

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

AirIO: Learning Inertial Odometry with Enhanced IMU Feature Observability

air-io.github.io

Yuheng Qiu^{*1}, Can Xu^{*1}, Yutian Chen¹, Shibo Zhao¹, Junyi Geng² and Sebastian Scherer¹

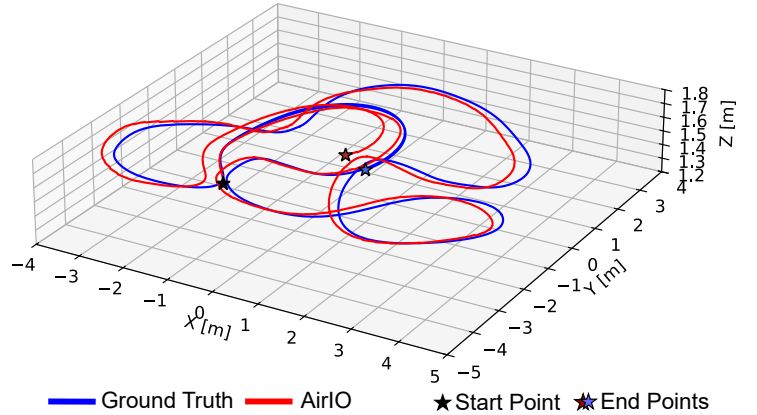
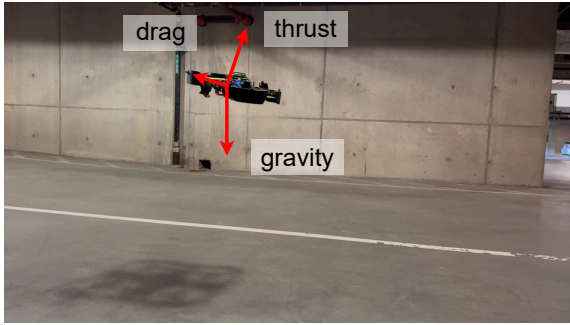


Fig. 1: When deploying learning-based IO on drones, we observed that preserving the IMU feature representation in the body frame while retaining gravitational acceleration improves the ATE by 73.9% on the Blackbird dataset. This significant improvement occurs because preserving the IMU data in the body frame, along with gravitational information, enhances observability and retains more kinematic details. Based on this finding, we propose AirIO, which outperforms state-of-the-art algorithms without extra information like control signals or additional sensors.

Abstract—Inertial odometry (IO) using only Inertial Measurement Units (IMUs) offers a lightweight and cost-effective solution for Unmanned Aerial Vehicle (UAV) applications, yet existing learning-based IO models often fail to generalize to UAVs due to the highly dynamic and non-linear-flight patterns that differ from pedestrian motion. In this work, we identify that the conventional practice of transforming raw IMU data to global coordinates undermines the observability of critical information in UAVs. By preserving the body-frame representation, our method achieves substantial performance improvements, with a 66.7% average increase in accuracy across three datasets. Furthermore, explicitly encoding attitude information into the motion network results in an additional 23.8% improvement over prior results. Combined with a data-driven IMU correction model (AirIMU) and an uncertainty-aware Extended Kalman Filter (EKF), our approach ensures robust state estimation under aggressive UAV maneuvers without relying on external sensors or control inputs. Notably, our method also demonstrates strong generalizability to unseen data, underscoring its potential for real-world UAV applications.

Index Terms—Aerial Systems; Perception and Autonomy; Deep Learning Methods; Localization

Manuscript received: February, 15, 2025; Revised April, 30, 2025; Accepted June, 1, 2025.

This paper was recommended for publication by Editor Giuseppe Loianno upon evaluation of the Associate Editor and Reviewers' comments.

^{*}Equal contribution.

¹Yuheng Qiu, Can Xu, Yutian Chen, Shibo Zhao, and Sebastian Scherer are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA {yuhengq, canxu, yutianch, shiboz, basti}@andrew.cmu.edu; ²Junyi Geng is with Department of Aerospace Engineering, Pennsylvania State University, University Park, PA, 16802, USA jgeng@psu.edu.

Digital Object Identifier (DOI): see top of this page.

©2026 IEEE

I. INTRODUCTION

INERTIAL Measurements Units (IMUs) are inexpensive and ubiquitous sensors that provide linear accelerations and angular velocities. Due to the size, weight, power, and cost (SWAP-C) constraints, IO based only on light and low-cost sensors is ideal for Unmanned Aerial Vehicles (UAVs) applications ranging from 3D mapping [1], exploration [2] to physical interaction [3]. Compared to exteroceptive sensors like vision and LiDAR, IMUs are unaffected by visual degradation factors such as motion blur [4] or dynamic object interruption [5], which are common in agile UAV flights [6]. Despite these advantages, current IO solutions struggle to accurately adapt to the complex motion models of UAVs due to the IMU inherent noise and bias, leading to large drift over time, particularly during agile maneuvers.

Most recent advances in learning-based IO have focused on pedestrian and legged robots [7]–[9], utilizing motion priors like stride length [10] and repetitive gait patterns [11], [12] beyond IMU pre-integration [13]. In contrast, multirotor UAV motion involves rigorous maneuvers affecting orientation and thrust, resulting in highly dynamic, nonlinear behavior without clear priors. Small attitude changes can cause significant variations in velocity and position, complicating the IO process and making pedestrian-focused methods less effective [6]. As a result, deploying learning-based IO on UAVs often requires additional sensors, such as tachometers [14] or control inputs like thrust commands [6]. This raises our main questions:

Why is learning IO not directly applicable to UAVs?
How can learning IO be effectively deployed for UAVs?

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

In this paper, we investigate the effective representation of learning-based IO in multirotor UAV motion and propose a solution that relies solely on IMU data. Through rigorous feature analysis and examination of UAV dynamics, we identify that the commonly used global-frame representation is less effective for dynamic, agile maneuvers due to the less observable attitude information compared to IMU data in the body-frame representation. To address this, we introduce AirIO, a learning-based IO method that explicitly encodes attitude information and predicts velocity using body-frame representation. We further integrate an EKF that fuses IMU pre-integration with the learning-based model for odometry estimation. We show that our approach outperforms state-of-the-art algorithms [6] in various scenarios and environments, even without additional sensors or control information.

The main contributions of this work are:

- 1) We conduct the analysis of the extracted features from different representations using t-Distributed Stochastic Neighbor Embedding (t-SNE) and principal component analysis (PCA). The results show that body-frame representation is more expressive and observable. Simply changing the input representation can significantly improve IO accuracy by an average of 66.7%.
- 2) Building on this, we develop AirIO, a learning-based IO that encodes the attitude information and fuses it with the body-frame IMU data for velocity prediction. This explicit encoding further improves accuracy by 23.8%. Additionally, we integrate an uncertainty-aware IMU preintegration model and a learned motion network into EKF for odometry estimation.
- 3) We validate our approach on extensive experiments. It outperforms existing IO algorithms without the need for additional sensors or control information. It also demonstrates generalizability to the unseen datasets.

