INERTIALBRIDGE: EFFICIENT STRUCTURE-AWARE CONDITIONAL TRANSPORT FOR INERTIAL KINEMATICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Dense inertial motion estimation requires accurate per-sample predictions at a manageable computational cost. We present **InertialBridge** (IB), a conditional bridge-matching (BM) framework that predicts 100 samples of 3D velocity $\vec{v}$ and angular rate $\vec{\omega}$ after observing a complete one-second IMU window, using a single network evaluation. Conditioning the drift on the source window makes every interior bridge target algebraically identifiable. A boundary-weighted scheduler $\pi_\rho = \rho\,\delta_0 + (1-\rho)\,\mathcal{U}[0,1]$ allocates training mass to the deployed query at $t = 0$. Our population analysis establishes one-step exactness of the optimal field and characterizes the boundary drift that determines its prediction. A matched ten-epoch, single-seed sweep yields 2.7–3.9 times higher position ATE under uniform scheduling than under the evaluated positive-boundary-mass schedules. IB parameterizes the drift with a Mamba3 state-space network conditioned on a frozen variational autoencoder (VAE), and uses a structured kinematic loss. Under our common RoNIN protocol, IB achieves the lowest mean position ATE and RTE among the evaluated methods on both subject splits; paired ATE tests do not detect a significant difference from EqNIO. IB uses $1.95\,\mathrm{M}$ parameters and $0.87\,\mathrm{GFLOPs}$ per window, compared with $5.03\,\mathrm{M}$ and $0.84\,\mathrm{GFLOPs}$ for EqNIO, while returning an estimate at every input sample. These results support efficient dense reconstruction of observed IMU windows.

1 INTRODUCTION

Inertial Measurement Units (IMUs) are foundational to navigation, robotics, and control, providing continuous high-frequency linear acceleration $\tilde{\vec{a}}$ and angular rate $\tilde{\vec{\omega}}$. However, ubiquitous Micro-Electro-Mechanical Systems (MEMS) IMUs suffer from severe noise and bias instability, which degrades kinematic estimation and makes reliable low-cost tracking difficult.

Conventional motion estimation relies on kinematic models coupled with Extended Kalman Filters (EKFs) (Li et al., 2023), which demand parameter tuning and assumptions about sensor placement. Learned estimators such as IONet (Chen et al., 2018a), RoNIN (Herath et al., 2020), the network of TLIO (Liu et al., 2020), and the equivariant EqNIO (Jayanth et al., 2025b) have advanced inertial odometry, but they regress a single velocity or displacement per window. With the 100-sample windows at 100 Hz used in our protocol, these networks return one estimate per second, so the motion within a window is not resolved at the output.

Dense estimators predict kinematics at every input sample. In our RoNIN evaluation (Table 1), the two strongest baselines sit at opposite ends of a trade-off. EqNIO, the recent State-Of-The-Art (SOTA) in learned inertial odometry, makes displacement regression robust to device heading through a canonical frame and subequivariance (Jayanth et al., 2025b); it is the most accurate baseline in our evaluation but returns one displacement per window, with $4.7\times$ the FLOPs of the equally sized TLIO network. DiffusionIMU denoises the inertial signal into a dense velocity sequence (Teng et al., 2025); it keeps the input resolution, but with over $200\times$ the FLOPs of the EqNIO across its 100 inference steps and the highest latency of all models, yet remains less accurate than EqNIO. Both buy accuracy with computation, and moving from DiffusionIMU to EqNIO also trades temporal resolution for accuracy. Flow Matching (FM) and Bridge Matching (BM) learn vector fields along straight paths (Lipman et al., 2023; Shi et al., 2023) and can produce dense outputs in few steps, but on paired data a drift that sees only the current state is ill-posed: where the paths of two pairs cross, it

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

must average their displacements and can transport a window to another window's target (Liu et al., 2023b; De Bortoli et al., 2023).

We propose InertialBridge (IB), a VAE-based Conditional BM (CBM) framework for dense reconstruction of observed IMU windows. After receiving a complete one-second window, IB predicts velocity $\vec{v}$ and angular-rate $\vec{\omega}$ at all 100 input samples in one network evaluation. Source conditioning makes every interior bridge target algebraically identifiable, separating interior reconstruction from prediction at the source boundary $t = 0$. The boundary-weighted scheduler π_ρ directly supervises this deployed query. Our population analysis establishes exact integration of the optimal field by a single Euler step and identifies the boundary drift as the determinant of the prediction. In a matched ten-epoch, single-seed sweep, positive boundary mass reduces one-step ATE relative to uniform scheduling. Under our common protocol, IB achieves the lowest mean position ATE and RTE among the evaluated methods on both RoNIN splits, with $2.6\times$ fewer parameters than EqNIO at similar FLOPs (Table 1); paired tests do not detect a significant ATE difference from EqNIO. The resulting formulation combines explicit supervision of the deployed predictor with dense full-window output.

The primary contributions of this work are as follows:

- A source-augmented bridge-matching objective with a boundary-weighted time scheduler π_ρ, and its population analysis for paired sensor-to-kinematics estimation: the source identifies every interior target, the optimum is exact under one Euler step (indeed any consistent explicit Runge–Kutta scheme), the deployed prediction is set by the boundary drift alone, which the uniform scheduler of BM leaves unsupervised (Propositions 1–3).
- InertialBridge, which parameterizes the drift with a Mamba3 state-space network, conditions it on a frozen-VAE latent, and trains it with a structured velocity and angular-rate loss.
- An evaluation on RIDI, RoNIN, OxIOD, and TLIO separating window-level from dense estimators, with per-window cost accounting, paired per-sequence tests, component ablations on both RoNIN splits, and a matched scheduler sweep characterizing the effect of boundary supervision (Table 5).

2 Related Work

2.1 Model-Based and Learned Inertial Odometry

Classical inertial tracking combines strapdown integration with EKF/UKF updates and domain-specific constraints such as ZUPT (Foxlin, 2005; Wang et al., 2015; Li et al., 2023). Learned estimators instead regress motion from IMU windows, including IONet (Chen et al., 2018a), RIDI (Yan et al., 2018), and RoNIN (Herath et al., 2020). Hybrid systems such as TLIO (Liu et al., 2020) place learned displacement and uncertainty estimates inside a filter that propagates the raw IMU at its native rate, and AirIMU learns dense corrections and uncertainties of the IMU signal itself (Qiu et al., 2023). These approaches differ in output space, output rate, and uncertainty assumptions, so we compare the learned networks under a common window and rate definition.

2.2 Generalization, Representations, and Efficient Backbones

Recent work addresses robustness by changing the representation or training distribution. AirIO combines learned inertial features with attitude-aware filtering for UAV odometry (Qiu et al., 2025), and explicit continuous bias dynamics offer a complementary correction strategy (Liu et al., 2025). EqNIO uses canonical displacement priors and subequivariance to improve geometric generalization (Jayanth et al., 2025b), and Neural Inertial Odometry from Lie Events encodes event streams through SE(3)-aware preintegration (Jayanth et al., 2025a). Tartan IMU scales inertial pretraining across platforms and uses lightweight adaptation (Zhao et al., 2025). InertialBridge is complementary: it keeps the windowed IMU input and changes the training objective and output resolution.

2.3 Transport Models for Paired Signals

DiffusionIMU adapts diffusion-style conditional denoising to inertial signals and predicts dense velocity (Teng et al., 2025). Flow Matching and Bridge Matching learn continuous-time transport between distributions (Lipman et al., 2023; Shi et al., 2023), and rectified flow straightens such

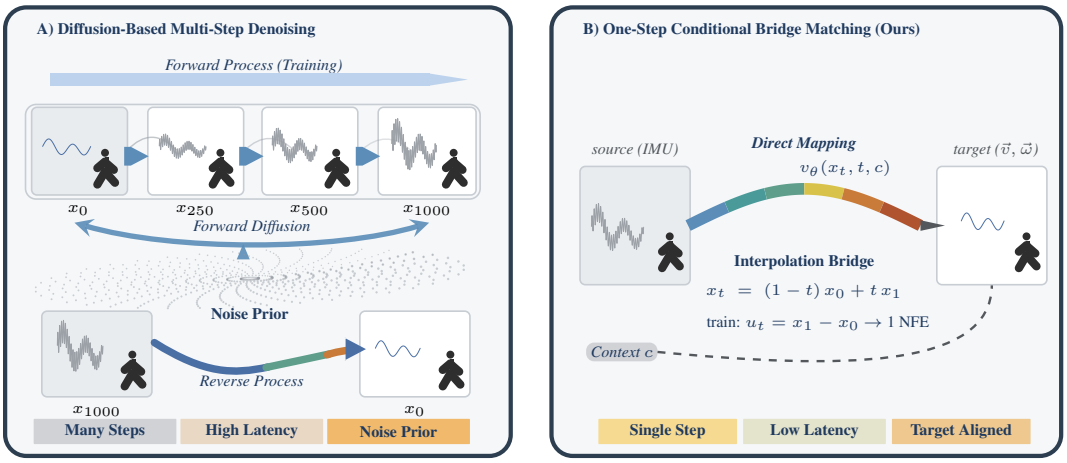

Figure 1: **Conceptual comparison.** (A) Diffusion models denoise a noise prior over many steps. (B) InertialBridge learns the drift v_θ of the bridge $x_t = (1-t)x_0 + tx_1$ from the paired IMU window x_0 to the kinematic target x_1, with $c = (x_0, z_c)$; inference takes one Euler step from $t = 0$.

paths to enable few-step sampling (Liu et al., 2023b). For paired data, I2SB and denoising diffusion bridge models transport a source to its target along stochastic bridges (Liu et al., 2023a; Zhou et al., 2024), InDI restores images by regressing along the straight interpolation between degraded and clean pairs (Delbracio & Milanfar, 2023), and stochastic interpolants admit data-dependent couplings (Albergo et al., 2024). Augmented bridge matching shows that a drift conditioned on the initial point preserves the training coupling (De Bortoli et al., 2023). One-step generation is otherwise obtained by consistency training (Song et al., 2023) or by distilling a trained bridge (Gushchin et al., 2025). Source conditioning (De Bortoli et al., 2023) and regression along straight paired interpolations (Delbracio & Milanfar, 2023) are therefore not new in themselves. IB combines them in the deterministic limit, augmenting the InDI-style input with the source and adding an explicit atom at the source boundary. What this specialization contributes is an exact characterization: every interior target becomes identifiable, one Euler step is exact at the optimum without distillation, and the scheduler parameter ρ controls supervision at the deployed boundary.

3 Methodology

Figure 1 and Figure 2 give the conceptual and structural overview. Given a paired source window x_0 and target kinematic window x_1, IB uses the deterministic bridge $x_t = (1-t)x_0 + tx_1$ and trains on the paired coupling, not on independent marginals. A U-shaped Mamba3 network maps $[x_t; x_0]$ to the drift, conditioned on t and on the posterior mean $z_c \in \mathbb{R}^{128}$ of a frozen VAE; frame conventions are in Appendix A.1.1. At inference, one Euler step from $t = 0$ outputs $(\vec{v}, \vec{\omega})$.

3.1 Augmented Conditional Bridge Matching

Paired bridge. Let $q(x_0, x_1)$ denote the joint distribution, sampled by the training set, of a gravity-aligned IMU window $x_0 = [\tilde{\vec{a}}; \tilde{\vec{\omega}}]$ (gravity retained) and its synchronized kinematic target $x_1 = [\vec{v}; \vec{\omega}]$, both in $\mathbb{R}^{6\times N}$. Each pair is joined by the deterministic linear bridge

$$x_t = (1-t)\,x_0 + t\,x_1, \qquad t \in [0,1], \tag{1}$$

whose conditional velocity is the constant displacement $\dot{x}_t = x_1 - x_0$. The bridge uses the loader numeric arrays in their common $6 \times N$ coordinates with shared per-channel scales across source, target, loss, and evaluation; it defines the training path scheduled by ρ.

