Learned IMU Bias Prediction for Invariant Visual Inertial Odometry

Abdullah Altawaitan[1], Jason Stanley[1], Sambaran Ghosal[1], Thai Duong[2], and Nikolay Atanasov[1]

***Abstract*—Autonomous mobile robots operating in novel environments depend critically on accurate state estimation, often utilizing visual and inertial measurements. Recent work has shown that an invariant formulation of the extended Kalman filter improves the convergence and robustness of visual-inertial odometry by utilizing the Lie group structure of a robot's position, velocity, and orientation states. However, inertial sensors also require measurement bias estimation, yet introducing the bias in the filter state breaks the Lie group symmetry. In this paper, we design a neural network to predict the bias of an inertial measurement unit (IMU) from a sequence of previous IMU measurements. This allows us to use an invariant filter for visual inertial odometry, relying on the learned bias prediction rather than introducing the bias in the filter state. We demonstrate that an invariant multi-state constraint Kalman filter (MSCKF) with learned bias predictions achieves robust visual-inertial odometry in real experiments, even when visual information is unavailable for extended periods and the system needs to rely solely on IMU measurements.**

***Index Terms*—Localization, Aerial Systems: Applications, Deep Learning Methods**

I. Introduction

MANY core robot autonomy functions, including mapping and control, depend on accurate state estimation. Visual-inertial odometry (VIO) [1], [2] offers a reliable and cost-effective approach to estimate the position, orientation, and velocity of mobile robots equipped with cameras and inertial measurement units (IMUs). Cameras can estimate pose displacements but are sensitive to lighting change and motion blur. IMUs, on the other hand, deliver high-frequency data independent of visual conditions but lead to estimate drift over time due to measurement bias. Thus, visual and inertial sensors complement each other effectively but estimating IMU bias is crucial for ensuring reliable state estimation, especially with poor or intermittent visual information.

Traditional VIO methods like the multi-state constraint Kalman filter (MSCKF) [3] include the IMU bias, together with the system position, orientation, and velocity, in the filter state and estimate it sequentially from sensor measurements. We explore an alternative formulation using a learned sequence-to-sequence model to predict the IMU bias based on a longer history of IMU measurements. Moreover, VIO systems typically model IMU bias as a random process driven by white noise rather than as an unknown term. This distinction impacts the observability properties of the VIO system: bias is observable when modeled as noise but unobservable when considered as an unknown term, as shown in [4]. In practice, IMU biases often exhibit slow time-varying drift rather than the rapid fluctuations characteristic of white noise. In this work, we design a neural network that predicts IMU biases directly from a sequence of previous IMU measurements, avoiding the need to include the bias in the filter state and allowing the use of an invariant Kalman filter formulation as we discuss later.

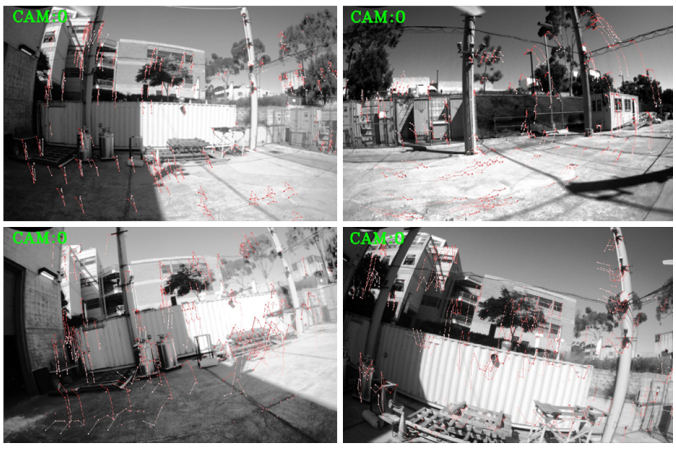


Fig. 1: Monocular images and keypoints from a quadrotor with a FLIR Chameleon camera and VectorNav VN-100 IMU.

First, we review learning-based methods that leverage IMU data for state estimation. IONet [5] uses a long-short-term memory (LSTM) network to predict velocities from buffered IMU measurements that are then integrated to estimate the 2D motion of pedestrians. RoNIN [6] continues this direction by presenting three different neural network architectures: Temporal Convolutional Network (TCN), Residual Network (ResNet), and LSTM to predict velocities which, when integrated with known orientation, yield 2D pedestrian motion estimates. TLIO [7] extends previous works to 3D and uses a ResNet to estimate pedestrian displacements in a local gravity-aligned frame and their uncertainty from a buffer of IMU measurements, which serve as measurements in an extended Kalman filter (EKF). While [5]–[7] focus on pedestrian motions, Zhang et al. [8] show that a series of neural networks

Manuscript received: May, 9, 2025; Revised August, 6, 2025; Accepted September, 11, 2025.

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

We gratefully acknowledge support from NSF CCF-2112665 (TILOS).

[1]The authors are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA, e-mails: {aaltawaitan,jtstanle,sghosal,natanasov}@ucsd.edu. A. Altawaitan is also affiliated with Kuwait University as a holder of a scholarship.

[2]This author is with the Department of Computer Science, Rice University, Houston, TX 77005, USA, e-mail: thaiduong@rice.edu.

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

can be used to estimate IMU bias, thrust correction, and integration errors for a quadrotor robot using IMU readings and motor speeds. Likewise, Cioffi et al. [9] use a TCN network to predict 3D relative position from thrust and gyroscope measurements for drone racing. However, the methods in [5]–[9] are trajectory-specific and cannot generalize to unseen trajectories at test time. Moreover, these methods often assume that the IMU measurements are transformed into the world frame using the ground-truth pose but, during deployment, the pose is typically estimated via Kalman filtering, making the transformation inaccurate. To address this limitation, Buchanan et al. [10] propose a neural network to predict the IMU bias directly instead of learning a motion model, enabling the system to generalize to unseen trajectories at test time. However, the network is trained with ground-truth IMU biases, which are unavailable in real-world scenarios. Qiu et al. [11] extend [10] by learning both IMU bias and measurement uncertainty through IMU preintegration in pose graph optimization. Denoising IMU Gyroscopes [12] learns only the gyroscope bias, which is not enough for accurate inertial integration, and evaluates rotational accuracy alone. In contrast, we learn both gyroscope and accelerometer biases and evaluate both translational and rotational accuracy. TLIO [7] trains a network to estimate relative positions directly from IMU measurements in a local gravity-aligned frame, implicitly learning the unobserved initial velocity in a time window from pedestrian motion patterns. Instead, we train a neural network to estimate IMU bias from past IMU measurements using a Lie algebra error between the integrated measurements and the ground-truth robot state. DIDO [8] learns biases separately for gyroscope and accelerometer and relies on a tachometer, prior knowledge of quadrotor parameters, and near hover flight assumption, and decouples rotational and translational dynamics to estimate relative positions and velocities for inertial-only odometry. DIDO overfits to motion patterns seen during training due to the unobserved initial velocity, similar to TLIO. Our method removes these requirements by predicting both biases with a single bias prediction network and integrating it into visual-inertial odometry. Finally, AirIMU [11] trains a network to estimate the IMU bias for an IMU-GPS pose graph optimization. However, AirIMU assumes access to ground-truth positions during deployment and does not claim real-time performance. In contrast, we learn the bias prediction model in the same way but integrate it with an invariant filter, enabling real-time visual-inertial odometry.

