PHIONet: Port Hamiltonian Inertial Odometry Network

Hechuan Shen[1], **Yingmin Liang**[2], **Jian Zhang**[1*]
[1]School of Computer Science, Wuhan University, [2]Dalian University of Technology
shc.email@qq.com

Abstract

Learning-based inertial odometry has shown strong potential for mitigating drift by extracting data-driven motion priors. However, existing methods primarily learn statistical mappings from inertial measurements to motion states, failing to consider the power balance between the specific-force input port and the predicted velocity. This limitation is particularly pronounced in aggressive MAV flight, where rapid accelerations, high angular rates, and aerodynamic dissipation can cause locally accurate velocity estimates to accumulate into trajectory drift. We propose PHIONet (Port Hamiltonian Inertial Odometry Network), which incorporates dissipative energy dynamics as a physically grounded prior. PHIONet translates the continuous port-Hamiltonian dynamics into a discrete energy-consistency residual over finite IMU windows, employing the DCM (Dissipative Coupling Module) to bridge IMU correction and velocity prediction. Experiments on the public Blackbird and EuRoC MAV datasets show that PHIONet achieves lower ATE and RTE than representative learning-based inertial odometry baselines.

1 Introduction

Inertial odometry is a fundamental capability for micro aerial vehicles operating in GPS-denied, visually degraded, or infrastructure-free environments. However, due to the double integration of inertial measurements, sensor noise and bias inevitably accumulate into rapid, unbounded drift along unobservable modes Trawny and Roumeliotis [2005].

To mitigate drift, learning-based inertial odometry has emerged as a powerful paradigm by extracting data-driven motion priors. Recent works have successfully constrained pedestrian motion by exploiting periodic gait patterns Chen *et al.* [2018]; Herath *et al.* [2020]. Some studies have extended these ideas to agile aerial platforms by encoding body-frame features and rotational invariants to enhance observability Qiu *et al.* [2025a]. Beyond motion priors, several learning-based approaches further attempt to mitigate drift by implicitly or explicitly modeling inertial noise and measurement uncertainty. While these methods improve statistical prediction accuracy, they fundamentally fail to account for the mechanical power balance induced by inertial excitation. This lack of physical grounding is particularly limiting for aggressive MAV flight, where rapid accelerations, high angular rates, and aerodynamic dissipation create a strong physical coupling between measured inertial excitation and body-frame velocity evolution. When this coupling is treated merely as a statistical correlation, locally accurate velocity predictions may still accumulate into larger trajectory drift during aggressive motion.

Although physics-informed learning frameworks provide a principled route for embedding physical structure into neural models, existing paradigms do not directly address the central challenge of IMU-only MAV inertial odometry: reducing trajectory drift from partial and noisy inertial excitation in an open, forced, and dissipative system. PINNs Raissi *et al.* [2019] typically assume residual-form

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

governing equations whose terms can be evaluated from available measurements, an assumption that breaks down when aerodynamic forces, actuation inputs, and sensor biases are unobserved or only partially observable. HNNs Greydanus *et al.* [2019], in contrast, impose conservative, energy-preserving dynamics, which conflicts with aggressive MAV flight where external energy injection and aerodynamic dissipation jointly shape the motion. These observations motivate a dissipative port hamiltonian perspective for learning-based MAV inertial odometry.

In this work, we propose PHIONet (Port Hamiltonian Inertial Odometry Network), a learning-based inertial odometry framework that incorporates physics-based constraints through the dissipative port hamiltonian formalism. Unlike conservative hamiltonian mechanics, dissipative port hamiltonian systems explicitly model energy dissipation and port-based interactions with external inputs. This structure is closely related to passivity: system energy evolves through dissipative losses and externally supplied power, rather than arbitrary spontaneous growth. This energy-balance structure provides a physically grounded prior for bridging measured inertial excitation with predicted body-frame velocity Van Der Schaft [2006]. To make this physical principle usable for neural inertial odometry, we discretize the port Hamiltonian power balance over finite IMU windows. The resulting discrete balance relates the predicted kinetic-energy variation to the accumulated work from corrected inertial excitation, gravitational effects, and aerodynamic dissipation. Based on this relation, PHIONet instantiates the port-Hamiltonian power balance as a discrete-time energy-consistency residual, which aligns IMU correction with velocity prediction, providing a structured training signal that improves velocity estimation and reduces drift accumulation.

The main contributions of our work are:

- **Theoretical Formulation:** We formulate MAV inertial odometry as a dissipative port hamiltonian system. To achieve this, we explicitly model energy exchanges under external forces and integrate rotational kinematics. This bridges continuous hamiltonian mechanics with discrete IMU sampling, yielding a physically motivated and computable discrete-time work–energy consistency constraint that serves as a principled structural prior for velocity estimation.
- **Physics-Informed Architecture:** Building on this theoretical foundation, we propose PHIONet. Our architecture couples IMU correction (Input Port) with velocity prediction (State Port) via the Dissipative Coupling Module (DCM). The proposed PHSLoss enforces this derived energetic consistency, effectively penalizing physical violations to suppress drift accumulation.
- **Experimental Validation:** We evaluate PHIONet on the public Blackbird and EuRoC MAV datasets. Among the evaluated representative learning-based inertial odometry baselines, PHIONet achieves the lowest average ATE and RTE. We further provide mechanism-oriented ablations and a diagnostic case study to analyze the roles of PHSLoss, DCM, and the learned dissipative correction.

2 Related Work

2.1 Classical inertial navigation and estimation

Classical inertial navigation estimates velocity and position by integrating the force and angular velocity measurements from the IMU. Since this process involves double integration, errors accumulate continuously even with small sensor noise and bias, causing the estimates to drift rapidly Trawny and Roumeliotis [2005]. Therefore, most model-based odometry systems further suppress drift by combining IMU propagation with external perception sensors Forster *et al.* [2015]; Qin *et al.* [2018]; Zhao *et al.* [2021]. While these works provide valuable references for inertial navigation, their reliance on external sensors fundamentally differs from the pure IMU inertial navigation setting addressed in this paper.

2.2 Learning-based inertial odometry

Learning-based IO aims to reduce drift by learning motion priors from inertial sequences. We review this line of work from two perspectives: state estimation and noise estimation.

