

Neural Inertial Odometry from Lie Events

Royina Karegoudra Jayanth Yinshuang Xu Evangelos Chatzipantazis Kostas Daniilidis Daniel Gehrig

Department of Computer Science, University of Pennsylvania

Philadelphia, PA 19104, USA

{royinakj, xuyin, vaghat, kostas, dgehrig}@seas.upenn.edu

Abstract—Neural displacement priors (NDP) can reduce the drift in inertial odometry and provide uncertainty estimates that can be readily fused with off-the-shelf filters. However, they fail to generalize to different IMU sampling rates and trajectory profiles, which limits their robustness in diverse settings. To address this challenge, we replace the traditional NDP inputs comprising raw IMU data with *Lie events* that are robust to input rate changes and have favorable invariances when observed under different trajectory profiles. Unlike raw IMU data sampled at fixed rates, Lie events are sampled whenever the norm of the IMU pre-integration change, mapped to the Lie algebra of the $SE(3)$ group, exceeds a threshold. Inspired by event-based vision, we generalize the notion of level-crossing on 1D signals to level-crossings on the Lie algebra and generalize binary polarities to normalized *Lie polarities* within this algebra. We show that training NDPs on Lie events incorporating these polarities reduces the trajectory error of off-the-shelf downstream inertial odometry methods by up to 21% with only minimal preprocessing. We conjecture that many more sensors than IMUs or cameras can benefit from an event-based sampling paradigm and that this work makes an important first step in this direction.

MULTIMEDIA MATERIAL

Open source code can be found here: https://github.com/RoyinaJayanth/NIO_Lie_Events.

I. INTRODUCTION

Visual inertial odometry (VIO) has become a staple of modern localization and navigation systems powering a diverse range of applications including Augmented and Virtual Reality (AR/VR) [9], autonomous driving, and robotics [17]. In short, it works by integrating accelerometer and gyroscope measurements from an inertial measurement unit (IMU) and correcting the resulting drift with observations from a standard frame camera [17]. However, the usefulness of these visual observations is often limited by the quality of the captured camera frames, which degrades significantly, especially in challenging lighting conditions and high-speed motion scenarios.

In seeking to overcome these limitations, a promising alternative has emerged, namely using *neural displacement priors* (NDPs). NDPs learn to map raw IMU measurements themselves to displacement and covariance terms, and these can then be used to correct drift, while circumventing the pitfalls of classical frame-based algorithms. Surprisingly, these priors have been shown to possess similar drift-correcting capabilities to frame-based observations [31], and have thus resulted in a resurgence of IMU-only, *i.e.* purely *Inertial Odometry* (IO) [31, 20, 6]. These neural priors generate

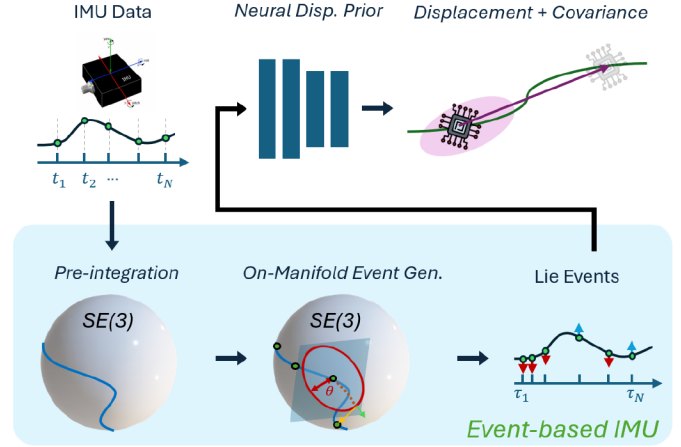


Fig. 1. Neural Inertial Odometry from Lie Events. We train Neural Displacement Priors (NDPs), which enable low-drift inertial odometry, with *Lie Events* derived from acceleration $\mathbf{a}(t_i)$ and angular rate $\boldsymbol{\omega}(t_i)$ measurements from an Inertial Measurement Unit (IMU). These events enhance the robustness of NDPs due to their favorable properties under varying sampling rates and trajectory profiles. To generate these events, we produce pre-integrations $\mathbf{x}(t)$ which reside in the special Euclidean group $SE(3)$ and then perform level-crossing on this signal which prompts the generalization of level-crossing and event polarities (red and blue arrows) to higher dimensional manifolds.

denoised displacement measurements with associated uncertainties by recognizing patterns in the IMU data and can be readily fused using off-the-shelf filters. However, *learning* generalizable priors has proven to be a challenging endeavor. This is because IMU data exhibits a high degree of variability as a result of differing IMU mount orientations, motion direction, and *motion patterns*. While orientation variability can be addressed via data augmentation [31], consistency losses [6] or equivariance [23], addressing the variability in motion patterns remains elusive. In particular, NDPs need to learn to ignore data variability which may arise from different gaits or speeds. Yet, modeling this variation for the purposes of enforcing consistency remains an open challenge.

In this work, we take on this challenge by training NDPs with *Lie Events* instead of raw IMU data, visualized in Fig. 1. Lie Events are generated when the change in IMU pre-integration exceeds a pre-specified threshold. IMU pre-integrations correspond to raw integrations of debiased IMU accelerometer and gyroscope measurements. Change is measured by projecting the endpoint of the pre-integrated path onto the Lie algebra of $SE(3)$ and taking its norm. We show both theoretically and empirically that this behavior imbues

the resulting events with favorable invariances with respect to different sampling rates and trajectory profiles, enhancing the robustness of downstream NDPs. To generate these events, we take inspiration from classical event-based vision [29, 38], and generalize notions of level-crossing and event polarities to arbitrary Lie groups. In particular, we generalize level-crossing on 1D signals to level-crossing in the Lie algebra centered at a reference element and introduce *Lie polarities* which generalize binary polarities to normalized elements in this Lie algebra. They characterize the direction of significant pre-integration change in the Lie algebra. Unlike their binary counterparts, Lie polarities are far richer since they can encode analog values. These generalizations reduce to the standard case in [29, 38] for 1D signals. Finally, we show that event generation partially canonicalizes the IMU data with respect to motion variations within an interval. This canonicalization transforms IMU sequences to a space with lower data variability and thereby enhances the generalizability of NDPs trained in this space. Our contributions are the following:

- We present a canonicalization scheme for IMU data which converts them into Lie Events. These Lie Events are directly compatible with a wide range of existing NDPs, and training on Lie Events enhances their robustness with only minimal preprocessing.
- We show both empirically and theoretically that these Lie Events possess favorable invariances with respect to motion variations within an interval, highlighting their benefit for NDP training.
- Finally, we derive the formalism necessary to generate Lie Events, which requires the extension of traditional notions of level-crossing and event polarities to arbitrary Lie groups. These tools are applicable in a wide range of settings, beyond IMU-based odometry.

To show the generality of our method we apply it to multiple down-stream NDPs and multiple datasets, where we show consistent error reductions, with only minimal preprocessing.

II. RELATED WORK

Neural Prior for IO: Learning-based inertial odometry (IO) effectively reduces integration drift by leveraging neural priors. Recent approaches [43, 1, 20] introduce neural priors via velocity regression. Specifically, RIDI [43] utilizes CNNs to predict velocity for IMU data refinement, PDRNet [1] applies recurrent networks for velocity regression, and RONIN [20] employs TCNs to directly integrate the predicted velocity for neural-based fusion. Some works [7, 31, 9, 40] learn the displacement prior. Specifically, TLIO [31] predicts both 3D displacements and diagonal covariances via a network, and incorporates them in an Extended Kalman Filter (EKF) via a measurement equation. RNIN-VIO [9] extends TLIO to continuous pedestrian motion estimation with visual inputs. In the meanwhile, there are some methods Brossard et al. [4], Buchanan et al. [5], Brossard et al. [3] that leverage the power of deep learning to denoise the motion and Steinbrener et al. [39] predicts the IMU biases. Our proposed Lie events

can be directly applied to TLIO [31], and enhance its robustness to variations in motion and sampling rate.

Canonicalization: Standardizing inputs [32, 44, 22] has long been widely practiced with both hand-crafted and learned techniques. Recently, canonicalization with equivariant models to improve generalization in off-the-shelf models has gained popularity. Kaba et al. [25] first theoretically proposed using Learned Canonicalization Functions to achieve equivariance with general off-the-shelf models. In practical applications, Li et al. [28], Baker et al. [2] canonicalize point clouds by constructing global and local frames, Deng et al. [10] introduces an equivariant canonicalization method in PointNet for 3D shape analysis and reconstruction, and Jayanth et al. [23] learns an equivariant frame to canonicalize IMU data with respect to reflections and rotations. Additionally, with the rise of large pretrained models, recent work [34] applies canonicalization as an input preprocessing step to such models.