The position, orientation, and velocity of a robot system evolve on a matrix Lie group and possess symmetries (or invariance) in the sense that certain transformations leave the system state unchanged. Barrau et al. [13] introduced an invariant EKF in which the estimation errors remain invariant under the action of a matrix Lie group. Hartley et al. [14] showed improved convergence and robustness of the invariant EKF in contact-aided inertial navigation, even when including bias terms in the filter state. Lin et al. [15] extend the latter work by developing an invariant state estimation approach using only onboard proprioceptive sensors. However, the inclusion of bias terms within the filter state breaks the Lie group symmetry, causing the linearized error dynamics to depend on the state estimates rather than remaining state-independent. Fornasier et al. [16] introduced an equivariant filter for VIO that integrates IMU bias and camera intrinsic-extrinsic parameters into a symmetry group structure. The approach achieves state-of-the-art accuracy and consistent estimation without the need for additional consistency enforcement techniques, e.g., observability constraint [17]. The equivariant filter extends the invariant filter by operating on homogeneous spaces, reducing to the invariant case with a specific choice of symmetry [18]. In this context, invariance corresponds to symmetries that leave the system state unchanged, whereas equivariance involves symmetries that change it in a structured manner [19].

Our contribution is a sequence-to-sequence neural network that predicts IMU biases directly from past inertial measurements, which enables three key capabilities. First, estimating the bias outside the filter state enables an invariant Kalman filter, whose state covariance evolution is independent of the state estimates. Second, the proposed method achieves real-time visual inertial odometry. Third, we demonstrate the performance of the bias prediction network in visually degraded scenarios where the system relies solely on IMU measurements for motion estimation. Our evaluation demonstrates that this approach yields bias estimates that are physically consistent and stable (unlike the fluctuating estimates obtained from a standard EKF), and achieves reliable state estimation, even with poor or no visual features for extended periods of time.

II. PROBLEM STATEMENT

Consider a robot equipped with an IMU and a camera. The IMU provides noisy measurements of angular velocity $\omega(t) \in \mathbb{R}^3$ and linear acceleration $a(t) \in \mathbb{R}^3$. The camera provides the pixel coordinates $z(t) \in \mathbb{R}^2$ of keypoints tracked across consecutive images. IMU and camera measurements are assumed to be generated synchronously at the same discrete time steps t_k.

Our goal is to estimate the robot's state at time t:

$$X(t) = \begin{bmatrix} R(t) & v(t) & p(t) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in SE_2(3), \tag{1}$$

where $R(t) \in SO(3)$, $v(t) \in \mathbb{R}^3$, and $p(t) \in \mathbb{R}^3$ denote the orientation, linear velocity, and position of the inertial frame relative to the global frame, respectively, and $SE_2(3)$ denotes the extended special Euclidean group [20].

The gyroscope and accelerometer measurements are corrupted by additive white noise $n^g(t), n^a(t) \in \mathbb{R}^3$ and time-varying bias $b^g(t), b^a(t) \in \mathbb{R}^3$, respectively:

$$\begin{aligned} \bar{\omega}(t) &= \omega(t) + b^g(t) + n^g(t), \\ \bar{a}(t) &= a(t) - R^\top(t)\, g + b^a(t) + n^a(t), \end{aligned} \tag{2}$$

where $\bar{\omega}(t) \in \mathbb{R}^3$ is the measurement of angular velocity in body-frame coordinates, $\bar{a}(t) \in \mathbb{R}^3$ is the measurement of linear acceleration in body-frame coordinates, and $g \in \mathbb{R}^3$ is the gravity vector in world-frame coordinates. Let $u(t) = (\omega(t), a(t))$ the denote noiseless measurements, $\bar{u}(t) = (\bar{\omega}(t), \bar{a}(t))$ denote the noisy measurements, and $b(t) = (b^g(t), b^a(t))$ denote the IMU bias.

The evolution of state $X(t)$ with input $u(t)$ is governed by a continuous-time motion model:

$$\dot{X}(t) = f(X(t), u(t)) = \begin{bmatrix} R(t)(\omega(t))_\times & R(t)a(t) + g & v(t) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3}$$

where the operator $(\cdot)_\times : \mathbb{R}^3 \to \mathfrak{so}(3)$ maps a vector in $\mathbb{R}^3$ to a 3×3 skew-symmetric matrix.

The IMU bias is typically modeled using a Brownian motion model (i.e., random walk) [10]:

$$\dot{b}(t) = \eta(t), \quad \eta(t) = \begin{bmatrix} \eta^g(t)^\top & \eta^a(t)^\top \end{bmatrix}^\top \in \mathbb{R}^6, \tag{4}$$

where η is the IMU bias noise. While this assumption provides a simple linear approximation of bias evolution, it might fail to capture complex behaviors. Instead, we consider learning a sequence-to-sequence parametrized model d_θ that maps a sequence of IMU measurements to their corresponding sequence of biases, offering a more expressive model than a random walk. To achieve this, given a set of raw measurements $\bar{u}^{(i)}_{0:N}$, we predict the corresponding IMU biases $\hat{b}^{(i)}_{0:N}$ using d_θ and roll out the IMU kinematics f in Eq. (3) with initial state $X^{(i)}_0$ and corrected measurements $\bar{u}^{(i)}_k - \hat{b}^{(i)}_k$, for $k = 0, \dots, N$. We assume both the IMU measurements $u(t)$ and bias $b(t)$ remain constant during the time interval $[t_k, t_{k+1})$.

Problem 1. Given dataset $\mathcal{D} = \{t^{(i)}_{0:N}, X^{(i)}_{0:N}, \bar{u}^{(i)}_{0:N}\}_{i=1}^D$, learn an IMU bias prediction model d_θ by determining the parameters θ that minimize the following:

$$\begin{aligned} \min_\theta \quad & \sum_{i=1}^{D} \sum_{k=1}^{N} c(\hat{X}^{(i)}_k, X^{(i)}_k) \\ \text{s.t.} \quad & \hat{X}^{(i)}_{k+1} = \text{ODESolver}(f, \hat{X}^{(i)}_k, \bar{u}^{(i)}_k - \hat{b}^{(i)}_k, t^{(i)}_{k+1} - t^{(i)}_k) \\ & \hat{b}^{(i)}_{0:N} = d_\theta(\bar{u}^{(i)}_{0:N}), \text{ for } k = 0, \dots, N \text{ and } i = 1, \dots, D, \end{aligned} \tag{5}$$

for a given initial state $\hat{X}^{(i)}_0 = X^{(i)}_0$. The cost function c may be chosen as a suitable distance metric on the $SE_2(3)$ manifold. Instead of using an ODE solver (e.g., Runge-Kutta [21]), we compute the integration of Eq. (3) on the $SE_2(3)$ group in closed-form (shown in Sec. III-B, Eq. (13)).

III. PRELIMINARIES

This section introduces background material that will be used throughout the paper.