2.2.1 State estimation

Across the literature, learned motion priors enter the estimation pipeline in different ways. Some models regress displacement or velocity end-to-end. Elsewhere, the prior is fused inside a tightly coupled filter, whereas optimization-based formulations recover the trajectory by minimizing an objective that incorporates the learned model. They mainly differ in how learned motion priors are represented and incorporated into the estimation pipeline. Such methods include robust double integration Yan *et al.* [2018] as well as sequence models that predict displacement or velocity Chen *et al.* [2018]; Wang *et al.* [2022]; Herath *et al.* [2022]. Many approaches further improve consistency by coupling learned predictions with estimators, including EKF-style backends Liu *et al.* [2020]; Sun *et al.* [2021] and batch optimization frameworks Buchanan *et al.* [2022]. Learning-based state estimation has also been explored using cascaded architectures that sequentially estimate orientation and translation from inertial data Deng *et al.* [2023]. Representation is also critical: RoNIN introduced a heading-agnostic coordinate frame (HACF) for pedestrian IO Herath *et al.* [2020], and RIO proposed rotation-equivariant training strategies Cao *et al.* [2022]; for wheeled platforms, RINS-W highlights the role of planar-motion constraints Brossard *et al.* [2019]. For multirotor UAVs, prior work explored AI-based preprocessing and down-sampling of high-rate inertial data Steinbrener *et al.* [2022] and UAV-specific learned IO models such as DIDO Zhang *et al.* [2022], learned IO for drone racing (IMO) Cioffi *et al.* [2023], and DIVE Bajwa *et al.* [2024]. Most closely related, Air-IO shows that preserving body-frame IMU representation and explicitly encoding attitude improves IMU feature observability for agile UAV motionQiu *et al.* [2025a].

2.2.2 Noise estimation

Due to the amplification of sensor noise and bias during integration, recent learning-based systems have progressively incorporated explicit noise modeling mechanisms. These include learning to correct inertial measurements or predicting uncertainties for downstream fusion applications Liu *et al.* [2020]; Qiu *et al.* [2025a]. Such approaches are often inspired by the error state modeling concepts found in classical inertial navigation systems Trawny and Roumeliotis [2005]. Air-IO further leverages AirIMU to pre-process raw IMU measurements and to estimate per-sample corrections and uncertainty for preintegration and EKF updates under aggressive maneuvers Qiu *et al.* [2023, 2025a]. Recently, there are some UAV-oriented approaches that address unmodeled effects by incorporating additional auxiliary cues such as thrust information Zhang *et al.* [2022]; Cioffi *et al.* [2023].

2.3 Summary

The above approaches have substantially advanced learning-based inertial odometry through improved motion priors and explicit noise modeling. However, they focus on the statistical level, and do not constrain learned inertial systems from an energetic perspective. In contrast, PHIONet focuses on physical consistency. By introducing a computable discrete-time work–energy consistency constraint on learning-based inertial odometry, it encourages energetically consistent predictions.

3 Dissipative Port Hamiltonian Structure

3.1 Hamiltonian Variation

In the standard port Hamiltonian formulation, the Hamiltonian includes both kinetic and potential energyVan Der Schaft [2006]; Van Der Schaft and Jeltsema [2014]:

$$H(\mathbf{v}, \mathbf{p}) = \underbrace{\frac{1}{2}\|\mathbf{v}\|^2}_{T} + \underbrace{(-\mathbf{g}^\top \mathbf{p})}_{V}, \tag{1}$$

However, tracking absolute position in pure inertial systems suffers from quadratic drift Titterton and Weston [2004]. Noting that the potential energy rate $\dot{V}$ represents the gravitational work rate. We model the UAV dynamics as a dissipative port Hamiltonian system with kinetic energy storage only, treating gravity as an external force performing work on the system. Assuming constant mass during short-term odometry intervals, the system stores energy solely in kinetic form:

$$H(\mathbf{v}) = \frac{1}{2}\|\mathbf{v}^W\|^2. \tag{2}$$

Since kinetic energy is invariant under rotation:

$$\|\mathbf{v}^W\|^2 = \|\mathbf{v}^B\|^2. \tag{3}$$

The change in Hamiltonian from t_i to t_j is:

$$\Delta H_{ij} = H_j - H_i = \frac{1}{2}\|\mathbf{v}_j^B\|^2 - \frac{1}{2}\|\mathbf{v}_i^B\|^2. \tag{4}$$

3.2 Work Done by External Forces

For UAV dynamics, the external forces acting on the system consist of thrust, gravity, and dissipative effects:

$$\mathbf{F}^B = \mathbf{R}_{WB}\mathbf{a}^B + \mathbf{g}^W - \mathbf{R}_{WB}\mathbf{D}(\mathbf{v^B}) \tag{5}$$

where $\mathbf{R}_{WB} \in SO(3)$ is the body-to-world rotation, $\mathbf{a}^B$ is the IMU specific force, $\mathbf{v}^B$ is the velocity in body frame, $\mathbf{F}^B$ is the external force acting on the system in the world frame, and $(\mathbf{v^B})^\top \mathbf{D}(\mathbf{v^B}) \geq 0$ represents velocity-dependent aerodynamic drag.

Accordingly, the instantaneous external power of the system is given by:

$$\dot{E} = \underbrace{(\mathbf{v}^B)^\top \mathbf{a}^B}_{\text{thrust}} + \underbrace{(\mathbf{v}^W)^\top \mathbf{g}^W}_{\text{gravity}} - \underbrace{(\mathbf{v}^B)^\top \mathbf{D}(\mathbf{v^B})}_{\text{dissipation}} \tag{6}$$

The power-conjugate pair $(\mathbf{a}^B, \mathbf{v}^B)$ identifies IMU acceleration as an energy input rate Ortega *et al.* [2002]. Integrating (6) over $[t_i, t_j]$, the change in energy is:

$$\Delta E_{ij} = \int_{t_i}^{t_j} (\mathbf{v}^B)^\top \mathbf{a}^B \, d\tau + \int_{t_i}^{t_j} (\mathbf{v}^W)^\top \mathbf{g}^W \, d\tau - \int_{t_i}^{t_j} (\mathbf{v}^B)^\top \mathbf{D}(\mathbf{v^B}) \, d\tau \tag{7}$$

The gravity work integral transforms to body frame:

$$\int_{t_i}^{t_j} (\mathbf{v}^W)^\top \mathbf{g}^W \, d\tau = \int_{t_i}^{t_j} (\mathbf{v}^B)^\top (\mathbf{R}^\top \mathbf{g}^W) \, d\tau. \tag{8}$$

Integrating the rotational kinematics $\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega}^B]_\times$ over each IMU sample interval yields:

$$\mathbf{R}_{k+1} = \mathbf{R}_k \cdot \mathrm{Exp}(\boldsymbol{\omega}_k^B \Delta t_k) \tag{9}$$

where $\boldsymbol{\omega}_k^B$ is the gyroscope measurement, Δt_k the sampling interval, and $\mathrm{Exp} : \mathfrak{so}(3) \to SO(3)$ the exponential map Sola *et al.* [2018].