Event-based Sensors: Event-based sensors are most prominently used in the computer vision community, where they address the bandwidth-latency tradeoff [12, 16] of algorithms based on fixed-rate images from standard frame-based image sensors. Capturing high-speed motion requires high framerates, and, as a result bandwidth, and reducing this framerate risks missing vital scene dynamics due to the increased latency. Event cameras like the dynamic vision sensor (DVS) [29], address this by adapting their sampling rate to the scene dynamics. They do this by responding only to changes, *i.e.* performing event-based sampling on the log intensity signal observed at a single pixel. Alternative cameras, like the Asynchronous Time-based Image Sensor (ATIS) [38], use the same event-based sampling technique but instead measure absolute intensities. Most recently, the work in [41] introduced the generalized event camera, which explored a large range of possible event cameras differing in the condition when an event is triggered, and what type of data is read out for each event. Beyond the vision domain, event-based sensing has been also generalized to asynchronous binaural spatial audition sensors [30], or tactile sensors [14]. A proof of concept presented in [37] shows the benefits of using event-guided depth-sensing, where an event camera is used to provide guidance to a structured light system for efficient scene reconstruction. This growing set of event-based sensors inspires the development of an event-based IMU, presented in this work. Similar to works in event-based control theory [24] which consider attitude consensus, our work focuses on generating events on the manifold. However, unlike [24], we use these events for inertial odometry and consider the full $SE(3)$ manifold instead of only $SO(3)$.

III. METHOD

The goal of this work is neural inertial odometry from a single IMU, comprised of an accelerometer providing linear acceleration measurements $\bar{a}(t_i) \in \mathbb{R}^3$, and a gyroscope providing angular velocity measurements $\bar{\omega}(t_i) \in \mathbb{R}^3$ at discrete times t_i . These measurements are related to the true

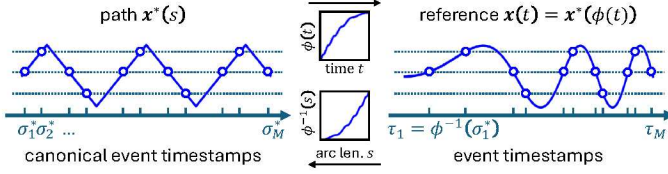


Fig. 2. Event timestamps τ_j generated from reference signal $\mathbf{x}(t) = \mathbf{x}^*(\phi(t))$ can be computed from the event timestamps σ_j^* of the path $\mathbf{x}^*(s)$, by applying the inverse mapping ϕ^{-1} . Moreover, references $\mathbf{x}_{\text{ref},j} = \mathbf{x}(\tau_j)$ and $\mathbf{x}_{\text{ref},j}^* = \mathbf{x}^*(\sigma_j^*)$ are equal and independent of a specific ϕ .

angular velocity $\boldsymbol{\omega}(t_i)$ and acceleration $\mathbf{a}(t_i)$ via

$$\bar{\boldsymbol{\omega}}(t_i) = \boldsymbol{\omega}(t_i) + \mathbf{b}^g(t_i) + \boldsymbol{\eta}^g(t_i) \quad (1)$$

$$\bar{\mathbf{a}}(t_i) = \mathbf{a}(t_i) - \mathbf{R}_{wb}^\top(t_i)\mathbf{g} + \mathbf{b}^a(t_i) + \boldsymbol{\eta}^a(t_i) \quad (2)$$

where $\mathbf{g} \in \mathbb{R}^3$ denotes the gravity direction pointing downward in world frame w , $\mathbf{R}_{wb}(t_i)$ is the transformation between body frame b and w at time t_i , and \mathbf{b}^g , \mathbf{b}^a and $\boldsymbol{\eta}^g$, $\boldsymbol{\eta}^a$ are IMU biases and noises respectively.

As in previous works [31] we assume access to a gravity direction and bias estimate via a filter, with which IMU measurements are gravity-compensated and bias-corrected:

$$\hat{\boldsymbol{\omega}}(t_i) = \hat{\mathbf{R}}_g(t_i)(\bar{\boldsymbol{\omega}}(t_i) - \hat{\mathbf{b}}^g(t_i)) \quad (3)$$

$$\hat{\mathbf{a}}(t_i) = \hat{\mathbf{R}}_g(t_i)(\bar{\mathbf{a}}(t_i) - \hat{\mathbf{b}}^a(t_i)) + \mathbf{g} \quad (4)$$

where $\hat{\mathbf{R}}_g(t_i)$ is the estimated gravity aligned frame, such that $\hat{\mathbf{R}}_g(t_i)\hat{\mathbf{R}}_{wb}^\top(t_i)\mathbf{g} = \hat{\mathbf{R}}_\gamma(t_i)\mathbf{g} = \mathbf{g}$ and $\hat{\mathbf{R}}_\gamma(t_i)$ is an unobservable yaw around the gravity axis. The resulting measurements thus mimic the true body rates $\boldsymbol{\omega}(t_i)$, $\mathbf{a}(t_i)$. Note that estimates of gravity and biases can be easily found by running an EKF [31].

We address inertial odometry with *neural displacement priors* that regress displacements and covariances from a sequence of IMU measurements regularly sampled at timestamps $t_i = i\Delta t$ within a time interval $\mathcal{I} = [0, T]$. We will denote these sampling times with $\mathcal{T} = (t_1, \dots, t_N)$ and $N = \lfloor \frac{T}{\Delta t} \rfloor$. We will use the shorthand $\hat{\boldsymbol{\omega}}(\mathcal{T})$ to denote the set $\{\hat{\boldsymbol{\omega}}(t_i)\}_{t_i \in \mathcal{T}}$. Then the displacement prior can be written as

$$\mathbf{d}, \boldsymbol{\Sigma} = \Phi(\hat{\boldsymbol{\omega}}(\mathcal{T}), \hat{\mathbf{a}}(\mathcal{T})). \quad (5)$$

Here $\mathbf{d} \in \mathbb{R}^3$ is the displacement between times $t = 0$ and $t = T$, and $\boldsymbol{\Sigma} \in \mathbb{R}^{3 \times 3}$ is its associated covariance. Note that our exposition also encompasses methods like [20] which predict averaged velocity \mathbf{v} over a time window. Displacement can be simply recovered via $\mathbf{d} = \mathbf{v}T$, with constant T .

In what follows, we will derive steps to make the above neural network robust to variations in IMU speed, thus enhancing its generalizability. In particular, we achieve this by generating canonicalized inputs $\hat{\boldsymbol{\omega}}(\mathcal{E})$, $\hat{\mathbf{a}}(\mathcal{E})$ sampled at specifically chosen *event timestamps* $\mathcal{E} = (\tau_1, \dots, \tau_M)$, and augmented by *Lie polarities* $\mathbf{p}(\mathcal{E})$ with $\mathbf{p}(\tau_j) \in \mathbb{S}^5 \subset \mathbb{R}^6$ via a suitable preprocessing step, using on-manifold event generation. These polarities are normalized vectors that encode the change direction of pre-integrations, and will be defined

later. We term these network inputs *Lie events*. In summary, our neural network uses inputs

$$\mathbf{d}, \boldsymbol{\Sigma} = \Phi(\hat{\boldsymbol{\omega}}(\mathcal{E}), \hat{\mathbf{a}}(\mathcal{E}), \mathbf{p}(\mathcal{E})). \quad (6)$$

In particular, we parametrize the uncertainty with a diagonal covariance $\boldsymbol{\Sigma} = \exp(\text{diag}(2u_x, 2u_y, 2u_z))$ with learnable u_x, u_y, u_z which ensures positive definiteness of $\boldsymbol{\Sigma}$. Next, we will study the impact of speed variations on the original neural network inputs, and then how to derive the new inputs above.

A. Modeling Speed Variations in IMU motion

Let the IMU trajectory¹ $\mathbf{T}(t) = (\mathbf{R}(t), \mathbf{t}(t)) \in SE(3)$ on the interval \mathcal{I} be decomposed into

$$\mathbf{T}(t) = \mathbf{T}^*(\phi(t)) \quad (7)$$

with *path*² $\mathbf{T}^*(s)$ parametrized by arc-length s and unit velocity $\|\mathbf{v}^*(s)\| = 1$ and *time parametrization* $\phi: \mathcal{I} \rightarrow [0, 1]$ with speed $\dot{\phi}(t) > 0$ and $\phi(0) = 0$ and $\phi(T) = 1$. Note that network outputs in Eq. (6) only depend on the path, since

$$\mathbf{d} = \mathbf{R}_0^\top(\mathbf{t}_T - \mathbf{t}_0) = \mathbf{R}_0^*\mathbf{t}_1^* - \mathbf{t}_0^* = \mathbf{d}^* \quad (8)$$

where $\mathbf{R}_0, \mathbf{t}_0$ and $\mathbf{R}_T, \mathbf{t}_T$ denote the pose at time $t = 0$ and $t = T$ respectively. Moreover, for methods like [20] which predict average velocity $\mathbf{v} = \frac{\mathbf{d}}{T} = \frac{\mathbf{d}^*}{T} = \mathbf{v}^*$ for a fixed and given T . Yet, the network inputs are affected by $\phi(t)$ via the kinematic relations

$$\boldsymbol{\omega}(t) = \left(\mathbf{R}^\top(t) \dot{\mathbf{R}}(t) \right)^\vee \quad \mathbf{a}(t) = \ddot{\mathbf{t}}(t) \quad (9)$$

where $(\cdot)^\vee$ maps a skew-symmetric matrix to a vector. Using the chain rule we find that

