\min_{\boldsymbol{\chi}} J(\boldsymbol{\chi}) = \min_{\boldsymbol{\chi}} \left(\sum_{k=0}^{n-1} \mathcal{R}_{\text{imu}}(k, k+1) + \sum_{m=1}^{m} \mathcal{R}_{\text{evt}}(m) + \sum_{(t,v)\in\mathcal{L}} \mathcal{R}_{\text{loop}}(t, v) \right). \tag{12}$$

For the event feature m observed at timestamps i and j, the event measurement residual can be expressed as

$$\mathcal{R}_{\text{evt}}(m) = \left\| \mathbf{r}_{\text{evt}}\left(\hat{z}_j^m, \lambda_m, \mathbf{s}_i, \mathbf{s}_j\right) \right\|^2_{\Sigma_{\text{evt}}} \tag{13}$$

with the reprojection residual defined as

$$\mathbf{r}_{\text{evt}} = \hat{z}_j^m - \psi\left(\mathbf{T}_{\text{evt}}^{-1} \mathbf{T}_w^j \mathbf{T}_w^{i\top} \mathbf{T}_{\text{evt}} \psi^{-1}(\lambda_m, \hat{z}_i^m) \right) \tag{14}$$

where Σ_{evt} represents the weight matrix for event measurement. $\hat{z}_j^m = [\hat{u}_j^m, \hat{v}_j^m]^\top$ and $\hat{z}_i^m$ represent the pixel locations of feature m in the NTS at timestamp j and i respectively. ψ, ψ^{-1} denote the projection and back-projection functions. $\mathbf{T}_{\text{evt}}$ is the extrinsic matrix between the event camera and IMU. $\mathbf{T}_w^i$, $\mathbf{T}_w^j$ are the transformation matrices from the world frame to the body frame at timestamps i and j, respectively.

For the loop closure residual, we consider the current frame t has successfully matched enough descriptors with the loop closure frame v in the keyframe database. Thus, we can construct the following residual constraint:

$$\mathcal{R}_{\text{loop}}(t, v) = \sum_{(t,v)\in\mathcal{L}} \rho\left(\left\| \mathbf{r}_{\text{loop}}\left(\hat{z}_v^m, \hat{z}_t^m, \hat{\mathbf{p}}_v, \hat{\mathbf{q}}_v\right) \right\|^2_{\Sigma_{\text{loop}}} \right) \tag{15}$$

where $\mathcal{L}$ represents the match set between the sliding window frames and the loop closure frames. ρ denotes a robust kernel function. $\hat{\mathbf{p}}_t, \hat{\mathbf{q}}_t$ represent the prior position and quaternion in the keyframe database.

IV. Evaluation

We evaluate our event feature detector/matcher and SuperEIO system through various experiments. All network training and simulated experiments are performed on a computer equipped with an AMD Ryzen 7 5800H processor, 16 GB of RAM, and an NVIDIA RTX 3070 Laptop GPU. Section IV-A details the qualitative and quantitative evaluation of our deep event feature detector and descriptor matcher. Section IV-B provides an extensive comparison of our SuperEIO system with other event-based methods on diverse public datasets, evaluating its generalization and adaptability. Section IV-C further demonstrates the robustness of our SuperEIO through additional evaluations on two extreme flight sequences. Finally, Sections IV-D and IV-E deploy our SuperEIO on Jetson Orin for real-world applications and time analysis, respectively.

A. Evaluation of Event Feature Detector and Matcher

1) Event Feature Detector: To evaluate our event feature detector, we conduct a qualitative comparison with the eHarris [5], eFast [6], Arc* [7], and SuperPoint [8] methods on DAVIS240C [38], Mono HKU [13], Stereo HKU [2], and VECtor [33] datasets, as illustrated in Fig. 3. We visualize the event features extracted by the five different detectors by projecting them onto the NTS for qualitative comparison, alongside the corresponding image for reference. As shown in the results across multiple datasets, our detector demonstrates higher accuracy and robustness in consistently extracting salient event features. In contrast, detectors like eHarris, eFast, and Arc* tend to generate redundant and ambiguous features, which become particularly pronounced under challenging conditions such as aggressive motion and HDR scenes, making them more susceptible to noise. On the other hand, SuperPoint is prone to detecting erroneous features in blank areas, since it is sensitive to noisy event streams. This limitation stems from its training on standard images, which inherently constrains its adaptability to event representation with complex temporal dynamics. In contrast, our method, trained directly on event streams, learns to detect more reliable features from salient structures.

For quantitative evaluation, we detect ground truth features on intensity frames from the DAVIS240C dataset using the OpenCV Shi-Tomasi detector. The aforementioned four event feature detectors are applied to detect event features in these sequences. The detected features are used to calculate the F1 score, which provides a comprehensive evaluation by balancing precision and recall. Fig. 4 shows that our detector consistently outperforms other methods, achieving the best F1 score in all sequences.