Based on (9), we define the relative rotation as:

$$\Delta \mathbf{R}_{ij} \triangleq \mathbf{R}_i^\top \mathbf{R}_j = \prod_{k=i}^{j-1} \mathrm{Exp}(\boldsymbol{\omega}_k^B \Delta t_k). \tag{10}$$

This provides the transformation between frames.[2]

With local linearization (assuming slowly-varying orientation over Δt):

$$\int_{t_i}^{t_j} (\mathbf{v}^B(\tau))^\top (\mathbf{R}(\tau)^\top \mathbf{g}^W) \, d\tau \approx (\mathbf{v}_i^B)^\top \mathbf{R}_i^\top \mathbf{g}^W \cdot \Delta t_{ij} \approx (\mathbf{v}_i^B)^\top \mathbf{g}_i^B \cdot \Delta t_{ij}. \tag{11}$$

This expresses gravity as work done on velocity, avoiding position integration. [3]

Therefore, the work of external forces can be written in discrete form as:

$$\Delta E_{ij} \approx \sum_{k=i}^{j-1} (\mathbf{v}_k^B)^\top \mathbf{a}_k^B \, \Delta t_k + \sum_{k=i}^{j-1} (\mathbf{v}_k^B)^\top \mathbf{g}_k^B \, \Delta t_k - \sum_{k=i}^{j-1} (\mathbf{v}_k^B)^\top \mathbf{D}(\mathbf{v}_k^B) \, \Delta t_k \tag{12}$$

[2] Our derivation integrates in world frame then transforms to body frame, avoiding explicit Coriolis terms.

[3] The approximation error is bounded by $\|\boldsymbol{\epsilon}_g\| \leq \frac{1}{2}\|\boldsymbol{\omega}\|\|\mathbf{g}\|\Delta t^2$, which is comparable to sensor noise at typical IMU rates ($\sim$0.01 m/s at 100 Hz).

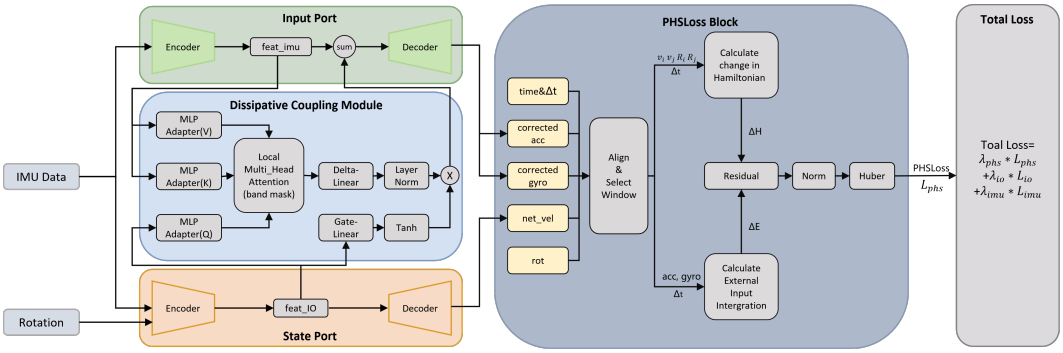

Figure 1: **PHIONet.** IMU data flows through two parallel branches: *Input Port:* outputs corrected IMU measurements, and *State Port:* outputs body-frame velocities. The *Dissipative Coupling Module* enables velocity-dependent dissipation compensation through cross-attention-based information flow from State Port to Input Port. Outputs from both branches feed into the *PHSLoss Block*, which compares Hamiltonian change ΔH with external work ΔE to penalize energy imbalance. The network is trained with combined supervision: physical consistency ($\mathcal{L}_{\text{phs}}$), velocity accuracy ($\mathcal{L}_{\text{io}}$), and trajectory accuracy ($\mathcal{L}_{\text{imu}}$).

3.3 Discrete-Time Work–Energy Consistency

In a port Hamiltonian system, the change in the Hamiltonian over a time interval equals the net work done by external forces

$$\Delta H_{ij} = \Delta E_{ij}. \tag{13}$$

Having expressed the Hamiltonian variation in terms of body-frame velocities (Eq.(4)) and approximated the external work in discrete time (Eq.(12)), we arrive at a compact work–energy consistency relation:

$$\frac{1}{2}\|\mathbf{v}_j^B\|^2 - \frac{1}{2}\|\mathbf{v}_i^B\|^2 = \sum_{k=i}^{j-1}(\mathbf{v}_k^B)^\top \mathbf{a}_k^B\,\Delta t_k + \sum_{k=i}^{j-1}(\mathbf{v}_k^B)^\top \mathbf{g}_k^B\,\Delta t_k - \sum_{k=i}^{j-1}(\mathbf{v}_k^B)^\top \mathbf{D}(\mathbf{v}_k^B)\,\Delta t_k. \tag{14}$$

This equality couples the predicted velocity trajectory with the corrected IMU input, providing a computable physical consistency constraint for learning-based inertial odometry.

4 PHIONet

Based on the computable physical consistency constraint set forth in the preceding Section 3, we propose PHIONet. PHIONet consists of two coupled estimation branches that jointly learn corrected IMU measurements and body-frame velocities under the proposed work–energy consistency constraint.

4.1 Overall

According to the physical consistency constraint, there are two complementary estimation tasks:

- **Input Port:** Based on the IMU input, clean measurements $\mathbf{a}^B$ can be obtained.
- **State Port:** Based on the IMU input, estimate body-frame velocities $\mathbf{v}^B$ can be obtained.

Figure 1 illustrates the complete PHIONet architecture. PHIONet augments two complementary branches with the dissipative coupling module. Additionally, Port Hamiltonian System Loss is used to satisfy the discrete work–energy consistency constraint.

4.2 Dissipative Coupling Module

As shown in Section 3, there is a dissipation term $(\mathbf{v}^B)^\top \mathbf{D}(\mathbf{v}^B)$ contained in the discrete energy balance (12). However, $\mathbf{D}(\mathbf{v}^B)$ is not able to be calculated. We therefore adopt an implicit dissipation

strategy:

$$\Delta E_{ij} = \sum_{k=i}^{j-1} (\mathbf{v}_k^B)^\top (\mathbf{a}_k^B + \delta \mathbf{a}_k^B)\, \Delta t_k + \sum_{k=i}^{j-1} (\mathbf{v}_k^B)^\top \mathbf{g}_k^B\, \Delta t_k. \tag{15}$$

in which $\delta\mathbf{a}$ is the learned dissipative correction serving to represent the dissipative effects by approximating $-\mathbf{D}(\mathbf{v}^B)$.

To accurately estimate $\delta\mathbf{a}$, the Input Port requires velocity information to model dissipation. We therefore propose the Dissipative Coupling Module (DCM), which enables the Input Port to receive velocity information from the State Port through a directed information flow. As shown in Fig. 1, the DCM adopts a gated cross-attention mechanism Qiu *et al.* [2025b] to realize a directed information flow from the State Port to the Input Port.

The query $\mathbf{Q}$ is derived from State Port (Air-IO) encoder features. The key $\mathbf{K}$ and value $\mathbf{V}$ are derived from Input Port (AirIMU) encoder features. This is illustrated as:

$$\mathbf{Q} = \phi_Q(\mathbf{f}_{\text{io}}), \quad \mathbf{K} = \phi_K(\mathbf{f}_{\text{imu}}), \quad \mathbf{V} = \phi_V(\mathbf{f}_{\text{imu}}) \tag{16}$$

where ϕ_Q, ϕ_K, ϕ_V are lightweight MLP adapters. This design ensures that velocity information from State Port queries relevant correction patterns from the Input Port.