II. RELATED WORK

A. Model-based Inertial Odometry

Traditional model-based inertial odometry (IO) methods rely on kinematic models [13] to estimate relative motion and are often integrated with exteroceptive sensors such as cameras [15] and LiDAR [1]. While these methods achieve high accuracy, their dependence on external sensors makes them vulnerable to disturbances caused by agile motion or dynamic environments. To address the SWAP challenge of lightweight UAV navigation, Ref [16] introduced a method that estimates tilt and velocity by fusing tachometer and IMU data. Although effective, this approach still relies on auxiliary sensors, limiting its applicability in environments where such sensors are unavailable or unreliable.

B. Learning Inertial Odometry

Recent advances in deep learning have sparked research in data-driven methods for learning velocity and displacement from inertial data [8], [10], [17], [18]. Approaches such as IONet [17], A2DIO [19], and NILoc [18] formulate inertial navigation as a sequential task by predicting relative position

displacements using IMU data. Subsequent work has incorporated these learned displacements into EKF [7], [20], [21] or batched optimization frameworks [22]. To ensure consistency in IMU measurements, many of these methods transform the input IMU data into global coordinates using ground-truth orientation. For pedestrian navigation, RoNIN [8] proposed a Heading-Agnostic Coordinate Frame (HACF), aligning the accelerometer's gravity vector with the z-axis and constraining rotation to the horizontal plane. This framework has been widely adopted for learning-based IO [7], [20] due to its effectiveness in simplifying pedestrian motion. Building on HACF, RIO [23] introduced rotation-equivalence data augmentation to self-supervise pose estimation after orientation alignment. Since pedestrians and wheeled vehicles predominantly move along the horizontal plane, HACF effectively removes yaw dependency from the representation, thereby reducing learning complexity and improving generalizability. Despite the success in pedestrian navigation and wheeled robots [8], [24], these representations are less effective for multirotor UAV motion, which requires more sophisticated dynamic modeling. In this paper, we show that the commonly used global coordinate representations hinder neural networks from effectively capturing the dynamic complexities of UAVs.

C. Inertial Odometry for Multirotor UAVs

Existing learning-based IO methods have primarily focused on pedestrian datasets, limiting their generalizability to platforms with more demanding motion dynamics, such as UAVs. Approaches like AI-driven pre-processing and down-sampling of high-speed inertial data have been proposed for better state estimation [25]. Due to its simplicity, most of the existing methods continue adopting the global coordinate frame, such as [6], [14], [26]. DIDO [14] estimates the thrust of UAVs using tachometer data and addresses unmodeled forces by incorporating a neural network. Building on this, the IMO [6] incorporates thrust information and IMU data within an EKF framework, which implicitly captures the drone dynamics. However, IMO relies on additional control signals to capture the drone's dynamics, as shown in Fig. 8, which may suffer from overfitting on training datasets. DIVE [26], a more recent work for UAV IO, encodes orientation alongside gravity-removed global-frame acceleration. While this representation improves the robustness of learning-based IO, our findings in Section. V indicate that it remains suboptimal for capturing UAV dynamics comprehensively.

III. METHODOLOGY

The goal of learning IO is to estimate relative position transform (velocity) from the IMU data in the local body IMU frame. Most existing methods employ transforming the IMU frame to global frame or removing gravity from the raw IMU data [8], [23], [26]. Our study reveals that these widely used representations, such as global coordinate frames, HACF, and gravity-removed frames, limit the effectiveness of IMU feature extraction. Thus, this section proposes a different feature representation and the corresponding state estimation design. It consists of three parts: feature representation analysis, attitude-encoded network design, and a tightly coupled EKF system.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

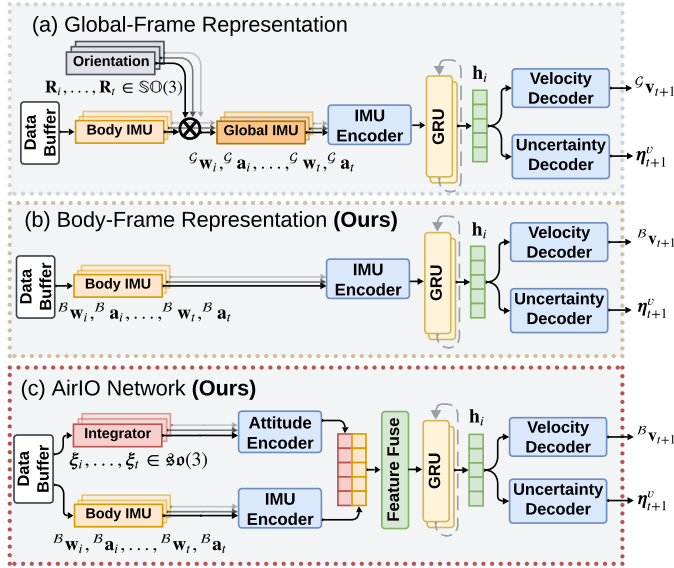


Fig. 2: (a) Existing learning IO usually transforms the IMU input to the global coordinate using ground-truth orientation. (b) We found that preserving the IMU data in the body coordinate, along with gravitational information, can significantly improve accuracy. (c) Our AirIO model leverages the body-frame IMU and explicitly encodes the UAV’s orientation to estimate the body-frame velocity.

A. IMU Coordinate Frame

Canonical Body Coordinate Frame: The IMU accelerometer measures the net force per unit mass acting on the sensor. Due to the measurement mechanism, it measures not only the object actual accelerations due to motion ${}^B\dot{\mathbf{v}} = \frac{{}^B\mathbf{F}_{net_i}}{m}$ in the sensor’s local coordinate frame (usually aligned with object body frame), but also the constant acceleration due to Earth’s gravity ${}^G\mathbf{g}$ directed toward the Earth’s center. Thus, the accelerometer measurement at timestamp i as:

$${}^B\mathbf{a}_i = \frac{{}^B\mathbf{F}_{net_i}}{m} + \mathbf{R}_i^\top {}^G\mathbf{g}, \quad (1)$$

where m is the object mass, and $\mathbf{R}_i \in \mathbb{SO}(3)$ is the rotation for vector from body frame \mathcal{B} to the global frame \mathcal{G} . $\mathcal{F}(\cdot)$ denotes quantity represented in frame \mathcal{F} .