Besides the F1 score, we also assess the distinctiveness of detected keypoints across frames using the projection error and valid percentage, as shown in Fig. 4. This involves detecting event features at timestamps t_1 and t_2, backprojecting t_2 features onto the t_1 imaging plane using ground truth poses, and establishing one-to-one correspondences via nearest-neighbor matching with a 5-pixel threshold. Reprojection error is then calculated as the Euclidean distance between matched features, while the valid percentage quantifies the proportion of valid matches. Our method achieves the lowest projection error in 3 out of 8 sequences and remains competitive across the rest. Although eFast and SuperPoint demonstrate slightly lower projection errors in *poster_6dof* and *boxes_translation*

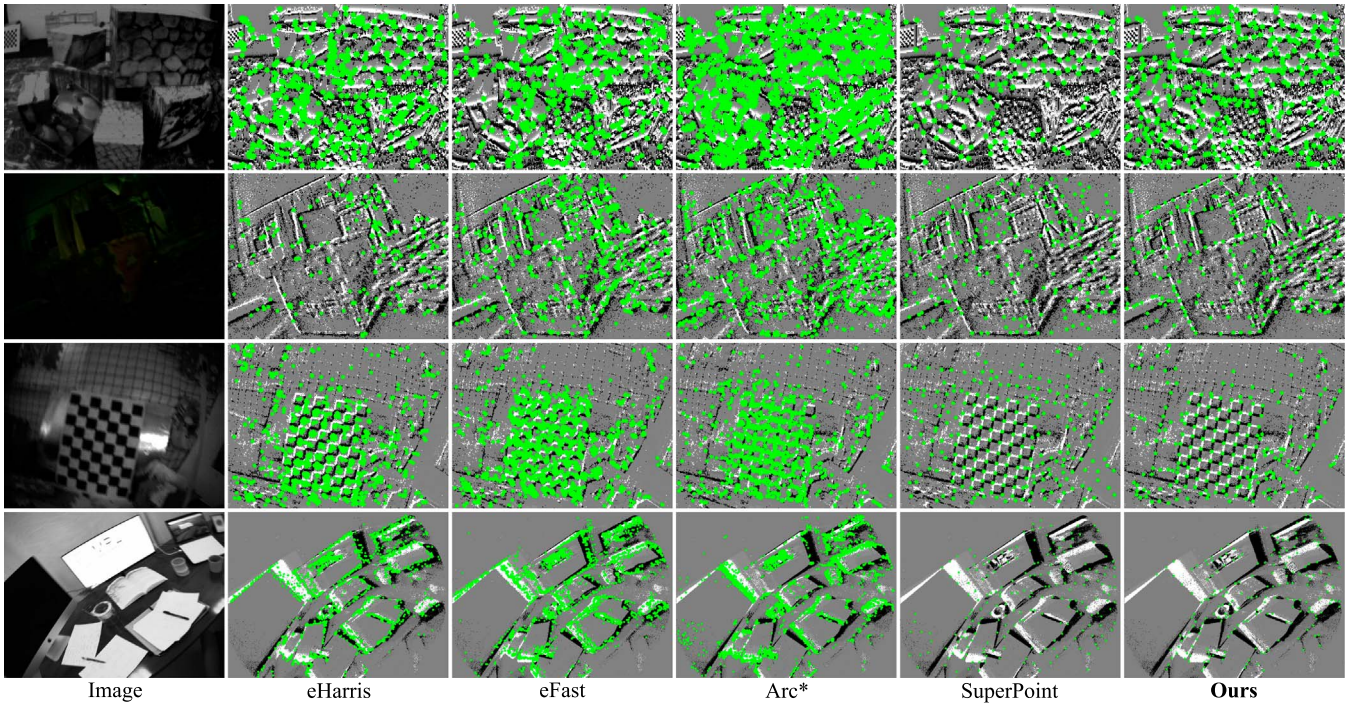

Fig. 3. Visual comparison of other event feature detectors, SuperPoint, and ours on multiple datasets with corresponding images. From top to bottom: DAVIS240C, Mono HKU, Stereo HKU, and VECtor.

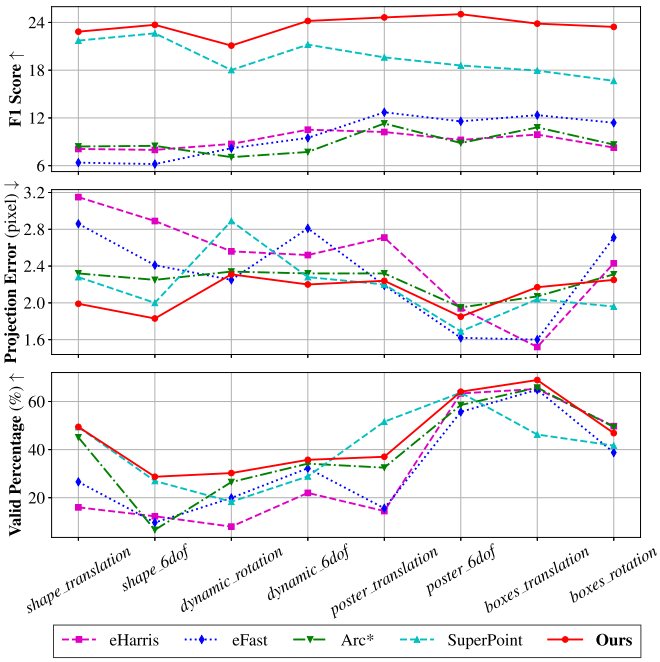

Fig. 4. Comparative performance curves of event feature detectors on various sequences. Our method achieves superior results across multiple metrics.

sequences, their valid percentages are notably lower than ours. This indicates that our event feature detector maintains robustness in complex, real-world scenarios, effectively balancing accuracy and reliability. Overall, our detector exhibits superior performance in both projection error and valid percentage metrics, making it more effective for practical applications.