To maintain temporal locality, we employ a band-masked attention:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V} \tag{17}$$

where $\mathbf{M}$ restricts each query to attend only within a local window of size W.

The output of attention is transformed into a residual feature by a MLP layer and the LayerNorm:

$$\Delta\mathbf{f} = \text{LayerNorm}(\phi_\Delta(\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}))) \tag{18}$$

where ϕ_Δ is a two-layer MLP. The output is mapped to the space of IMU feature by ϕ_Δ.

The residual is then modulated by a velocity-driven gate and fused with the original feature:

$$\mathbf{f}_{\text{fused}} = \mathbf{f}_{\text{imu}} + \tanh(\mathbf{W}_g \mathbf{f}_{\text{io}}) \odot \Delta\mathbf{f} \tag{19}$$

The gate projection $\mathbf{W}_g$ is initialized near zero, ensuring minimal perturbation to pre-trained features at the beginning of training. The fused feature $\mathbf{f}_{\text{fused}}$ is then decoded to produce $(\mathbf{a}^B + \delta\mathbf{a}^B)$.

4.3 Port Hamiltonian System Loss

Section 4.2 introduces the implicit dissipation strategy. As shown in Eq. (15), the dissipative effects are absorbed into the learned correction $\delta\mathbf{a}^B$. In order to navigate the network toward energy-balanced outputs during training, we construct the Port Hamiltonian System Loss (PHSLoss).

Firstly, we construct the energy residual by evaluating the power flow along the discretized trajectory. Formally, this is given by:

$$r_{ij} = \underbrace{(\bar{\mathbf{v}}^B)^\top (\hat{\mathbf{v}}_j^B - \hat{\mathbf{v}}_i^B)}_{\Delta H_{ij}\ \text{(Kinetic Variation)}} - \underbrace{(\bar{\mathbf{v}}^B)^\top \sum_{k=i}^{j-1} \left(\Delta\mathbf{R}_{i,k}\hat{\mathbf{a}}_k^B + \mathbf{g}_i^B\right) \Delta t_k}_{\Delta E_{ij}\ \text{(Work Done)}} \tag{20}$$

where $\bar{\mathbf{v}}^B = \frac{1}{2}(\hat{\mathbf{v}}_i^B + \hat{\mathbf{v}}_j^B)$ represents the state trajectory average. The work done by external force term accumulates the corrected specific force $\hat{\mathbf{a}}_k^B$. Additionally, specific forces are rotated to the k-th frame via $\Delta\mathbf{R}_{i,k}$. The gravity term $\mathbf{g}_i^B$ is applied over each timestep Δt_k.

The residual projects the discrepancy between velocity evolution and accumulated inertial work onto a scalar work–energy balance. Minimizing $|r_{ij}|$ encourages the predicted velocity and corrected IMU excitation to remain energetically consistent over each finite window. This method effectively acts as a physics-informed regularizer that couples the velocity evolution with the input excitation.

We also apply a Huber loss Huber [1992] to the residual for robustness:

$$\mathcal{L}_{\text{PHS}} = \frac{1}{|\mathcal{W}|} \sum_{(i,j)\in\mathcal{W}} \rho_\delta\big(|r_{ij}|\big) \tag{21}$$

Table 1: Trajectory accuracy comparison with baselines on Blackbird and EuRoC MAV.

Dataset	Seq.	Baseline		AirIMU		RoNIN		TLIO		IMO		Air-IO		PHIONet	
		ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE
Blackbird	Ampersand	41.138	5.026	24.738	4.379	5.467	5.321	4.665	4.078	17.664	10.039	2.303	1.145	**1.586**	**0.618**
	Sid	13.108	2.055	16.832	2.131	18.581	10.665	7.323	8.427	9.967	6.992	0.717	**0.527**	**0.577**	0.595
	Oval	15.365	2.400	8.312	1.553	20.399	12.120	1.276	1.045	5.670	2.863	0.976	0.583	**0.435**	**0.358**
	Sphinx	3.429	2.713	2.828	2.024	11.537	7.762	2.005	2.408	5.629	5.395	1.145	1.008	**1.042**	**0.875**
	BentDice	28.307	2.342	54.261	4.390	11.453	6.792	2.119	1.747	6.143	4.519	1.360	1.000	**1.124**	**0.668**
	Avg.	20.269	2.907	21.394	2.895	13.487	8.532	3.478	3.541	9.015	5.962	1.300	0.853	**0.953**	**0.623**
EuRoC	MH02	447.222	34.594	55.382	4.447	5.902	2.121	7.281	2.451	5.710	2.339	4.917	0.936	**2.545**	**0.927**
	MH04	164.018	23.514	24.335	3.126	8.586	4.542	8.626	5.498	7.576	5.016	2.726	1.093	**2.621**	**1.013**
	V103	162.442	16.624	18.834	3.375	3.240	1.695	7.863	2.580	15.654	3.151	3.844	1.519	**3.001**	**1.453**
	V202	269.712	28.139	181.852	23.215	7.445	2.303	6.260	2.783	6.054	2.765	4.823	1.303	**4.216**	**1.290**
	V101	340.849	30.543	113.587	10.844	8.576	1.918	4.814	1.946	2.934	1.628	2.917	1.137	**2.657**	**1.035**
	Avg.	276.849	26.683	78.798	9.002	6.750	2.516	6.969	3.052	7.586	2.980	3.845	1.198	**3.008**	**1.144**

where $\mathcal{W}$ is the set of time windows, and ρ_δ is the Huber function:$\rho_\delta(s) = \begin{cases} \frac{1}{2}s^2, & s \le \delta \\ \delta(s - \frac{1}{2}\delta), & s > \delta \end{cases}$.

As illustrated in Fig. 1, the Input Port and State Port operate on different temporal resolutions: the corrected IMU maintains T frames at the original sensor rate, while the velocity output contains L frames after encoder downsampling (typically $L \approx T/9$ due to the stride-3 convolutions in both branches). PHSLoss must bridge these two time axes to compute the energy residual.

Let $\mathbf{t}^{\text{imu}} \in \mathbb{R}^T$ denote the input port timestamps and $\mathbf{t}^{\text{vel}} \in \mathbb{R}^L$ denote the state port timestamps. For each velocity frame $l \in [1, L]$, we define a nearest-neighbor mapping:

$$\text{align}(t) = \arg\min_{k \in [1,T]} |t_k^{\text{imu}} - t| \tag{22}$$

As a result, we construct overlapping windows on the velocity time axis. For a window size M, we form a window (i, j) for each start index $i \in [1, L - M]$. The complete window construction and residual computation procedure is provided in Appendix A.