Global Coordinate Frame: Many existing methods transform the IMU data from body frame \mathcal{B} to HACF or global frame \mathcal{G} by applying an estimated transform rotation $\hat{\mathbf{R}}_i$, e.g.

$${}^G\mathbf{a}_i = \hat{\mathbf{R}}_i \frac{{}^B\mathbf{F}_{net_i}}{m} + \hat{\mathbf{R}}_i \mathbf{R}_i^\top {}^G\mathbf{g} \quad (2)$$

The rotation $\hat{\mathbf{R}}_i$ can often be obtained by simple IMU preintegration or the estimation results from an EKF, which is often accurate enough, such that $\hat{\mathbf{R}}_i \mathbf{R}_i^\top \approx \mathbf{I}$.

Notice that both (1) and (2) represents the same physical process. The key difference lies in how the attitude information is coupled with different components: in (1), it is coupled with the gravitational force, whereas in (2), it is coupled with the motion force. In (1), since the gravity acceleration ${}^G\mathbf{g}$ is a constant, the the motion force and attitude form a linear combination. In contrast. in (2), the attitude couples with motion force in a nonlinear manner, with gravity acting as an additional constant that does not contribute extra useful information. Thus, the motion force and attitude are intuitively

easier to estimate in (1), while (2) requires a more complex network structure for estimation.

B. Representation Analysis

We perform feature analysis to verify our intuition. Specifically, we extract the latent features \mathbf{h}_i from the IMU feature encoder for the input under different representations in Fig. 2. We conduct multiple feature analyses to assess their effectiveness and representativeness.

Principal Component Analysis PCA is a statistical method that reduces the dimensionality of a data set by replacing correlated variables with a smaller set of uncorrelated principal components. To analyze the expressivity of the input representation, we performed PCA on the feature latent space for different input coordinates. We concatenate the latent features from all samples to form a feature matrix and perform PCA on the latent feature matrix to analyze its underlying structure by computing the singular value of the normalized feature matrix in a numerically stable way. Each singular value represents the corresponding magnitude scaling of one principal feature component. We then quantify the cumulative explained variance (or the energy coverage) for the top k principal components in PCA. Specifically, we evaluate on two datasets: the widely used UAV dataset Blackbird [27] collected for agile autonomous flight, which covers large UAV maneuvers, and our custom simulation dataset Pegasus, which was collected in the NVIDIA IssacSim simulator with Pegasus autopilot [28], featuring diverse maneuvers in an ideal environment without external disturbance, allowing us to work with cleaner data.

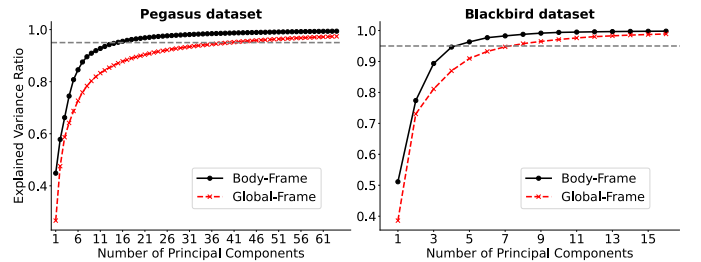


Fig. 3: The cumulative explained variance of the latent features for the Pegasus dataset (Left) and Blackbird dataset (Right). In both datasets, the latent features derived from body frames (black line) require fewer principal components to achieve comparable total energy.

Fig. 3 shows the cumulative energy coverage of the latent feature derived from two different IMU input representations: global and body frames. The red and black curves represent the energy coverage of the top k principal components. Specifically, for the Blackbird dataset, it shows that the top $k = 5$ principal components in body-frame representation already cover 95% of the total energy. In contrast, global-frame representation requires the top $k = 8$ principal components to achieve the same level of energy. Similarly, in the Pegasus datasets, the body-frame representation requires top $k = 15$ principal components and the global frame require top $k = 40$ principal components to cover 95% of the total energy. It clearly shows that models trained on the body representation input can achieve comparable performance with

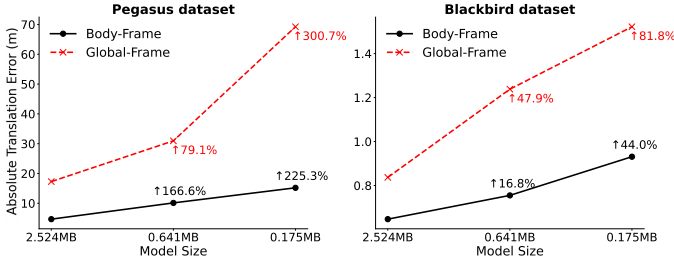


Fig. 4: The ATE for body-frame and global-frame models at different model sizes on Pegasus dataset (Left) and Blackbird dataset (Right). The text denotes the relative percentage increase in ATE.

fewer features, highlighting the efficiency and expressivity of this representation in capturing the essential features. It also implies that models trained under body frame have better compressibility and can be designed more lightweight. Fig. 4 shows the comparison of the absolute translation error (ATE) for the models at different scales under both body and global frames. As the model size decreases, the prediction performance for model under body-frame representation degrades more smoothly and consistently yields lower errors. See Appendix. G for more studies on the model compressibility.

t-SNE [29] We also perform t-SNE on Pegasus dataset, a dimensionality reduction approach that projects high-dimension data into a low-dimensional map to analyze the feature representation qualitatively. As shown in Fig. 5, based on the network prediction (here is the velocity), we color code each data point by the velocity magnitude from low to high. In the global frame, the points exhibit a highly entangled distribution and different velocity levels overlap, indicating poor separability and less representative encoding. In contrast, the feature distribution of the body coordinate frame is well-separated, where different velocity levels form more coherent clusters. This suggests that the body frame facilitates a more discriminative and effective representation of latent features.

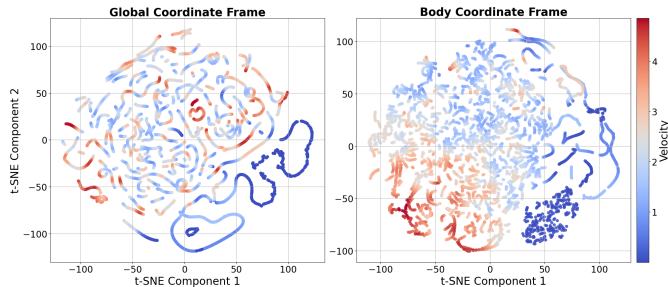


Fig. 5: t-SNE analysis on Pegasus dataset with each point colored based on the velocity magnitude from low to high, represented by blue to red.

In summary, the analysis from Section. III-A and Section. III-B indicates that i). Good prior information will help for the estimation since it couples with the motion force in a linear or nonlinear manner, as shown in (1)(2). ii). The nonlinear coupling between the motion force and attitude in the global-frame representation complicates the estimation problem. Body-frame representation proves to be more expressive and discriminative, leading to a light network for prediction. Notice that while the analysis is indeed applicable to general robots via standard IMU equations, UAVs present a compelling test case due to their agile motion. Unlike many mobile or

legged robots with mainly yaw-dominant movement, UAVs often exhibit significant roll and pitch maneuvers, resulting in more pronounced impact of frame representation difference.