$$\boldsymbol{\omega}(t) = \dot{\phi}(t) \boldsymbol{\omega}^*(\phi(t)) \quad (10)$$

$$\mathbf{a}(t) = \dot{\phi}(t)^2 \mathbf{a}^*(\phi(t)) + \ddot{\phi}(t) \mathbf{v}^*(\phi(t)). \quad (11)$$

where canonical angular rate and acceleration are defined as $\boldsymbol{\omega}^*(t) = (\mathbf{R}'^\top(s) \mathbf{R}'(s))^\vee$ and $\mathbf{a}'(s) = \mathbf{t}''(s)$. Note the use of $(\cdot)'$ to denote derivatives, as these are no longer temporal derivatives. Thus the network inputs $\hat{\boldsymbol{\omega}}(\mathcal{T})$ and $\hat{\mathbf{a}}(\mathcal{T})$ depend on ϕ in the same form as above: First, ϕ modulates the function evaluation time, *i.e.* shifts the evaluation time from \mathcal{T} to $\phi(\mathcal{T})$, and second it modulates the magnitude of the measurement via its derivative. We perform partial canonicalization of the existing inputs and provide additional canonical inputs by introducing new event timestamps $\mathcal{E} = (\tau_1, \dots, \tau_M)$ and Lie polarities $\mathbf{p}(\mathcal{E})$. Crucially, these event timestamps depend on the trajectory $\mathbf{T}(t)$ taken by the IMU and, by construction, will depend on the event timestamps $\mathcal{E}^* = (\sigma_1^*, \dots, \sigma_M^*)$ derived from the *path* $\mathbf{T}^*(s)$ via $\mathcal{E} = \phi^{-1}(\mathcal{E}^*)$ ³. Likewise, we will construct polarities $\mathbf{p}(\tau_j)$ that only depend on the path, and

¹Subscripts *wb* (body to world transformation) are omitted for brevity.

²We distinguish between *path* which is simply the sequence of points, and *trajectory* which depends on time.

³Note that since $\dot{\phi}(t) > 0$ the function ϕ is invertible.

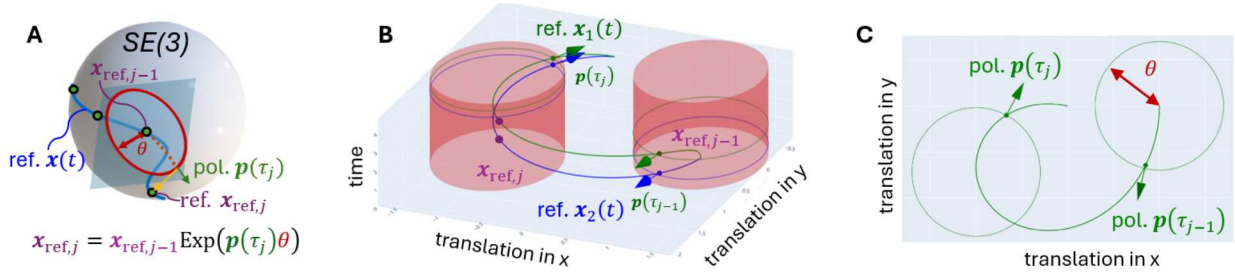


Fig. 3. Illustration of on-manifold event generation. (A) For a reference signal $\mathbf{x}(t)$ on $SE(3)$ we find the tangent space at the last reference pose $\mathbf{x}_{\text{ref},i}$. In this space, we find the moment it exits the θ -ball and record the *polarity* $\mathbf{p}(\tau_j)$ with unit norm perpendicular to that ball, and update the reference pose to $\mathbf{x}_{\text{ref},j} = \mathbf{x}_{\text{ref},j-1} \text{Exp}(\mathbf{p}(\tau_j)\theta)$. (B) Two trajectories $\mathbf{x}_1(t) = \mathbf{x}^*(\phi_1(t))$ and $\mathbf{x}_2(t) = \mathbf{x}^*(\phi_2(t))$ on the simplified manifold \mathbb{R}^2 with time parametrizations ϕ_1, ϕ_2 . (C) After projecting both trajectories onto the xy -plane, the reference signals $\mathbf{x}_{\text{ref},j}, \mathbf{x}_{\text{ref},j-1}$ and polarities $\mathbf{p}(\tau_j), \mathbf{p}(\tau_{j-1})$ are equal.

are thus equal to the polarities $\mathbf{p}^*(\sigma_j^*)$ derived from the path. As a result, the inputs to the network become

$$\omega(\mathcal{E}) = \dot{\phi}(\mathcal{E})\omega^*(\mathcal{E}^*) \quad (12)$$

$$\mathbf{a}(\mathcal{E}) = \dot{\phi}(\mathcal{E})^2 \mathbf{a}^*(\mathcal{E}^*) + \ddot{\phi}(\mathcal{E})\mathbf{v}^*(\mathcal{E}^*) \quad (13)$$

$$\mathbf{p}(\mathcal{E}) = \mathbf{p}^*(\mathcal{E}^*) \quad (14)$$

simplifying the dependence on ϕ . Note that $\mathbf{p}^*(\mathcal{E}^*), \mathbf{v}^*(\mathcal{E}^*), \omega^*(\mathcal{E}^*), \mathbf{a}^*(\mathcal{E}^*)$ are *canonicalized* i.e. only depend on path $\mathbf{T}^*(s)$, not $\mathbf{T}(t)$, and that the polarity is completely independent of ϕ . Thus networks trained on these quantities exhibit better generalization to variation in motion ϕ . Next, we discuss how to construct \mathcal{E} and $\mathbf{p}(\tau_j)$ with the properties outlined above.

B. Generating IMU Events

To construct the event timestamps \mathcal{E} we take inspiration from event-based cameras [15]. These cameras have independent pixels that trigger an event whenever the difference of the log intensity $\log(I(t))$ at a specific pixel with respect to some reference $\log(I_{\text{ref}})$ exceeds a threshold θ . We generalize this notion by identifying the pixel intensity with a new variable termed reference signal $\mathbf{x}(t) \doteq I(t)$. Thus, given that an event was triggered at the last time step τ_{j-1} and reference $\mathbf{x}_{\text{ref},j-1}$, the next event timestamp and reference can be written as

$$\tau_j = \min_{t > \tau_{j-1}} \{t : \theta \leq \|\log(\mathbf{x}(t)) - \log(\mathbf{x}_{\text{ref},j-1})\|\}, \quad (15)$$

and $\mathbf{x}_{\text{ref},j} = \mathbf{x}(\tau_j)$. Furthermore, event cameras report a polarity $\mathbf{p}(\tau_j)$ which can be defined as

$$\mathbf{p}(\tau_j) = \frac{\mathbf{x}_{\text{ref},j} - \mathbf{x}_{\text{ref},j-1}}{\|\mathbf{x}_{\text{ref},j} - \mathbf{x}_{\text{ref},j-1}\|} \in \{-1, 1\} \quad (16)$$

i.e. the sign of the change since the last reference. Using the polarities, the references can be reconstructed via

$$\mathbf{x}_{\text{ref},j} = \mathbf{x}_{\text{ref},j-1} \text{Exp}(\theta \mathbf{p}(\tau_j)) \quad (17)$$

Setting the initial event timestamp $\tau_1 = 0$, and reference to $\mathbf{x}_{\text{ref},1} \doteq \mathbf{x}(\tau_1)$ we can generate event timestamps τ_1, \dots, τ_M , references $\mathbf{x}_{\text{ref},1}, \dots, \mathbf{x}_{\text{ref},M}$, as well as event polarities $\mathbf{p}(\tau_2), \dots, \mathbf{p}(\tau_M)$ recursively from a given signal $\mathbf{x}(t)$. We will show that such a sampling scheme has the desired property $\mathcal{E} = \phi^{-1}(\mathcal{E}^*)$. In particular, as the log intensity $L(t)$ at a given pixel, we can let the reference signal $\mathbf{x}(t)$ depend

on a canonical reference $\mathbf{x}(t) = \mathbf{x}^*(\phi(t))$. We can inductively prove the following theorem (see appendix, Sec. I):

Theorem 1. *If $\mathbf{x}(t) = \mathbf{x}^*(\phi(t))$ with $\phi'(t) > 0$ and $\phi(0) = 0, \phi(T) = 1$, then $\tau_j = \phi^{-1}(\sigma_j^*)$ and $\mathbf{x}_{\text{ref},j} = \mathbf{x}_{\text{ref},j}^*$, where σ_j^* and $\mathbf{x}_{\text{ref},j}^*$ are generated from $\mathbf{x}^*(s)$.*

The implications of this theorem are that for all j we can simply find the event timestamps τ_j by computing σ_j^* from the path $\mathbf{x}^*(s)$ and applying ϕ^{-1} . Note that, since $\mathbf{x}_{\text{ref},j} = \mathbf{x}_{\text{ref},j}^*$, the polarity $\mathbf{p}(\tau_j)$, defined in Eq. 16, only depends on events on the path $\mathbf{x}^*(s)$, i.e. $\mathbf{p}(\tau_j) = \mathbf{p}^*(\sigma_j^*)$. We visualize these relations in Fig. 2. This theorem establishes the core invariance property of Lie polarities (dependent on $\mathbf{x}_{\text{ref},j}$) to variations in parametrization $\phi(t)$. Moreover, it shows that we can derive such polarities without needing to directly access unobservable $\phi(t)$ and $\mathbf{x}^*(s)$ by simply performing event-based sampling on observable $\mathbf{x}(t)$. Crucially, we never explicitly compute $\mathbf{x}^*(s)$ or $\phi(t)$. Having established these event properties, we turn to finding a suitable reference signal $\mathbf{x}(t)$ for IMU inertial odometry. We will show that the codomain of this reference is $SE(3)$, and thus, we extend the notion of event-based sampling to manifolds.