4.4 Training Strategy

In order to train different parts of PHIONet, we design a three-stage training strategy. In Stage 1, the Input Port (AirIMU) and State Port (Air-IO) are pre-trained independently on their respective tasks for 100 epochs each. In Stage 2, both pre-trained networks are frozen and only the DCM is trained for 5 epochs, allowing the gate values to increase from their near-zero initialization. In Stage 3, the Input Port, State Port, and DCM are jointly fine-tuned for 30 epochs. The total loss for joint training combines three components:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{io}}\mathcal{L}_{\text{IO}} + \lambda_{\text{imu}}\mathcal{L}_{\text{IMU}} + \lambda_{\text{phs}}\mathcal{L}_{\text{PHS}} \tag{23}$$

where $\mathcal{L}_{\text{IO}}$ supervises velocity estimation, $\mathcal{L}_{\text{IMU}}$ supervises IMU correction via trajectory integration, and $\mathcal{L}_{\text{PHS}}$ enforces the energy balance constraint.

5 Experiments

5.1 Datasets

We evaluate PHIONet on two MAV datasets: Blackbird Antonini *et al.* [2020] and EuRoC MAV Burri *et al.* [2016]. Blackbird contains aggressive quadrotor flights with synchronized IMU and motion-capture ground truth, while EuRoC MAV contains indoor MAV sequences with 200 Hz IMU measurements and ground-truth trajectories. We follow the official data processing procedures and dataset splits, and report Blackbird results only on evaluation trajectories that do not appear in training. Additional dataset details are provided in Appendix B.

5.2 Metrics

We use absolute trajectory error (ATE) and relative trajectory error (RTE) between estimated positions $\{\hat{\mathbf{p}}_t\}_{t=1}^N$ and ground truth $\{\mathbf{p}_t\}_{t=1}^N$: $\text{ATE} = \sqrt{\frac{1}{N}\sum_{t=1}^N \|\hat{\mathbf{p}}_t - \mathbf{p}_t\|^2}$, $\text{RTE} =$

$\sqrt{\frac{1}{N-L}\sum_{t=1}^{N-L}\|(\hat{\mathbf{p}}_{t+L}-\hat{\mathbf{p}}_t)-(\mathbf{p}_{t+L}-\mathbf{p}_t)\|^2}$. For relative trajectory error (RTE), we evaluate over a fixed temporal window of 5 seconds. The window length is set to $L = 500$ frames on Blackbird (100 Hz) and $L = 1000$ frames on EuRoC (200 Hz).

5.3 Evaluation Protocol

PHIONet is trained and evaluated on Blackbird and EuRoC separately using the same network architecture and identical evaluation metrics. The network takes only IMU measurements as input, while ground-truth trajectories are used only for training supervision and evaluation. During evaluation, body-frame velocity estimates are transformed to the world frame and integrated in an open-loop manner to reconstruct trajectories. Further details are provided in Appendix C.

5.4 Implementation Details

5.4.1 Training

During training and validation, each sequence is split into fixed-length windows using a sliding window strategy. On each dataset, PHIONet is trained following the procedure described in Section 4. The checkpoint with the lowest validation loss is selected for evaluation. The training objective consists of three terms: Air-IO velocity supervision, AirIMU correction supervision, and the proposed PHSLoss. All experiments are run in double precision. Detailed hyperparameters and dataset-specific loss weights are provided in Table 3 of Appendix D.

5.4.2 Inference

During validation and evaluation, PHIONet is run on each sequence as a whole. Inference is performed with a batch size of 1. The network processes the full sequence in a single forward pass.

PHIONet outputs body-frame velocity estimates together with their associated timestamps. Due to the use of strided temporal encoders, the output rate is lower than the IMU sampling rate. As a result, velocity predictions are produced at sparse time steps. Both the predicted velocities and their timestamps are stored for later processing. The computational cost and single-window inference latency of PHIONet are reported in Appendix E.

5.4.3 Evaluation

The network outputs velocity predictions together with their timestamps. These timestamps do not match the IMU time grid. Before trajectory reconstruction, the velocities are aligned to the IMU timestamps. We use one-dimensional linear interpolation on each axis to obtain velocities at every IMU time step.

The interpolated velocities are expressed in the body frame. They are rotated to the world frame using the orientation at each time step. The trajectory is then obtained by integrating the velocities over time.

5.5 Result

Table 1 reports ATE and RTE on the Blackbird and EuRoC MAV datasets. We compare PHIONet against Baseline (raw-IMU dead reckoning via kinematic integration), IMU noise correction method (AirIMU Qiu *et al.* [2023]), end-to-end learning method (RoNIN Herath *et al.* [2020]), EKF-fused learning method (TLIO Liu *et al.* [2020]), UAV-specific method with additional thrust cues (IMO Cioffi *et al.* [2023]), and the strong IMU-only UAV baseline(Air-IO Qiu *et al.* [2025a]).

Across both datasets, PHIONet achieves the lowest average ATE and RTE, showing that introducing an explicit physical-consistency objective improves trajectory accuracy beyond motion priors and statistical noise modeling alone.

5.6 Ablation Study

We conduct ablations on the Blackbird evaluation sequences to examine the two core components of PHIONet: PHSLoss and DCM. Specifically, we consider two groups of variants: one replaces

Table 2: Ablation Study on Blackbird.

Seq.	PHIONet		DCM Only		PHSLoss Only		Concat-MLP		Cross-Attention		Smoothness		Acceleration	
	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE	ATE	RTE
Ampersand	**1.586**	**0.618**	1.758	0.795	1.880	0.807	1.718	0.688	1.598	0.631	2.327	1.082	1.735	0.639
Sid	**0.577**	0.595	0.892	0.571	0.838	0.563	0.663	**0.484**	0.648	0.485	3.457	4.137	0.810	0.501
Oval	**0.435**	**0.358**	0.709	0.542	0.734	0.561	0.708	0.509	0.554	0.432	2.520	2.032	0.774	0.563
Sphinx	**1.042**	**0.875**	1.181	1.001	1.173	1.009	1.072	0.902	1.096	0.930	2.367	2.990	1.101	0.971
BentDice	1.124	**0.668**	**1.080**	0.749	1.125	0.760	1.215	0.732	1.243	0.714	2.468	2.663	1.224	0.741
Avg.	**0.953**	**0.623**	1.124	0.731	1.150	0.740	1.075	0.663	1.028	0.640	2.628	2.581	1.129	0.683

PHSLoss with alternative consistency objectives, while the other replaces DCM with simpler coupling modules. All variants keep the remaining training protocol unchanged. Detailed definitions of the ablated variants are provided in Appendix F.

Table 2 shows that PHIONet achieves the lowest average ATE and RTE among all variants. The DCM replacement variants, Concat-MLP and Cross-Attention, consistently underperform PHIONet, indicating that the proposed DCM provides a more effective coupling design than generic residual fusion or ungated directed attention. The PHSLoss replacement variants, Smoothness and Acceleration, also degrade performance, suggesting that PHSLoss is more effective than acceleration-form velocity smoothing or time-normalized vector consistency over the same finite windows.