A. Lie Group Operators

Let X denote an element of the extended special Euclidean Lie group $SE_2(3)$, with structure defined in Eq. (1). The corresponding Lie algebra $\mathfrak{se}_2(3)$ consists of 9×9 matrices:

$$\xi^\wedge = \begin{bmatrix} (\xi^R)_\times & \xi^v & \xi^p \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; \xi = \begin{bmatrix} \xi^R \\ \xi^v \\ \xi^p \end{bmatrix}, \; \xi^R, \xi^v, \xi^p \in \mathbb{R}^3. \tag{6}$$

The vector $\xi \in \mathbb{R}^9$ parametrizes the Lie algebra via the hat operator $(\cdot)^\wedge : \mathbb{R}^9 \to \mathfrak{se}_2(3)$, while the vee operator $(\cdot)^\vee : \mathfrak{se}_2(3) \to \mathbb{R}^9$ is its inverse. A group element $X \in SE_2(3)$ is related to an algebra element $\xi^\wedge \in \mathfrak{se}_2(3)$ through the exponential $\exp(\cdot) : \mathfrak{se}_2(3) \to SE_2(3)$ and logarithm $\log(\cdot) : SE_2(3) \to \mathfrak{se}_2(3)$ maps:

$$X = \exp(\xi^\wedge), \quad \xi^\wedge = \log(X), \tag{7}$$

where $\exp(\xi^\wedge)$ admits a closed-form expression:

$$\exp(\xi^\wedge) = \begin{bmatrix} \Gamma_0(\xi^R) & \Gamma_1(\xi^R)\xi^v & \Gamma_1(\xi^R)\xi^p \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $\Gamma_0(\cdot)$ and $\Gamma_1(\cdot)$ denote the $SO(3)$ exponential map and left Jacobian, respectively (see [14]).

We consider right-invariant error associated with left perturbation. The group error state $\tilde{X}$ and retraction $\xi \oplus \hat{X}$ are defined as follows:

$$\tilde{X} = X\hat{X}^{-1}, \quad \xi \oplus \hat{X} = \exp(\xi^\wedge)\hat{X}, \tag{8}$$

where the vector ξ represents a perturbation in $\mathfrak{se}_2(3)$. For $X \in SE_2(3)$, the adjoint map is defined as $\text{Ad}_X(\xi^\wedge) = X\xi^\wedge X^{-1}$ and its matrix representation can be written as:

$$\text{Ad}_X = \begin{bmatrix} R & 0 & 0 \\ (v)_\times R & R & 0 \\ (p)_\times R & 0 & R \end{bmatrix}. \tag{9}$$

Please refer to [22] for further details.

B. Multi-state Constraint Kalman Filter (MSCKF)

The MSCKF [3] is a VIO method that marginalizes landmark positions instead of incorporating them in the filter state, thereby avoiding to build a map of 3D landmark positions. The MSCKF maintains a sliding window of past sensor poses to triangulate keypoints via least-squares optimization using geometric constraints from multiple images.

The filter state consists of the robot state $X_k \in SE_2(3)$, IMU bias b_k at time t_k, and a window of W historical states $X_{k-1} \dots, X_{k-W}$. Given inertial measurement $\bar{u}_k$, the mean of the state $\hat{X}(t)$ and of the bias $\hat{b}(t)$ are propagated as:

$$\dot{\hat{X}}(t) = f(\hat{X}(t), \bar{u}_k - \hat{b}(t)), \quad \dot{\hat{b}}(t) = 0. \tag{10}$$

In the remainder of the paper, we omit the time dependence of the variables for readability. We denote estimated quantities with $\hat{(\cdot)}$ and error quantities with $\tilde{(\cdot)}$. The MSCKF uses decoupled error states $\exp(\xi^R_\times) = R\hat{R}^\top$, $\xi^v = v - \hat{v}$, $\xi^p = p - \hat{p}$, and $\tilde{b} = b - \hat{b}$ to propagate the IMU covariance $P \in \mathbb{R}^{15\times 15}$ through the linearized error-state dynamics:

$$\begin{bmatrix} \dot{\xi} \\ \dot{\tilde{b}} \end{bmatrix} = A \begin{bmatrix} \xi \\ \tilde{b} \end{bmatrix} + G \begin{bmatrix} n \\ \eta \end{bmatrix}, \tag{11}$$

where $n = \begin{bmatrix} n^{g\top} & n^{a\top} & n^{v\top} \end{bmatrix} \in \mathbb{R}^9$ is the noise associated with angular velocity, linear acceleration, linear velocity. Here, A and G represent the Jacobians resulting from linearizing the error-state dynamics around the filter state estimate [3]. Thus, the IMU covariance evolution is governed by the continuous-time Riccati equation [14]:

$$\dot{P} = AP + PA^\top + Q, \quad Q = G\text{Cov}(n)G^\top. \tag{12}$$

Filter propagation: To propagate the means of the state $\hat{X}_k$ and of the bias $\hat{b}_k$ between t_k and t_{k+1}, with the assumption that the IMU measurement $\bar{u}_k$ remains constant over the time interval $\Delta t_k = t_{k+1} - t_k$, [14] provides closed-form integration of Eq. (10):

$$\begin{aligned}
\hat{R}_{k+1} &= \hat{R}_k \Gamma_0((\bar{\omega}_k - \hat{b}_k^g)\Delta t_k), \\
\hat{v}_{k+1} &= \hat{v}_k + g\Delta t_k + \hat{R}_k \Gamma_1((\bar{\omega}_k - \hat{b}_k^g)\Delta t_k)(\bar{a}_k - \hat{b}_k^a), \\
\hat{p}_{k+1} &= \hat{p}_k + \hat{v}_k \Delta t_k + \frac{1}{2} g \Delta t_k^2 \\
&\quad + \hat{R}_k \Gamma_2((\bar{\omega}_k - \hat{b}_k^g)\Delta t_k)(\bar{a}_k - \hat{b}_k^a),
\end{aligned} \tag{13}$$

where $\Gamma_2(\phi)$ is defined in [14]. To obtain the covariance of $\hat{X}_k$ and $\hat{b}_k$, we approximate the integration of Eq. (12):

$$\begin{aligned}
P_{k+1} &= \Phi_k P_k \Phi_k^\top + Q_k^d, \qquad \Phi_k = \exp(A_k \Delta t_k), \\
Q_k^d &\approx \Phi_k Q_k \Phi_k^\top \Delta t_k, \qquad Q_k = G_k \mathrm{Cov}(n) G_k^\top.
\end{aligned} \tag{14}$$

Since past states remain constant, their covariance entries are propagated using an identity Jacobian and zero process noise, as described in [23].