To assess the impact of latent dimensions in the feature detector, we conduct a channel-level ablation study, where each four-value channel setting corresponds to the latent dimensions of the input layer and three backbone blocks. We evaluate three variants: a compact version [Ours-S, channels=[32, 32, 128, 128]], a larger version [Ours-L, channels=[64, 64, 256, 256]], and a balanced design [Ours, channels=[64, 64, 128, 128]]. Results in Fig. 5(a) demonstrate a clear tradeoff between computational efficiency and accuracy. Ours-S, the most lightweight variant, consistently underperforms across all sequences, yielding the highest mean projection error (2.35 px) and the lowest average valid percentage (34.01%), which indicates that its limited channel capacity hinders the detection of accurate and reliable features. Conversely, Ours-L, despite having the highest channel count, does not achieve superior feature quality. Its average valid percentage remains low at 37.15%, only marginally surpassing Ours-S and being substantially lower than our method (45.13%). This indicates that simply increasing the number of channels, while it may enhance repeatability, primarily leads to significant parameter growth. Ours achieves a competitive projection error (2.11 px) alongside a markedly higher valid percentage. This demonstrates that our balanced design yields features with an optimal blend of distinctiveness and viewpoint invariance, making it the most parameter-efficient architecture.

Moreover, we compare absolute trajectory error (ATE) for the SuperEIO system using all three variants of our detectors (Ours-S, Ours-L, and Ours) versus SuperPoint, measuring

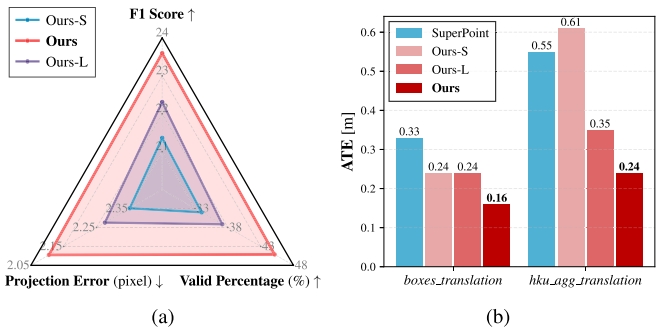

Fig. 5. Ablation study of our detector. (a) Performance of our detector with different channel configurations. (b) ATE [m] with SuperPoint and our detector in different scenarios.

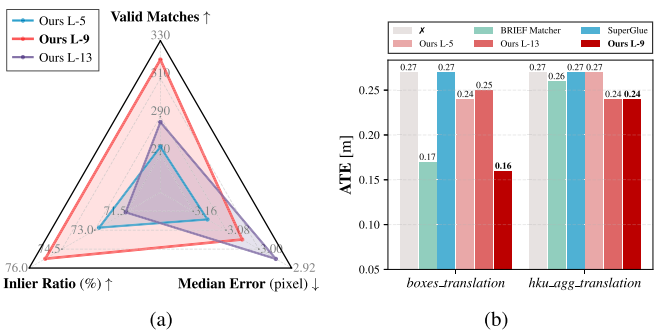

Fig. 6. Ablation study of our matcher. (a) Performance of our matcher with different layer configurations. (b) ATE [m] with different loop closure methods in different scenarios.

errors through complete trajectory alignment in SE(3). Fig. 5(b) shows that SuperEIO with our detector outperforms SuperPoint. This performance gap arises because SuperPoint is trained on standard images and struggles with NTS, failing to detect richer features from event data. Fewer repetitive features of SuperPoint lead to fewer matched pairs, reducing inter-frame transformation accuracy and degrading odometry precision. Meanwhile, the balanced design of our final detector (Ours) proves most effective, outperforming both its compact (Ours-S) and wider (Ours-L) counterparts.

2) Event Descriptor Matcher: To evaluate the impact of matching network depth, we conduct a layer-level ablation study by varying the number of layers (L) to 5, 9 (used in our matcher), and 13 on the DAVIS240C datasets, while employing a fixed event-based detector. The performance is quantified using three metrics: the number of valid matches, the inlier ratio, and the median reprojection error. The number of valid matches refers to the set of correspondences that are identified as inliers through both RANSAC outlier rejection and geometric verification by reprojection error. The inlier ratio is the proportion of these verified inliers relative to the total number of initial matches, and the median reprojection error is calculated by projecting 3-D points, which are derived from triangulation, back to the first frame. The experimental results, as presented in Fig. 6(a), demonstrate that the 9-layer configuration provides the most balanced and robust performance across various sequences. When the depth is reduced to 5 layers, a clear performance degradation is observed across the benchmark. This shallow model exhibits a significant drop in valid matches and a lower inlier ratio, underscoring its insufficient capacity for the task. Conversely, increasing the depth to 13 layers fails to yield a consistent advantage. The marginally lower median reprojection error comes at the cost of a clearly decreased inlier ratio and no gain in valid matches, indicating that added depth fails to enhance accuracy meaningfully while increasing computational complexity. Consequently, the 9-layer architecture is validated as the optimal design choice, effectively balancing matching accuracy and generalization for the task.

Moreover, to evaluate our event descriptor matcher for the whole system, we focus on challenging scenarios from both DAVIS240C (with aggressive motion) and Stereo HKU (with combined motion / HDR conditions). We conduct

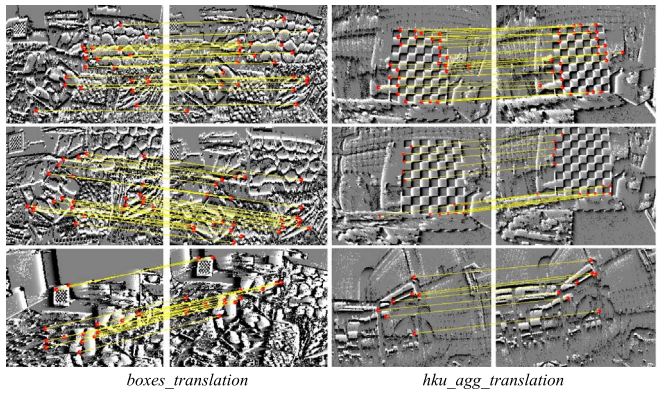

Fig. 7. Examples of our event descriptor matches in loop closure under *boxes_translation* and *hku_agg_translation* sequences.