Overall, these results suggest that PHSLoss and DCM provide complementary benefits: PHSLoss supplies the finite-window physical consistency signal, while DCM provides a structured pathway for coupling IMU correction with velocity prediction.

5.7 Mechanistic Analysis

To further inspect whether the proposed physical coupling induces interpretable correction behavior, we provide a diagnostic case study in Appendix G. This analysis examines two complementary aspects: finite-window work–energy alignment and the velocity-aligned projection of the learned acceleration correction. The case study shows that, on a diagnostic closed-loop flight sequence, PHIONet reduces the typical work–energy mismatch and learns a correction with drag-like dissipative behavior. This analysis is intended to make the learned mechanism observable, while the primary quantitative evidence remains the benchmark comparison and ablation results.

6 Conclusion

We introduced PHIONet, a dissipative port Hamiltonian framework for learning-based MAV inertial odometry. PHIONet moves beyond purely statistical motion priors by enforcing a finite-window work–energy consistency relation between corrected IMU excitation and predicted body-frame velocity. This principle is implemented through PHSLoss and DCM, which jointly couple the Input Port for IMU correction and the State Port for velocity prediction.

Experiments on Blackbird and EuRoC MAV show that PHIONet achieves lower average ATE and RTE than representative inertial odometry baselines. Ablation results further show that PHSLoss and DCM contribute complementary improvements, and the mechanistic case study suggests that the learned correction can exhibit a drag-like, velocity-aligned dissipative pattern. These findings support the central claim that finite-window energetic consistency provides a principled physical prior for reducing drift under aggressive MAV motion.

The current formulation still relies on finite-window approximations and available orientation estimates, and it represents dissipation implicitly rather than as an explicit aerodynamic model. Future work will explore joint orientation–velocity estimation, richer dissipative parameterizations, and real-time onboard deployment. More broadly, this work points toward energy-consistent learning as a promising direction for robotic systems in which estimation, dynamics, and control are tightly coupled.

References

Amado Antonini, Winter Guerra, Varun Murali, Thomas Sayre-McCord, and Sertac Karaman. The blackbird uav dataset. *The International Journal of Robotics Research*, 39(10-11):1346–1364, 2020.

Angad Bajwa, Charles Champagne Cossette, Mohammed Ayman Shalaby, and James Richard Forbes. Dive: Deep inertial-only velocity aided estimation for quadrotors. *IEEE Robotics and Automation Letters*, 9(4):3728–3734, 2024.

Martin Brossard, Axel Barrau, and Silvere Bonnabel. Rins-w: Robust inertial navigation system on wheels. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2068–2075. IEEE, 2019.

Russell Buchanan, Marco Camurri, Frank Dellaert, and Maurice Fallon. Learning inertial odometry for dynamic legged robot state estimation. In *Conference on robot learning*, pages 1575–1584. PMLR, 2022.

Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

Xiya Cao, Caifa Zhou, Dandan Zeng, and Yongliang Wang. Rio: Rotation-equivariance supervised learning of robust inertial odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6614–6623, 2022.

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.

Giovanni Cioffi, Leonard Bauersfeld, Elia Kaufmann, and Davide Scaramuzza. Learned inertial odometry for autonomous drone racing. *IEEE Robotics and Automation Letters*, 8(5):2684–2691, 2023.

Xiangyu Deng, Shenyue Wang, Chunxiang Shan, Jinjie Lu, Ke Jin, Jijunnan Li, and Yandong Guo. Data-driven based cascading orientation and translation estimation for inertial navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3381–3388. IEEE, 2023.

Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. *Robotics: Science and Systems Proceedings*, 2015.

Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. Openvins: A research platform for visual-inertial estimation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4666–4672. IEEE, 2020.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

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)*, pages 3146–3152. IEEE, 2020.

Sachini Herath, David Caruso, Chen Liu, Yutian Chen, and Yasutaka Furukawa. Neural inertial localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6604–6613, 2022.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

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.

Romeo Ortega, Arjan Van Der Schaft, Bernhard Maschke, and Gerardo Escobar. Interconnection and damping assignment passivity-based control of port-controlled hamiltonian systems. *Automatica*, 38(4):585–596, 2002.

Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

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.

Yuheng Qiu, Can Xu, Yutian Chen, Shibo Zhao, Junyi Geng, and Sebastian Scherer. Airio: Learning inertial odometry with enhanced imu feature observability. *arXiv preprint arXiv:2501.15659*, 2025.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *arXiv preprint arXiv:2505.06708*, 2025.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Joan Sola, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018.

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. IEEE, 2022.

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.

David Titterton and John L Weston. *Strapdown inertial navigation technology*, volume 17. IET, 2004.

Nikolas Trawny and Stergios I Roumeliotis. Indirect kalman filter for 3d attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, 2:2005, 2005.

Arjan Van Der Schaft and Dimitri Jeltsema. Port-hamiltonian systems theory: An introductory overview. *Foundations and Trends® in Systems and Control*, 1(2-3):173–378, 2014.

Arjan Van Der Schaft. Port-hamiltonian systems: an introductory survey. In *Proceedings of the international congress of mathematicians*, volume 3, pages 1339–1365. Marta Sanz-Sole, Javier Soria, Juan Luis Verona, Joan Verdura, Madrid, Spain, 2006.

Yingying Wang, Hu Cheng, and Max Q-H Meng. A2dio: Attention-driven deep inertial odometry for pedestrian localization based on 6d imu. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 819–825. IEEE, 2022.

Hang Yan, Qi Shan, and Yasutaka Furukawa. Ridi: Robust imu double integration. In *Proceedings of the European conference on computer vision (ECCV)*, pages 621–636, 2018.

Kunyi Zhang, Chenxing Jiang, Jinghang Li, Sheng Yang, Teng Ma, Chao Xu, and Fei Gao. Dido: Deep inertial quadrotor dynamical odometry. *IEEE Robotics and Automation Letters*, 7(4):9083–9090, 2022.

Shibo Zhao, Hengrui Zhang, Peng Wang, Lucas Nogueira, and Sebastian Scherer. Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8729–8736. IEEE, 2021.

Appendix

A PHSLoss Window Construction

Algorithm 1 summarizes the implementation procedure for constructing PHSLoss windows and computing the corresponding work–energy residuals. Since the corrected IMU sequence and predicted velocity sequence are produced at different temporal resolutions, the procedure first aligns velocity-window endpoints to the nearest IMU timestamps, then integrates the corrected inertial excitation over the matched IMU interval.

B Datasets

B.1 Blackbird

Our experiments use the Blackbird aggressive-flight dataset Antonini *et al.* [2020]. The data were collected during highly dynamic quadrotor flights and provide synchronized IMU measurements together with motion-capture ground truth.[4] The dataset contains high-rate IMU signals, motor RPM measurements, and motion-capture trajectories, all processed using the official pipeline released with the dataset. We follow the training, validation, and evaluation splits provided by Blackbird. Since the evaluation split includes both trajectories that overlap with the training set and additional sequences, we report results only on those evaluation trajectories that do not appear in training.