Filter update: Consider a keypoint $z_{k,m} \in \mathbb{R}^2$, obtained from an image keypoint detection algorithm such as FAST [24], associated with landmark $\ell_m \in \mathbb{R}^3$ and state X_k. The variables are related by the measurement model [25]:

$$z_{k,m} = h(X_k, \ell_m) + \rho_{k,m}, \tag{15}$$

where h is the image projection of landmark ℓ_m and $\rho_{k,m}$ is the keypoint detection noise. Define the keypoint error for each measurement as $e_{k,m} = z_{k,m} - h(\hat{X}_k, \hat{\ell}_m)$. After applying the left null-space projection step from [3], let $\hat{e}$, H, and V represent the stacked errors, measurement Jacobians, and noise covariances for all landmarks. We update the mean $\hat{\mathcal{X}}_k = (\hat{X}_k, \hat{b}_k, \hat{X}_{k-1}, \ldots, \hat{X}_{k-W})$, and covariance $\mathcal{P}_k$ as:

$$\begin{aligned}
\hat{\mathcal{X}}_{k+1} &= (K\hat{e}) \oplus \hat{\mathcal{X}}_{k+1}, \\
\mathcal{P}_{k+1} &= (I - KH)\mathcal{P}_{k+1}(I - KH)^\top + KVK^\top, \\
K &= \mathcal{P}_k H^\top \left(H\mathcal{P}_k H^\top + V\right)^{-1}.
\end{aligned} \tag{16}$$

IV. Learning IMU Bias for VIO

Typically, the IMU bias is not directly observable from a single IMU measurement. Instead, bias estimation requires integrating a sequence of IMU measurements and comparing against an external sensor for ground truth (e.g., motion capture system). This sequence dependence motivates us to predict IMU biases from a sequence of raw IMU measurements via a neural network model and using the ground-truth state for training. If ground-truth poses are unavailable, a camera sensor can be used to track keypoints across frames to estimate relative poses $X_{k+1}^{-1} X_k$, which can be used as ground-truth. Recent works propose using deep learning architectures, such as Convolutional Neural Network (CNN), ResNet, or transformer, to capture temporal dependencies and patterns in IMU measurements [5]–[11]. In this work, we learn a sequence-to-sequence neural network model d_θ, mapping a sequence of IMU measurements $\bar{u}_{k-L}^{(i)}, \ldots, \bar{u}_k^{(i)}$ to a corresponding sequence of IMU bias estimates $(\hat{b}_{k-L}^{(i)}, \ldots, \hat{b}_k^{(i)})$:

$$d_\theta(\bar{u}_{k-L}^{(i)}, \ldots, \bar{u}_k^{(i)}) = (\hat{b}_{k-L}^{(i)}, \ldots, \hat{b}_k^{(i)}). \tag{17}$$

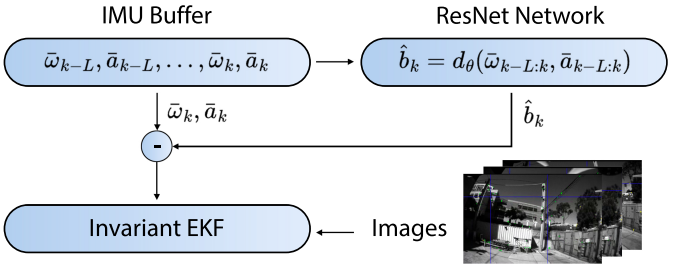


Fig. 2: Block diagram of our system.

Due to the correlation between angular velocity and acceleration at slow-varying velocities, e.g., observed in [12], we use a single model d_θ to infer the IMU bias instead of separate models for the gyroscope and accelerometer.

To optimize θ, we partition the collected trajectories into non-overlapping D segments, each consisting of N samples of ground-truth state $X_k^{(i)}$ and raw IMU measurements $\bar{u}_k^{(i)}$. For each segment, we feed the raw IMU measurements into the neural network model d_θ to predict the corresponding bias estimates $\hat{b}_{0:N}^{(i)}$ as in Eq. (17), which are expected to initially be inaccurate. These predicted biases are then used to correct the raw IMU measurements with $\bar{u}_k^{(i)} - \hat{b}_{0:N}^{(i)}$. Then, we roll out an estimated state trajectory $\hat{X}_{1:N}^{(i)}$ with the corrected IMU measurements $\bar{u}_k^{(i)} - \hat{b}_{0:N}^{(i)}$ and an initial state $X_0^{(i)}$ using Eq. (13). To update the neural network parameters θ, we use a cost function $c(\hat{X}_k^{(i)}, X_k^{(i)})$ that measures the discrepancy between the predicted state $\hat{X}_k^{(i)}$ and the ground-truth $X_k^{(i)}$. As states lie on $SE_2(3)$, we compute the group error $\tilde{X}_k^{(i)} = X_k^{(i)} \hat{X}_k^{(i)^{-1}} \in SE_2(3)$, map it onto the Lie algebra $\xi_k^{(i)\wedge} = \log(\tilde{X}_k^{(i)}) \in \mathfrak{se}_2(3)$, and calculate the norm of its vector representation $\xi_k^{(i)} = \log(\tilde{X}_k^{(i)})^\vee \in \mathbb{R}^9$ as follows:

$$c(\hat{X}_k^{(i)}, X_k^{(i)}) = \left\| \log\left(X_k^{(i)} \hat{X}_k^{(i)^{-1}}\right)^\vee \right\|_h. \tag{18}$$

We use the Huber loss $\|\cdot\|_h$ in the cost function definition above to prioritize early trajectory estimates, which are less corrupted by the noise inherent in the IMU measurements which accumulates with long-horizon integration of Eq. (13). Alternatively, the network can be trained with a loss on relative poses $X_{k+1}^{-1} X_k$, which delivers similar results. The 9-dimensional state error vector evaluated by the Huber loss is weighted by 10^3 for orientation, 10^2 for position, and 10^1 for velocity to give the three components equal influence despite their different units. We use the Adam optimizer [26] with learning rate 1×10^{-3} to iteratively optimize the parameters θ. Our approach provides high-frequency bias estimates compared to updating the bias only at the lower-frequency update step, typically at the camera frame rate. In addition, our model decouples the bias prediction from visual information, which can be unreliable for extended periods.

We use the invariant EKF [13] to track the filter state mean $\hat{X}_k, \ldots, \hat{X}_{k-W}$ on the matrix Lie group $SE_2(3)$, while the covariance P_k is propagated in the corresponding Lie algebra. Relying on learned bias predictions from d_θ allows us to formulate the IMU dynamics without introducing the bias in

the filter state:

$$\dot{X} = f(X, \bar{u}_k - b) - Xn^{\wedge}, \tag{19}$$

where n is the propagation noise as in Eq. (11). From [13], the deterministic system f satisfies the group-affine property. Thus, using the group error defined in Eq. (8), the linearized IMU error-state dynamics:

$$\dot{\xi} = A\xi + \mathrm{Ad}_X n, \quad A = \begin{bmatrix} 0 & 0 & 0 \\ (g)_\times & 0 & 0 \\ 0 & I & 0 \end{bmatrix}, \tag{20}$$

can be propagated as in Eq. (12) with $G = \mathrm{Ad}_X$. Note that for the deterministic system $\dot{\xi} = A\xi$, since the Jacobian A is state-independent, the error propagation is independent of the state estimate. When noise is brought to the system, the state covariance mapping remains state-independent, whereas the noise mapping depends on the state estimate as in Eq. (14), which is an advantage over the standard EKF. To summarize, as shown in Fig. 2, for a given state mean $\hat{X}_k$ and covariance P_k with raw IMU measurements $\bar{u}_{k-L}, \ldots, \bar{u}_k$, we have:

$$\begin{aligned} (\hat{b}_{k-L}, \ldots, \hat{b}_k) &= d_\theta(\bar{u}_{k-L}, \ldots, \bar{u}_k), \\ \Phi_k &= \exp(A\Delta t_k), \\ P_{k+1} &= \Phi_k P_k \Phi_k^\top + \Phi_k \mathrm{Ad}_{X_k} \mathrm{Cov}(n) \mathrm{Ad}_{X_k}^\top \Phi_k^\top \Delta t_k, \\ \text{and} \quad & \text{Eq. (13)}. \end{aligned} \tag{21}$$