C. On-manifold Event Generation

Several options exist for selecting the reference signal as long as it can be written in terms of $\mathbf{x}^*(s)$ as $\mathbf{x}(t) = \mathbf{x}^*(\phi(t))$. This excludes acceleration- or velocity-like signals since these transform as \mathbf{a}, ω (i.e. depend on $\dot{\phi}, \phi$), but includes distance-like signals, e.g. the distance over time from some selected origin. We use relative IMU pose estimates over the time interval, which can be approximated via IMU pre-integrations $\mathbf{x}(t) = (\tilde{\mathbf{R}}(t), \tilde{\mathbf{t}}(t))$. These terms are the on-manifold forward Euler integrated solution $\tilde{\mathbf{R}}(t), \tilde{\mathbf{t}}(t)$ to (9), using raw IMU measurements $\tilde{\mathbf{a}}(t_i), \tilde{\omega}(t_i)$. Forster et al. [13] provides formulae for these, given initial pose $(\tilde{\mathbf{R}}(t_1), \tilde{\mathbf{t}}(t_1)) = (\tilde{\mathbf{R}}(t_1), \tilde{\mathbf{t}}(t_1))$ and velocity $\tilde{\mathbf{v}}(t_1) = \mathbf{v}_0$, namely

$$\tilde{\mathbf{R}}(t_{i+1}) = \tilde{\mathbf{R}}(t_i) \text{Exp}(\{\tilde{\omega}(t_i) - \hat{\mathbf{b}}^g(t_1)\} \Delta t) \quad (18)$$

$$\tilde{\mathbf{v}}(t_{i+1}) = \tilde{\mathbf{v}}(t_i) + \tilde{\mathbf{R}}(t_i)(\tilde{\mathbf{a}}(t_i) - \hat{\mathbf{b}}^a(t_1)) \Delta t + \mathbf{g} \Delta t \quad (19)$$

$$\begin{aligned} \tilde{\mathbf{t}}(t_{i+1}) &= \tilde{\mathbf{t}}(t_i) + \tilde{\mathbf{v}}(t_i) \Delta t + \frac{1}{2} \mathbf{g} \Delta t^2 \\ &+ \frac{1}{2} \tilde{\mathbf{R}}(t_i)(\tilde{\mathbf{a}}(t_i) - \hat{\mathbf{b}}^a(t_1)) \Delta t^2 \end{aligned} \quad (20)$$

which provides finite samples $\tilde{\mathbf{R}}(t_i), \tilde{\mathbf{t}}(t_i)$ for $t_i \in \mathcal{T}$. We define the continuous signal $\mathbf{x}(t)$ by interpolating between these samples. In particular, between samples i and $i+1$ we construct the geodesic path defined on $[t_i, t_{i+1}]$ as

$$\mathbf{x}(t) = \mathbf{x}(t_i) \text{Exp} \left(\frac{t - t_i}{t_{i+1} - t_i} \text{Log}(\mathbf{x}^{-1}(t_i) \mathbf{x}(t_{i+1})) \right) \quad (21)$$

Here $\text{Log} : SE(3) \rightarrow \mathfrak{se}(3)$ is the log map, which sends relative poses to elements in the tangent space, and $\text{Exp} : \mathfrak{se}(3) \rightarrow SE(3)$ is the (inverse) exponential map. In Sec. IV-D, we show that event generation is robust to IMU rate variations, and interpolation errors.

While during test time we use the EKF state to set \mathbf{v}_0 , during training we use ground truth, and add random perturbations to \mathbf{v}_0 to simulate filter uncertainty. Since pre-integrations mimic the true pose of the IMU, they approximately depend on pre-integration paths $\tilde{\mathbf{R}}^*(s), \tilde{\mathbf{t}}^*(s)$ via

$$\tilde{\mathbf{t}}(t) = \tilde{\mathbf{t}}^*(\phi(t)) \quad \tilde{\mathbf{R}}(t) = \tilde{\mathbf{R}}^*(\phi(t)), \quad (22)$$

While previously our reference signal was simply a 1-D signal (log intensity), the above pre-integrations naturally reside in $SE(3)$. Thus notions such level-crossing, and polarities need to be generalized. We start off by replacing the notion of signal change between $\mathbf{x}_{\text{ref},j-1}$ and $\mathbf{x}(t)$ with the geodesic distance in $SE(3)$, and use this to replace the L_2 -norm in Eq. 15

$$\tau_j = \min_{\tau > \tau_{j-1}} \left\{ \tau : \theta \leq \left\| \text{Log} \left(\mathbf{x}_{\text{ref},j-1}^{-1} \mathbf{x}(t) \right) \right\| \right\} \quad (23)$$

and $\mathbf{x}_{\text{ref},j} = \mathbf{x}(\tau_j)$. Note that this equation mimics Eq. (34) in [24] applied to event-based attitude control on $SO(3)$. We extend this to $SE(3)$ and introduce the notion of event polarities in this space, which we use to enhance the robustness of our method. We visualize the event generation process in Fig. 3 (a). We see that on the manifold, event generation amounts to finding the timestamp when the signal projected to the tangent space centered at $\mathbf{x}_{\text{ref},j-1}$ steps outside of a ball of dimension 6. At the point where this crossing happens we record the unit vector perpendicular to this sphere and define it as being the *Lie polarity* of the event at time τ_j :

$$\mathbf{p}(\tau_j) = \frac{\text{Log}(\mathbf{x}_{\text{ref},j-1}^{-1} \mathbf{x}_{\text{ref},j})}{\left\| \text{Log}(\mathbf{x}_{\text{ref},j-1}^{-1} \mathbf{x}_{\text{ref},j}) \right\|} \in \mathbb{S}^5 \quad (24)$$

where \mathbb{S}^5 denotes the 6D unit sphere. we can reconstruct the original signal with $\mathbf{p}(\tau_j)$ recursively

$$\mathbf{x}_{\text{ref},j} = \mathbf{x}_{\text{ref},j-1} \text{Exp}(\theta \mathbf{p}(\tau_j)). \quad (25)$$

These closely mimic the formulae for event generation in Eqs. (15) and (16), but are valid for arbitrary Lie groups. Moreover, when $\mathbf{x}(t) \in \mathbb{R}$, Eqs. (23), (24), and (25) above reduce to Eqs. (15), (16), and (17) and polarities $\mathbf{p}(\tau_j) \in \mathbb{S}^0 = \{-1, 1\}$. Note also that Theorem 1 can be seamlessly transferred to this extension, as is proven in the appendix, Sec. I.

In Fig. 3 (b) and (c) we illustrate the invariance of polarities to time reparametrizations by considering a toy example of a dot moving in \mathbb{R}^2 without orientation. We visualize two reference signals $\mathbf{x}_1(t) = \mathbf{x}^*(\phi_1(t))$ and $\mathbf{x}_2(t) = \mathbf{x}^*(\phi_2(t))$, with

different speed profiles. The tangent planes around reference poses in \mathbb{R}^2 can be visualized as planes perpendicular to time, and the θ ball can be visualized as a cylinder parallel to time. Polarities are 2D unit vectors perpendicular to the cylinder. Projecting this image onto the xy -plane (Fig. 3 (c)) reveals that the polarities only depend on the path $\mathbf{x}^*(s)$, not ϕ .

D. Network Inputs

As our method generates a variable number of events it is not directly compatible with off-the-shelf neural displacement priors [31] which are designed to handle a fixed number of IMU measurements. Inspired by event-based vision, we use event stacks [35]. We convert Lie events $\hat{\omega}(\mathcal{E}), \hat{\mathbf{a}}(\mathcal{E}), \mathbf{p}(\mathcal{E})$ to tensors $E_{a,\omega}, E_{\mathbf{p}} \in \mathbb{R}^{B \times 6}$ with $B = 200$ via:

$$E_{a,\omega}[b] = \frac{1}{n_b} \sum_{j=1}^M [\hat{\mathbf{a}}(\tau_j) \|\hat{\omega}(\tau_j)\| \delta[b - j^*]] \quad (26)$$

$$E_{\mathbf{p}}[b] = \frac{1}{l_b} \sum_{j=1}^M \mathbf{p}(\tau_j) \delta[b - j^*] \quad (27)$$

where M is the number of events triggered on the interval $[0, T]$, and δ is the Kronecker delta. Note that M can be much smaller than B , leading to sparsity in the inputs. Future work could tackle leveraging this sparsity for efficient processing. Acceleration and angular rates have been concatenated. Here n_b denotes the number of samples mapped to index b and l_b denotes the norm of the vector at index b so as to normalize the polarity. Furthermore, $j^* = \left\lfloor \frac{j-1}{M-1} (B-1) \right\rfloor$ maps the index of the event into a range $[0, B-1]$. Note that we linearly interpolate $\hat{\omega}, \hat{\mathbf{a}}$ to the event timestamp. We concatenate $E_{a,\omega}$ and $E_{\mathbf{p}}$ along the channel dimension, resulting in an input tensor of shape $B \times 12$.