B.2 EuRoC MAV

We also evaluate on the EuRoC MAV dataset Burri *et al.* [2016]. The dataset consists of indoor MAV flight sequences with synchronized IMU measurements at 200 Hz and corresponding ground-truth trajectories.[5] All experiments follow the official data processing procedure and sequence partitioning provided with the dataset. For the `V1_01` sequence, we use the reprocessed orientation estimates released with OpenVINS Geneva *et al.* [2020].

C Evaluation Protocol

C.1 Dataset-wise Training and Evaluation

PHIONet is trained and evaluated on Blackbird and EuRoC separately. The same network architecture is used for both datasets, and the evaluation metrics are kept identical. Training, validation, and evaluation follow the dataset splits described above. In practice, the network operates directly on IMU measurements. Ground-truth trajectories are not used as inputs and are only accessed during training and evaluation.

C.2 Trajectory Reconstruction and Evaluation

At evaluate time, the network is run over each sequence to produce body-frame velocity estimates. These velocities are transformed to the world frame using the orientation at each time step. The resulting velocities are then integrated to obtain the trajectory. This evaluation is performed in an open-loop setting. As a result, drift accumulates over time. The same procedure is applied to all methods to ensure a fair comparison.

D Implementation Details

This section provides the detailed training settings and hyperparameters used in our experiments. The same optimization configuration is used for both Blackbird and EuRoC unless otherwise specified. Dataset-specific differences mainly come from the loss weights and the PHSLoss window size, which are selected according to the temporal characteristics and training stability of each dataset.

[4] `https://github.com/mit-aera/Blackbird-Dataset`

[5] `https://ethz-asl.github.io/datasets`

Table 3: Training settings and hyperparameters.

Setting	Blackbird	EuRoC
Window length W	1000 IMU frames	
Training step size	3 frames	
Validation step size	10 frames	
Optimizer	Adam	
Learning rate	1×10^{-4}	
Weight decay	1×10^{-4}	
Learning rate scheduler	ReduceLROnPlateau	
Scheduler factor	0.5	
Scheduler patience	5	
Minimum learning rate	1×10^{-6}	
Numerical precision	float64	
Air-IO loss weight $\lambda_{\mathrm{Air-IO}}$	100.0	100.0
AirIMU loss weight $\lambda_{\mathrm{AirIMU}}$	1.0	0.1
PHS loss weight λ_{PHS}	0.5	0.1
PHS window size	1	3

E Compute Resources and Onboard Suitability

We report the computing environment, inference complexity, and deployment-oriented runtime characteristics of PHIONet. This information is provided to make the computational requirements of the experiments transparent and to clarify whether the proposed network is suitable for modern UAV platforms.

E.1 Compute environment

All experiments are conducted on a local workstation with a single GPU. The hardware and software environment is summarized in Table 4. Training and evaluation are performed in double precision, following the numerical setting used by the AirIMU/Air-IO codebase.

Table 4: Compute environment used for training and evaluation.

Item	**Specification**
Workstation	Local desktop workstation
Operating system	Windows 10, 64-bit
CPU	Intel Core i9-14900K
CPU cores / threads	24 cores / 32 logical processors
System memory	128 GB
GPU	NVIDIA GeForce RTX 5070 Ti
GPU memory	16 GB
CUDA / PyTorch CUDA	CUDA 12.8 in PyTorch
Python	3.10.18
PyTorch	2.10.0.dev20251021+cu128
PyPose	0.6.8
Numerical precision	FP64 for training and evaluation

E.2 Inference complexity

We benchmark PHIONet on a single sliding window of length $T = 1000$, corresponding to 10 seconds of IMU data at 100 Hz on Blackbird. The benchmark measures only the network forward pass and excludes data loading, trajectory reconstruction, optional filtering, and file I/O. The results are summarized in Table 5.

Algorithm 1 Window Construction and Residual Computation

Require: Corrected IMU $(\hat{\mathbf{a}}^B, \hat{\boldsymbol{\omega}}^B) \in \mathbb{R}^{T\times 3}$, timestamps $\mathbf{t}^{\text{imu}} \in \mathbb{R}^T$
Require: Predicted velocity $\hat{\mathbf{v}}^B \in \mathbb{R}^{L\times 3}$, timestamps $\mathbf{t}^{\text{vel}} \in \mathbb{R}^L$
Require: Rotation sequence $\mathbf{R} \in SO(3)^T$, window size M, gravity $\mathbf{g}^W$
Ensure: Residual set $\mathcal{R}$

$\mathcal{R} \leftarrow \emptyset$
for $i = 1$ to $L - M$ **do**
  $j \leftarrow i + M$ ▷ Velocity window endpoints
  $t_i \leftarrow t_i^{\text{vel}}, \quad t_j \leftarrow t_j^{\text{vel}}$
  $\hat{\mathbf{v}}_i^B \leftarrow \hat{\mathbf{v}}^B[i], \quad \hat{\mathbf{v}}_j^B \leftarrow \hat{\mathbf{v}}^B[j]$
  Step 1: Map to IMU indices via nearest neighbor
  $s \leftarrow \text{align}(t_i), \quad e \leftarrow \text{align}(t_j)$
  if $e \le s$ **then**
    continue ▷ Invalid window
  end if
  Step 2: Extract IMU subsequence
  $\hat{\mathbf{a}}_{s:e}^B \leftarrow \hat{\mathbf{a}}^B[s:e]$
  $\hat{\boldsymbol{\omega}}_{s:e}^B \leftarrow \hat{\boldsymbol{\omega}}^B[s:e]$
  $\Delta t_k \leftarrow t_{k+1}^{\text{imu}} - t_k^{\text{imu}}, k \in [s, e)$
  Step 3: Compute cumulative rotations
  $\Delta\mathbf{R}_s \leftarrow \mathbf{I}$
  for $k = s$ to $e - 1$ **do**
    $\Delta\mathbf{R}_{k+1} \leftarrow \Delta\mathbf{R}_k \cdot \text{Exp}(\hat{\boldsymbol{\omega}}_k^B \Delta t_k)$
  end for
  Step 4: Compute gravity in body frame
  $\mathbf{g}_i^B \leftarrow \mathbf{R}[s]^\top \mathbf{g}^W$
  Step 5: Compute energy residual
  $\bar{\mathbf{v}}^B \leftarrow \frac{1}{2}(\hat{\mathbf{v}}_i^B + \hat{\mathbf{v}}_j^B)$
  $\Delta H_{ij} \leftarrow (\bar{\mathbf{v}}^B)^\top (\hat{\mathbf{v}}_j^B - \hat{\mathbf{v}}_i^B)$
  $\Delta E_{ij} \leftarrow (\bar{\mathbf{v}}^B)^\top \sum_{k=s}^{e-1} (\Delta\mathbf{R}_k \hat{\mathbf{a}}_k^B + \mathbf{g}_i^B)\Delta t_k$
  $r_{ij} \leftarrow \Delta H_{ij} - \Delta E_{ij}$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{r_{ij}\}$
end for
return $\mathcal{R}$

E.3 Suitability for modern UAV platforms

PHIONet is lightweight relative to the temporal span of its input. A single $T = 1000$ window corresponds to 10 seconds of IMU data, while the measured GPU forward latency is 6.89 ms. The forward pass therefore occupies less than 0.1% of the input duration:

$$\frac{6.89 \text{ ms}}{10 \text{ s}} = 6.89 \times 10^{-4}.$$

Even on CPU, the measured forward latency is 29.01 ms for the same 10-second window. This indicates that the neural network itself is computationally lightweight and compatible with real-time or near-real-time execution on modern onboard computing platforms.