V. Evaluation

We present our choice of neural network architecture for IMU bias prediction in Sec. V-A. Then, we evaluate our method against state-of-the-art VIO baselines using both publicly available dataset and *Aerodrome* dataset with challenging motions in Sec. V-B. We demonstrate the robustness of our method under challenging conditions where visual features are temporarily lost, requiring the filter to rely solely on IMU measurements, in Sec. V-C. Finally, in Sec. V-D, we evaluate our method as an inertial-only odometry approach and compare it against two popular inertial-odometry baselines.

Datasets: We evaluate on the public *EuRoC* dataset [27] and our *Aerodrome* dataset. The *EuRoC* dataset provides 200 Hz IMU, 20 Hz camera, and 100 Hz ground truth from a quadrotor operating at maximum speeds of 2.3 m/s. Per [12], MH and VR1 were captured on consecutive days, while VR2 was acquired later. Thus, we train on (MH01, MH03, MH04), validate on (MH02, MH05), and test on (V102, V103, V202). The *Aerodrome* dataset consists of five trajectories with 200 Hz IMU data, 25 Hz camera, and 100 Hz ground truth from a quadrotor operating at maximum speeds of 5.4 m/s. We train on A01, validate on A02, and test on (A03, A04, A05).

Metrics: To assess IMU bias prediction accuracy in Sec. V-A, we compute the cost defined in Eq. (5) over the test data along with the average error norms of the orientation $\|\xi^R\| = \|\log(\hat{R}R^\top)^\vee\|$, velocity $\|\xi^v\| = \|\hat{v} - v\|$, and position $\|\xi^p\| = \|\hat{p} - p\|$. For quantitative trajectory evaluation in Sec. V-B, V-C, and V-D, we report the Absolute Trajectory Error (ATE) in translation and rotation and the Relative Error (RE) in translation. These metrics are defined in [28].

TABLE I: IMU Noise Parameters

Parameter	Symbol	EuRoC	Aerodrome	Unit
Gyro. Noise Density	σ_g	$1e^{-2}$	$1e^{-2}$	$\frac{\text{rad}}{\text{s}} \frac{1}{\sqrt{Hz}}$
Gyro. Random walk	σ_{bg}	$8e^{-4}$	$6e^{-4}$	$\frac{\text{rad}}{\text{s}^2} \frac{1}{\sqrt{Hz}}$
Accel. Noise Density	σ_a	$3e^{-2}$	$1e^{-1}$	$\frac{\text{m}}{\text{s}^2} \frac{1}{\sqrt{Hz}}$
Accel. Random Walk	σ_{ba}	$2e^{-4}$	$7e^{-3}$	$\frac{\text{m}}{\text{s}^3} \frac{1}{\sqrt{Hz}}$

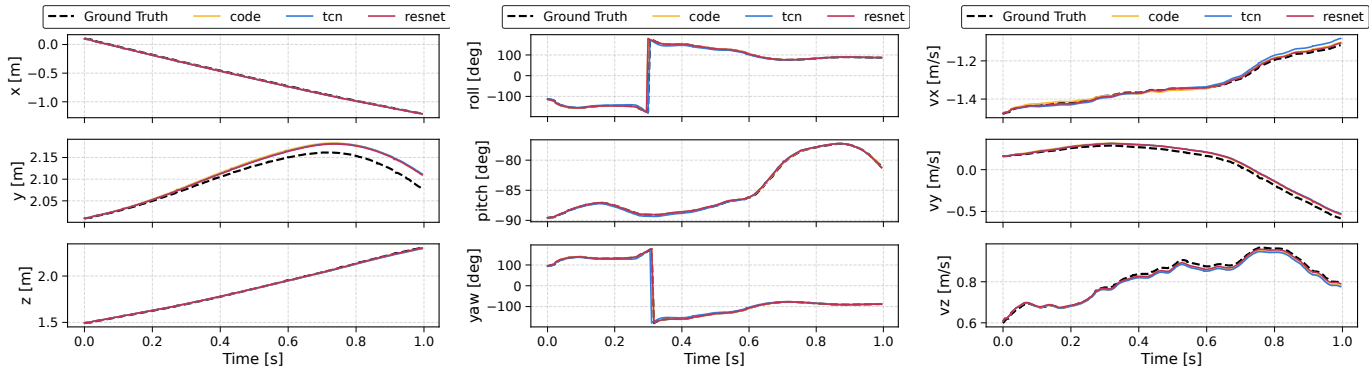
Baselines: We evaluate our approach against three VIO methods: MSCKF [23], a monocular multi-state constraint Kalman filter with IMU bias estimated in the filter state, MSCEqF [16], an equivariant formulation of the monocular MSCKF with IMU bias estimated in the filter state, and LBMSCKF, an MSCKF with learned bias. In addition, we evaluate our approach as an inertial-only odometry against two learning-based inertial odometry methods: IMO [9], which combines a TCN to estimate relative positions from IMU and thrust inputs with an EKF, and TLIO [7], which uses a ResNet to estimate relative position displacements in a local gravity-aligned frame and its uncertainty from IMU measurements. In both methods, the IMU bias is estimated within the filter state.

A. Model Architecture Choice and Implementation Details

We investigate the choice of neural network architecture for IMU bias prediction that obtains the best state prediction on the *EuRoC* and *Aerodrome* datasets. Motivated by the sequential dependence in Sec. IV, we compare three commonly used sequential architectures: ResNet [7], TCN [9], and CodeNet [11] as candidate models for d_θ.

Implementation Details: The neural network architectures compared in this evaluation have a comparable number of trainable parameters: $300K$ for ResNet, $500K$ for TCN, and $400K$ for CodeNet. Given a sequence of IMU measurements sampled at 200 Hz over a one-second window ($L = 200$), along with an initial state X_0, each network estimates biases $\hat{b}_{k-L}, \ldots, \hat{b}_k$, as in Eq. (17), and corrects for the raw measurements before integration as in Eq. (13). The ResNet follows its original design, predicting a single bias estimate assumed constant $\hat{b}_k = \hat{b}_i$ for all $i = k - L, \ldots, k$ throughout the prediction window. The implementation of TCN in [9] outputs a 3-dimensional vector. We modified the TCN architecture to predict a sequence of 6-dimensional output, representing both gyroscope and accelerometer biases. In our implementation, the filter state maintains up to $W = 11$ past states, enabling real-time operation with filter update steps at 20 Hz, while neural network inference runs at 200 Hz on every incoming IMU measurement using an overlapping one-second sliding window. Table I presents the IMU noise parameters used for the filters in all experiments. The computation times, recorded onboard the quadrotor with an Intel i7 NUC, are listed in Table V, showing that bias inference adds negligible overhead to the total processing time.