Augmenting the drift with the source. Bridge and flow matching train a drift $v(x_t, t)$ that sees only the current state (Lipman et al., 2023; Shi et al., 2023). On a paired bridge this input is

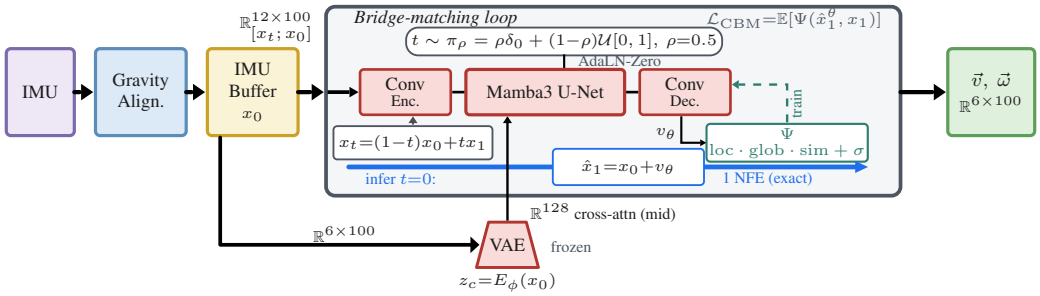

Figure 2: **Architecture of InertialBridge.** One-step mapping of loader-processed IMU data to dense velocity and angular-rate sequences $(\vec{v}, \vec{\omega})$ through a VAE-conditioned Mamba3 backbone.

ambiguous: when the straight paths of two pairs cross, both pairs reach the same x_t but require different displacements $x_1 - x_0$, and no function of (x_t, t) can fit both. For squared error and network input c, the loss splits into a part that training can reduce and a part that it cannot:

$$\mathbb{E}\left\|v(c) - (x_1 - x_0)\right\|^2 = \underbrace{\mathbb{E}\left\|v(c) - \mathbb{E}[x_1 - x_0 \mid c]\right\|^2}_{\text{reducible}} + \underbrace{\mathbb{E}\ \mathrm{tr}\,\mathrm{Cov}\left[x_1 - x_0 \mid c\right]}_{\text{irreducible}}. \tag{2}$$

The second term measures how much the correct displacement varies among pairs that present the same input c. With $c = (x_t, t)$ it is positive wherever paths cross: the loss cannot reach zero, the gradients carry label noise, and the learned field averages the crossing pairs, so a window may be transported to another window's target (Liu et al., 2023b). Latent diffusion models condition inpainting and super-resolution by concatenating the encoded observation to the noisy latent channel-wise (Rombach et al., 2022), and SR3 and Palette condition on the source image in the same way (Saharia et al., 2023; 2022). We likewise concatenate x_0 with x_t channel-wise, as denoising diffusion bridge models do with their source endpoint (Zhou et al., 2024). The input then determines the displacement, $x_1 - x_0 = (x_t - x_0)/t$ for every $t > 0$, so the second term of Equation 2 is zero and every interior sample has a single correct answer. The interior problem is thus solvable algebraically from the inputs; the only predictive problem is the one at $t = 0$. At $t = 0$ the two inputs coincide ($x_t = x_0$), so the concatenation does not change the best one-step prediction; it changes how interior samples train the shared network (subsection 3.2). A frozen VAE pretrained to reconstruct IMU windows encodes the source into its posterior mean $z_c = E_\phi(x_0) \in \mathbb{R}^{128}$, which is injected by cross-attention (subsection 3.3); the full drift is $v_\theta(x_t, t, x_0, z_c)$.

Training objective and time scheduler. We parameterize the drift through the terminal state it implies, $\hat{x}_1^\theta = x_t + (1 - t)\, v_\theta(x_t, t, x_0, z_c)$, and minimize

$$\mathcal{L}_{\mathrm{CBM}}(\theta) = \mathbb{E}_{t\sim\pi_\rho}\, \mathbb{E}_{(x_0,x_1)\sim q}\Big[\Psi\big(\hat{x}_1^\theta,\, x_1\big)\Big], \tag{3}$$

where Ψ is the structured loss defined below, ϕ is held fixed, and π_ρ is the time scheduler. FM and BM draw the training time from the uniform scheduler $\mathcal{U}[0, 1]$ (Lipman et al., 2023; Shi et al., 2023), and diffusion models likewise tune the training distribution of noise levels (Karras et al., 2022). Because inference queries the drift only at $t = 0$, we use the boundary-weighted scheduler

$$\pi_\rho = \rho\,\delta_0 + (1 - \rho)\,\mathcal{U}[0, 1], \quad F_\rho(t) = \rho + (1 - \rho)\,t, \quad t = S_\rho(u) = \max\Big(0, \tfrac{u-\rho}{1-\rho}\Big),\ u \sim \mathcal{U}[0, 1], \tag{4}$$

with cumulative distribution F_ρ on $[0, 1]$ and sampler S_ρ for $\rho < 1$. The scheduler places an atom of mass $\rho \in [0, 1]$ at the source boundary and spreads the rest uniformly along the bridge, so the objective splits into the deployed task and an interior term,

$$\mathcal{L}_{\mathrm{CBM}} = \rho\,\mathcal{L}_{\mathrm{bnd}} + (1-\rho)\,\mathcal{L}_{\mathrm{int}}, \quad \mathcal{L}_{\mathrm{bnd}} = \mathbb{E}_q\, \Psi\big(\hat{x}_1^\theta\big|_{t=0}, x_1\big), \quad \mathcal{L}_{\mathrm{int}} = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_q\, \Psi\big(\hat{x}_1^\theta, x_1\big). \tag{5}$$

At $\rho = 0$ it reduces to the uniform scheduler of standard BM. Since $\hat{x}_1^\theta - x_1 = (1-t)\big(v_\theta - (x_1 - x_0)\big)$, for squared error each interior sample is displacement regression reweighted by $(1 - t)^2$. We use $\rho = 0.5$ and analyze this choice in Appendix A.1.11.

Inference. A single Euler step from the source boundary yields

$$\hat{x}_1 = x_0 + v_\theta(x_0, 0, x_0, z_c), \tag{6}$$

i.e., one network evaluation per window. subsection 3.2 shows why one step suffices for this bridge.

Structured loss. For each output $s \in \mathcal{S} = \{\vec{v}, \vec{\omega}\}$, let $P^s, G^s \in \mathbb{R}^{3\times N}$ be the predicted and target sequences. The discrepancy Ψ in Equation 3 combines a frame-wise error, a window-sum error that integration accumulates, and a per-axis temporal cosine that rewards the waveform shape:

$$\mathcal{L}^s_{\text{loc}} = \tfrac{1}{3N}\big\|P^s - G^s\big\|_1, \quad \mathcal{L}^s_{\text{glob}} = \tfrac{1}{3}\Big\|\textstyle\sum_{i=1}^{N}\big(P^s_i - G^s_i\big)\Big\|_1, \quad \mathcal{L}^s_{\text{sim}} = 1 - \tfrac{1}{3}\textstyle\sum_{c=1}^{3}\cos\big(P^s_{c,:}, G^s_{c,:}\big). \tag{7}$$

The terms are balanced by homoscedastic uncertainty weighting (Kendall et al., 2018),

$$\Psi = \mathcal{L}_{\text{total}} = \sum_{s\in\mathcal{S}}\sum_{k\in\{\text{loc,glob,sim}\}}\big[\exp(-\sigma_{s,k})\,\mathcal{L}^s_k + \sigma_{s,k}\big], \tag{8}$$

with log-variances $\sigma_{s,k}$ learned jointly with θ. Every term is minimal when the prediction equals the target (the cosine uses a small constant in its denominator, so all-zero axes are defined), so Ψ meets the assumption of Appendix A.1.10; the propositions treat the learned weights σ as fixed. Implementation details are given in Appendix A.1.4.

3.2 What the Augmented Bridge Learns

We characterize the population minimizer of Equation 3, treating the uncertainty weights in Ψ as fixed and assuming only that $\Psi(y, y') \geq \Psi(y', y')$. Let

$$m_\Psi(x_0) = \arg\min_y \ \mathbb{E}\big[\Psi(y, x_1) \mid x_0\big] \tag{9}$$

denote the Ψ-Bayes estimator of the target given the window; it is $\mathbb{E}[x_1 \mid x_0]$ for squared error and the coordinate-wise conditional median for absolute error, and for the structured loss below it is an M-estimator that we assume to be unique. Proofs are deferred to Appendix A.1.10.

Proposition 1 (Optimal drift)**.** *Over measurable drifts, Equation 3 is minimized by*

$$v^\star(x_t, t, x_0) = \begin{cases} (x_t - x_0)/t, & t \in (0, 1], \\ m_\Psi(x_0) - x_0, & t = 0. \end{cases} \tag{10}$$

Proposition 2 (One-step exactness)**.** *Every line $x_t = x_0 + t\,w$ is an integral curve of $v^\star$ on $(0, 1]$, and the boundary value selects $w = m_\Psi(x_0) - x_0$. Hence every consistent explicit Runge–Kutta scheme on $[0, 1]$ started at x_0 returns $\hat{x}_1 = m_\Psi(x_0)$, for any order and any number of steps.*

For $t > 0$ Proposition 1 is elementary, since the target on the bridge in Equation 1 is identified by construction and gives the interior branch of Equation 10; its content lies in what it implies for one-step deployment. Two consequences explain the empirical behaviour of the model. *(i) Solver invariance.* Additional steps or higher-order solvers cannot improve the optimum; the differences in Appendix A.1.7 measure approximation error of the learned interior field, not discretization error. *(ii) The boundary determines the prediction.* Interior samples identify the family of straight lines through x_0 but not which line is taken; that choice is made entirely by the drift at $t = 0$, which receives training signal only through the atom $\rho\,\delta_0$ of the scheduler. Under the uniform scheduler of BM ($\rho = 0$) the deployed value receives no direct supervision and is reached only through the continuity of the network as $t \to 0$. The source augmentation does not change this boundary optimum: a drift without the source input has the same value $m_\Psi(x_0) - x_0$ at $t = 0$, because there $x_t = x_0$. All one-step variants in Table 2 therefore share one population optimum and differ only in how training reaches it.

Why interior samples can help. The scheduler splits off the interior term $\mathcal{L}_{\text{int}}$ (Equation 5).

Proposition 3 (Consistency of the interior term)**.** *For every $\rho \in (0, 1]$, all minimizers of Equation 3 over a sufficiently rich function class agree at $t = 0$ with $m_\Psi(x_0) - x_0$.*

Unlike a generic auxiliary loss, $\mathcal{L}_{\text{int}}$ therefore introduces no bias in the deployed predictor for an unrestricted function class: the two terms constrain disjoint parts of the input domain, and $v^\star$ minimizes both. A network continuous in t cannot represent the jump of $v^\star$ at $t = 0$, so small-t samples also act on the boundary value; for squared error they pull it toward $\mathbb{E}[x_1 - x_0 \mid x_0]$, which is the boundary optimum, whereas for the structured loss this pull is a finite-capacity effect. With finite capacity, the parameters are shared and $\mathcal{L}_{\text{int}}$ acts on the deployed predictor only through the common representation. It asks the network to reconstruct x_1 from x_0 together with the partial reveal $x_t - x_0 = t\,(x_1 - x_0)$, a task whose supervision is noise-free by Equation 2 and whose difficulty increases continuously as $t \to 0$ until it coincides with the deployed task. We hypothesize that this graded task regularizes the shared representation, and the ρ sweep tests it.