quantitative ablation studies on the *boxes_translation* and *hku_agg_translation* sequences using ATE with SE(3) alignment. We compare the ATE of our SuperEIO system under different loop closure configurations: without any loop closure, with the BRIEF [39] matcher from [40], with SuperGlue [9], and with all three variants of our matchers (Ours L-5, Ours L-9, Ours L-13). As shown in Fig. 6(b), the system with our matcher (Ours L-9) shows the best accuracy among other configurations. Ablation on the depth of our matcher reveals that the L-5 variant, while improving upon no loop closure, underperforms relative to our final L-9 model, indicating insufficient capacity for robust loop closure. The L-13 variant shows competitive performance on *hku_agg_translation* but is inconsistent, failing to match the accuracy of L-9 across both scenarios. Specifically, compared with the system without loop closure, applying the BRIEF matcher for loop closure significantly improves performance in *boxes_translation*. Furthermore, our method outperforms SuperGlue by a significant margin, reducing ATE by 41% in the *boxes_translation* and 11% in *hku_agg_translation*. This performance gap can be attributed to the fact that image-based SuperGlue struggles to match event-based descriptors effectively. In contrast, our matcher is specifically trained in the event domain, allowing it to handle event-driven data more accurately.

TABLE I
MPE [%] COMPARISON OF OUR SUPEREIO WITH OTHER EVENT-BASED METHODS ON VARIOUS DATASETS

Sequence		Zhu's [24] (w/o LC)	USLAM [12] (w/o LC)	Mono-EIO [13] (w/ LC)	PL-EIO [26] (w/ LC)	C2F-EFIO [28] (w/o LC)	**SuperEIO (w/o LC)**	**SuperEIO (w/ LC)**
DAVIS240C [38]	*boxes_translation*	2.69	0.76	0.34	**0.26**	0.91	0.63	0.36
	boxes_6dof	3.61	0.44	0.61	**0.43**	0.70	0.50	0.48
	dynamic_translation	1.90	0.59	**0.26**	1.13	0.37	0.98	0.49
	dynamic_6dof	4.07	**0.38**	0.43	1.18	0.45	0.50	0.48
	hdr_boxes	1.23	0.67	0.40	0.51	0.53	**0.37**	**0.37**
	poster_6dof	3.56	0.30	**0.26**	0.94	0.52	0.49	0.48
Mono HKU [13]	*vicon_hdr1*	-	1.49	0.59	0.67	0.81	0.99	**0.58**
	vicon_hdr2	-	1.28	0.74	**0.45**	1.01	1.41	0.46
	vicon_darktolight1	-	1.33	0.81	0.78	0.82	0.73	**0.49**
	vicon_lighttodark1	-	1.79	**0.29**	0.42	1.21	1.04	0.69
	vicon_dark1	-	1.75	1.02	0.64	1.60	0.53	**0.24**
	vicon_dark2	-	1.10	0.66	0.62	1.33	0.54	**0.35**
Stereo HKU [2]	*hku_agg_translation*	-	16.22	0.58	0.44	0.42	0.45	**0.41**
	hku_agg_flip	-	11.15	*failed*	4.17	**1.87**	2.48	2.37
	hku_agg_walk	-	*failed*	5.42	1.34	**1.27**	1.62	1.38
	hku_hdr_slow	-	*failed*	1.24	**0.25**	0.30	1.06	0.47
	hku_hdr_tran_rota	-	*failed*	0.76	0.84	**0.57**	0.69	0.65
	hku_dark_normal	-	*failed*	0.60	0.80	0.81	0.69	**0.40**
VECtor [33]	*robot_fast*	-	**0.68**	3.26	2.36	*failed*	1.33	0.76
	desk_fast	-	0.82	1.83	1.56	0.77	0.49	**0.47**
	mountain_fast	-	1.61	1.04	2.33	*failed*	0.53	**0.44**
	hdr_fast	-	4.30	7.02	2.49	*failed*	0.64	**0.56**
	school_scooter	-	4.98	*failed*	4.98	10.78	**1.98**	**1.98**
	units_scooter	-	4.77	*failed*	2.03	2.56	**1.54**	1.68
Average Performance		2.84	2.19	1.24	1.17	1.38	0.93	**0.71**

Note: Average performance is computed over successful runs, excluding "failed" and "-" cases. w/ and w/o LC represent with and without loop closure. Best in **bold** and second best underlined.

Moreover, we execute the complete SuperEIO pipeline on representative sequences, saving some detected loop closure matches for evaluation. As demonstrated in Fig. 7, these results reliably identify recurring scenes and establish sufficient descriptor matches under such extreme conditions.

B. Evaluation of SuperEIO Pipeline

We compare our SuperEIO with other state-of-the-art event odometry [12], [13], [24], [26], [28] on four publicly available benchmarks, which feature diverse camera resolutions and cover a wide range of scenarios, including DAVIS240C [38], Mono HKU [13], Stereo HKU [2], and VECtor [33] datasets.

For quantitative accuracy analysis, we use the mean position error (MPE, %) by aligning estimated trajectories with ground truth via 6-DOF SE(3) transformation. The alignment methods varied by dataset for consistency with previous work: full trajectory alignment for Stereo HKU and Vector, and 5-second alignments for DAVIS240C and Mono HKU as described in [12], [26]. For nonopen-source algorithms, we adopt the accuracy metrics reported in their publications, with "-" indicating missing data. For open-source algorithms, we also utilize the reported results on identical datasets. For the remaining datasets, results are obtained via released codes, where *failed* denotes unsuccessful runs. Both USLAM [12] and C2F-EFIO [28] have full and pure event-based configurations, while we evaluate the pure event configuration for fair comparisons. Table I demonstrates the accuracy comparison of our SuperEIO (with and without loop closure) with other event-based odometry on various datasets.