Training Results: Table II summarizes the state prediction accuracy achieved with bias correction using different network architectures, measured by the metrics defined earlier. ResNet shows average improvements in test, velocity, and position losses, while TCN achieves the best rotation estimation performance on average. We observe a comparable performance



(a) Position (b) Orientation (c) Velocity

Fig. 3: Predicted position, orientation, and velocity over a 1-second window from initial state X_0 on the *Aerodrome* dataset, comparing CodeNet, TCN, and ResNet predictions against ground-truth.

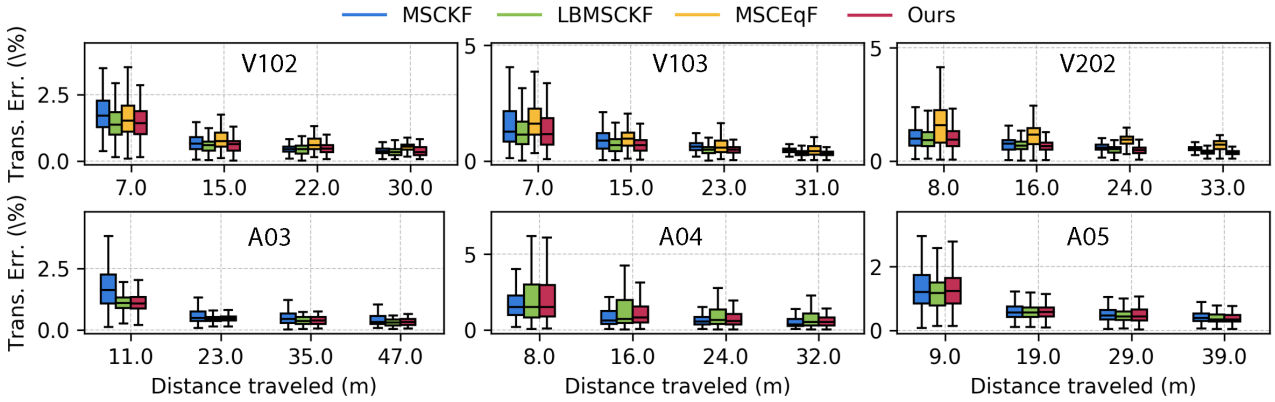


Fig. 4: RE on *EuRoC* and *Aerodrome* over 2.5%, 5%, 7.5%, and 10% traveled distances of the total trajectory, averaged over five runs.

TABLE II: IMU bias learning with different network architectures.

Metrics	Dataset	TCN	CodeNet	ResNet
Test loss in Eq. (18)	EuRoC	0.022	0.023	**0.022**
$\|\log(\hat{R}R^\top)^\vee\|$ (avg.)	EuRoC	1.76×10^{-6}	$\mathbf{1.70 \times 10^{-6}}$	2.3×10^{-6}
$\|\hat{v} - v\|$ (avg.)	EuRoC	7.46×10^{-4}	8.01×10^{-4}	$\mathbf{7.26 \times 10^{-4}}$
$\|\hat{p} - p\|$ (avg.)	EuRoC	1.27×10^{-4}	1.35×10^{-4}	$\mathbf{1.24 \times 10^{-4}}$
Test loss in Eq. (18)	Aerodrome	0.013	**0.005**	**0.005**
$\|\log(\hat{R}R^\top)^\vee\|$ (avg.)	Aerodrome	6.88×10^{-3}	$\mathbf{1.69 \times 10^{-3}}$	2.27×10^{-3}
$\|\hat{v} - v\|$ (avg.)	Aerodrome	2.37×10^{-2}	2.66×10^{-2}	$\mathbf{2.32 \times 10^{-2}}$
$\|\hat{p} - p\|$ (avg.)	Aerodrome	1.12×10^{-2}	$\mathbf{1.03 \times 10^{-3}}$	$\mathbf{1.03 \times 10^{-3}}$

TABLE III: Ablation on window size L for *Aerodrome* dataset.

Metrics	Architecture	$L = 50$	$L = 100$	$L = 150$	$L = 200$
$\|\log(\hat{R}R^\top)^\vee\|$ (avg.)	TCN	1.1×10^{-2}	8.6×10^{-3}	7.8×10^{-3}	$\mathbf{6.9 \times 10^{-3}}$
$\|\hat{v} - v\|$ (avg.)	TCN	6.9×10^{-2}	4.7×10^{-2}	4.6×10^{-2}	$\mathbf{2.4 \times 10^{-2}}$
$\|\hat{p} - p\|$ (avg.)	TCN	7.5×10^{-2}	4.3×10^{-2}	3.9×10^{-2}	$\mathbf{1.1 \times 10^{-2}}$
$\|\log(\hat{R}R^\top)^\vee\|$ (avg.)	CodeNet	8.9×10^{-3}	3.2×10^{-3}	1.8×10^{-3}	$\mathbf{1.7 \times 10^{-3}}$
$\|\hat{v} - v\|$ (avg.)	CodeNet	7.0×10^{-2}	4.8×10^{-2}	4.6×10^{-2}	$\mathbf{2.7 \times 10^{-2}}$
$\|\hat{p} - p\|$ (avg.)	CodeNet	7.5×10^{-2}	4.4×10^{-2}	3.8×10^{-2}	$\mathbf{1.0 \times 10^{-3}}$
$\|\log(\hat{R}R^\top)^\vee\|$ (avg.)	ResNet	1.1×10^{-2}	4.3×10^{-3}	2.3×10^{-3}	$\mathbf{2.2 \times 10^{-3}}$
$\|\hat{v} - v\|$ (avg.)	ResNet	7.0×10^{-2}	4.7×10^{-2}	4.5×10^{-2}	$\mathbf{2.3 \times 10^{-2}}$
$\|\hat{p} - p\|$ (avg.)	ResNet	7.5×10^{-2}	4.3×10^{-2}	3.9×10^{-2}	$\mathbf{1.0 \times 10^{-3}}$

across the architectures, with ResNet showing a slight edge. A qualitative evaluation is provided in Fig. 3, illustrating the predicted position, orientation, and linear velocity over a one-second interval starting from a known initial state X_0. Visually, these predictions align with the quantitative results reported in Table II. Therefore, we use the ResNet architecture to evaluate our IMU bias prediction for VIO in the remainder of the paper. In addition, we provide an ablation study on the window length L, reported in Table III. The error decreases with longer windows, so we set $L = 200$ in our model for VIO evaluation.