3.3 State-Space Parameterization of the Drift Network

Self-attention over an N-frame window costs $\mathcal{O}(N^2)$, whereas inertial navigation integrates measured rates through a first-order linear recurrence. We therefore parameterize v_θ with Mamba3 (Gu & Dao, 2024; Dao & Gu, 2024; Lahoti et al., 2026), a selective state-space model with linear cost in N. For input features u_k at frame k, each head updates a state h_k as

$$h_k = \alpha_k h_{k-1} + (1 - \lambda_k)\,\Delta_k \alpha_k\, B_{k-1} u_{k-1}^\top + \lambda_k \Delta_k\, B_k u_k^\top, \qquad y_k = h_k^\top C_k + D\,u_k, \tag{11}$$

with decay $\alpha_k = \exp(\Delta_k A_k)$, where the step $\Delta_k > 0$, rate $A_k < 0$, and mixing weight $\lambda_k \in (0, 1)$ are computed from u_k, and B_k, C_k are rotated by data-dependent angles (Lahoti et al., 2026). With $\alpha_k = 1$ and $\lambda_k = \frac{1}{2}$, Equation 11 is the trapezoidal rule for integrating $B_k u_k$. The learned decay $\alpha_k < 1$ lets the state discard accumulated error. Because the scan is causal, alternate blocks process the time-reversed sequence, so the stack combines forward and backward passes over the window. This is a structural motivation; no block is constrained to integrate. The network stacks two down, one middle, and two up blocks with U-Net skip connections.

Conditioning the drift network. The sequence is conditioned on the timestep t and the VAE posterior mean $z_c = E_\phi(x_0)$ through two separate paths.

The timestep path uses adaptive layer normalization with zero-initialized gates (AdaLN-Zero) (Peebles & Xie, 2023): an MLP maps t to scale (γ_m), shift (b_m), and gates (g_m, g_a):

$$\tilde{x}_k = (1 + \gamma_m(t)) \odot \text{LayerNorm}(x_k) + b_m(t), \quad x'_k = x_k + g_m(t) \odot \text{Mamba}(\tilde{x}_k). \tag{12}$$

The z_c path applies cross-attention (Vaswani et al., 2017) after token mixing in the middle block; with one key–value token, the attention weight is one and the layer is a frame-shared map $A(z_c)$:

$$y_k = x'_k + g_a(t) \odot \text{CrossAttn}\big(\text{LayerNorm}(x'_k), z_c\big) = x'_k + g_a(t) \odot A(z_c). \tag{13}$$

All blocks carry the AdaLN-Zero path; only the middle block carries the z_c path.

4 Evaluation

4.1 Experimental Setup & Evaluation Metrics

Datasets. We evaluate on four inertial odometry benchmarks: RIDI (Yan et al., 2018), RoNIN (Herath et al., 2020), OxIOD (Chen et al., 2018b), and TLIO (Liu et al., 2020). Their references come from different pipelines, including Tango and VICON for OxIOD, ARCore for RoNIN, and an MSCKF-based estimate for TLIO (Mourikis & Roumeliotis, 2007), so the supervision error is benchmark-dependent. The main comparison uses the native RoNIN seen and unseen subject splits; the pooled setting over all benchmarks and the exact partitions are given in Appendix A.1.6, and the windowing, source–target construction, and training settings in Appendix A.1.1.

Metrics and baselines. We report ATE and RTE for global and local position accuracy (Sturm et al., 2012; Herath et al., 2020; Liu et al., 2020; Rao et al., 2022; Teng et al., 2025), the trajectory length ratio (TLR) for scale, and the mean cosine similarity (MCS) of step displacements for direction (Appendix A.1.5). Predicted and reference velocities are integrated from a common origin without further alignment; for window-level methods, the displacement of each window is placed at its last

Table 1: Error and per-window inference cost on RoNIN seen (S) and unseen (U) subjects (mean ± std over test sequences). Res. is the output resolution: window-level methods return one displacement per 100-sample window, dense methods one estimate per sample. Orientation, available for IONet and IB only, is near chance level and not marked. Cost covers the full map from the IMU window to the output at batch size 1 (Appendix A.1.2); for DiffusionIMU it is for 100 inference steps, 100× the measured per-step cost as for DDIM in Table 2, and its errors are from one deterministic pass. Baselines are cited in subsection 4.1.

Method	Res.	Pos. ATE (m) ↓		Pos. RTE (m) ↓		Ori. ATE (rad)		Params	FLOPs	Lat.	Mem.
		S	U	S	U	S	U	(M)	(G)	(ms)	(MB)
IONet	1 Hz	6.09±2.39	6.13±2.85	3.01±1.39	4.29±1.90	2.25±0.10	2.27±0.06	1.16	0.40	0.59	55.5
RoNINResNet	1 Hz	6.06±2.35	6.75±4.44	2.98±1.21	4.27±1.93	–	–	15.57	0.72	4.98	95.7
TLIO	1 Hz	4.90±2.35	6.81±5.14	2.87±1.32	4.26±1.99	–	–	5.03	0.18	1.89	55.4
EqNIO (O(2))	1 Hz	3.54±1.54	4.64±2.72	2.72±1.27	3.97±1.96	–	–	5.03	0.84	3.88	55.5
RoNINLSTM	100 Hz	4.02±1.66	5.47±3.87	2.74±1.31	4.22±2.02	–	–	0.21	0.16	0.71	35.4
RoNINTCN	100 Hz	4.58±2.26	6.76±4.83	2.77±1.36	4.34±2.14	–	–	0.14	0.54	1.32	1.2
DiffusionIMU	100 Hz	3.86±1.72	5.90±3.96	2.82±1.16	4.37±1.94	–	–	1.86	196	949	40.6
IB (Ours)	100 Hz	**3.39±1.53**	**4.55±2.43**	**2.36±1.22**	**3.80±2.02**	2.03±0.21	2.16±0.13	1.95	0.87	5.92	41.8

sample, so they are evaluated at the window rate. ± is the standard deviation over test sequences, and every model is trained once. Differences between two models on RoNIN are tested with a two-sided Wilcoxon signed-rank test over the 32 paired test sequences of each split; p_{W} denotes its p-value. This captures variation over test sequences but not over training runs. Baselines are IONet (Chen et al., 2018a), the replicated RoNINResNet, RoNINLSTM, and RoNINTCN (Herath et al., 2020), the TLIO network without its filter (Liu et al., 2020), EqNIO (Jayanth et al., 2025b), and the reimplemented DiffusionIMU (Teng et al., 2025), whose cost is reported for 100 inference steps, matching its 100 training timesteps, and whose errors are from one deterministic pass.

4.2 Comparison with Baselines under a Common Protocol

Table 1 compares position error and per-window cost on RoNIN. EqNIO has the lowest ATE among window-level baselines (3.54 m seen; 4.64 m unseen). IB returns a dense estimate for each sample, with ATE/RTE of 3.39 ± 1.53/2.36 ± 1.22 m on seen subjects and 4.55 ± 2.43/3.80 ± 2.02 m on unseen subjects. Against EqNIO, the paired mean ATE differences of 0.14 m and 0.10 m favor IB (from unrounded per-sequence values), with Wilcoxon $p_{\mathrm{W}} = 0.37$ and 0.66, respectively. IB has lower mean ATE than the other dense methods shown; the strongest of those reach 3.86 m (DiffusionIMU) on seen and 5.47 m (RoNINLSTM) on unseen subjects. Against the evaluated one-pass DiffusionIMU predictor, the paired comparison is significant on unseen subjects ($p_{\mathrm{W}} = 0.006$) but not on seen subjects ($p_{\mathrm{W}} = 0.12$). IB uses 1.95 M parameters and 0.87 GFLOPs per window, versus 5.03 M and 0.84 GFLOPs for EqNIO; its measured latency is 5.92 vs. 3.88 ms. DiffusionIMU's 100-pass cost is extrapolated from single-pass measurements to 196 GFLOPs and 949 ms; its reported errors are measured with one deterministic pass. These quantities describe the evaluated predictor and the projected iterative cost, respectively. Orientation errors on RoNIN remain high for both methods with an orientation output, so this comparison focuses on position; pooled orientation results appear in Table 4.

When every method is trained on the pooled benchmarks (Table 4), no method dominates: IB has the lowest ATE and RTE on OxIOD (V), the second-lowest aggregate RTE, and lower orientation ATE than IONet on every split, well below chance on OxIOD, RIDI, and TLIO (0.47 to 1.08 rad), and ranks third in aggregate ATE, 0.21 m behind EqNIO and 0.02 m behind RoNINResNet. All comparisons hold under the stated split and metric protocol.

4.3 Ablation of Component Contributions

Table 2 changes one component of IB at a time on RoNIN, with every variant trained under the same early-stopping rule. Replacing the structured loss with MSE raises ATE from 3.39 to 5.18 m on seen and from 4.55 to 6.77 m on unseen subjects. Replacing every Mamba3 block by a transformer with the same conditioning paths and a similar parameter count roughly doubles the error (6.75 and 7.61 m) at lower latency (3.36 ms). Replacing z_c by a constant zero token raises ATE on both splits

Table 2: Ablations on RoNIN seen (S) and unseen (U) subjects (mean ± std over test sequences; same early-stopping rule for all). Checkmarks mark active components: VAE latent z_c, source stacking (Stack), boundary atom (Bnd.); flow matching (one step) and DDIM start from $\mathcal{N}(0, I)$. Cost as in Table 1, Params include the frozen VAE encoder (0.11 M), DDIM latency is steps times per-step latency. Bold: lowest position error; orientation ATE (1.81–2.17 rad, near chance) is not marked and orientation RTE is omitted.

Variant	VAE	Stack	Bnd.	Pos. ATE (m)		Pos. RTE (m)		Ori. ATE (rad)		Params	FLOPs	Lat.
				S	U	S	U	S	U	(M)	(G)	(ms)
IB (Ours, $\rho = 0.5$)	✓	✓	✓	3.39±1.53	**4.55±2.43**	2.36±1.22	3.80±2.02	2.03±0.21	2.16±0.13	1.95	0.87	5.92
Components												
MSE loss	✓	✓	✓	5.18±2.07	6.77±3.71	2.94±1.28	4.46±2.14	2.04±0.21	2.14±0.15	1.95	0.87	5.93
Transformer backbone	✓	✓	✓	6.75±3.64	7.61±4.12	4.84±2.86	6.47±3.19	2.03±0.23	2.03±0.19	2.10	0.96	3.36
Zero token for z_c	–	✓	✓	3.80±1.76	5.01±2.82	2.44±1.25	3.79±2.04	1.99±0.21	2.09±0.19	1.84	0.86	5.77
Boundary rate												
$\rho = 0$ (uniform)	✓	–	–	4.10±1.93	5.33±3.03	2.45±1.33	3.86±1.99	1.99±0.25	2.10±0.17	1.95	0.86	5.87
Direct regression	✓	–	–	3.42±1.60	5.53±3.77	2.33±1.28	3.85±1.88	2.07±0.17	2.06±0.17	1.92	0.84	5.85
Learning paradigm												
Flow matching	✓	✓	✓	**3.20±1.46**	5.33±4.44	**2.24±1.27**	**3.71±1.84**	2.04±0.19	2.13±0.17	1.95	0.87	5.92
DDIM, 10 steps	✓	✓	–	4.30±1.92	5.57±2.51	2.52±1.24	**3.71±1.64**	2.05±0.21	2.10±0.14	1.95	8.65	58.9
DDIM, 40 steps	✓	✓	–	5.23±2.10	6.13±2.59	2.61±1.24	3.74±1.64	2.09±0.15	2.13±0.09	1.95	34.6	235.6

(3.80 and 5.01 m), consistent with z_c supplying a window-level summary of the source. Orientation errors of all variants lie between 1.81 and 2.17 rad, so the ablations move position only.

4.4 Time-Scheduler and Learning-Paradigm Analysis

Time scheduler. The boundary-biased bridge ($\rho = 0.5$) directly supervises the deployed input through the boundary atom. The uniform-scheduler row ($\rho = 0$) has higher ATE than IB (4.10 vs. 3.39 m seen; 5.33 vs. 4.55 m unseen) and also disables source stacking. The matched ten-epoch, single-seed sweep in Table 5 isolates the scheduler with source stacking active in every row: uniform scheduling yields 2.7–3.9 times the ATE of the evaluated positive-boundary-mass schedules. This result supports explicit supervision at the deployed query, consistent with consequence (ii) of subsection 3.2. Mixed schedules improve selected ATE values over $\rho = 1$, whereas pure boundary regression gives the lowest RTE and validation error; the short sweep therefore identifies metric-dependent effects of interior training rather than a consistent advantage over regression. Additional Euler steps or higher-order solvers do not reduce the aggregate ATE of the pooled model (Appendix A.1.7), consistent with the population one-step result in consequence (i).