Note on Timestamps: We do not feed in raw time stamps into the network since they vary with the unobservable speed $\phi(t)$ from the trajectory, not the path $\mathbf{x}^*(s)$. They would thus re-introduce data variability, which the network would need to learn to ignore. We believe the network can recover the displacement by leveraging event polarities. Eq. (25) shows that the path $\mathbf{x}^*(s)$, which is sufficient to find, \mathbf{d} can be recovered from polarities alone and without raw time stamps. We argue that the network learns to correct this reconstruction with priors and additional information from $\mathbf{a}(\tau_j)$ and $\omega(\tau_j)$.

IV. EXPERIMENTS

First, we present a toy experiment to validate the result of Theorem 1 in Sec. IV-A, before demonstrating the generality of our approach by applying it to the neural inertial odometry frameworks TLIO [31] (Sec. IV-B) and RoNIN [20] (Sec. IV-C). Furthermore, we will present a sensitivity study that highlights the robustness of our approach with respect to IMU rate variations in Sec. IV-D, and round off with ablation and hyperparameter sensitivity studies in Sec. IV-E.

Ref. $\mathbf{x}(t)$	$\phi_\alpha(t)$	Correction	Contrast Threshold		
			$\theta = 0.005$	$\theta = 0.01$	$\theta = 0.02$
pre-integration	t^2	no	2.18	3.62	5.52
	$t^{0.5}$	no	2.48	3.89	5.61
	t^2	yes	1.15	1.73	2.45
	$t^{0.5}$	yes	0.95	1.11	1.12
ground truth	t^2	no	5.2	10.4	2.39
	$t^{0.5}$	no	0.73	1.37	2.79
	t^2	yes	0.13	0.16	0.21
	$t^{0.5}$	yes	0.02	0.03	0.04

TABLE I

CHAMFER DISTANCE IN % BETWEEN EVENT TIMESTAMPS $\phi_\alpha(\mathcal{E})$ AND CANONICAL EVENT TIMESTAMPS \mathcal{E}^* . NO CORRECTION DENOTES THAT \mathcal{E} WAS NOT REMAPPED USING ϕ_α .

A. Toy Example

Here, we study the event timestamps \mathcal{E} generated from reference signals based on the pre-integration of IMU accelerations and angular rates, and show their relation to event timestamps \mathcal{E}^* generated from canonical pre-integrations. We use IMU data from the TLIO test dataset [31], which comprises 60 hours of 1 kHz IMU data, with 200 Hz ground truth trajectory data from MSCKF [36]. It was gathered from five individuals performing a broad range of activities, for example, walking, stair traversal, and kitchen organization. More dataset details are in the appendix, Sec. V.

We denote non-overlapping 1-second ground truth trajectories as paths $\mathbf{T}^*(s)$ and generate synthetic new trajectories $\mathbf{T}_\alpha(t) = \mathbf{T}^*(\phi_\alpha(t))$, where $\phi_\alpha(t) = (\frac{t}{\Delta t})^\alpha \in [0, 1]$ with $\Delta t = 1$ s and $\alpha \in \{0.5, 1.0, 2.0\}$. We then generate artificial IMU measurements by fitting a spline through the poses and finding derivatives. We apply bias, noise, and bias drift consistent with the noise magnitudes found in [31]. We then perform pre-integration on IMU data from the path $\hat{\mathbf{R}}^*(s), \hat{\mathbf{t}}^*(s)$ and trajectories $\hat{\mathbf{R}}_\alpha(t), \hat{\mathbf{t}}_\alpha(t)$. Finally, we generate events using different thresholds $\theta \in \{0.005, 0.01, 0.02\}$ producing canonical event timestamps \mathcal{E}^* and non-canonical event timestamps \mathcal{E}_α . Finally, we map \mathcal{E}_α to the canonical setting, i.e. compute $\phi_\alpha(\mathcal{E}_\alpha)$ and compare them with \mathcal{E}^* via the chamfer distance. For comparison, we also report the chamfer distance without remapping. We compare with noiseless ground truth poses as the reference signal to show the lower error bound.

Tab. I shows that event generation is capable of sampling the correct measurement even after time dilation. The smallest threshold 0.005 achieves an error of 1.15 % for a remapping of $\phi(t) = t^2$. We also see that using ground truth significantly reduces this error to about 0.1% in the same setting validating the theory for ideal references. Future research can benefit from more accurate pre-integration schemes.

B. Application to TLIO

TLIO is a learning-based inertial odometry algorithm that estimates the 6-DoF IMU pose with an extended Kalman filter (EKF). Propagation is performed by integrating raw accelerometer and gyroscope measurements, while measurement updates are performed with neural network-based displacement and covariance predictions $\mathbf{d} \in \mathbb{R}^3$ and $\Sigma \in \mathbb{R}^{3 \times 3}$. The network takes 1-second windows of 200 IMU measurements as input with shape 200×6 . IMU data is bias-corrected

using factory calibration bias values, and gravity aligned using the EKF orientation state. EKF details and propagation and measurement equations are provided in Sec. II, and network details in Sec. IV of the appendix.

Implementation Details We pre-integrate accelerometer and gyroscope measurement sequences of length 200 and obtain the orientation and position estimates in a gravity-aligned world frame. We use the EKF estimates of orientation, position, and initial velocity at the beginning of the time window, termed clone states, to initialize the pre-integration during testing. We then generate event timestamps and polarities using this reconstructed trajectory and a threshold $\theta = 0.01$. Sec. IV-E shows a sensitivity analysis motivating the choice of θ . For consistency, we map the polarity into the gravity-aligned frame after generation.

Datasets: We train our method on the TLIO Dataset [31], described in the appendix, and evaluate it on the TLIO Test Set and the Aria Everyday Activities (Aria) Datasets [33]. Aria is an egocentric dataset collected using the Project Aria glasses [11] and comprises a left (800 Hz) and right (1 kHz) IMU with ground truth position and orientation. It contains 7.3 hours of data and includes a wide range of wearers engaged in everyday activities like reading, exercising, and relaxing. More dataset details are in the appendix, Sec. V.

Training Details: We train the first 10 epochs we train with a component-wise MSE loss [31], so that the displacement prediction converges, and we then, for 40 epochs, switch to a Maximum Likelihood Error (MLE) loss [31] which incorporates both the predicted displacement and diagonal covariance. We use a learning rate of 10^{-4} , the Adam optimizer [26], and a batch size of 1024. We train all our models on NVIDIA a40 GPUs. During training, we use ground truth initial velocities \mathbf{v}_0 for bootstrapping pre-integration, and perturb it with uniform noise $\mathbf{n}_{\mathbf{v}_0} \sim \mathcal{U}[-0.5, 0.5]$. To simulate noise in the gravity-aligned frame, we perturb the gravity direction by 5° uniformly at random and apply a random yaw. Furthermore, we add uniform noise $\mathbf{n}_\omega \sim \mathcal{U}[-0.05, 0.05]$ and $\mathbf{n}_a \sim \mathcal{U}[-0.2, 0.2]$ to angular rates, and accelerations simulating uncertainties in bias estimates. We also add noise $\mathbf{n}_p \sim \mathcal{U}[-0.5, 0.5]$ to polarities before renormalizing.

Baselines: For a fair comparison, we implement three variants of TLIO [31], which, similar to our method, target time scale robustness. During training, we randomly subsample the 200 Hz IMU data to rates $r \in \{20, 40, 100, 200\}$. Since natively TLIO requires an input of 200 samples, we convert the subsampled data back to 200 by interpolation (denoted *TLIO + interp.*) or by using the event stack in Eq. (26) (denoted *TLIO + splat.*). Note that our method (denoted *TLIO + events*) is not trained with rate augmentation.

Metrics: We report the following metrics used in [31, 6, 20], and written out in detail in the appendix, Sec. VI:

- *Mean Squared Error (MSE)* between predicted ($\hat{\mathbf{d}}(t_i)$) and ground truth displacements ($\mathbf{d}(t_i)$) averaged over the trajectory.
- *Absolute Translation Error (ATE)*, i.e. the RMSE between estimated ($\hat{\mathbf{t}}(t_i)$) and ground truth positions ($\mathbf{t}(t_i)$).