The DAVIS240C records rapid yet spatially constrained 6-DOF motions under limited indoor areas using a single DAVIS240C (240×180) camera. Our SuperEIO system achieves significantly higher accuracy than Zhu's [24]. The gap may arise since the EM-based feature tracking in their system is not well matched with the traditional image-based FAST corner detection, potentially leading to feature drift under rapid motion conditions. Meanwhile, our approach achieves competitive results compared with other algorithms.

The Mono/Stereo HKU dataset captures more challenging sequences, particularly those involving aggressive motion and significant HDR conditions, using a single DAVIS346 camera and a stereo rig of DAVIS346 cameras (346×260) with Vicon ground truth for each configuration, respectively. Compared with other methods, ours achieves the best results in 6 out of 12 sequences and the second best in 3 others. The results also reveal that our superiority is particularly evident in scenes with HDR. This is due to the highly distinctive and repeatable features from our self-supervised detector, enabling more reliable tracking than conventional handcrafted event features. Despite successful initialization, both USLAM and Mono-EIO fail on

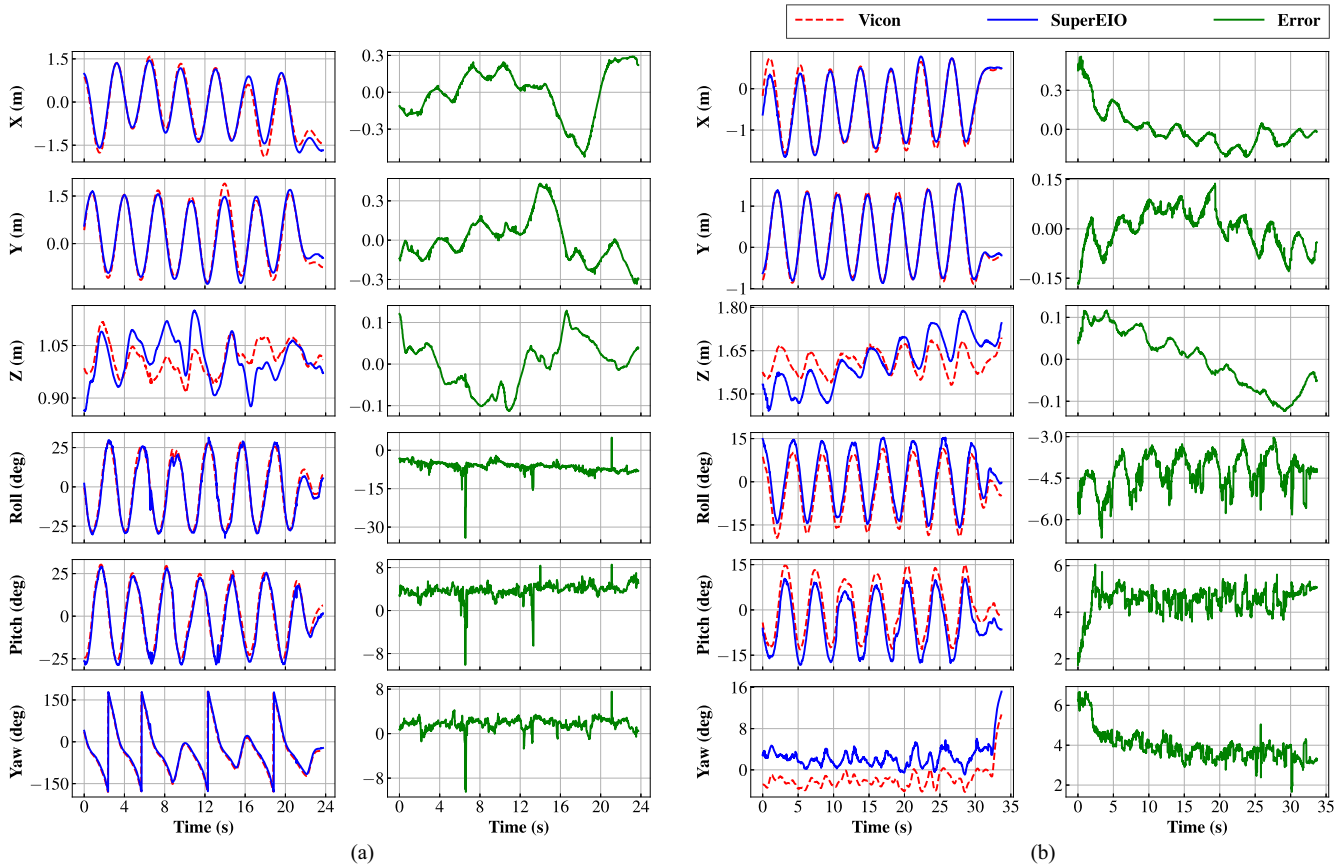

Fig. 8. Comparison of SuperEIO estimated position (X, Y, Z), attitude (roll, pitch, yaw), and corresponding errors with vicon ground truth: (a) *aggressive_flight* sequence; (b) *dark_flight* sequence.

some sequences since significant drift accumulates under aggressive motion and severe HDR conditions, eventually leading to tracking loss.