Learning paradigms. With the same backbone, flow matching from a noise prior has lower ATE than IB on seen subjects (3.20 vs. 3.39 m) but higher ATE on unseen ones (5.33 vs. 4.55 m). This reversal suggests possible overfitting to seen subjects and better unseen-subject generalization by IB in this comparison, although the variants differ in more than the objective (Appendix A.1.13). DDIM degrades as steps increase (4.30 m at 10 steps, 5.23 m at 40) at 10 to 40 times the latency of IB. Direct regression without source stacking matches IB on seen subjects (3.42 m) but not on unseen ones (5.53 m).

5 Discussion

Output resolution and latency. The window-level models return one displacement per 100-sample window, whereas IB, the sequence regressors, and DiffusionIMU return one estimate per input sample. Before IB, the dense models in Table 1 were less accurate than the best window-level model; on RoNIN, IB attains window-level position accuracy at the input resolution. This is a property of the output within a window, not of the update cadence: IB is evaluated once per non-overlapping window, its time-reversed blocks and the VAE make the estimate of every sample depend on the whole window, and all 100 estimates become available when the window is complete. The advantage therefore applies where a dense reconstruction of an observed window is the desired product, such as offline trajectory reconstruction or delayed smoothing: IB obtains it in one evaluation, whereas EqNIO

would have to slide its window with stride one, 100 evaluations or 84 GFLOPs (388 ms at batch size 1) per second of data. A causal estimate at every new sample would require stride-one sliding for IB as well (86.5 GFLOPs, 592 ms), which removes the cost advantage; IB has not been trained or evaluated for causal streaming. Because ATE and RTE are computed on integrated trajectories over non-overlapping windows, they also do not reward the within-window structure of a dense estimate.

Ground-truth fidelity. The two ends of the bridge play different roles. IMU noise enters through the source x_0 and is corrected by the drift; reference errors enter through the target x_1 and are learned as signal, so supervision accuracy limits the attainable error. Table 4 is consistent with this: IB has its lowest ATE on OxIOD with VICON reference (0.42 m) and its highest on RoNIN (3.32 m seen, 5.14 m unseen), whose reference comes from phone-based ARCore tracking (Herath et al., 2020), although RoNIN sequences are also longer, and ATE grows with length.

Seen–unseen gap. On RoNIN, the ATE of IB rises from 3.39 m on seen to 4.55 m on unseen subjects (Table 1), a gap similar to EqNIO's (1.15 vs. 1.10 m, from unrounded means) and smaller than DiffusionIMU's (2.04 m); IONet and RoNINResNet have smaller gaps at higher errors. The unseen spread of IB (2.43 m) is the smallest in the table. As RoNIN subject identity is confounded with recording conditions and reference quality, the gap is not attributable to overfitting alone.

Limitations. The evaluation is centered on human-motion datasets and does not establish transfer to wheeled robots, UAVs, or event-based sensors. Every model is trained once; the reported standard deviations and paired tests describe variation among test sequences, not variation across training runs. The paired tests do not establish superiority over EqNIO, and the benefit of interior samples remains untested against a matched regression control. Under pooled multi-benchmark training, IB is not the most accurate method in aggregate position error (Table 4). Orientation on RoNIN is close to chance level for every method, so orientation is supported only in the pooled setting. TLIO and EqNIO are evaluated without their filters. IB is non-causal within a window and has not been evaluated for causal streaming. The deterministic bridge returns a point estimate; predictive uncertainty and downstream filtering remain future work.

6 Conclusion

We introduced InertialBridge (IB), a conditional bridge-matching framework for dense velocity and angular-rate reconstruction from complete IMU windows. Source conditioning makes the interior targets identifiable, while the boundary-weighted scheduler directly supervises the deployed predictor. Our population analysis characterizes this boundary prediction and establishes exact one-step integration of the optimal field. Under the common RoNIN protocol, IB achieves the lowest mean position ATE and RTE among the evaluated methods, with $2.6\times$ fewer parameters than EqNIO at similar FLOPs; paired tests do not detect a significant ATE difference from EqNIO. Together, the analysis and experiments support single-evaluation dense reconstruction with explicit boundary supervision. Converged multi-seed comparisons with matched regression and sample-level kinematic evaluation are the next steps toward characterizing the contribution of interior training and the fidelity of within-window predictions.

AI USE STATEMENT

Large language models were used to assist with editing and text refinement. They were not used to generate experimental data, results, or code for the reported experiments. The authors reviewed and verified all AI-assisted text and take full responsibility for the content of the paper.

ETHICS STATEMENT

The study uses previously released inertial-odometry datasets (RIDI, RoNIN, OxIOD, and TLIO) under the terms of their original releases; they are not redistributed, and no new human-participant data were collected. Low-cost inertial odometry may support navigation and activity-aware interfaces without external cameras or GPS, but the same capability can enable non-consensual indoor tracking or behavioral profiling from smartphone and wearable IMUs. We therefore recommend explicit user consent, OS-level sensor permissions, on-device processing where possible, and evaluation for demographic and device-specific performance differences; on-device inference alone does not guarantee privacy.

REPRODUCIBILITY STATEMENT

section 3 and Alg. 1 specify the model and training procedure, Appendix A.1.4 the loss, Appendix A.1.1 and Appendix A.1.6 the data protocol and splits, and Appendix A.1.5 the metrics; proofs of all propositions are given in Appendix A.1.10. Anonymized code and configuration files are provided in the supplementary material.

REFERENCES

Michael S. Albergo, Mark Goldstein, Nicholas M. Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. IONet: Learning to cure the curse of drift in inertial odometry. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018a. doi: 10.1609/aaai.v32i1.12102. URL https://ojs.aaai.org/index.php/AAAI/article/view/12102.

Changhao Chen, Peijun Zhao, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham, and Niki Trigoni. OxIOD: The dataset for deep inertial odometry. *arXiv preprint arXiv:1809.07491*, 2018b. URL https://arxiv.org/abs/1809.07491.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2405.21060.

Valentin De Bortoli, Guan-Horng Liu, Tianrong Chen, Evangelos A. Theodorou, and Weilie Nie. Augmented bridge matching. *arXiv preprint arXiv:2311.06978*, 2023.

Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *Transactions on Machine Learning Research (TMLR)*, 2023.

Eric Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6):38–46, 2005. doi: 10.1109/MCG.2005.140.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

Nikita Gushchin, David Li, Daniil Selikhanovych, Evgeny Burnaev, Dmitry Baranchuk, and Alexander Korotin. Inverse bridge matching distillation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21471–21496, 2025.

Sachini Herath, Hang Yan, and Yasutaka Furukawa. RoNIN: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 3146–3152. IEEE, 2020. doi: 10.1109/ICRA40945.2020.9196860. URL https://arxiv.org/abs/1905.12853.

Royina Karegoudra Jayanth, Yinshuang Xu, Evangelos Chatzipantazis, Kostas Daniilidis, and Daniel Gehrig. Neural inertial odometry from Lie events. In *Proceedings of Robotics: Science and Systems XXI*, 2025a. doi: 10.15607/RSS.2025.XXI.143. URL https://www.roboticsproceedings.org/rss21/p143.html.

Royina Karegoudra Jayanth, Yinshuang Xu, Ziyun Wang, Evangelos Chatzipantazis, Kostas Daniilidis, and Daniel Gehrig. EqNIO: Subequivariant neural inertial odometry. In *International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=PDMP8ETt16.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 26565–26577, 2022. URL https://arxiv.org/abs/2206.00364.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018. doi: 10.1109/CVPR.2018.00781.

Aakash Lahoti, Kevin Y. Li, Berlin Chen, Caitlin Wang, Aviv Bick, J. Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations (ICLR)*, 2026.

Hongpeng Li, Haichao Liu, Zhen Li, Chenxi Li, Zhaozong Meng, Nan Gao, and Zonghua Zhang. Adaptive threshold-based ZUPT for single IMU-enabled wearable pedestrian localization. *IEEE Internet of Things Journal*, 10(13):11749–11760, 2023. doi: 10.1109/JIOT.2023.3243296.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Ben Liu, Tzu-Yuan Lin, Wei Zhang, and Maani Ghaffari. Debiasing 6-DOF IMU via hierarchical learning of continuous bias dynamics. In *Proceedings of Robotics: Science and Systems XXI*, 2025. doi: 10.15607/RSS.2025.XXI.009. URL https://www.roboticsproceedings.org/rss21/p009.html.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I^2SB: Image-to-image Schrödinger bridge. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22042–22062, 2023a.

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. doi: 10.1109/LRA.2020.3007421. URL https://ieeexplore.ieee.org/document/9134860/.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023b.

Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE international conference on robotics and automation*, pp. 3565–3572. IEEE, 2007. doi: 10.1109/ROBOT.2007.364024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Yuheng Qiu, Chen Wang, Can Xu, Yutian Chen, Xunfei Zhou, Youjie Xia, and Sebastian Scherer. AirIMU: Learning uncertainty propagation for inertial odometry. *arXiv preprint arXiv:2310.04874*, 2023. URL https://arxiv.org/abs/2310.04874.

Yuheng Qiu, Can Xu, Yutian Chen, Shibo Zhao, Junyi Geng, and Sebastian Scherer. AirIO: Learning inertial odometry with enhanced IMU feature observability. *IEEE Robotics and Automation Letters*, 10(9):9368–9375, 2025. doi: 10.1109/LRA.2025.3581130.

Bingbing Rao, Ehsan Kazemi, Yifan Ding, Devu M Shila, Frank M Tucker, and Liqiang Wang. CTIN: Robust contextual transformer network for inertial navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 5413–5421, 2022. doi: 10.1609/aaai.v36i5.20479. URL https://ojs.aaai.org/index.php/AAAI/article/view/20479.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022. doi: 10.1109/CVPR52688.2022.01042.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, 2022. doi: 10.1145/3528233.3530757.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023. doi: 10.1109/TPAMI.2022.3204461.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger bridge matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 62183–62223, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/c428adf74782c2092d254329b6b02482-Abstract-Conference.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2011.13456.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252, 2023.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, 2012. doi: 10.1109/IROS.2012.6385773.

Xiaoqiang Teng, Chenyang Li, Shibiao Xu, Zhihao Hao, Deke Guo, Jingyuan Li, Haisheng Li, Weiliang Meng, and Xiaopeng Zhang. DiffusionIMU: Diffusion-based inertial navigation with iterative motion refinement. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 8787–8795, 2025. doi: 10.24963/ijcai.2025/977. URL https://www.ijcai.org/proceedings/2025/977.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Zhelong Wang, Hongyu Zhao, Sen Qiu, and Qin Gao. Stance-phase detection for ZUPT-aided foot-mounted pedestrian navigation system. *IEEE/ASME Transactions On Mechatronics*, 20(6): 3170–3181, 2015. doi: 10.1109/TMECH.2015.2430357.

Hang Yan, Qi Shan, and Yasutaka Furukawa. RIDI: Robust IMU double integration. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 621–636, 2018. doi: 10.1007/978-3-030-01261-8_38. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Hang_Yan_RIDI_Robust_IMU_ECCV_2018_paper.html.

Shibo Zhao, Sifan Zhou, Raphael Blanchard, Yuheng Qiu, Wenshan Wang, and Sebastian Scherer. Tartan IMU: A light foundation model for inertial positioning in robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22520–22529, 2025. doi: 10.1109/CVPR52734.2025.02097. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhao_Tartan_IMU_A_Light_Foundation_Model_for_Inertial_Positioning_in_CVPR_2025_paper.html.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *International Conference on Learning Representations (ICLR)*, 2024.