		NN evaluation						NN+EKF evaluation											
Model	rate aug.	TLIO Dataset		Aria Right		Aria Left		TLIO Dataset				Aria Right				Aria Left			
		MSE* (m ²)	ATE* (m)	MSE* (m ²)	ATE* (m)	MSE* (m ²)	ATE* (m)	ATE (m)	RTE (m)	Drift (%)	AYE (°)	ATE (m)	RTE (m)	Drift (%)	AYE (deg)	ATE (m)	RTE (m)	Drift (%)	AYE (°)
TLIO	✗	0.013	1.660	0.018	1.240	0.019	1.314	<u>1.410</u>	0.102	1.486	<u>1.832</u>	0.865	0.119	2.629	1.481	1.069	0.160	2.749	<u>5.203</u>
+ interp.	✓	0.013	1.519	0.019	1.195	0.020	1.326	<u>1.410</u>	0.109	1.302	<u>1.836</u>	<u>0.837</u>	<u>0.117</u>	2.456	1.441	<u>0.944</u>	0.163	2.335	5.214
+ splat.	✓	<u>0.014</u>	1.498	0.019	1.216	0.020	<u>1.292</u>	1.440	0.107	<u>1.242</u>	1.788	0.876	0.114	2.479	1.462	0.965	0.161	<u>2.425</u>	5.142
+ events (ours)	✗	0.015	1.445	0.015	1.099	0.015	1.064	1.282	<u>0.106</u>	0.953	1.863	0.836	<u>0.115</u>	<u>2.459</u>	<u>1.456</u>	0.942	0.165	<u>2.425</u>	5.215

TABLE II

COMPARISON OF OUR METHOD AGAINST BASELINES TLIO WITH AND WITHOUT DATA AUGMENTATION ON THE TLIO DATASET [31], AND ARIA EVERYDAY DATASETS [33]. NOTE HERE LEFT AND RIGHT DENOTE THE LEFT AND RIGHT IMUS IN THE ARIA DATASETS.

- Relative Translation Error (RTE), *i.e.* local differences between $\mathbf{t}(t_i)$ and $\bar{\mathbf{t}}(t_i)$ over a 1-second or (1 minute for RoNIN) window δt .
- Absolute Yaw Error (AYE), *i.e.* the yaw RMSE.
- Translational drift over the total distance traveled.

To limit the impact of outliers we report the median error over the different trajectories of the dataset. Furthermore, to evaluate the impact of the neural network and EKF separately, we evaluate two types of trajectories produced by our method: The first uses simple integration of the network outputs, and the corresponding metrics are denoted with a *, and the second uses the EKF, and is denoted without a *.

Results: We report the results in Tab. II, and trajectory plots in Sec. VII of the appendix. First, we see that, without EKF, events improve ATE* on all datasets. They reduce the ATE* on the TLIO Dataset by 13%, on Aria Right by 11%, and on Aria Left by 19% compared to base TLIO. Moreover, it reduces MSE* on most datasets, reducing by 17% on Aria Right, by 21% on Aria Left, but having a 15% higher MSE* than base TLIO on the TLIO Dataset. As in [31] we argue that the correlation between MSE and ATE is not exact, since other methods may produce displacement estimates with lower variance, but higher bias, introducing larger overall drift.

Our method's EKF accuracy (ATE), is 10% higher on the TLIO Dataset, 3% on the Aria Right Dataset, and 12% on the Aria Left Dataset compared to base TLIO. We also see that our method outperforms base TLIO in terms of drift but is outcompeted by methods that use data augmentation.

C. Application to RoNIN

RoNIN [19] is a 2D inertial navigation method designed for IMU-based pedestrian tracking. It works by regressing velocity predictions, $\hat{\mathbf{v}}(t) \in \mathbb{R}^2$ expressed in the local gravity aligned frame, from 1-second sequences of IMU measurements using a ResNet-18 [18]. These velocities are regressed with a 25ms stride, transformed into the global frame and then integrated using known orientations from the IMU.

Implementation Details: As in [20] we use the RoNIN dataset, which features 200 Hz IMU data. We convert sets of 200 measurements into events, using the same parameters as for TLIO but a higher contrast threshold $\theta = 0.1$ due to the higher average accelerations. We also use the last velocity prediction as initialization for pre-integration.

Datasets: We train our model on the public portion of the RoNIN dataset [20] which comprises 200 Hz IMU and ground truth trajectory data collected from pedestrians with varied

Model	Training Data	RONIN-U		RIDI-T		OxIOD	
		ATE* (m)	RTE* (m)	ATE* (m)	RTE* (m)	ATE* (m)	RTE* (m)
RoNIN + ResNet	100%	5.14	4.37	1.63	1.91	3.46	4.39
RoNIN + LSTM	100%	5.46	<u>3.95</u>	2.00	2.64	3.02	4.18
RoNIN + TCN	100%	5.87	4.39	1.66	1.95	3.56	4.98
RoNIN + ResNet	50%	5.57	4.38	1.19	1.75	3.52	4.42
+ J	50%	5.02	4.23	1.13	1.65	3.59	4.43
+ TTT	50%	5.05	4.14	1.04	1.53	2.92	3.67
+ J + TTT	50%	5.07	4.17	1.03	1.51	2.96	3.74
+ events (ours)	50%	5.35	4.63	1.03	1.97	1.52	1.91
Integration	-	458.06	117.06	31.06	37.53	1941.41	848.55

TABLE III

APPLICATION OF OUR METHOD TO THE RONIN ARCHITECTURE, AND COMPARISON ON THE RONIN, RIDI, AND THE OXFORD INERTIAL ODOMETRY DATASETS (OXIOD). NOTE, * INDICATES TRAJECTORY RECONSTRUCTION WITHOUT AN EKF, AS REPORTED IN [20, 6].

placements and devices. We report results on the RoNIN test set, and also on the RIDI [43] and Oxford Inertial Odometry Dataset (OxIOD) [8]. RIDI comprises pedestrian data at 200 Hz, with four different sensor placements, and includes forward, backward, and sideways walking motion as well as accelerating and decelerating motions. This dataset uses Umeyama alignment [42] before calculating metrics. The OxIOD dataset also comprises smartphone-based pedestrian data with various motion modes, devices, and device placements. More dataset details are in the appendix, Sec. V.

Training Details: Training mirrors [20] with 50% of the training data that is public, minimizing the robust velocity loss proposed in [20]. We use ADAM [26] with a batch size of 128 and an initial learning rate of 10^{-4} and a decay by a factor of 0.1 if the validation loss does not decrease in 10 epochs. The maximum number of epochs is 120. Furthermore, the linear layers use dropout with $p = 0.5$. As in [20] we omit gravity perturbation, and bias noise, and leave other noise sources and magnitudes the same.

Baselines: We compare our method with RoNIN operating on raw IMU data with three different temporal processing modules, namely ResNet-18 [18], LSTM [21] and TCN [27], and also compare against RIO [6]. RIO adds two additional strategies to RoNIN which enforce approximate yaw equivariance. The first one, termed joint training (denoted with +J), optimizes an auxiliary equivariant consistency loss between predictions observed under different yaws. The other strategy, adopted during testing and termed adaptive test-time-training (denoted with +TTT) performs gradient descent steps on the current model at 20 Hz, based on the equivariant consistency loss derived from a buffer of 128 past IMU measurements. Backpropagation is only performed on losses between samples

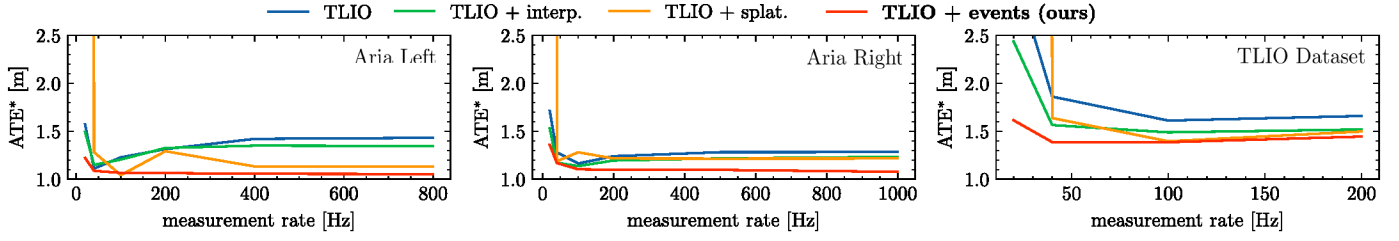


Fig. 4. IMU rate sensitivity analysis. Each method is trained on the TLIO training set. Methods + interp. and + splat. were trained with IMU rate augmentation and TLIO and TLIO + events were trained without data rate augmentation.

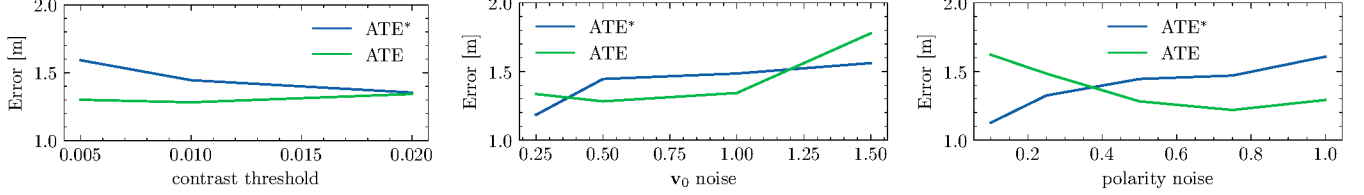


Fig. 5. Hyperparameter Sensitivity: (left) the contrast threshold determines the distance to the reference when events are fired, (middle) initial velocity \mathbf{v}_0 and (right) polarity $\mathbf{p}(\tau_j)$ noise determine the range of uniform noise perturbing these quantities during training. In particular, \mathbf{v}_0 noise affects pre-integration.

that have a significant predicted velocity mismatch. Furthermore, test-time training and model resetting are controlled via a separate ensemble of models that estimate prediction uncertainty online. Finally, we report the results of a naive double integration baseline (denoted with *Integration*).