C. AirIO Motion Network & Training

Based on the previous analysis, we introduce a separate encoding of the drone's attitude $\xi \in \mathfrak{so}(3)$ to the motion network beyond the IMU encoder with body-frame representation, shown in Fig. 2(c). After fusing this information, the network predicts the body-frame velocity ${}^B\hat{v}$ and the corresponding uncertainty $\hat{\eta}^v$. The motion network can be written as: $({}^B\hat{v}, \hat{\eta}^v) = f_{\theta}({}^B\mathbf{w}, {}^B\mathbf{a}, \xi)$.

Inspired by [30], we map the drone's orientation to the $\mathfrak{so}(3)$ Lie algebra space for compact and continuous representation of 3D rotations and smoother network gradient to mitigate numerical instability. We then leverage a CNN encoder to encode $\mathfrak{so}(3)$.

After concatenating the features from the attitude encoder and the IMU encoder, we employ bi-directional GRU layers to extract the latent feature \mathbf{h}_i and model the temporal dependencies in the drone dynamics. We leverage this feature to estimate the body-frame velocity and the corresponding uncertainty by two MLP decoders.

To jointly supervise the velocity and the corresponding uncertainty ${}^B\hat{v}_i$, we designed a combined loss function:

$$\mathcal{L} = \mathcal{L}_{\text{Huber}} + \lambda \mathcal{L}_C, \quad (3)$$

where the $\lambda = 1e^{-4}$. The $\mathcal{L}_{\text{Huber}}$ is the Huber loss between the predicted body-frame velocity ${}^B\mathbf{v}_i$ and the body-frame ground-truth velocity ${}^B\hat{v}_i$, defined as:

$$\mathcal{L}_{\text{Huber}} = \begin{cases} \frac{1}{2} ({}^B\mathbf{v}_i - {}^B\hat{v}_i)^2 & \text{if } |{}^B\mathbf{v}_i - {}^B\hat{v}_i| < \delta \\ \delta \cdot (|{}^B\mathbf{v}_i - {}^B\hat{v}_i| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}, \quad (4)$$

where we set the $\delta = 0.005$.

To supervise the uncertainty of the velocity, we employ an uncertainty loss inspired by [7], assuming that the estimated velocity uncertainty follows a Gaussian distribution:

$$\mathcal{L}_C = ({}^B\mathbf{v}_i - {}^B\hat{v}_i) \hat{\Sigma}_i^{v-1} ({}^B\mathbf{v}_i - {}^B\hat{v}_i)^T + \ln(\det \hat{\Sigma}_i^v), \quad (5)$$

where $\hat{\Sigma}_i^v$ is 3×3 covariance matrix for the i^{th} data. Similar to the setting in [7], we simplified the covariance as $\hat{\Sigma}_i^v = \text{diag}(\hat{\eta}_i^{v2})$, where $\hat{\eta}_i^v$ is learned uncertainty from network.

For orientation ξ , we use the ground truth during training and replace it with EKF-estimated values during testing.

D. Extended Kalman Filter

Different from the existing IO methods [6], [7] which employ a motion network to capture relative translation and maintain historic states to optimize relative pose for state pairs. AirIO network models the body-frame velocity. As a result, we only need to constrain the velocity of the current frame. which significantly reduces the number of state parameters and alleviates the need for frame culling. This setup isolates the current state of the drone from the previous state, which simplifies the EKF structure. The full state of the filter is $\mathbf{X} = (\mathbf{R}_i, {}^G\mathbf{v}_i, {}^G\mathbf{p}_i, \mathbf{b}_{a_i}, \mathbf{b}_{g_i})$, where \mathbf{b}_{a_i} and

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

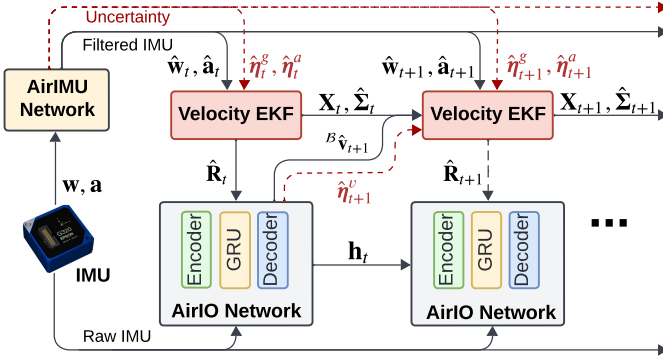


Fig. 6: AirIO system pipeline. A tightly coupled EKF that fuses the AirIO motion network with a learning-based IMU preintegration network AirIMU. Instead of using constant uncertainty parameters, we employ learned uncertainty $\hat{\eta}^v$ from the AirIO network and the learned gyroscope uncertainty $\hat{\eta}^g$ and accelerometer uncertainty $\hat{\eta}^a$ from the AirIMU in our EKF system.

\mathbf{b}_{g_i} are the bias of the accelerometer and the gyroscope respectively. Following the common setup of the error-based filtering method, we define the error-state of the current filter as $\delta \mathbf{X} = (\delta \xi_i, \delta^G \mathbf{v}_i, \delta^G \mathbf{p}_i, \delta \mathbf{b}_{a_i}, \delta \mathbf{b}_{g_i})$. For the rotation error, we define $\delta \xi_i = \log_{SO3}(\mathbf{R}_i \hat{\mathbf{R}}_i^{-1}) \in \mathfrak{so}(3)$, where $\log_{SO3}(\cdot)$ denotes logarithm map operator.

AirIMU feature correction and uncertainty estimation

To filter the noise and, more importantly, quantify the uncertainty of the IMU pre-integration, we leverage AirIMU [30] to pre-process the raw IMU data. This model estimates the corrections of the gyroscope and the accelerometer, and also the uncertainty of the IMU sensor:

$$(\hat{\sigma}_{g_i}, \hat{\sigma}_{a_i}) = g_{\theta}(\mathbf{w}_i, \mathbf{a}_i), \quad (\hat{\eta}_{g_i}, \hat{\eta}_{a_i}) = \Sigma_{\theta}(\mathbf{w}_i, \mathbf{a}_i), \quad (6)$$

where the θ is the parameter of the neural network. The AirIMU network is trained by a differentiable IMU integrator and covariance propagator. We later filtered the IMU measurement by the predicted correction $\hat{\mathbf{a}} = \mathbf{a} + \hat{\sigma}_{a_i}$, $\hat{\mathbf{w}} = \mathbf{w} + \hat{\sigma}_{g_i}$. In the EKF system, the learned uncertainty $(\hat{\eta}_{g_i}, \hat{\eta}_{a_i})$ is leveraged in the sensor covariance modeling.