The VECtor dataset utilizes a hardware-synchronized stereo event camera (640 × 480) system to acquire sequences featuring both small-scale and large-scale indoor scenes with complex illumination conditions. Our SuperEIO performs better than other competing methods in 4 out of 6 sequences. In the *units_scooter* sequence, our method without loop closure outperforms the full system. This can be attributed to the scene containing numerous visually similar yet geographically different locations, which can lead to false positive loop closures and thus degrade trajectory accuracy. C2F-EFIO fails on several *fast* sequences even after successful initialization, as it loses tracking within seconds under highly aggressive motion. Moreover, Mono-EIO fails on the *scooter* sequences due to large textureless regions (e.g., white walls), which provide insufficient features for reliable initialization.

Overall, our method shows superior performance compared with other state-of-the-art algorithms, which can be attributed to the superior robustness of our learning-based event feature detector, providing more stable and dominant features. In addition, our loop closure thread employs a more accurate and robust event-based descriptor matcher, resulting in lower average errors across all four datasets.

C. Evaluation on Complex Quadrotor Flight

To further demonstrate the robustness of SuperEIO, we evaluate it in challenging aggressive motion and darkness flight scenarios. To assess positional accuracy, we calculate the ATE and plot both aligned trajectories and their error profiles along all three axes. For rotation analysis, we calculate attitude errors by computing the relative rotations between estimated and ground truth poses, then converting them into roll-pitch-yaw (RPY) angles for illustration. Fig. 8 presents a comparative analysis between the estimated and ground truth trajectories, along with error plots under two scenarios.

In *aggressive_flight* sequences, the drone executes 1.5 m radius circular trajectories with rapid yaw variations in a 5 m × 8 m room, with an average speed of 2.0 m/s. Our system maintains acceptable precision with an ATE of 0.24 m, and constrains positional errors within 0.4 m across all axes. During aggressive maneuvers, although there are moments of fluctuation in the attitude angles, the system is still able to track the changes in attitude angles effectively overall. In *dark_flight* scenarios, the drone executes 1.5 m radius circular patterns with constant yaw under indoor lights-off conditions, with an average speed of 1.5 m/s, achieving an ATE of 0.15 m. SuperEIO demonstrates precise position estimation with errors within 0.15 m along the y, z axes, and within 0.6 m along the x axis. For attitude

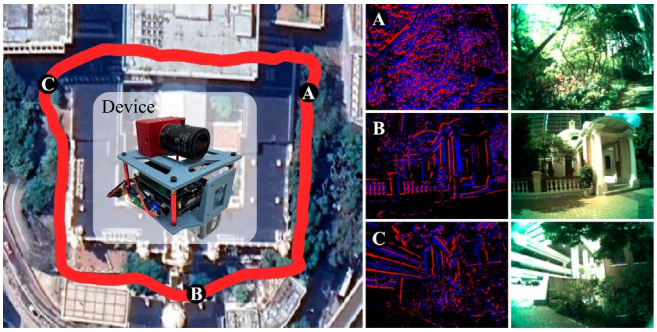

Fig. 9. Our SuperEIO operates in real-time on jetson orin under large-scale scenes. Left: our compact handheld devices and the estimated trajectories are accurately aligned with google maps. Right: Visualization of event streams (used for trajectory estimation) and standard images (for scene reference only).

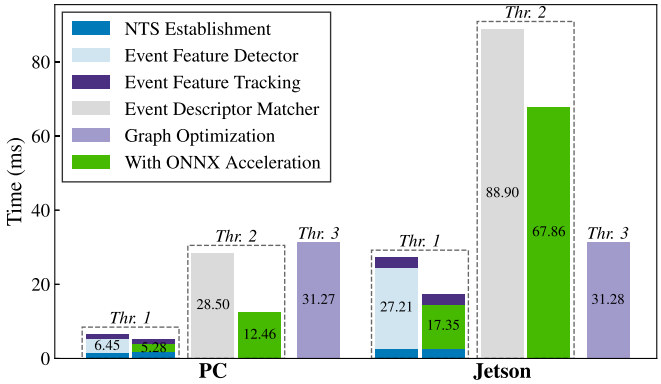

Fig. 10. Computational efficiency [ms] per event stream at a resolution of 346×260 on different modules in independent threads. With ONNX acceleration, the detector and matcher are accelerated by $1.22\times$ and $2.29\times$ on PC, respectively. On jetson, the corresponding speed-ups are $1.57\times$ and $1.31\times$.

estimation, the system achieves remarkable roll, pitch, and yaw error within 6°.

D. Real World Onboard Evaluation

In this section, we deploy SuperEIO on our handheld platform in outdoor scenes. As shown in Fig. 9, our hardware setup consists of a DAVIS346 event camera, a Jetson Orin, and a portable power supply. We conduct localization testing by traversing a closed rectangular path ($80\,\text{m} \times 75\,\text{m}$) around the HKU main building, completing a full loop back to the origin. Our SuperEIO achieves real-time performance on resource-constrained devices while maintaining localization accuracy, as shown by the trajectory alignment with Google Maps, which demonstrates robustness in large-scale environments.

E. Time Analysis

In this section, we analyze the time consumption of the main modules in our SuperEIO. We first evaluate the computational complexity of the core neural networks deployed in our system. The analysis reveals that our event feature detector has a computational complexity of 6.6 GFLOPs and contains 1.3 M parameters, demonstrating a computation-intensive design well-suited for real-time inference. Meanwhile, the event descriptor matcher employed in the loop closure thread demands 12.3 GFLOPs and 12.0 M parameters, leveraging extensive representational capacity to ensure reliable place recognition under appearance changes. This distinct distribution of complexity reflects the different operational requirements of the front-end and the loop closure thread, where the detector prioritizes computational efficiency for real-time operation while the matcher employs extensive parameterization to ensure accurate place recognition across varying conditions.