Results: We report results in Tab. III, and trajectory plots in Sec. VII for the appendix. Note that the first three rows are results reported in [19] which use 100% of the training data. Other methods are retrained with the public training data, which is only 50%. We observe that, despite this mismatch, in terms of ATE*, our method outperforms RONIN on 50% data on the RONIN dataset, on RIDI, and on OxIOD. Test-time training and yaw consistency methods can be used complementarily to our method to further improve results.

Discussion: Similar to the results when applied to TLIO, the above results highlight the strength of using events instead of regular IMU samples. Moreover, we see that our method outperforms (i) methods that were trained with more data on RIDI-T and OxIOD (see row 1-3 in Tab. III) and (ii) methods that incorporate yaw consistency strategies on OxIOD. In particular, using yaw consistency strategies is an orthogonal approach to improving inertial odometry, which would additionally strengthen our result. Finally, we note that our method may benefit from a weak autoregression form by using the predicted velocity from the previous timestamp for pre-integration. However, compared to models that also perform autoregression, *i.e.* RoNIN+TCN or RoNIN+LSTM, our method achieves a lower ATE*.

D. Rate Sensitivity Study

In this section, we study the robustness of our method to varying IMU measurement rates which simulate differing time scales. We generate lower-rate IMU data by subsampling the TLIO data (200Hz) to $r = \{20, 40, 100, 200\}$, the Aria Left data (800Hz) to $r = \{20, 40, 100, 200, 400, 800\}$ and the Aria Right data (1000Hz) to $r = \{20, 40, 100, 200, 500, 1000\}$. This subsampling strategy tests the robustness of our event generation method to sampling errors. For the TLIO baselines

ev.	pol.	noise	ref. $\mathbf{x}(t)$	Manifold	TLIO Dataset (NN)		TLIO Dataset (NN+EKF)			
					MSE* (m^2)	ATE* (m)	ATE (m)	RTE (m)	Drift (%)	AYE (°)
✗	-	-	-	-	0.013	1.660	1.410	0.102	1.486	1.832
✗	✗	-	position	\mathbb{R}^3	0.014	1.515	1.724	0.110	1.580	1.846
✓	✗	-	pose	$SO(3) \times \mathbb{R}^3$	0.015	1.421	1.622	0.108	1.091	1.919
✓	✗	-	pose	$SE(3)$	0.016	1.425	1.420	0.108	0.982	1.862
✓	✓	✗	pose	$SE(3)$	0.005	0.912	6.080	0.240	3.140	1.700
✓	✓	✓	pose	$SE(3)$	0.015	1.445	1.282	0.106	0.953	1.863

TABLE IV
ABLATION ON EVENT GENERATION, POLARITY, AND (POLARITY) NOISE.

(base, +interp. and +splat.) we convert the IMU data to an equivalent of 200 inputs per 1 second of data with interpolation or event stack generation. We report the ATE* in Fig. 4, and a table with the full results in the appendix, Sec. VIII. Note that despite not being trained with data augmentation our method remains stable over differing rates, outperforming the baselines and TLIO. Most methods degrade heavily at low rates, while our method degrades gracefully.

E. Ablation Study and Hyperparameter Sensitivity Analysis

In this section, we ablate the design choices in the methodology, and show sensitivity analyses of the specific chosen hyperparameters. First, we study the impact of the reference signal $\mathbf{x}(t)$ and its associated manifold by comparing the ATE and ATE* on the TLIO test set. We select reference signals that monitor only the translation component $\tilde{\mathbf{t}}(t)$ of the pre-integration (labeled with “position”) and those that monitor the full pose $\tilde{\mathbf{R}}(t), \tilde{\mathbf{t}}(t)$ (labeled “pose”). While “position” admits only the \mathbb{R}^3 manifold, “pose” admits both the direct product manifold $SO(3) \times \mathbb{R}^3$ and the semidirect product manifold $SE(3) = SO(3) \ltimes \mathbb{R}^3$. These differ in the definition of the Log and Exp map used previously. We also study the impact of *Lie polarities* and the necessity of adding polarity noise. We report results in Tab. IV. We see that of the different considered manifolds, $SE(3)$ yields the best ATE* and ATE. This points to the fact that using the full pose and associated manifold yields the most informative events. Using polarities without noise yields very low ATE* but high ATE when deployed with the EKF. We argue that the neural

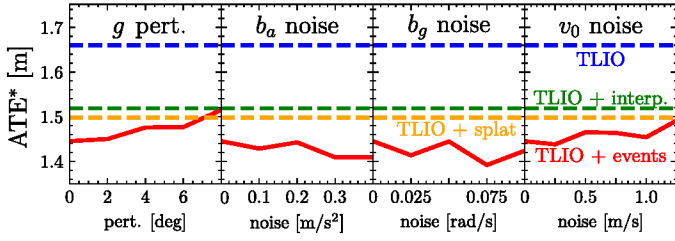


Fig. 6. Sensitivity to pre-integration errors. Baseline ATE* in dashed lines.

Testing Data	Training Data			
	50%	30%	20%	10%
RoNIN-U	(5.35/4.63)	(6.11/4.89)	(5.83/4.96)	(5.73/5.00)
RIDI-T	(1.03/1.97)	(1.23/2.50)	(1.22/2.44)	(1.43/2.77)
OxIOD	(1.52/1.91)	(1.48/2.82)	(1.52/2.18)	(1.51/2.48)

TABLE V
DATA EFFICIENCY OF OUR METHOD (ATE*/RTE*) IN METERS.

network learns to overfit to specific training trajectories, and thus becomes sensitive to noise in its predictions. Adding polarity noise during training significantly reduces the ATE when deployed with the EKF. Next, we study the sensitivity of our method to hyperparameters. Specifically, we vary the contrast threshold $\theta \in \{0.005, 0.1, 0.2\}$, polarity noise range $r_p \in \{0.1, 0.25, 0.5, 0.75, 1.0\}$ and initial velocity noise range $r_{v_0} \in \{0.25, 0.5, 1.0, 1.5\}$ applied during training and report results in Fig. 5, with tables with the full results in the appendix, Sec. VIII. First, we find an optimal ATE at $\theta = 0.01$. While too large thresholds may reduce the sensitivity of the method, too low thresholds may be more susceptible to noise and increase computational complexity. ATE reaches an optimum for initial velocity noise range $r_{v_0} = 0.5$. As previously discussed, using too low noise makes the network overfit to the specific trajectories and sensitive to noise, leading to excessively low ATE* but high ATE. Finally, the optimal polarity noise range is around $r_p = 0.75$. We found that this value yielded marginally worse results on other datasets, so we opted for $r_p = 0.5$.

F. Runtime, Robustness and Generalization

Runtime: Running TLIO for 1 second of data takes 1.14 ms on an NVIDIA GeForce RTX 4090 Laptop GPU, with an Intel i9 processor, enabling real-time measurement updates at the 20 Hz used in the experiments. The network (baseline TLIO, i.e. 1D ResNet-18, PyTorch) takes 1.06 ms. The C++-implemented event generation (including pre-integration) takes 76 μ s (4.22 μ s/event), and stacking takes 7.62 μ s (0.331 μ s/event). With an average event rate of 74 Hz (vs. 200 Hz for TLIO), our method reduces redundancy.

Sensitivity to IMU Pre-integration Accuracy: Inaccurate bias, gravity, and initial velocity estimation degrade pre-integration accuracy. To minimize this we train with initial velocity, gravity, and bias noise. Fig. 6 shows the sensitivity of ATE* (lower is better) with respect to gravity, bias, and initial velocity noise perturbation for pre-integration, on the TLIO test set, following the scheme in Sec. IV-B “training details”. These introduce drift, $SO(3)$ perturbations, and elucidate the role of gravity. Gravity and v_0 perturbations lead to graceful degradation, still outperforming other baselines, and

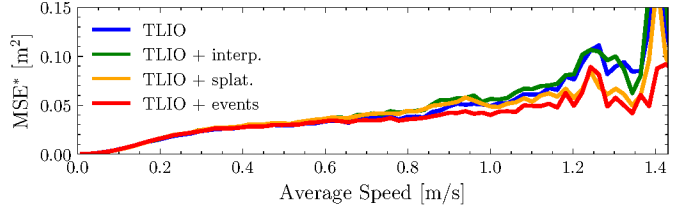


Fig. 7. Aria dataset MSE vs. speed for models trained on the TLIO dataset.

bias noises show little to no degradation. Slight improvements may stem from the fact that we train with noise. Overall this shows low impact of bias-induced drift, and higher sensitivity to gravity and velocity estimation errors.