A Appendix

A.1 Additional Experimental Details

A.1.1 Data and Training Protocol

All benchmarks are resampled to 100 Hz and cut into 100-sample windows with stride 33 for training and validation and 100 for testing; shorter windows are zero-padded. Each window pairs the source $x_0 = [\tilde{\vec{a}}; \tilde{\vec{\omega}}]$, with gravity retained, with the target $x_1 = [\vec{v}; \vec{\omega}]$ obtained by finite differences of reference positions and loader quaternions; frames follow each dataset's convention (Appendix A.1.6). Training uses batch size 128, $\rho = 0.5$, and a random planar (XY) rotation applied jointly to source and target; inference is one Euler step. Remaining hyperparameters are listed in the supplementary material.

A.1.2 RoNIN Orientation and Cost Protocol

IB has lower orientation ATE than IONet on RoNIN seen (2.03 vs. 2.25 rad) and unseen subjects (2.16 vs. 2.27 rad; Table 1). Given the orientation-alignment limitations documented for RoNIN (Herath et al., 2020), these differences should be interpreted cautiously when assessing IB's relative effectiveness. Cost is measured per 100-sample window at batch size 1 on one NVIDIA RTX 3090 with locked clocks (core 1800 MHz, memory 9751 MHz). Latency is the mean of 1000 synchronized CUDA forward passes after 50 warm-up passes and covers the full map from the IMU window to the model output, including input canonicalization for EqNIO and the frozen VAE encoder for IB; for DiffusionIMU, trained over 100 diffusion timesteps, FLOPs and latency are 100 times the measured single-pass values; peak memory is the allocator peak of the timed section with only the benchmarked model loaded.

Paired position comparison. For each RoNIN split and baseline B, pair sequences by identifier and let $d_j = \mathrm{ATE}_{\mathrm{IB},j} - \mathrm{ATE}_{B,j}$ for the 32 shared test sequences, using the per-sequence position ATE of Appendix A.1.5. Rank the nonzero $|d_j|$ (average tied ranks R_j); the Wilcoxon signed-rank statistic is

$$W^+ = \sum_{j:d_j>0} R_j, \qquad W^- = \sum_{j:d_j<0} R_j. \tag{14}$$

The reported p_{W} is the two-sided Wilcoxon p-value for differences symmetric about zero, computed separately for each split and baseline. These comparisons use sequences, not independent training runs; no multiple-comparison adjustment is applied.

A.1.3 Training Procedure

Algorithm 1 Augmented conditional bridge matching: one training iteration

Input: paired windows $(x_0, x_1) \sim q$ with $x_0 \in [\tilde{\vec{a}}, \tilde{\vec{\omega}}]$ and $x_1 \in [\vec{v}, \vec{\omega}]$; boundary rate ρ; frozen pretrained VAE encoder E_ϕ; EMA decay β.
Initialize: drift network v_θ, log-variances $\sigma = \{\sigma_{s,k}\}$ of Equation 8, EMA weights $\bar{\theta} \leftarrow \theta$.
for each training iteration **do**
  Sample a minibatch $(x_0, x_1) \sim q$ and rotate x_0 and x_1 by the same random planar (XY) rotation.
  $z_c \leftarrow E_\phi(x_0)$ {posterior mean, no gradient}
  Sample $t \sim \mathcal{U}[0,1]$ and set $t \leftarrow 0$ with probability ρ $\{t \sim \pi_\rho\}$
  $x_t \leftarrow (1-t)\, x_0 + t\, x_1$
  $\hat{x}_1^\theta \leftarrow x_t + (1-t)\, v_\theta\big([x_t; x_0], t, z_c\big)$ {$[\cdot;\cdot]$: channel concatenation}
  Take a gradient step on (θ, σ) to decrease $\mathcal{L}_{\text{total}}\big(\hat{x}_1^\theta, x_1\big)$ {Equation 3, Equation 8}
  $\bar{\theta} \leftarrow \beta\, \bar{\theta} + (1-\beta)\, \theta$
end for
Output: EMA weights $\bar{\theta}$, deployed through Equation 6: $\hat{x}_1 = x_0 + v_{\bar{\theta}}\big([x_0; x_0], 0, z_c\big)$.

Alg. 1 lists one training iteration. Each iteration draws paired windows, places them on the straight bridge at $t \sim \pi_\rho$, and regresses the terminal state implied by the drift onto x_1 (Equation 3). The VAE encoder is frozen and supplies its posterior mean, so the conditioning is deterministic. The log-variances in Equation 8 are optimized jointly with θ, and validation and testing use the EMA weights.

A.1.4 Loss Implementation

Each ℓ_1 term in Equation 7 is implemented as a Huber loss on residuals rescaled by $1/\kappa$ with threshold $\delta = \kappa$, where $\kappa = 10^{-3}$ for $\vec{v}$ and $\kappa = 10^{-3}\pi/180$ for $\vec{\omega}$. Since $H_\kappa(r/\kappa) = r^2/(2\kappa^2)$ for $|r| \le \kappa^2$ and $H_\kappa(r/\kappa) = |r| - \kappa^2/2$ otherwise, the loss is quadratic only for residuals below κ^2 and otherwise equals the ℓ_1 residual up to an additive constant; the cosine term is passed through the same function with $\kappa = 10^{-3}$. Constant factors do not change the minimizer in θ: multiplying $\mathcal{L}_k^s$ by $c > 0$ shifts the optimal $\sigma_{s,k}$ by $\log c$ and leaves the remaining objective unchanged up to a constant. The window sum is therefore computed without the factor Δt that would convert it into an integrated displacement or rotation. The cosine in $\mathcal{L}_{\text{sim}}^s$ is taken between the length-N time series of each axis, so it is invariant to the scale of that axis and complements the two ℓ_1 terms. Explicitly, with $r_{c,i}^s = P_{c,i}^s - G_{c,i}^s$ and $H_\delta(u) = \frac{1}{2}u^2$ for $|u| \le \delta$ and $\delta(|u| - \delta/2)$ otherwise, the implemented terms replacing those in Equation 7 are

$$\widetilde{\mathcal{L}}_{\text{loc}}^s = \frac{1}{3N}\sum_{c,i} H_{\kappa_s}(r_{c,i}^s/\kappa_s), \qquad \widetilde{\mathcal{L}}_{\text{glob}}^s = \frac{1}{3}\sum_{c} H_{\kappa_s}\left(\sum_i r_{c,i}^s/\kappa_s\right),$$
$$\widetilde{\mathcal{L}}_{\text{sim}}^s = H_{10^{-3}}\left(\overline{\mathcal{L}}_{\text{sim}}^s/10^{-3}\right), \qquad \Psi_{\text{impl}} = \sum_{s,k}\left[e^{-\sigma_{s,k}}\widetilde{\mathcal{L}}_k^s + \sigma_{s,k}\right]. \tag{15}$$

Here $s \in \{\vec{v}, \vec{\omega}\}$ and $k \in \{\text{loc}, \text{glob}, \text{sim}\}$; $\overline{\mathcal{L}}_{\text{sim}}^s$ is the batch mean of the per-window cosine discrepancy in Equation 7. Local and global terms are batch-averaged after Huber, while the cosine term is batch-averaged before it. Training minimizes the expectation of this batch-level objective over sampled minibatches and bridge times. The population propositions analyze the per-example objective in Equation 3; the implemented cosine term introduces batch coupling through its aggregation before the nonlinear Huber transform.

A.1.5 Evaluation Metrics

To assess trajectory fidelity, we employ Absolute Trajectory Error (ATE) and Relative Trajectory Error (RTE) (Sturm et al., 2012), which are widely adopted in recent literature (Herath et al., 2020; Liu et al., 2020; Rao et al., 2022; Teng et al., 2025). Furthermore, we introduce Trajectory Length Ratio (TLR) and Mean Cosine Similarity (MCS) to quantify scale drift and fine-grained structural consistency, respectively. Let $\mathbf{P} = \{\vec{P}_i\}_{i=1}^N$ and $\mathbf{G} = \{\vec{G}_i\}_{i=1}^N$ denote the predicted and ground-truth position vectors at time step i, with step-wise displacements defined as $\Delta\vec{P}_i = \vec{P}_{i+1} - \vec{P}_i$ and $\Delta\vec{G}_i = \vec{G}_{i+1} - \vec{G}_i$.

Absolute Trajectory Error (ATE):

Evaluates global consistency by the root-mean-square error between predicted and ground-truth positions, both integrated from the same origin without further alignment and averaged over the three axes, i.e. a per-axis RMSE that is $1/\sqrt{3}$ of the Euclidean-norm ATE; the same evaluator is used for every method:

$$\text{ATE} = \sqrt{\frac{1}{3N}\sum_{i=1}^{N}\left\|\vec{P}_i - \vec{G}_i\right\|_2^2} \tag{16}$$

Relative Trajectory Error (RTE_Δ):

Measures local drift by comparing relative displacements over a horizon Δ, capturing the accumulation of error over shorter intervals. The evaluator sets $\Delta = \min(60 \cdot f_s,\, N-1)$, i.e. a 60-second window at the sequence sampling rate f_s (the full trajectory when shorter

than 60 s):

$$\mathrm{RTE}_{\Delta} = \sqrt{\frac{1}{3(N-\Delta)} \sum_{i=1}^{N-\Delta} \left\| (\vec{P}_{i+\Delta} - \vec{P}_i) - (\vec{G}_{i+\Delta} - \vec{G}_i) \right\|_2^2} \tag{17}$$

Trajectory Length Ratio (TLR):

Measures macroscopic scale drift by comparing instantaneous path lengths:

$$\mathrm{TLR} = \frac{\sum_{i=1}^{N-1} \left\| \Delta \vec{P}_i \right\|_2}{\sum_{i=1}^{N-1} \left\| \Delta \vec{G}_i \right\|_2} \tag{18}$$

A value of TLR $= 1$ signifies perfect length matching, while TLR < 1 and TLR > 1 indicate path underestimation and overestimation (due to noise), respectively. Because ATE and RTE can be artificially minimized by models that conservatively under-predict movement on bounded or closed-loop trajectories, TLR serves as a critical counterbalance to penalize scale collapse.

Mean Cosine Similarity (MCS):

Assesses frame-by-frame directional alignment by averaging the cosine of the angle between corresponding displacement vectors; it is reported only for dense methods, since window-level methods produce one displacement per window:

$$\mathrm{MCS} = \frac{1}{N-1} \sum_{i=1}^{N-1} \frac{\Delta \vec{P}_i^{\top} \Delta \vec{G}_i}{\max\left(\|\Delta \vec{P}_i\|_2, \epsilon\right) \max\left(\|\Delta \vec{G}_i\|_2, \epsilon\right)} \tag{19}$$

where $\epsilon = 10^{-8}$ keeps zero-length steps defined (they contribute 0) and MCS $\in [-1, 1]$. While ATE evaluates macroscopic drift, MCS is used as a proxy for frame-level directional consistency. It does not by itself establish recovery of raw sensor dynamics or signal denoising.

For orientation, the predicted angular rate is integrated on $SO(3)$ and converted to Euler angles; orientation ATE and RTE apply the same formulas to the Euler-angle sequences, with angles wrapped to $[-\pi, \pi]$ for RTE, and orientation TLR compares the accumulated angle lengths.

A.1.6 Dataset Analysis

Table 3: **Hybrid dataset composition and splits.** Sequences and samples per split, with the motion type (planar 2D or full 3D) and the ground-truth source. Datasets: OxIOD (Chen et al., 2018b), RIDI (Yan et al., 2018), RoNIN (Herath et al., 2020), and TLIO (Liu et al., 2020).

Dataset	Annotation	Motion	Train		Val		Test	
			Seq.	Samples	Seq.	Samples	Seq.	Samples
OxIOD (V)	VICON	2D	138	124,015	22	20,555	6	584
OxIOD (T)	Tango	2D	138	12,173	22	732	6	303
RIDI	Tango	2D	66	39,586	10	5,966	18	2,172
RoNIN-S	ARCore	2D	72	240,895	16	49,365	32	17,713
RoNIN-U	ARCore	2D	–	–	–	–	32	18,217
TLIO	MSCKF	3D	283	550,063	35	67,998	36	11,379
Total	–	–	**697**	**966,732**	**105**	**144,616**	**130**	**50,368**

The composition of the hybrid dataset for InertialBridge is summarized in Table 3. We categorize the sequences by motion type: while OxIOD, RIDI, and RoNIN focus on planar pedestrian movement (2D), the TLIO dataset introduces more complex 3D trajectories. As shown in Table 4, the different motion profiles and ground-truth sources may contribute to estimation variance.