B. Comparison to MSCKF and MSCEqF

We use the inertial noise parameters in Table I and disable the SLAM features of the MSCKF baseline [23] for a fair comparison. We omitted MSCEqF from the *Aerodrome* evaluation due to issues in the open-source code with the keypoint detector failing to extract image features. Table IV and Fig. 4 present results on the *EuRoC* and *Aerodrome* datasets using the ATE and RE metrics. MSCKF and LBMSCKF achieve nearly the same accuracy on *EuRoC*, while on *Aerodrome*, which includes maneuvers up to 5.4 m/s, LBMSCKF outperforms MSCKF across all sequences. Our method achieves the best or second-best ATE results on all sequences, except on V102, thereby outperforming the baselines on average. Similarly, Fig. 4 shows lower mean RE for our method across all sequences. In addition, we achieve lower variance errors compared to the baselines, suggesting better reliability. Overall, our method exhibits a slight edge in performance in normal scenarios. Both the learned bias prediction and the invariant filter formulation contribute complementary improvements in the performance. In the next subsection, we examine the robustness of our approach against the MSCKF under challenging conditions, where visual features are temporarily lost.

C. Comparison to MSCKF in Extreme Scenarios

To demonstrate the benefits of our IMU bias learning approach, we evaluate it in scenarios where visual features

TABLE IV: Evaluation on *EuRoC* and *Aerodrome*: best in bold, second-best underlined, averaged over five runs per sequence.

Metrics	Sequence	MSCKF [23]	MSCEqF [16]	LBMSCKF	Ours
ATE trans. [m]	V102	**0.111**	0.140	0.122	0.129
ATE rot. [deg]	V102	3.931	**1.470**	2.877	2.193
ATE trans. [m]	V103	0.158	0.164	0.157	**0.140**
ATE rot. [deg]	V103	**0.858**	3.598	1.834	1.584
ATE trans. [m]	V202	0.148	0.182	0.153	**0.125**
ATE rot. [deg]	V202	2.372	**1.707**	3.973	2.198
ATE trans. [m]	A03	0.220	-	**0.142**	**0.142**
ATE rot. [deg]	A03	1.524	-	**0.524**	0.641
ATE trans. [m]	A04	1.355	-	0.742	**0.386**
ATE rot. [deg]	A04	4.841	-	**3.508**	3.860
ATE trans. [m]	A05	0.201	-	0.191	**0.190**
ATE rot. [deg]	A05	1.382	-	**0.773**	0.821

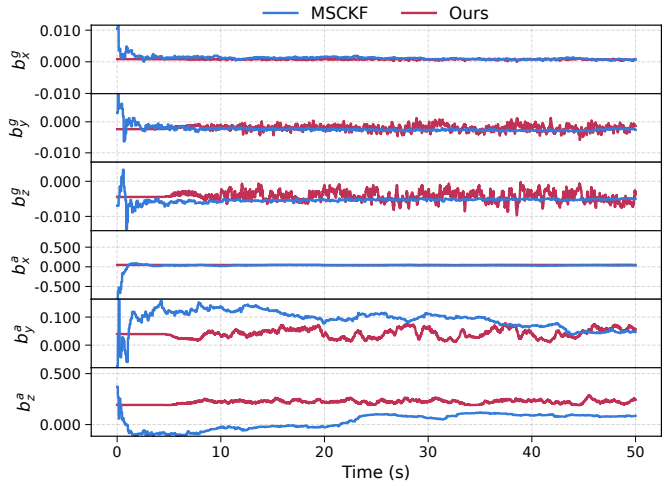


Fig. 5: Comparison of IMU bias estimates over time on *Aerodrome* A03 sequence between MSCKF (blue) and our network (red).

are temporarily lost, requiring the filter to rely solely on the IMU measurements for motion estimation. We introduced a single visual feature failure point of durations 1, 2, 3, and 4 seconds to each of the A03, A04, and A05 sequences of the *Aerodrome* dataset. In Fig. 6, we show the ATE in translation averaged over all three sequences and provide a sample trajectory estimate for the A03 sequence. Our proposed method outperforms the MSCKF, particularly in instances where the filter's bias estimation is inaccurate. The MSCKF estimates bias by minimizing visual measurement residuals, an approach that may not yield the true IMU bias [10]. In contrast, our method predicts the IMU bias independently of visual information, resulting in improved reliability. Therefore, accurate IMU bias estimation becomes critical during visual feature blackouts to maintain reliable inertial integration. To illustrate this, Fig. 5 demonstrates that under normal conditions with consistent visual measurements, MSCKF bias estimates often converge to values already predicted by our method. Additionally, while the true IMU bias typically exhibits slow time-varying behavior, the MSCKF bias estimates fluctuate, which is inconsistent with expected physical behavior.

D. Comparison with IMO

So far, our IMU bias prediction model was used in the MSCKF propagation step. Here, we use it in the update step and benchmark against IMO [9] for patterned motion without visual information on the Blackbird dataset [29]. IMO predicts relative positions from world-frame IMU with a neural network and uses them as update measurements for an EKF.

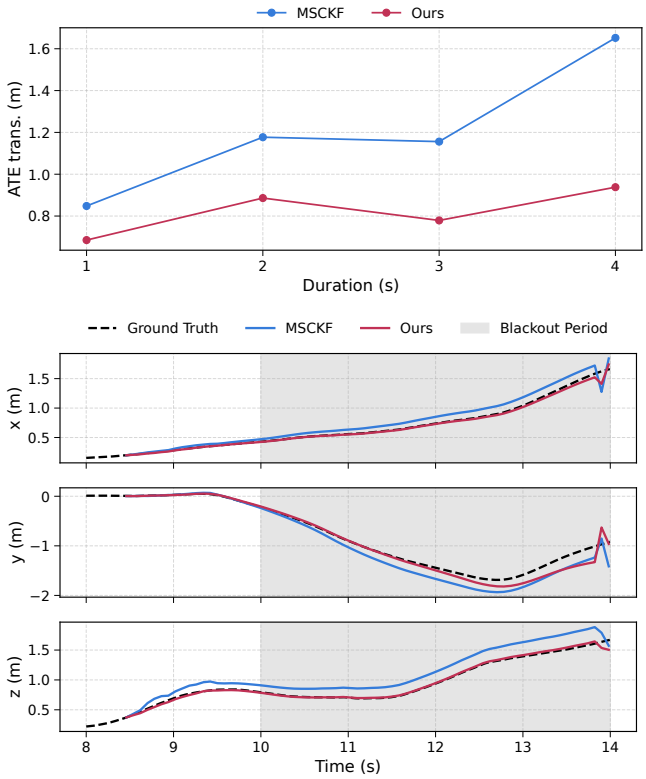


Fig. 6: (Top) Translational ATE under different visual blackout durations, averaged across *Aerodrome* sequences. (Bottom) Position estimates on *Aerodrome* A03 with a 4-second blackout at 10 seconds.

TABLE V: Computation times in milliseconds [ms] of our pipeline.

Processing Step	Min	Mean	Max
Tracking	6.83	18.06	67.67
Propagation	0.28	0.77	2.48
Update	0.03	9.19	50.01
Bias inference	**0.56**	**2.18**	**9.10**
Retriangulation and marginalization	4.35	10.11	22.47
Total	14.99	38.14	102.62

IMO also uses a sliding window, similar to the MSCKF, to attenuate large error instances from the network predictions.