Filter Propagation The kinematic motion model is:

$$\begin{aligned} \mathbf{R}_{i+1} &= \mathbf{R}_i \cdot \text{Exp}(\hat{\mathbf{w}}_i - \mathbf{b}_{g_i}) \Delta t, \\ \mathbf{v}_{i+1} &= \mathbf{v}_i + \mathbf{R}_i (\hat{\mathbf{a}}_i - \mathbf{b}_{a_i}) \Delta t, \\ \mathbf{p}_{i+1} &= \mathbf{p}_i + \mathbf{v}_i \Delta t + \frac{1}{2} \Delta t^2 \mathbf{R}_i (\hat{\mathbf{a}}_i - \mathbf{b}_{a_i}), \\ \mathbf{b}_{a_{i+1}} &= \mathbf{b}_{a_i}, \mathbf{b}_{g_{i+1}} = \mathbf{b}_{g_i}, \end{aligned} \quad (7)$$

Here, the $\text{Exp}(\cdot)$ denotes mapping from the log space to exponential space. The linearized propagation model is:

$$\delta \mathbf{X}_{i+1} = \mathbf{A}_i \cdot \delta \mathbf{X}_i + \mathbf{B}_i \cdot \mathbf{n}_i, \quad (8)$$

where $\mathbf{n}_i = [\hat{\eta}_{g_i}, \hat{\eta}_{a_i}, \eta_{b_g}, \eta_{b_a}]^T$. The $\hat{\eta}_{g_i}, \hat{\eta}_{a_i}$ are the learned IMU uncertainty from the AirIMU network [30] with the i -th frame. The η_{b_g}, η_{b_a} are the random walk uncertainty set as a constant across different frames. The corresponding covariance propagation of the state covariance \mathbf{P}_{i+1} follows:

$$\mathbf{P}_{i+1} = \mathbf{A}_i \mathbf{P}_i \mathbf{A}_i + \mathbf{B}_i \mathbf{W}_i \mathbf{B}_i \quad (9)$$

where the \mathbf{W}_i is the covariance matrix of the i -th input, including the learned uncertainty of the IMU $\hat{\eta}_{g_i}, \hat{\eta}_{a_i}$ and sensor bias random walk η_{b_g}, η_{b_a} .

Filter Update The measurement update is as follows:

$$h(\mathbf{X}) = \hat{\mathbf{R}}_i^T \cdot \mathbf{v}_i = \hat{\mathbf{v}}_i + \hat{\eta}_{v_i}. \quad (10)$$

where η_{v_i} is a random variable that follow the Gaussian distribution $\mathcal{N}(0, \hat{\Sigma}_i^v)$ learned by the network. The Jacobian matrix \mathbf{H} is computed as:

$$\begin{aligned} \mathbf{H}_{\delta v_i} &= \frac{\partial h(\mathbf{X})}{\partial \delta \mathbf{v}_i} = \hat{\mathbf{R}}_i^T \\ \mathbf{H}_{\delta \xi_i} &= \frac{\partial h(\mathbf{X})}{\partial \delta \xi_i} = \hat{\mathbf{R}}_i^T [\mathbf{v}_i]_{\times} \end{aligned} \quad (11)$$

Finally, the Kalman Gain and the update can be computed:

$$\begin{aligned} \mathbf{K} &= \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T + \hat{\Sigma}_i^v)^{-1} \\ \mathbf{X} &\leftarrow \mathbf{X} \oplus (\mathbf{K} (h(\mathbf{X}) - \hat{\mathbf{v}}_i)) \\ \mathbf{P} &\leftarrow (\mathbf{I} - \mathbf{K} \mathbf{H}) \mathbf{P} (\mathbf{I} - \mathbf{K} \mathbf{H})^T + \mathbf{K} \hat{\Sigma}_i^v \mathbf{K}^T \end{aligned} \quad (12)$$

Operator \oplus denotes the additional operation except for the orientation, where the update performs $\mathbf{R} \leftarrow \text{Exp}(\xi) \cdot \mathbf{R}$.

IV. EXPERIMENT

A. Experiment Setup

Datasets We evaluate the proposed approach on two quadrotor datasets: the real-world EuRoC dataset [31] and the challenging drone racing dataset Blackbird [27]. EuRoC datasets are a widely used benchmark for evaluating odometry. Blackbird dataset is an indoor flight dataset, presenting more aggressive maneuvers and high-speed dynamics. For ablation studies, we additionally include our custom simulated dataset collected using the NVIDIA IsaacSim simulator with Pegasus autopilot [28]. The Pegasus dataset also presents diverse maneuvers but in the ideal environment without external disturbance, allowing us to work with cleaner data for ground-truth accuracy.

Baseline Several state-of-the-art methods are selected for comparison. For model-based IO, we include *Baseline* [13], the IMU preintegration approach in raw IMU data, and *Air-IMU* [30], a hybrid method that corrects the raw IMU noise with a data-driven method and integrates it with the IMU kinematic function. For learning-based IO, we compare against an end-to-end trained model *RoNIN* [8] and an EKF-fused algorithm *TLIO* [7]. For learning-based IO methods designed for UAVs, we evaluate *IMO* [6], a control signal-embedded IO specifically developed for drone racing, which incorporates additional thrust signals to better capture drone motion.

Evaluation Metrics We use the following metrics for evaluation: 1) **Absolute Translation Error** (ATE, m): the average Root Mean Squared Error (RMSE) between the estimated and ground-truth positions over all time points:

$$\text{ATE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_i - \hat{\mathbf{p}}_i\|_2^2}, \quad (13)$$

2) **Relative Translation Error** (RTE, m): the average RMSE of the relative displacements over predefined time intervals.

$$\text{RTE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left\| \mathbf{p}_{i+\Delta t} - \mathbf{p}_i - \mathbf{R}_i \hat{\mathbf{R}}_i^T (\hat{\mathbf{p}}_{i+\Delta t} - \hat{\mathbf{p}}_i) \right\|_2^2}, \quad (14)$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I: The ATE (Unit: m) and RTE (Unit: m) on the Blackbird dataset. **Seen** are sequences where the training and testing datasets are derived from the same trajectory. **Unseen** are those where the testing dataset includes sequences never used in training, demonstrating the model’s generalizability.