Meanwhile, we conduct real-world time analysis across different hardware platforms to validate the practical implications. Our system primarily consists of three parallel processing threads: front-end (thread 1), loop closure (thread 2), and graph optimization (thread 3). Fig. 10 presents the processing time per event stream for 346×260 resolutions on both PC and Jetson. The front-end consists of NTS establishment, event feature detector, and event feature tracking. Even on the resource-constrained Jetson device, it still enables real-time processing of event streams within $\Delta t < 33$ ms, ensuring completion before the next data arrival. Meanwhile, the ONNX-accelerated event feature detector significantly reduces the processing time. The loop closure processes matching efficiently, maintaining reasonable processing times. Crucially, the ONNX-accelerated event descriptor matcher achieves a $2.29\times$ and $1.31\times$ improvement on PC and Jetson, respectively, compared with its nonaccelerated version. The most time-consuming module is graph optimization, as it integrates information from the front-end and loop closure to achieve pose estimation. Its computation does not affect the odometry frequency, as SuperEIO maintains IMU-rate pose estimation through continuous IMU propagation. Overall, the system achieves real-time performance on edge devices, effectively balancing computational efficiency and accuracy.

V. CONCLUSION

In this article, we proposed SuperEIO, a novel EIO framework that integrates self-supervised event feature learning networks with IMU measurements for robust pose estimation. Our system employed a CNN for event feature detection in the tracking front-end and a GNN for event descriptor matching to enable loop closure. All learning-based networks were trained on synthetic data and demonstrated strong generalization in real-world scenarios. By converting these networks to ONNX models and optimizing them with TensorRT, SuperEIO achieves real-time performance on edge devices. Extensive evaluations showed that our method outperformed state-of-the-art event-based odometry, particularly under aggressive motion and HDR conditions. Future work will focus on developing a unified deep event tracking pipeline and designing a lightweight, end-to-end architecture for edge devices.

REFERENCES

[1] G. Gallego et al., "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[2] P. Chen, W. Guan, and P. Lu, "ESVIO: Event-based stereo visual inertial odometry," *IEEE Robot. Automat. Lett.*, vol. 8, no. 6, pp. 3661–3668, Jun. 2023.

[3] J. Fan, X. Liu, Y. Ou, P. Zhang, C. Zhou, and Z. Hou, "Underwater robot self-localization method using tightly coupled events, images, inertial, and acoustic fusion," *IEEE Trans. Ind. Electron.*, vol. 72, no. 5, pp. 5126–5135, May 2025.

[4] S. Schraml, A. N. Belbachir, and H. Bischof, "An event-driven stereo system for real-time 3-D 360 panoramic vision," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 418–428, Jan. 2016.

[5] V. Vasco, A. Glover, and C. Bartolozzi, "Fast event-based Harris corner detection exploiting the advantages of event-driven cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS),* Piscataway, NJ, USA: IEEE Press, 2016, pp. 4144–4149.

[6] E. Mueggler, C. Bartolozzi, and D. Scaramuzza, "Fast event-based corner detection," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2017, pp. 1–11.

[7] I. Alzugaray and M. Chli, "Asynchronous corner detection and tracking for event cameras in real time," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3177–3184, Oct. 2018.

[8] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 224–236.

[9] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4938–4947.

[10] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "AirSLAM: An efficient and illumination-robust point-line visual SLAM system," vol. 41, pp. 1673–1692, 2025, *arXiv:2408.03520.*

[11] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza, "EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real time," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 593–600, Apr. 2017.

[12] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robot. Automat. Letters*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[13] W. Guan and P. Lu, "Monocular event visual inertial odometry based on event-corner using sliding windows graph-based optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 2438–2445.

[14] S. Klenk, M. Motzet, L. Koestler, and D. Cremers, "Deep event visual odometry," in *Proc. Int. Conf. 3D Vis. (3DV)*, Piscataway, NJ, USA: IEEE Press, 2024, pp. 739–749.

[15] X. Clady, S.-H. Ieng, and R. Benosman, "Asynchronous event-based corner detection and matching," *Neural Netw.*, vol. 66, pp. 91–106, Jun. 2015.

[16] A. Glover, A. Dinale, L. D. S. Rosa, S. Bamford, and C. Bartolozzi, "luvHarris: A practical corner detector for event-cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 10087–10098, Dec. 2022.

[17] J. Manderscheid, A. Sironi, N. Bourdis, D. Migliore, and V. Lepetit, "Speed invariant time surface for learning to detect corner points with event-based cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10245–10254.

[18] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "HOTS: A hierarchy of event-based time-surfaces for pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, Jul. 2017.

[19] Z. Huang, L. Sun, C. Zhao, S. Li, and S. Su, "EventPoint: Self-supervised interest point detection and description for event-based camera," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 5396–5405.

[20] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3D reconstruction and 6-DOF tracking with an event camera," in *Proc. Eur. Conf. Comput. Vis.*, New York, NY, USA: Springer, 2016, pp. 349–364.

[21] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1394–1414, 2018.

[22] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433–1450, Oct. 2021.

[23] J. Niu, S. Zhong, X. Lu, S. Shen, G. Gallego, and Y. Zhou, "ESVO2: Direct visual-inertial odometry with stereo event cameras," vol. 41, pp. 2164–2183, 2025, *arXiv:2410.09374.*

[24] A. Zihao Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5391–5399.

[25] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2017.

[26] W. Guan, P. Chen, Y. Xie, and P. Lu, "PL-EVIO: Robust monocular event-based visual inertial odometry with point and line features," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 4, pp. 6277–6293, Oct. 2024.