Our method differs by predicting IMU biases directly from the IMU measurements, correcting the measurements, and integrating to obtain relative position estimates. The biases estimated in this section no longer represent the physical IMU biases alone. Instead, they represent the physical biases plus a correction that compensates the unobservable initial velocity so that the integrated position increments align with the ground truth. We compute the relative velocity and position increments $\Delta v_{ij} = v_j - v_i$ and $\Delta p_{ij} = p_j - p_i$ as:

$$\Delta v_{ij} = \sum_{k=i}^{j-1} \left(R_k(\bar{a}_k - \hat{b}_k^a) + g \right) \Delta t_k, \tag{22}$$

$$\Delta p_{ij} = \sum_{k=i}^{j-1} (v_i + \Delta v_{ik}) \Delta t_k + \frac{1}{2} \left(R_k(\bar{a}_k - \hat{b}_k^a) + g \right) \Delta t_k^2.$$

Because Δp_{ij} depends on the unobservable initial velocity v_i, both IMO and our approach implicitly compensate the term $\sum_{k=i}^{j-1} v_i \Delta t_k$. We train only the accelerometer bias with the

TABLE VI: Inertial odometry on *Blackbird Clover* over 30 seconds.

Metrics	Sequence	TLIO [7]	IMO [9]	Ours
ATE x-axis [m]	Clover	7.340	**0.357**	0.431
ATE y-axis [m]	Clover	3.139	**0.207**	0.299
ATE z-axis [m]	Clover	2.332	**0.061**	0.571
ATE trans. [m]	Clover	8.318	**0.417**	0.776
ATE rot. [deg]	Clover	75.341	**3.028**	7.736

loss $\|\Delta p_{ij} - \Delta\hat{p}_{ij}\|^2$, where

$$\Delta\hat{p}_{ij} = \sum_{k=i}^{j-1} \Delta v_{ik}\Delta t_k + \frac{1}{2}\left(R_k(\bar{a}_k - \hat{b}_k^a) + g\right)\Delta t_k^2.$$

Note that $\Delta\hat{p}_{ij}$ omits the initial velocity term. Following [9], we use ground-truth R_k during training and the estimated $\hat{R}_k$ from the filter at deployment. IMO performs well on in-distribution data, achieving an ATE of 0.418 in translation and 3.028 in rotation. This accurate performance arises from directly estimating relative positions from rotated IMU measurements in the world frame, simplifying the network's learning task. However, IMO struggles with out-of-distribution data since it implicitly learns initial velocities specific to patterned motion. In contrast, our method achieves an ATE of 0.772 in translation and 14.156 in rotation, yielding relatively accurate estimates along the x and y axes, as presented in Table VI. Our method is less accurate along the z axis due to the double integration of accelerometer measurements $\bar{a}_k$ along with gravitational acceleration g and bias compensation $\hat{b}_k^a$, which makes the network's learning task more challenging. However, IMO is designed mainly for improving state estimation around known trajectories, e.g., for drone racing. Meanwhile, the primary strength of our approach lies in generalizing to unseen data through IMU bias estimation, which enables the use of an invariant filter in VIO, shown in Sec. V-B and V-C.

VI. CONCLUSION

We developed a learning-based invariant filter for VIO, estimating the IMU bias externally to the filter state via a neural network. This allows us to preserve the system's invariance and achieve robustness over traditional VIO methods, especially in visually degraded scenarios. Future work will focus on learning the measurement uncertainty to assess the reliability of the IMU measurements more accurately and in addition to using additional information as input to the neural network, such as past images or image features.

REFERENCES

[1] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *IEEE International Conference on Robotics and Automation*, pp. 2502–2509, 2018.

[2] G. Huang, "Visual-inertial navigation: A concise review," in *IEEE International Conference on Robotics and Automation*, pp. 9572–9582, 2019.

[3] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-Aided Inertial Navigation," in *IEEE International Conference on Robotics and Automation*, pp. 3565–3572, 2007.

[4] J. Hernandez, K. Tsotsos, and S. Soatto, "Observability, Identifiability and Sensitivity of Vision-Aided Inertial Navigation," in *IEEE International Conference on Robotics and Automation*, pp. 2319–2325, 2015.

[5] C. Chen, X. Lu, A. Markham, and N. Trigoni, "IONet: Learning to Cure the Curse of Drift in Inertial Odometry," in *AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[6] S. Herath, H. Yan, and Y. Furukawa, "RoNIN: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, & New Methods," in *IEEE International Conference on Robotics and Automation*, pp. 3146–3152, 2020.

[7] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, "TLIO: Tight Learned Inertial Odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5653–5660, 2020.

[8] 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.

[9] G. Cioffi, L. Bauersfeld, E. Kaufmann, and D. Scaramuzza, "Learned Inertial Odometry for Autonomous Drone Racing," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2684–2691, 2023.

[10] 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.

[11] Y. Qiu, C. Wang, X. Zhou, Y. Xia, and S. Scherer, "AirIMU: Learning uncertainty propagation for inertial odometry," *arXiv preprint: 2310.04874*, 2023.

[12] M. Brossard, S. Bonnabel, and A. Barrau, "Denoising IMU Gyroscopes with Deep Learning for Open-Loop Attitude Estimation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4796–4803, 2020.

[13] A. Barrau and S. Bonnabel, "The Invariant Extended Kalman Filter as a Stable Observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2016.

[14] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-Aided Invariant Extended Kalman Filtering for Robot State Estimation," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 402–430, 2020.

[15] T.-Y. Lin, T. Li, W. Tong, and M. Ghaffari, "Proprioceptive Invariant Robot State Estimation," *arXiv preprint arXiv:2311.04320*, 2023.

[16] A. Fornasier, P. van Goor, E. Allak, R. Mahony, and S. Weiss, "MSCEqF: A Multi State Constraint Equivariant Filter for Vision-Aided Inertial Navigation," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 731–738, 2023.

[17] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based Rules for Designing Consistent EKF SLAM Estimators," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, 2010.

[18] A. Fornasier, Y. Ge, P. van Goor, R. Mahony, and S. Weiss, "Equivariant symmetries for inertial navigation systems," *arXiv preprint arXiv:2309.03765*, 2023.

[19] A. Fornasier, "Equivariant Symmetries for Aided Inertial Navigation," *arXiv preprint arXiv:2407.14297*, 2024.

[20] M. Brossard, A. Barrau, P. Chauchat, and S. Bonnabel, "Associating Uncertainty to Extended Poses for on Lie Group IMU Preintegration With Rotating Earth," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 998–1015, 2021.

[21] J. R. Dormand and P. J. Prince, "A Family of Embedded Runge-Kutta Formulae," *Journal of computational and applied mathematics*, vol. 6, no. 1, pp. 19–26, 1980.

[22] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2024.

[23] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *IEEE International Conference on Robotics and Automation*, pp. 4666–4672, 2020.

[24] E. Rosten, R. Porter, and T. Drummond, "Faster and Better: A Machine Learning Approach to Corner Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2008.

[25] M. Shan, Q. Feng, and N. Atanasov, "OrcVIO: Object Residual Constrained Visual-Inertial Odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5104–5111, 2020.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint: 1412.6980*, 2017.

[27] 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.

[28] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual (-Inertial) Odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7244–7251, 2018.

[29] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, and S. Karaman, "The Blackbird UAV Dataset," *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1346–1364, 2020.