The frame transform follows dataset-specific conventions rather than a single universal body- or world-frame convention. The RIDI and RoNIN loaders rotate sensor vectors with calibrated quaternions, OxIOD relies on its sensor-to-world rotation, and TLIO consumes stored preprocessed arrays directly. The acceleration signal retains gravity rather than undergoing a universal gravity-removal step, and step-based segmentation and auxiliary rotation-noise augmentation are disabled. These distinctions must be respected in any metric recomputation or cross-dataset comparison.

A.1.7 Inference Solver Analysis

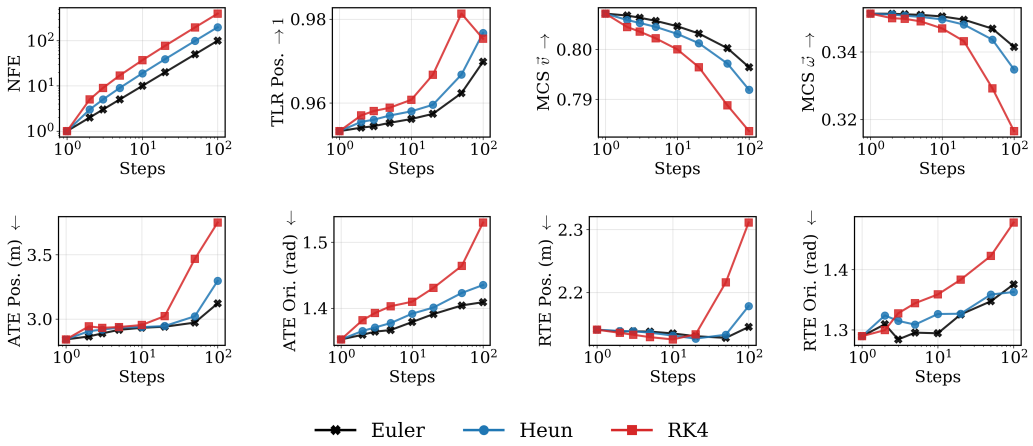

Figure 3: **Solver and step-count sweep** of the pooled IB model of Table 4, aggregated over all test sequences (inference only, same checkpoint). NFE is the number of network evaluations; the cost of one evaluation is given in Table 1. With one step the three solvers coincide and reproduce Table 4.

We conduct a timestep ablation using the kinematic metrics ATE, RTE, and MCS; image-generation inference schedules are not assumed to transfer to this task.

We evaluate three numerical integration schemes for Bridge Matching inference: the first-order Euler method (Lipman et al., 2023), the second-order Heun method (Karras et al., 2022), and the fourth-order Runge-Kutta (RK4) (Song et al., 2021). All three solvers take their last step as an Euler step, so a sweep with T steps from 1 to 1000 with evaluations for Euler, Heun, and RK4, respectively (Figure 3); the cost therefore grows linearly with T.

With one step, the three solvers coincide at an aggregate position ATE of 2.84 m. Additional steps increase ATE for every solver, to 3.12, 3.30, and 3.75 m at $T = 100$ for Euler, Heun, and RK4, and lower the MCS of $\vec{v}$ and $\vec{\omega}$; RTE changes by at most 0.02 m for $T \leq 20$. This matches consequence (i) of subsection 3.2: at the optimum all solvers return the same prediction, and additional steps expose the approximation error of the learned interior field. We therefore use the Euler method with a single step ($T = 1$). Boundary-sampling sensitivity is analyzed in Appendix A.1.11.

A.1.8 Inference Analysis by Datasets

Figure 4 shows how the reported ATE, RTE, TLR, and MCS values change with Euler inference steps across the evaluated datasets. Position ATE at one step equals the entries of Table 4 on every split. One step gives the lowest ATE on RoNIN-S, RoNIN-U, TLIO, and in aggregate; on OxIOD (V), OxIOD (T), and RIDI a few steps lower ATE slightly (to 0.40, 1.44, and 1.49 m, from 0.42, 1.61, and 1.50 m), where the OxIOD (T) test split contains two sequences. At 100 steps ATE is higher than at one step on every split, with the largest increase on RoNIN-U (5.14 to 5.73 m). We therefore use one-step inference, which Proposition 2 shows is exact at the optimum.

A.1.9 Comparative Analysis of Baseline Performance across Datasets

Table 4 reports results on the hybrid setting across six test splits. IB's lowest positional ATE (0.42 ± 0.10 m) and RTE (0.30 ± 0.03 m) occur on OxIOD (V), while RoNIN-U has its highest positional ATE (5.14 ± 3.33 m). The larger variance may reflect the split's device and annotation heterogeneity.

In aggregate, EqNIO has the lowest positional ATE (2.63 ± 1.97 m) and RoNINResNet the lowest RTE (2.11 ± 1.83 m); IB follows with ATE 2.84 ± 2.54 m and RTE 2.14 ± 1.95 m, third and second, respectively, so no single method dominates position under this pooled protocol. For orientation, IB has lower ATE than IONet on every split (1.35 vs. 1.98 in aggregate). Both orientation-output

Table 4: **Position and orientation performance on the hybrid setting.** All methods are trained on the union of RIDI, RoNIN, OxIOD (VICON and Tango), and TLIO and tested on each split and on RoNIN-U. Entries are mean ± std over test sequences; best per column and block in **bold**; – marks a metric the method does not output. Orientation is reported for IONet and IB, the two methods with an orientation output; orientation ATE and RTE are in radians. Values use a different protocol from Table 1 and should not be compared across tables.

Method	OxIOD (V)	OxIOD (T)	RIDI	RoNIN-S	RoNIN-U	TLIO	All
Position ATE (m) ↓							
IONet	0.53±0.28	1.80±0.78	1.86±1.09	4.48±2.05	5.32±3.46	1.59±1.36	3.31±2.78
RoNINResNet	1.63±2.01	1.47±0.47	**1.43±0.79**	3.40±1.54	4.93±3.46	**1.34±1.17**	2.82±2.56
RoNINLSTM	2.57±3.70	1.80±0.63	1.80±0.98	3.90±1.55	4.94±2.72	1.66±1.40	3.14±2.37
RoNINTCN	1.74±1.39	3.20±1.01	1.74±0.83	4.57±2.37	6.25±4.25	1.98±1.46	3.73±3.21
TLIO	1.90±2.18	1.98±0.57	1.76±1.00	3.48±1.72	5.29±3.92	1.35±1.23	3.00±2.82
EqNIO (O(2))	0.58±0.35	1.46±0.60	1.75±1.39	**3.16±1.46**	**4.33±2.06**	1.39±1.19	**2.63±1.97**
DiffusionIMU	1.04±0.92	**1.41±0.08**	2.09±0.91	3.70±1.51	5.78±4.64	1.97±1.27	3.38±3.05
IB (Ours)	**0.42±0.10**	1.61±0.85	1.50±0.61	3.32±1.38	5.14±3.33	1.38±1.34	2.84±2.54
Position RTE (m) ↓							
IONet	0.36±0.05	1.35±0.61	1.62±0.65	2.63±1.36	4.06±1.96	1.16±2.02	2.33±2.03
RoNINResNet	0.76±0.77	1.18±0.23	**1.49±0.70**	**2.32±1.22**	3.67±1.66	**1.06±1.93**	**2.11±1.83**
RoNINLSTM	1.21±1.46	1.13±0.12	1.84±0.87	2.65±1.35	4.16±1.85	1.25±2.09	2.45±2.03
RoNINTCN	0.80±0.35	2.28±0.33	1.87±1.03	2.68±1.38	4.02±1.93	1.44±2.04	2.48±1.97
TLIO	0.86±0.92	1.41±0.11	1.70±0.78	2.38±1.28	3.86±1.84	1.09±1.94	2.23±1.92
EqNIO (O(2))	0.31±0.14	1.29±0.44	1.91±1.37	2.38±1.23	**3.63±1.64**	1.19±1.91	2.20±1.84
DiffusionIMU	0.59±0.29	1.98±0.12	1.99±0.95	2.93±1.24	4.34±1.96	1.51±1.95	2.65±1.99
IB (Ours)	**0.30±0.03**	**1.12±0.37**	**1.49±0.62**	2.33±1.29	3.81±1.86	1.08±2.02	2.14±1.95
Position TLR → 1							
IONet	**0.97±0.03**	1.04±0.03	**0.98±0.03**	**0.97±0.03**	0.90±0.07	0.93±0.09	0.94±0.07
RoNINResNet	0.94±0.03	**1.00±0.03**	**0.98±0.02**	0.96±0.02	0.90±0.06	0.90±0.09	0.93±0.07
RoNINLSTM	1.06±0.11	1.02±0.03	1.24±0.10	1.35±0.19	1.36±0.23	1.46±0.24	1.35±0.23
RoNINTCN	1.13±0.11	1.02±0.02	1.22±0.11	1.31±0.17	1.28±0.17	1.34±0.16	1.29±0.17
TLIO	0.94±0.03	1.01±0.03	**0.98±0.03**	0.96±0.03	0.89±0.08	0.90±0.09	0.93±0.07
EqNIO (O(2))	0.96±0.02	**1.00±0.01**	**0.98±0.03**	**0.97±0.03**	0.91±0.06	0.88±0.10	0.93±0.08
DiffusionIMU	0.88±0.03	0.93±0.02	0.91±0.02	0.91±0.03	0.85±0.06	0.87±0.07	0.88±0.06
IB (Ours)	0.96±0.02	1.02±0.03	**0.98±0.03**	**0.97±0.03**	**0.92±0.07**	**0.95±0.07**	**0.95±0.06**
Position MCS ↑							
IONet	–	–	–	–	–	–	–
RoNINResNet	–	–	–	–	–	–	–
RoNINLSTM	0.92±0.20	0.85±0.32	0.68±0.46	0.53±0.58	0.48±0.60	0.42±0.59	0.50±0.59
RoNINTCN	0.88±0.25	0.82±0.35	0.74±0.43	0.57±0.56	0.52±0.59	0.43±0.59	0.53±0.58
TLIO	–	–	–	–	–	–	–
EqNIO (O(2))	–	–	–	–	–	–	–
DiffusionIMU	0.98±0.05	**0.88±0.29**	0.93±0.21	0.75±0.48	0.71±0.50	0.65±0.48	0.73±0.48
IB (Ours)	**0.99±0.05**	**0.88±0.29**	**0.96±0.18**	**0.79±0.44**	**0.76±0.45**	**0.86±0.31**	**0.81±0.41**
Orientation ATE (rad) ↓							
IONet	1.67±0.17	1.91±0.04	1.80±0.15	2.10±0.11	2.12±0.10	1.86±0.37	1.98±0.27
IB (Ours)	**0.81±0.55**	**1.08±0.04**	**0.77±0.43**	**2.06±0.24**	**2.06±0.15**	**0.47±0.40**	**1.35±0.81**
Orientation RTE (rad) ↓							
IONet	1.97±0.66	**1.37±0.18**	1.85±0.56	2.11±0.34	2.10±0.29	1.77±0.43	1.95±0.44
IB (Ours)	**0.62±0.55**	1.45±0.58	**0.81±0.92**	**1.92±0.29**	**2.03±0.32**	**0.38±0.64**	**1.29±0.92**
Orientation TLR → 1							
IONet	2.55±0.24	4.05±1.59	5.59±2.17	2.33±0.89	2.27±0.86	1.36±0.94	2.54±1.79
IB (Ours)	**0.88±0.13**	**0.34±0.07**	**0.98±0.38**	**0.39±0.20**	**0.35±0.10**	**0.99±0.09**	**0.65±0.36**
Orientation MCS ↑							
IONet	–	–	–	–	–	–	–
IB (Ours)	**0.83±0.30**	**0.33±0.57**	**0.94±0.19**	**0.13±0.58**	**0.08±0.59**	**0.99±0.07**	**0.35±0.64**