Seq.	Baseline [‡]		AirIMU [‡]		RoNIN [†]		TLIO [†]		IMO ^{†*}		AirIO Net		AirIO EKF		
	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE	
Seen	Clover	15.769	2.027	6.163	0.294	3.074	1.367	1.464	0.797	0.381	0.681	0.434	0.368	0.367	0.391
	Egg	66.17	7.677	13.293	3.801	2.449	2.552	2.227	2.398	1.153	0.828	0.713	0.391	0.408	0.344
	halfMoon	19.165	3.386	4.174	0.746	1.263	0.864	0.956	0.475	0.761	0.24	0.491	0.256	0.457	0.253
	Star	20.49	4.556	4.347	1.933	3.15	3.266	0.68	0.784	2.13	3.066	0.442	0.401	0.477	0.341
	Winter	15.781	2.395	7.445	0.743	1.031	1.089	0.616	0.755	0.219	0.206	0.348	0.147	0.307	0.164
Avg.	27.475	4.008	7.084	1.503	2.193	1.828	1.189	1.042	0.929	1.004	0.486	0.312	0.403	0.299	
Unseen	Ampersand	41.138	5.026	24.738	4.379	5.467	5.321	4.665	4.078	17.664	10.039	2.303	1.145	2.334	1.154
	Sid	13.108	2.055	16.832	2.131	18.581	10.665	7.323	8.427	9.967	6.992	0.717	0.527	0.874	0.49
	Oval	15.365	2.4	8.312	1.553	20.399	12.12	1.276	1.045	5.67	2.863	0.976	0.583	0.828	0.54
	Sphinx	3.429	2.713	2.828	2.024	11.537	7.762	2.005	2.408	5.629	5.395	1.145	1.008	1.178	1.033
	BentDice	28.307	2.342	54.261	4.39	11.453	6.792	2.119	1.747	6.143	4.519	1.360	1.000	1.331	0.955
Avg.	20.269	2.907	21.403	2.896	13.487	8.532	3.478	3.541	9.015	5.962	1.300	0.853	1.309	0.834	

* Leverage thrust information as the input data for training and inference.

† Leverage ground truth orientation to transform the input data. ‡ Dead-reckoning IMU pre-integration.

Our evaluation adopts a 5-second interval configuration.

We define the improvement percentage as the relative reduction in error compared to the baseline. Specifically, it is computed as the difference between the error of baseline and the proposed method, divided by the baseline error.

Training Details We use the Adam optimizer with an initial learning rate of 0.001. A learning rate scheduler is implemented following the ReduceLROnPlateau, with a patience of 5 epochs and a decay factor of 0.2. The batch size is 128. Our training and testing window size is 1000 frames (equal to 5 seconds in the EuRoC dataset). In the CNN encoder, we include a dropout layer with $p = 0.5$ to reduce overfitting.

B. Results

1) *Evaluation on Blackbird Dataset:* We employ the same sequences from Blackbird as used in IMO [6] and adopt their same train-test split configuration. We name these sequences as **SEEN** sequences because the training and testing data are from the same sequence. To evaluate generalization, we select five additional trajectories from BlackBird dataset as **UNSEEN** sequences, which are excluded from the training set. More details on separation are in Appendix. B1.

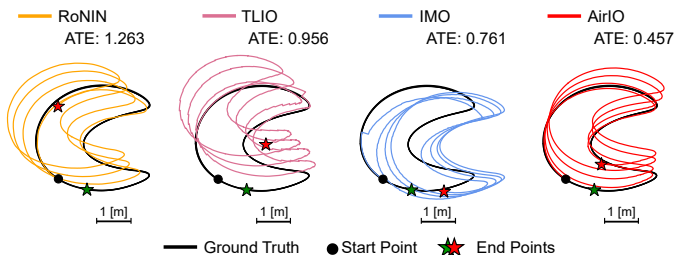


Fig. 7: Trajectories of **SEEN** sequence halfMoon from the Blackbird dataset. AirIO, relying solely on IMU, outperforms IMO by 30% in ATE.

For the seen sequences, as shown in Table. I, RoNIN and TLIO fail to capture the aggressive drone maneuvers as they are designed for pedestrian navigation with clear slow motion patterns. AirIO significantly outperforms these baselines, improving by 66.1% in ATE and 71.3% in RTE over TLIO. Furthermore, AirIO exceeds IMO by 56.6% in ATE and 70.2% in RTE relying solely on IMU measurement without integrating external thrust information. We also show that our algorithm can outperform the multi-sensor fusion algorithms

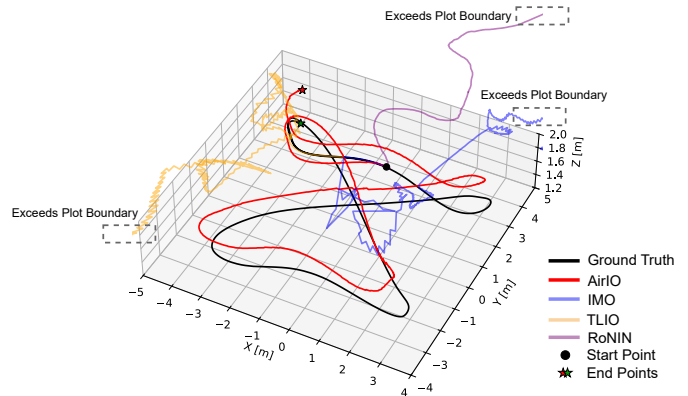


Fig. 8: Trajectories of **UNSEEN** sequence sid from the Blackbird dataset by RoNIN, TLIO, IMO, and AirIO.

like the visual-inertial odometry on the drone racing data. For more details, please check Appendix. H.

In addition, it is worth noting that AirIO demonstrates remarkable generalizability to unseen sequences, as shown in Fig. 8. While other methods suffer from significant drift, AirIO consistently maintains precise estimations. The baseline methods tend to overfit the training data, performing well within the training distribution but failing to generalize to UNSEEN data. Compared to IMO, AirIO shows a substantial improvement of 85.5% in ATE and 86.0% in RTE.

2) *Evaluation on EuRoC Dataset:* We then evaluate the proposed method on EuRoC dataset, as shown in Table. II. AirIO outperforms both RoNIN and TLIO by 52.9% and 54.4% in ATE, respectively. Thanks to effective uncertainty modeling, AirIO leverages uncertainties learned from both the IMU kinematic model and motion network, ensuring optimal data fusion and improved performance. After integrating EKF, AirIO further improves accuracy, showing an average additional 17.4% improvement in ATE.

TABLE II: The ATE (Unit: m) and RTE (Unit: m) on EuRoC dataset.

Seq.	RoNIN [†]		TLIO [†]		AirIO Net		AirIO EKF	
	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE
MH02	5.902	2.121	7.281	2.451	4.917	0.936	2.478	0.987
MH04	8.586	4.542	8.626	5.498	2.726	1.093	2.308	1.005
V103	3.240	1.695	7.863	2.580	3.844	1.519	3.05	1.552
V202	7.445	2.303	6.260	2.783	4.823	1.303	4.206	1.313
V101	8.576	1.918	4.814	1.946	2.917	1.137	3.844	1.103
Avg.	6.75	2.516	6.969	3.052	3.846	1.198	3.177	1.192

† Leverage ground truth orientation to transform the input data for inference.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

3) *Real time deployment and Analysis*: We evaluate the real-time performance of our algorithm on both an NVIDIA Jetson Orin AGX and a desktop with a low-end RTX 2060 GPU, using an IMU operating at 200 Hz. On Orin, AirIO achieves an average runtime of 74.23 ms, while on the desktop, it runs in 28.92 ms. These results indicate the feasibility of our algorithm for real drone deployment.