Generalization: Tab. V reports our method trained on smaller subsets of the RoNIN training set. Our method’s performance stays relatively stable on both in- (RoNIN-U), and out-of-distribution data (RIDI-T, OxIOD). Notably, performance on RIDI-T degrades gracefully, while fluctuating on the other datasets. Even with 10% our method can learn reasonably well, underlining generalization, which we ascribe to the canonicalization capabilities of events.

Out-of-distribution data: We report the distribution of MSE versus average speed, for models trained on the TLIO dataset and tested on the out-of-distribution Aria datasets (Fig. 2). Our method outcompetes other methods with lower MSE at higher speeds, supporting the claim that events confer canonicalization with respect to speed variations. This targets a key limitation of existing methods, and shows improvements in edge cases with large accelerations.

V. LIMITATIONS AND CONCLUSION

Limitations: Pre-integrations as reference signals are efficient to compute, but suffer from slight drift and sampling noise, which may impact event generation. Yet, we show that event generation is robust to sampling noise. Finding better-behaved reference signals is the subject of future work.

Conclusion: This work enhances the robustness of neural inertial odometry methods by training the underlying *neural displacement priors* with *Lie events* generated from IMU pre-integrations. We showed both theoretically and empirically that these events have favorable canonicalization properties with respect to motion and rate variations in IMU data which simplifies network training. To generate events on pre-integrations, we generalize notions of level-crossing and event polarities to arbitrary Lie groups, and in particular, the special Euclidean group $SE(3)$. To this end, we introduce new concepts such as *Lie polarities*. Training on these events effectively reduces the tracking error of off-the-shelf neural inertial odometry methods with only minimal input pre-processing. As a result of this study, we believe that many more sensors than IMUs and cameras may benefit from event-based sampling, opening the door to exciting new applications.

ACKNOWLEDGEMENTS

We gratefully acknowledge support by the following grants: NSF FRR 2220868, NSF IIS-RI 2212433, ONR N00014-22-1-2677 and SNF 225354.

REFERENCES

- [1] Omri Asraf, Firas Shama, and Itzik Klein. Pdrnet: A deep-learning pedestrian dead reckoning framework. *IEEE Sensors Journal*, 22(6):4932–4939, 2022. doi: 10.1109/JSEN.2021.3066840.
- [2] Justin Baker, Shih-Hsin Wang, Tommaso de Fernex, and Bao Wang. An explicit frame construction for normalizing 3d point clouds. In *Forty-first International Conference on Machine Learning*, 2024.
- [3] Martin Brossard, Axel Barrau, and Silvère Bonnabel. Aiimu dead-reckoning. *IEEE Transactions on Intelligent Vehicles*, 5(4):585–595, 2020. doi: 10.1109/TIV.2020.2980758.
- [4] Martin Brossard, Silvère Bonnabel, and Axel Barrau. Denoising imu gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robotics and Automation Letters*, 5(3):4796–4803, 2020. doi: 10.1109/LRA.2020.3003256.
- [5] Russell Buchanan, Varun Agrawal, Marco Camurri, Frank Dellaert, and Maurice Fallon. Deep imu bias inference for robust visual-inertial odometry with factor graphs. *IEEE Robotics and Automation Letters*, 8(1): 41–48, 2023. doi: 10.1109/LRA.2022.3222956.
- [6] Xiya Cao, Caifa Zhou, Dandan Zeng, and Yongliang Wang. Rio: Rotation-equivariance supervised learning of robust inertial odometry. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 6614–6623, 2022.
- [7] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [8] Changhao Chen, Peijun Zhao, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham, and Niki Trigoni. Oxiod: The dataset for deep inertial odometry. In *arXiv*, 2018.
- [9] Danpeng Chen, Nan Wang, Runsen Xu, Weijian Xie, Hujun Bao, and Guofeng Zhang. Rnin-vio: Robust neural inertial navigation aided visual-inertial odometry in challenging scenes. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 275–283. IEEE, 2021.
- [10] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.
- [11] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023.
- [12] Benedek Forrai, Takahiro Miki, Daniel Gehrig, Marco Hutter, and Davide Scaramuzza. Event-based agile object catching with a quadrupedal robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, June 2023.
- [13] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Trans. Robot.*, 33(1): 1–21, 2017. doi: 10.1109/TRO.2016.2597321.
- [14] Niklas Funk, Erik Helmut, Georgia Chalvatzaki, Roberto Calandra, and Jan Peters. Evetac: An event-based optical tactile sensor for robotic manipulation. *IEEE Trans. Robot.*, 40:3812–3832, 2024. doi: 10.1109/TRO.2024.3428430.
- [15] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. doi: 10.1109/TPAMI.2020.3008413.
- [16] Daniel Gehrig and Davide Scaramuzza. Low latency automotive vision with event cameras. *Nature*, 2024.
- [17] Patrick Geneva, Kevin Eickenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. OpenVINS: A research platform for visual-inertial estimation. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, Paris, France, 2020.
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 770–778, 2016. doi: 10.1109/cvpr.2016.90.
- [19] Sachini Herath, Hang Yan, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3146–3152. IEEE, 2020.
- [20] Sachini Herath, Hang Yan, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3146–3152. IEEE, 2020.
- [21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.
- [22] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.
- [23] Royina Karegoudra Jayanth, Yinshuang Xu, Ziyun Wang, Evangelos Chatzipantazis, Daniel Gehrig, and Kostas Daniilidis. Eqnio: Subequivariant neural inertial odometry. *arXiv preprint arXiv:2408.06321*, 2024.
- [24] Xin Jin, Yang Shi, Yang Tang, and Xiaotai Wu. Event-triggered attitude consensus with absolute and relative attitude measurements. *Automatica*, 122:109245, 2020. ISSN 0005-1098. doi: <https://doi.org/10.1016/j.automatica.2020.109245>. URL <https://www.sciencedirect.com/science/article/pii/S000510982030443X>.
- [25] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance

- with learned canonicalization functions. In *International Conference on Machine Learning*, pages 15546–15566. PMLR, 2023.
- [26] Diederik P. Kingma and Jimmy L. Ba. Adam: A method for stochastic optimization. *Int. Conf. Learn. Representations (ICLR)*, 2015.
- [27] Colin Lea, René Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified approach to action segmentation. In Gang Hua and Herve Jegou, editors, *Eur. Conf. Comput. Vis. (ECCV)*, pages 47–54. Springer Verlag, 2016.
- [28] Feiran Li, Kent Fujiwara, Fumio Okura, and Yasuyuki Matsushita. A closer look at rotation-invariant deep point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16218–16227, 2021.
- [29] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 dB 15 μ s latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits*, 43(2):566–576, 2008. doi: 10.1109/JSSC.2007.914337.
- [30] Shih-Chii Liu, André van Schaik, Bradley A. Minch, and Tobi Delbruck. Asynchronous binaural spatial audition sensor with $2 \times 64 \times 4$ channel output. *IEEE Trans. Biomed. Circuits Syst.*, 8(4):453–464, 2014. doi: 10.1109/TBCAS.2013.2281834.
- [31] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020.
- [32] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.
- [33] Zhaoyang Lv, Nickolas Charron, Pierre Moulon, Alexander Gamino, Cheng Peng, Chris Sweeney, Edward Miller, Huixuan Tang, Jeff Meissner, Jing Dong, et al. Aria everyday activities dataset. *arXiv preprint arXiv:2402.13349*, 2024.
- [34] Arnab Kumar Mondal, Siba Smarak Panigrahi, Oumar Kaba, Sai Rajeswar Mudumba, and Siamak Ravanbakhsh. Equivariant adaptation of large pretrained models. *Advances in Neural Information Processing Systems*, 36:50293–50309, 2023.
- [35] S.M. Mostafavi I., Lin Wang, Yo-Sung Ho, and Kuk-Jin Yoon Yoon. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.
- [36] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3565–3572, 2007.
- [37] Manasi Muglikar, Diederik Paul Moeys, and Davide Scaramuzza. Event guided depth sensing. In *2021 International Conference on 3D Vision (3DV)*, pages 385–393, 2021. doi: 10.1109/3DV53792.2021.00048.
- [38] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. An asynchronous time-based image sensor. In *IEEE Int. Symp. Circuits Syst. (ISCAS)*, pages 2130–2133, 2008. doi: 10.1109/ISCAS.2008.4541871.
- [39] Jan Steinbrener, Christian Brommer, Thomas Jantos, Alessandro Fornasier, and Stephan 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)*, pages 6084–6090, 2022. doi: 10.1109/ICRA46639.2022.9811989.
- [40] Scott Sun, Dennis Melamed, and Kris Kitani. Idol: Inertial deep orientation-estimation and localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6128–6137, 2021.
- [41] Varun Sundar, Matt Dutson, Andrei Ardelean, Claudio Bruschini, Edoardo Charbon, and Mohit Gupta. Generalized event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2024.
- [42] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4), 1991.
- [43] Hang Yan, Qi Shan, and Yasutaka Furukawa. Ridi: Robust imu double integration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [44] Cem Yüceer and Kemal Oflazer. A rotation, scaling, and translation invariant pattern classification system. *Pattern recognition*, 26(5):687–710, 1993.