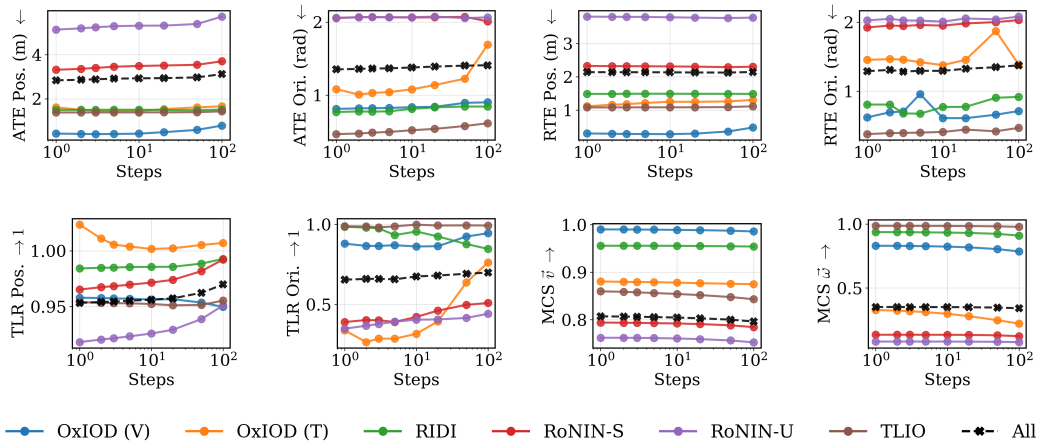

Figure 4: **Per-dataset sensitivity to the number of Euler steps** for the pooled IB model of Table 4 (inference only, same checkpoint as Figure 3); "All" is the aggregate over all test sequences.

methods degrade on RoNIN; IB's orientation ATE is 2.06 rad on both RoNIN splits versus 0.47 rad on TLIO. The RoNIN authors report switching to reference orientations during training when estimated device orientations have end-sequence alignment error above 20° (Herath et al., 2020), highlighting the sensitivity of this benchmark to orientation alignment.

IONet's orientation TLR is far from one on every split, reaching 5.59 ± 2.17 on RIDI, which indicates a scale mismatch of its integrated orientation under this evaluation rather than an architectural effect.

A.1.10 Probabilistic View and Proofs for Section 3.2

Throughout, $(X_0, X_1) \sim q$, $X_t = (1-t)X_0 + tX_1$, and $U = X_1 - X_0$ denotes the bridge displacement. We assume that $\Psi(y, y') \geq \Psi(y', y')$ for all y, y', that the uncertainty weights in Ψ are fixed, and that the minimizer in Equation 9 exists, is unique, and admits a measurable selection.

Markov and augmented drifts. For squared error, the minimizer of $\mathbb{E}\|v(c) - U\|^2$ over measurable v is the conditional mean $\mathbb{E}[U \mid c]$ (Equation 2). With the Markov input $c = (X_t, t)$ this is $v^{\mathrm{M}}(x, t) = \mathbb{E}[U \mid X_t = x]$. Its ODE reproduces the marginal laws of X_t (Lipman et al., 2023), but its endpoint map need not reproduce the training coupling: where straight paths cross, the flow rewires the pairs (Liu et al., 2023b). With the augmented input $c = (X_t, t, X_0)$ the minimizer is $v^{\mathrm{A}}(x, t, x_0) = \mathbb{E}[U \mid X_t = x, X_0 = x_0]$. For $t \in (0, 1]$ the conditional law of X_1 given $(X_t, X_0) = (x, x_0)$ is the Dirac mass at $x_0 + (x - x_0)/t$, so $v^{\mathrm{A}} = (x - x_0)/t$; for $t = 0$ it is $q(x_1 \mid x_0)$. The noise-free augmented bridge therefore returns a point estimate from $q(x_1 \mid x_0)$ rather than a sample from it. Propositions 1–3 make this precise for a general Ψ.

Proof of Proposition 1. Write $c = (x_t, t, x_0)$ and $\hat{y}(c) = x_t + (1-t)\, v(c)$. Because v is an arbitrary measurable function of c, Equation 3 equals $\mathbb{E}_c\big[\mathbb{E}[\Psi(\hat{y}(c), X_1) \mid c]\big]$ and is minimized by minimizing the inner conditional risk separately for each c. For $t \in (0, 1)$, $X_1 = x_0 + (x_t - x_0)/t$ is determined by c, and $v^\star = (x_t - x_0)/t$ gives $\hat{y} = \big(x_t - (1-t)x_0\big)/t = X_1$, which attains the smallest value $\Psi(X_1, X_1)$. For $t = 1$, $\hat{y} = x_t = X_1$ for every v, so $v^\star$ is again a minimizer. For $t = 0$, $c = (x_0, 0, x_0)$, $\hat{y} = x_0 + v$, and the conditional law of X_1 is $q(x_1 \mid x_0)$; by Equation 9 the inner risk $\mathbb{E}[\Psi(x_0 + v, X_1) \mid x_0]$ is minimized by $v = m_\Psi(x_0) - x_0$. □

Proof of Proposition 2. Fix x_0 and, for $w \in \mathbb{R}^{6\times N}$, let $\ell_w(t) = x_0 + t\, w$. For $t \in (0, 1]$, $v^\star(\ell_w(t), t, x_0) = (\ell_w(t) - x_0)/t = w = \dot{\ell}_w(t)$, so every ℓ_w is an integral curve on $(0, 1]$: the field is singular at $t = 0$, and the interior dynamics alone do not determine w. At the boundary, $v^\star(x_0, 0, x_0) = m_\Psi(x_0) - x_0 =: w^\star$. Consider an explicit Runge–Kutta method with coefficients (a_{ij}, b_i, c_i) satisfying the consistency conditions $\sum_i b_i = 1$ and $c_i = \sum_{j<i} a_{ij}$, which hold for Euler,

Heun, and RK4. Suppose the current iterate satisfies $x_n = \ell_{w^\star}(t_n)$. If the stages $k_1, \dots, k_{i-1}$ all equal $w^\star$, stage i is evaluated at the state $x_n + h\sum_{j<i} a_{ij}k_j = \ell_{w^\star}(t_n + c_i h)$ and time $t_n + c_i h$, hence $k_i = w^\star$, by the interior formula if $t_n + c_i h > 0$ and by the boundary value otherwise. By induction all stages equal $w^\star$, and $x_{n+1} = x_n + h\sum_i b_i k_i = \ell_{w^\star}(t_n + h)$. Starting from $x_0 = \ell_{w^\star}(0)$, every step sequence that ends at $t = 1$ therefore returns $\ell_{w^\star}(1) = m_\Psi(x_0)$. □

Proof of Proposition 3. Call a function class $\mathcal{F}$ sufficiently rich if it contains $v^\star$ and is closed under gluing along $t = 0$: for $v_1, v_2 \in \mathcal{F}$, the function equal to v_1 on $\{t = 0\}$ and to v_2 on $\{t > 0\}$ belongs to $\mathcal{F}$; the class of all measurable drifts is an example. Under π_ρ, $\mathcal{L}_{\mathrm{bnd}}$ depends only on the restriction of v to $\{t = 0\}$ and $\mathcal{L}_{\mathrm{int}}$ only on its restriction to $\{t > 0\}$, because the uniform component assigns zero mass to $t = 0$. Let $\hat{v} \in \mathcal{F}$ minimize $\mathcal{L}_{\mathrm{CBM}} = \rho\,\mathcal{L}_{\mathrm{bnd}} + (1-\rho)\,\mathcal{L}_{\mathrm{int}}$ with $\rho \in (0, 1]$, and let $\tilde{v} \in \mathcal{F}$ glue $v^\star$ on $\{t = 0\}$ to $\hat{v}$ on $\{t > 0\}$. Then $\mathcal{L}_{\mathrm{int}}(\tilde{v}) = \mathcal{L}_{\mathrm{int}}(\hat{v})$ and, by Proposition 1, $\mathcal{L}_{\mathrm{bnd}}(\tilde{v}) \le \mathcal{L}_{\mathrm{bnd}}(\hat{v})$. Optimality of $\hat{v}$ and $\rho > 0$ force $\mathcal{L}_{\mathrm{bnd}}(\hat{v}) = \mathcal{L}_{\mathrm{bnd}}(v^\star)$, so $x_0 + \hat{v}(x_0, 0, x_0)$ minimizes the conditional risk in Equation 9 for q-almost every x_0 and, by uniqueness, equals $m_\Psi(x_0)$. For $\rho = 0$ the boundary restriction does not enter the objective and is left undetermined. □

Stochastic extension. With a Brownian bridge $X_t = (1-t)X_0 + tX_1 + \sigma\sqrt{t(1-t)}\,\varepsilon, \varepsilon \sim \mathcal{N}(0, I)$, the target is no longer determined by (X_t, X_0) for $t \in (0, 1)$, and a bridge model that takes the source endpoint as input samples from $q(x_1 \mid x_0)$ (Zhou et al., 2024). This variant would provide predictive samples and uncertainty at the cost of multi-step inference; the limit $\sigma \to 0$ recovers the estimator studied here. We leave it to future work.

Scope. The statements above concern population minimizers for fixed uncertainty weights. They do not imply that a finite network learns the interior field accurately, that $\mathcal{L}_{\mathrm{int}}$ improves the deployed predictor at finite capacity, or that m_Ψ integrates into low-drift trajectories; the ρ sweep tests sensitivity to boundary supervision but does not isolate the benefit of interior samples.

A.1.11 Boundary-Biased Timestep Sampling Analysis by Datasets

Training draws $t \sim \pi_\rho$ (Equation 3): each window receives $t \sim \mathcal{U}[0, 1]$, which is set to 0 with probability ρ. Splitting the objective by mixture component gives

$$\mathcal{L}_{\mathrm{CBM}} = \rho\,\underbrace{\mathbb{E}_q\,\Psi\big(x_0 + v_\theta(x_0, 0, x_0, z_c),\, x_1\big)}_{\mathcal{L}_{\mathrm{bnd}}} + (1-\rho)\,\underbrace{\mathbb{E}_{t\sim\mathcal{U}(0,1]}\,\mathbb{E}_q\,\Psi\big(x_t + (1-t)\,v_\theta(x_t, t, x_0, z_c),\, x_1\big)}_{\mathcal{L}_{\mathrm{int}}}.$$

$\mathcal{L}_{\mathrm{bnd}}$ is exactly the deployed one-step task (Equation 6), and $\mathcal{L}_{\mathrm{int}}$ is the interior reconstruction task. By Proposition 3, positive boundary mass yields the same population predictor m_Ψ at $t = 0$; ρ therefore does not change what the model should predict, only how training signal is divided between the deployed input and the interior. At $\rho = 0$ the deployed value is learned only by extrapolation $t \to 0$.

The same analysis explains the solver results in Appendix A.1.7. By Proposition 2, all consistent explicit Runge–Kutta schemes return $m_\Psi(x_0)$ under $v^\star$, so additional steps or higher order cannot improve on one Euler step at the optimum. One Euler step queries the network only at $t = 0$, whereas multi-step and higher-order solvers also query it at $t > 0$, where the learned field receives mass $(1-\rho)$ and has its own approximation error. Differences between solvers therefore measure the interior approximation error, not discretization error; whether this error grows with ρ at finite capacity is an empirical question that the solver curves and the ρ sweep should answer together.

A sweep in which only ρ changes, with source stacking in every row, is reported in Appendix A.1.12.

One-step distillation frameworks such as Inverse Bridge Matching Distillation (IBMD) (Gushchin et al., 2025) offer an alternative route to single-step inference. Concatenating x_0 does not leak the target at deployment: at $t = 0$ the network receives only $(x_0, 0, x_0, z_c)$, and the partial reveal $x_t - x_0 = t\,(x_1 - x_0)$ is present only for $t > 0$ during training. The relevant risk is instead that the network relies on this reveal and transfers poorly to $t = 0$; the boundary atom $\rho\,\delta_0$ supervises the deployed input directly.