V. ABLATION STUDY

We perform an ablation study to investigate the impact of different feature representations. Experiments with the following input representation scenarios are conducted on the real-world dataset EuRoC and the simulated dataset Pegasus.

- i. **Body**: The network takes body-frame IMU measurements defined in (1) as input.
- ii. **Global**: The network takes global-frame IMU measurements as input. The global-frame IMU is transformed from raw IMU data, as defined in (2).
- iii. **Body + Attitude**: Building upon *i.*, the drone orientation is encoded as auxiliary inputs to the network.
- iv. **Global + Attitude**: Building upon *ii.*, the drone orientation is encoded as auxiliary inputs to the network.
- v. **Body – $\mathbf{R}_i^T \mathbf{g}$** : Building upon *i.*, we consider the motion acceleration by removing the rotation-coupled gravity term $\mathbf{R}_i^T \mathbf{g}$ from (1).
- vi. **Global – \mathbf{g}** : Building upon *ii.*, we consider the motion acceleration by removing the gravity term \mathbf{g} from (2).

To ensure a fair comparison, scenarios *i.*, *iii.*, and *v.* leverage ground-truth orientation to transform estimated velocity into global frame.

Table. III shows the results of the ablation study for all variants. More results can be found in Appendix. F. From these results, we address the following questions:

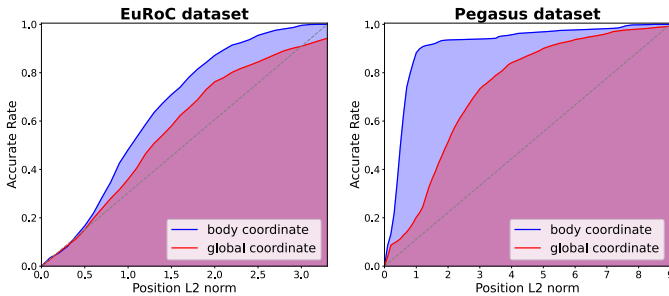


Fig. 9: Accuracy AUC of EuRoC dataset (Left) and Pegasus dataset (Right).

A. How Effective is the Body-frame Representation?

Compared to *ii.* Global, *i.* Body demonstrates significantly higher precision on the Pegasus. This new input representation results in an average ATE improvement of 74.1%. Scenario *i.* on EuRoC dataset and Blackbird Unseen sequences exhibit similar trends.

We further evaluate the system performance under two representation using the Accuracy Area Under the Curve (AUC) metric, which quantifies the performance across various error thresholds. Specifically, we calculate the relative position error over a 5-second interval. As shown in Fig. 9, on both datasets,

TABLE III: Ablation study on the EuRoC and Pegasus datasets comparing different feature representations. Evaluation metric: ATE (Unit: m).

	Seq.	Global (2) – \mathbf{g}	Global	Body (1) – $\mathbf{R}_i^T \mathbf{g}$	Global +Attitude	Body	Body +Attitude
EuRoC	MH02	18.048	17.892	16.136	9.522	4.537	2.531
	MH04	15.904	6.895	8.977	4.572	3.561	2.250
	V103	7.007	7.086	6.586	4.068	6.824	3.107
	V202	9.662	3.496	6.161	10.274	3.248	4.310
	V101	8.142	15.11	7.082	7.117	5.478	4.287
	Avg.	11.753	10.096	8.988	7.110	4.730	3.297
Pegasus	TEST_1	22.115	9.689	9.534	6.151	5.225	3.418
	TEST_2	15.745	15.185	14.729	5.373	4.489	5.719
	TEST_3	19.467	26.960	3.580	14.545	3.734	1.786
	Avg.	19.109	17.278	9.281	8.690	4.483	3.641
Blackbird	Amperсанд	21.93	17.863	18.639	14.487	1.977	2.492
	Sid	11.866	7.915	5.686	2.307	1.108	0.651
	Oval	7.212	6.743	1.656	7.354	1.353	0.913
	Sphinx	3.327	3.572	2.117	2.784	1.266	1.163
	BentDice	15.5	8.876	6.989	9.626	1.915	1.249
	Avg.	11.967	8.994	7.018	7.312	1.524	1.294

scenario *i.* achieves higher accuracy across all error thresholds and reaches near-perfect accuracy much more quickly. These results highlight the effectiveness of body-frame representation which efficiently captures implicit attitude information of IMU data. Simply encoding the body frame IMU features improves the system performance.

B. How Effective is the Attitude Encoding?

As illustrated in Fig. 10, on the EuRoC dataset, *iii.* Body + Attitude achieves a further 30.3% improvement over the scenario *i.*, outperforming scenario *ii.* by 67.3% in ATE. Additionally, we evaluate attitude encoding on global-frame representation *iv.* Global + Attitude. This also yields an improvement of approximately 29.6% over scenario *ii.* on EuRoC dataset, validating the efficiency of encoding attitude information. Encoding the drone’s attitude information enhances the network’s ability for state estimation.

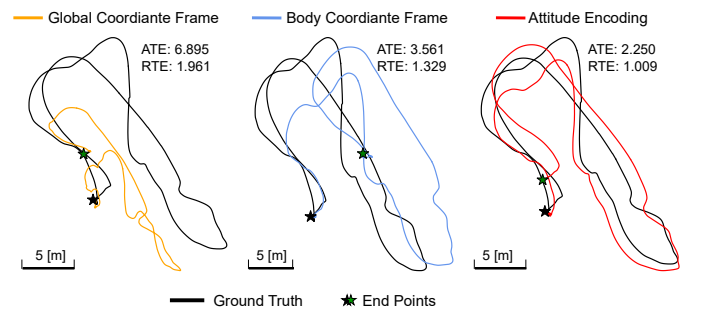


Fig. 10: Performance of MH_04_difficult in EuRoC dataset across different representations. *ii.* Global (Left) is suboptimal. Using IMU data in *i.* Body (Center) improves significantly. Encoding UAV orientation *iii.* Body + Attitude (Right) further improves the performance.