[27] W. Guan, P. Chen, H. Zhao, Y. Wang, and P. Lu, "EVI-SAM: Robust, real-time, tightly-coupled event–visual–inertial state estimation and 3D dense mapping," *Adv. Intell. Syst.*, vol. 6, no. 12, 2024, Art. no. 2400243.

[28] B. Choi, H. Lee, and C. G. Park, "Event-frame-inertial odometry using point and line features based on coarse-to-fine motion compensation," *IEEE Robot. Automat. Lett.*, vol. 10, no. 3, pp. 2622–2629, Mar. 2025.

[29] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 989–997.

[30] A. Soliman, F. Bonardi, D. Sidibé, and S. Bouchafa, "DH-PTAM: A deep hybrid stereo events-frames parallel tracking and mapping system," *IEEE Trans. Intell. Veh.*, vol. 10, no. 1, pp. 336–345, Jan. 2025.

[31] R. Pellerito et al., "Deep visual odometry with events and frames," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS),* Piscataway, NJ, USA: IEEE Press, 2024, pp. 8966–8973.

[32] W. Guan, F. Lin, P. Chen, and P. Lu, "DEIO: Deep event inertial odometry," 2025, *arXiv:2411.03928.*

[33] L. Gao et al., "VECtor: A versatile event-centric benchmark for multi-sensor SLAM," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8217–8224, Jul. 2022.

[34] P. Chen et al., "ECMD: An event-centric multisensory driving dataset for SLAM," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 407–416, Jan. 2024.

[35] T.-Y. Lin et al., "Microsoft coco: Common objects in context," in *Proc. Comput. Vis.–ECCV 2014: 13th Eur. Conf.*, Zurich, Switzerland. New York, NY, USA: Springer, 2014, pp. 740–755.

[36] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Video to events: Recycling video datasets for event cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2020, pp. 3586–3595.

[37] G. Peyré et al., "Computational optimal transport: With applications to data science," *Foundations Trends® in Mach. Learn.*, vol. 11, nos. 5–6, pp. 355–607, 2019.

[38] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, 2017.

[39] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proc. Comput. Vis.–ECCV: 11th Eur. Conf. Comput. Vis., Heraklion*, Crete, Greece. New York, NY, USA: Springer, 2010, pp. 778–792.

[40] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

Peiyu Chen received the B.Sc. degree in automation from Nanjing University of Science and Technology, China, in 2020, and the M.Sc. degree in computer control & automation from Nanyang Technological University, Singapore, in 2022, respectively. He is currently working toward the Ph.D. degree in robotics with the Adaptive Robotic Controls Lab (ArcLab), University of Hong Kong, Hong Kong SAR, China.

His research interests include event-based visual odometry, robotics, SLAM, and others.

Fuling Lin (Graduate Student Member, IEEE) received the B.Eng. and M.Sc. degrees in mechanical engineering from Tongji University, Shanghai, China, in 2019 and 2022, respectively. He is currently working toward the Ph.D. degree in robotics with the Adaptive Robotic Controls Lab (ArcLab), University of Hong Kong, Hong Kong SAR, China.

His research interests include robotics and computer vision.

Weipeng Guan received the bachelor's degree in electronic science & technology and the master's degree in control theory and control engineering from the South China University of Technology, Guangzhou, South, in 2016, 2019, respectively, and the Ph.D. degree in robotics from the University of Hong Kong, Hong Kong SAR, China, in 2025.

He was a Research Assistant with the Chinese Academy of Sciences focusing on deep learning and computer vision. He also had a short but highly rewarding experience at the Chinese University of Hong Kong, Hong Kong, China. His research interests include robotics perception and navigation.

Dr. Guan served as a Consultant or an Intern for several reputable companies, such as Samsung Electronics, Huawei Technologies, TCL, etc. He has published over 70 research articles in international journals and conferences. He also holds more than 50 authorized patents. In 2021, he was selected for the Elsevier and Stanford University's list of the World's Top 2% Scientists.

Yi Luo received the B.Eng. degree in nuclear engineering from the Southeast University, Nanjing, China, in 2020. He is currently working toward the Ph.D. degree in robotics with the Adaptive Robotic Controls Laboratory (ArcLab), University of Hong Kong, Hong Kong SAR, China.

His research interests include uncrewed aerial vehicles (UAVs), robotics, and control systems.

Peng Lu (Member, IEEE) received the B.Sc. degree in automatic control and the M.Sc. degree in nonlinear flight control from the Northwestern Polytechnical University, Xi'an, China, in 2010 and 2013, respectively, and the Ph.D. degree in flight control from Delft University of Technology, Delft, The Netherlands, in 2016.

He continued his journey on flight control with Delft University of Technology. After that, he shifted a bit from flight control and started to explore control for ground/construction robotics with ETH Zurich, Zurich, Switzerland. He was a Postdoctoral Researcher with ADRL Lab, Zurich, in 2016. He also had a short but nice journey at University of Zurich, Zurich, and ETH Zurich (RPG Group), where he was working on vision-based control for UAVs as a Postdoc Researcher. He was an Assistant Professor in autonomous UAVs and robotics with Hong Kong Polytechnic University, Hong Kong, China, prior to joining the University of Hong Kong, Hong Kong SAR, China, in 2020.

Prof. Lu was a recipient of the several awards, such as 3rd place in 2019 IROS autonomous drone racing competition and Best Graduate Student Paper finalist in AIAA GNC. He is currently an Associate Editor for IROS and a Session Chair/Co-Chair for conferences like IROS and AIAA GNC for several times. He also gave a number of invited/keynote speeches at multiple conferences, universities, and research institutes.