A.1.12 BOUNDARY-RATE SWEEP

The learning-paradigm variants in Table 6 differ in more than one component. To isolate the timestep distribution, the runs in Table 5 share the dataset manifests, initialization and seed, frozen VAE, Mamba3 backbone, source stacking, structured loss, optimizer, batch size 128, and checkpoint-selection rule, and differ only in π_ρ of Equation 3. Each run is trained for 10 epochs, so all errors are higher than those of the converged models in Table 2.

Table 5: Boundary-rate sweep on RoNIN with source stacking in every row (single seed, 10 epochs, one Euler step from $t = 0$). Val. is the validation planar position error used to select the checkpoint; ATE and RTE are in meters (mean $\pm$ standard deviation over test sequences).

ρ	Val.	ATE (m) ↓		RTE (m) ↓	
		Seen	Unseen	Seen	Unseen
0 (uniform)	0.303	21.70 ± 7.17	24.81 ± 10.58	6.02 ± 1.79	7.38 ± 2.92
0.2	0.214	$\mathbf{5.62 \pm 2.48}$	8.75 ± 4.96	3.20 ± 1.44	4.83 ± 2.53
0.3	0.211	6.13 ± 2.20	9.30 ± 5.84	3.17 ± 1.37	4.73 ± 2.36
0.5	0.198	6.26 ± 2.13	$\mathbf{7.64 \pm 3.76}$	2.98 ± 1.26	4.38 ± 2.17
1 (regression)	**0.191**	6.56 ± 2.65	8.36 ± 4.23	$\mathbf{2.87 \pm 1.22}$	$\mathbf{4.35 \pm 2.01}$

With source stacking active in all runs, uniform scheduling ($\rho = 0$) yields 2.7–3.9 times the ATE of the evaluated positive-boundary-mass schedules. This contrast supports direct supervision of the deployed boundary query in the matched training regime. Relative to pure boundary regression ($\rho = 1$), every mixed schedule reduces seen-subject ATE, and $\rho = 0.5$ also reduces unseen-subject ATE; $\rho = 1$ gives the lowest RTE and validation error. The effects of interior training are thus metric-dependent in this ten-epoch, single-seed comparison. A converged multi-seed comparison would estimate their consistency across training runs.

A.1.13 EXTENDED LEARNING PARADIGM ABLATIONS

Table 6: Direction and scale metrics on RoNIN seen (S) and unseen (U) subjects (mean over test sequences) for the ablation variants in Table 2 and the $\rho = 1$ boundary-regression variant. Bold marks the best value per column (MCS ↑, TLR → 1).

Variant	Pos. MCS ↑		Pos. TLR → 1		Ori. MCS ↑		Ori. TLR → 1	
	S	U	S	U	S	U	S	U
IB (Ours, $\rho = 0.5$)	**0.79**	**0.76**	0.97	0.91	**0.13**	0.07	0.36	0.35
Zero token for z_c	**0.79**	**0.76**	0.96	0.91	0.12	0.07	0.37	0.30
$\rho = 1$ (regression)	**0.79**	**0.76**	0.97	0.92	0.12	**0.08**	0.45	0.41
$\rho = 0$ (uniform)	**0.79**	0.75	0.97	0.91	0.09	0.05	0.46	0.55
Direct regression	**0.79**	**0.76**	0.97	0.90	**0.13**	**0.08**	0.36	0.30
Flow matching	**0.79**	**0.76**	0.97	0.91	0.12	0.07	0.44	0.36
DDIM, 10 steps	0.78	0.74	0.97	0.93	0.10	0.07	**1.01**	0.96
DDIM, 40 steps	0.77	0.72	**0.99**	**0.95**	0.10	0.07	1.03	**1.03**

Table 6 presents a detailed ablation study of our learning paradigm, analyzing the impact of three core architectural components:

- **Variational Autoencoding (VAE):** Is intended to provide a compact source representation; domain-generalization effects are not isolated here.
- **Observation Stacking (Obs.):** Supplies source information to the drift network; its benefit is confounded with the learning objective in the current rows.
- **Boundary-Biased Timestep Sampling (Bnd.):** Emphasizes source-boundary fit but does not establish that shortcut learning is mitigated.

The reported comparison uses the VAE module by default for most variants; all paradigms share the optimizer configuration, learning rate, and batch size of 128.

Additionally, we made structural adaptations to specific baselines to align with their underlying formulations. For the IB variant without the VAE, z_c is replaced by a constant zero token and the network is otherwise unchanged. For Direct Regression, we disabled observation stacking; for the IB variant with uniform timestep sampling ($\rho = 0$), both observation stacking and boundary

biasing were disabled. Because direct regression and uniform sampling also change observation stacking, these rows are descriptive; Appendix A.1.12 isolates the bridge. For DDIM, we adopted x_0-parameterization over ϵ-parameterization to improve training stability.

A.1.14 Trajectories Where EqNIO Is the Strongest Baseline

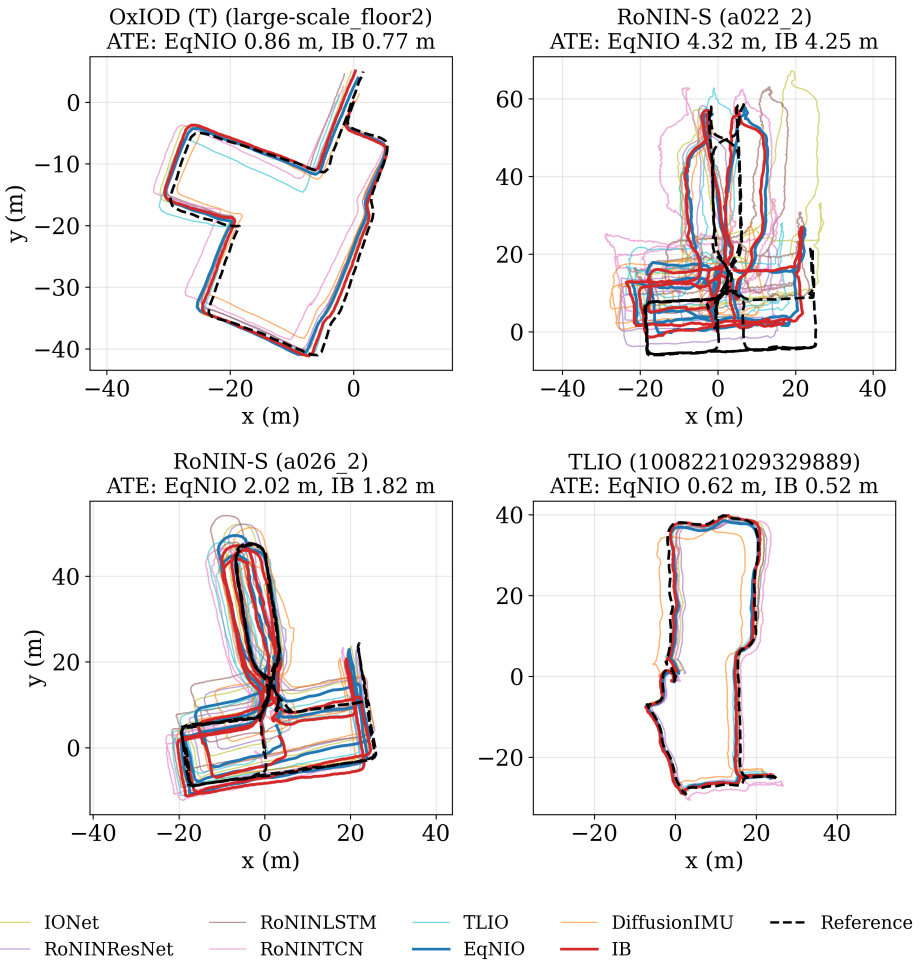

Figure 5: **Planar trajectories on test sequences where EqNIO is the most accurate baseline** under the pooled setting of Table 4. A sequence is shown if EqNIO has a lower position ATE than every other baseline and IB is at most 5% above EqNIO; all four such sequences are shown, and on each of them IB has the lower ATE. Trajectories are integrated from a common origin without alignment; window-level methods change position once per 100-sample window.

Figure 5 compares IB with EqNIO on the sequences where EqNIO is the strongest baseline, selected by the rule stated in the caption rather than by the error of IB. On these sequences the dense estimate of IB follows the reference at the input resolution with an error no larger than that of EqNIO, which updates once per window. The examples are qualitative and do not replace the sequence-level statistics of Table 4.

On these sequences the failing baselines drift by 2.5 to 4.1 times the error of EqNIO, while IB stays within 10% of EqNIO (Figure 6); on the TLIO sequence IB is also below EqNIO (0.57 vs. 0.62 m). IB reaches this accuracy with estimates at the input resolution, whereas EqNIO returns one displacement per window.

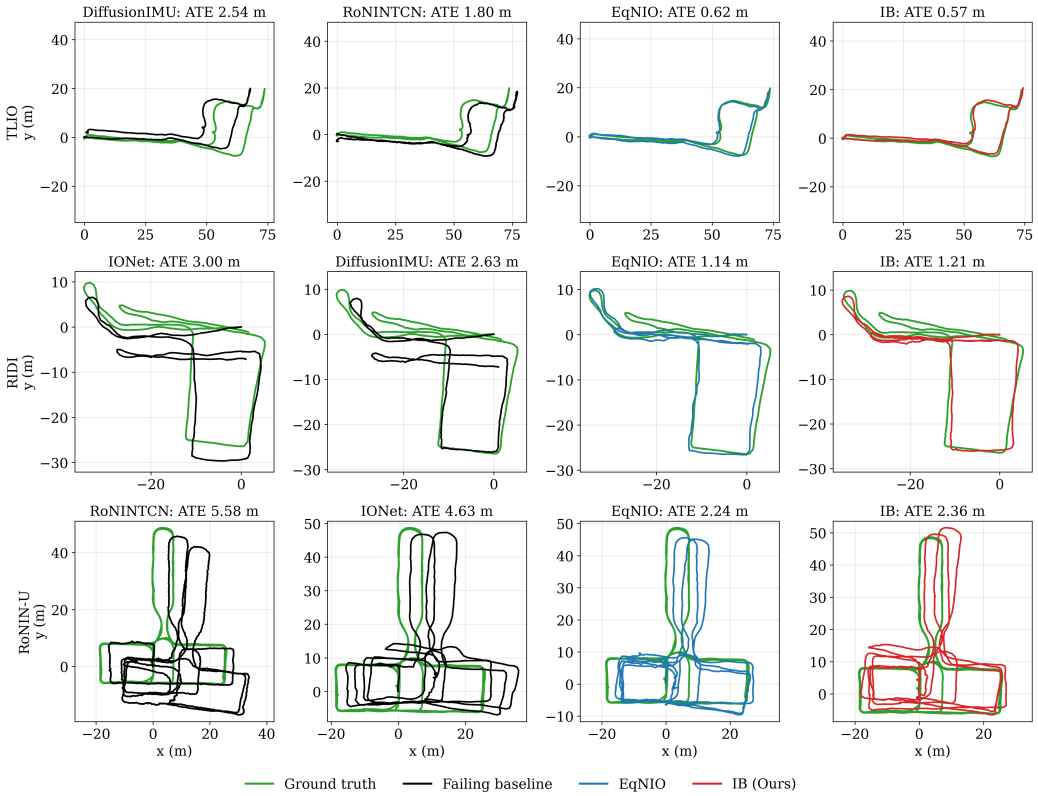

Figure 6: **Test sequences on which baselines fail and IB matches EqNIO** (pooled setting of Table 4). A sequence qualifies if some baseline has at least twice the position ATE of EqNIO and IB is within 10% of EqNIO; nine sequences qualify, and the one with the largest failure ratio is shown for each of TLIO, RIDI, and RoNIN-U. Each row shows the two baselines with the highest ATE on that sequence, EqNIO, and IB against the ground truth, integrated from a common origin without alignment.