C. Shall We Keep Gravity in the Feature Representation?

Many learning-based IO methods remove gravity from the IMU encoding [26]. However, as discussed in Section. III-A, gravitational acceleration is a critical component for body-frame representation as it explicitly embeds drone rotation through the term $\mathbf{R}_i^T \mathbf{g}$, as shown in 1. After removing this rotation-coupled gravity term, scenario *i.* loses essential

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

attitude information. Table. III demonstrates that \mathbf{v} . Body – $\mathbf{R}_i^{Tg} \mathbf{g}$ causes significant performance degradation: the ATE increases by 90.0% on EuRoC, 107.0% on Pegasus dataset, and 360.5% on Blackbird’s Unseen sequences.

VI. CONCLUSION & DISCUSSION

In this work, we investigate an effective representation of learning-based IO for multirotor UAVs and propose a solution that relies solely on IMU data. We identify that the commonly used global-frame representation is less effective for dynamic, agile maneuvers due to the less observable attitude information compared to IMU data represented in body-frame. Based on this, we develop the AirIO system by explicitly encoding attitude information and integrating it with an EKF. We show that our approach outperforms the state of the art [6] in various scenarios and environments without relying on additional sensors or control information. Our method achieves on average 66.7% improvement in accuracy. Furthermore, explicitly encoding attitude information into the motion network results in an additional 23.8% improvement. We validate the real-time capability of AirIO on an NVIDIA Jetson AGX Orin. Future work includes deploying the system on real UAVs for closed-loop evaluation.

REFERENCES

- [1] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, “Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8729–8736.
- [2] Y. Hu, J. Geng, C. Wang, J. Keller, and S. Scherer, “Off-policy evaluation with online adaptation for robot exploration in challenging environments,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 6, pp. 3780–3787, 2023.
- [3] X. Guo, G. He, J. Xu, M. Mousaei, J. Geng, S. Scherer, and G. Shi, “Flying calligrapher: Contact-aware motion and force planning and control for aerial manipulation,” *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 11194–11201, 2024.
- [4] S. Zhao, Y. Gao, T. Wu, D. Singh, R. Jiang, H. Sun, M. Sarawata, Y. Qiu, W. Whittaker, I. Higgins, Y. Du, S. Su, C. Xu, J. Keller, J. Karhade, L. Nogueira, S. Saha, J. Zhang, W. Wang, C. Wang, and S. Scherer, “Subt-mrs dataset: Pushing slam towards all-weather environments,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 22647–22657.
- [5] Y. Qiu, C. Wang, W. Wang, M. Henein, and S. Scherer, “Airdos: Dynamic slam benefits from articulated objects,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8047–8053.
- [6] G. Cioffi, L. Bauersfeld, E. Kaufmann, and D. Scaramuzza, “Learned inertial odometry for autonomous drone racing,” in *IEEE Robotics and Automation Letters (RA-L)*, 2023.
- [7] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, “Thio: Tight learned inertial odometry,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5653–5660, 2020.
- [8] S. Herath, H. Yan, and Y. Furukawa, “Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods,” in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 3146–3152.
- [9] R. Buchanan, V. Agrawal, M. Camurri, F. Dellaert, and M. Fallon, “Deep imu bias inference for robust visual-inertial odometry with factor graphs,” *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 41–48, 2022.
- [10] H. Yan, Q. Shan, and Y. Furukawa, “Ridi: Robust imu double integration,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 621–636.
- [11] S. Yang, Z. Zhang, B. Bokser, and Z. Manchester, “Multi-imu proprioceptive odometry for legged robots,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 774–779.
- [12] X. Teng, P. Xu, D. Guo, Y. Guo, R. Hu, H. Chai, and D. Chuxing, “Arpdr: An accurate and robust pedestrian dead reckoning system for indoor localization on handheld smartphones,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10888–10893.
- [13] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation,” in *Robotics: Science and Systems XI*, 2015.
- [14] K. Zhang, C. Jiang, J. Li, S. Yang, T. Ma, C. Xu, and F. Gao, “Dido: Deep inertial quadrotor dynamical odometry,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9083–9090, 2022.
- [15] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE transactions on robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [16] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, “Imu-based inertia estimation for a quadrotor using newton-euler dynamics,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3861–3867, 2020.
- [17] C. Chen, X. Lu, A. Markham, and N. Trigoni, “Ionet: Learning to cure the curse of drift in inertial odometry,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [18] S. Herath, D. Caruso, C. Liu, Y. Chen, and Y. Furukawa, “Neural inertial localization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6604–6613.
- [19] Y. Wang, H. Cheng, and M. Q.-H. Meng, “A2dio: Attention-driven deep inertial odometry for pedestrian localization based on 6d imu,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 819–825.
- [20] S. Sun, D. Melamed, and K. Kitani, “Idol: Inertial deep orientation-estimation and localization,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, 2021, pp. 6128–6137.
- [21] X. Deng, S. Wang, C. Shan, J. Lu, K. Jin, J. Li, and Y. Guo, “Data-driven based cascading orientation and translation estimation for inertial navigation,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 3381–3388.
- [22] R. Buchanan, M. Camurri, F. Dellaert, and M. Fallon, “Learning inertial odometry for dynamic legged robot state estimation,” in *Conference on robot learning*. PMLR, 2022, pp. 1575–1584.
- [23] X. Cao, C. Zhou, D. Zeng, and Y. Wang, “Rio: Rotation-equivariance supervised learning of robust inertial odometry,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6614–6623.
- [24] M. Brossard, A. Barrau, and S. Bonnabel, “Rins-w: Robust inertial navigation system on wheels,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2068–2075.
- [25] J. Steinbrener, C. Brommer, T. Jantos, A. Fornasier, and S. Weiss, “Improved state propagation through ai-based pre-processing and down-sampling of high-speed inertial data,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6084–6090.
- [26] A. Bajwa, C. C. Cossette, M. A. Shalaby, and J. R. Forbes, “Dive: Deep inertial-only velocity aided estimation for quadrotors,” *IEEE Robotics and Automation Letters*, 2024.
- [27] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, and S. Karaman, “The blackbird dataset: A large-scale dataset for uav perception in aggressive flight,” in *International Symposium on Experimental Robotics*. Springer, 2018, pp. 130–139.
- [28] M. Jacinto, J. Pinto, J. Patrikar, J. Keller, R. Cunha, S. Scherer, and A. Pascoal, “Pegasus simulator: An isaac sim framework for multiple aerial vehicles simulation,” in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2024, pp. 917–922.
- [29] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [30] Y. Qiu, C. Wang, C. Xu, Y. Chen, X. Zhou, Y. Xia, and S. Scherer, “Airimu: Learning uncertainty propagation for inertial odometry,” 2023.
- [31] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The euroc micro aerial vehicle datasets,” *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.
- [32] P. Geneva, K. Eickenhoff, W. Lee, Y. Yang, and G. Huang, “Openvins: A research platform for visual-inertial estimation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.
- [33] Y. Peng, C. Chen, and G. Huang, “Quantized visual-inertial odometry,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 17954–17960.