The reported latency measures the PHIONet forward pass rather than an end-to-end online navigation system. In our experiments, trajectory evaluation is conducted offline by processing each sequence as a whole and reconstructing the trajectory from the predicted velocities. Nevertheless, the single-window benchmark shows that the learned module itself is lightweight: one $T = 1000$ window corresponds to 10 seconds of IMU data, while the GPU forward latency is 6.89 ms. This suggests that PHIONet is computationally suitable for modern UAV companion computers, although a fully online deployment would additionally require sensor buffering, timestamp synchronization, streaming inference, trajectory integration, and optional filtering.

Table 5: Single-window inference complexity of PHIONet. The benchmark uses one $T = 1000$ IMU window.

Metric	Value
Input window length	1000 IMU frames
Equivalent input duration	10 s at 100 Hz
Output velocity tokens	112
Parameter size	13.0 MB in FP64
Forward complexity	0.338 GFLOPs / window
GPU latency	6.89 ms / window
GPU throughput	145.25 windows/s
CPU latency	29.01 ms / window
CPU throughput	34.48 windows/s

F Ablation Variant Details

This section provides implementation details for the ablated variants used in Table 2. All variants follow the same data split, training schedule, optimizer, checkpoint selection, and evaluation protocol as PHIONet. Unless otherwise specified, the Input Port and State Port are kept unchanged.

F.1 Concat-MLP

This variant keeps PHSLoss unchanged but replaces DCM with a residual Concat-MLP fusion module. The State-Port feature is linearly interpolated to match the temporal length of the Input-Port feature when their temporal resolutions differ:

$$\tilde{\mathbf{f}}_{io} = \mathrm{Interp}(\mathbf{f}_{io}, L_{imu}),$$

where L_{imu} denotes the temporal length of $\mathbf{f}_{imu}$. In practice, this interpolation acts as a safeguard because the joint model usually aligns the two feature streams before fusion. The fused feature is computed as

$$\mathbf{f}_{\mathrm{fused}} = \mathbf{f}_{imu} + \alpha\, \phi_{\mathrm{mlp}} \left([\mathbf{f}_{imu}, \tilde{\mathbf{f}}_{io}] \right),$$

where α is a learnable scalar residual parameter without sigmoid or tanh constraint. This variant examines whether generic residual feature fusion is sufficient to replace the proposed DCM.

F.2 Cross-Attention

This variant keeps PHSLoss and the directed State-to-Input cross-attention structure, including the local band attention mask, but removes the element-wise velocity-conditioned gate in DCM. The attention residual $\Delta\mathbf{f}$ is computed as in Eq. (18), while the gated fusion in Eq. (19) is replaced by an ungated residual fusion:

$$\mathbf{f}_{\mathrm{fused}} = \mathbf{f}_{imu} + \alpha\Delta\mathbf{f},$$

where α is a learnable scalar residual parameter. This variant examines the role of the element-wise gate while preserving the directed cross-attention pathway.

F.3 Smoothness

This variant keeps the DCM architecture unchanged but replaces PHSLoss with an acceleration-form velocity-smoothness regularizer. Specifically, it penalizes the squared finite-difference acceleration of the predicted body-frame velocity:

$$\mathcal{L}_{\mathrm{smooth}} = \frac{1}{L-1} \sum_{t=1}^{L-1} \left\| \frac{\hat{\mathbf{v}}^B_{t+1} - \hat{\mathbf{v}}^B_t}{t_{t+1} - t_t} \right\|_2^2.$$

In our implementation, this smoothness term is weighted by 0.01. This variant examines whether a simple acceleration-form smoothing prior can substitute for the proposed physical consistency objective.

F.4 Acceleration

This variant keeps DCM unchanged but replaces PHSLoss with a time-normalized version of the same window-level residual. For each finite window (i, j), we reuse the same timestamp alignment, preintegration, rotation handling, and gravity compensation as PHSLoss. Let $\Delta\mathbf{v}_{ij}^{pre}$ denote the velocity increment obtained from the corrected IMU over the aligned interval, and let

$$\Delta\mathbf{v}_{ij}^{state} = \Delta\mathbf{R}_{ij}\hat{\mathbf{v}}_j^B - \hat{\mathbf{v}}_i^B - \mathbf{g}_i^B(t_j - t_i)$$

denote the corresponding state-side velocity increment. The replacement residual is

$$\mathbf{r}_{ij}^{acc} = \frac{\Delta\mathbf{v}_{ij}^{pre} - \Delta\mathbf{v}_{ij}^{state}}{t_j - t_i},$$

and the loss is

$$\mathcal{L}_{\text{acc}} = \frac{1}{|\mathcal{W}|}\sum_{(i,j)\in\mathcal{W}} \rho_\delta\left(\left\|\mathbf{r}_{ij}^{acc}\right\|_2\right).$$

This variant examines whether a time-normalized vector consistency objective over the same finite windows can replace the proposed PHSLoss objective.

F.5 Summary

The Concat-MLP and Cross-Attention variants replace or modify DCM while keeping PHSLoss fixed. The Smoothness and Acceleration variants replace PHSLoss while keeping DCM fixed. Together, these ablations clarify the empirical contribution of PHSLoss and DCM under a unified training and evaluation protocol.

G Mechanistic Case Study on Work–Energy Alignment and Learned Dissipation

G.1 Diagnostic Sequence and Residual Definition

We provide a diagnostic case study to inspect whether PHIONet produces physically interpretable corrections in addition to improving trajectory accuracy. This appendix is intended as a within-sequence mechanism audit rather than dataset-wide statistical evidence of residual reduction. We use the Blackbird Oval sequence because its closed-loop motion covers a broad and smooth speed range, making finite-window work–energy alignment and velocity-dependent dissipative correction visually inspectable.

Following (20), we compute the finite-window work–energy residual

$$r_{ij} = \Delta H_{ij} - \Delta E_{ij},$$

where ΔH_{ij} denotes the kinetic-energy variation induced by the predicted body-frame velocity, and ΔE_{ij} denotes the accumulated work from corrected inertial excitation and gravity over the same window. A smaller $|r_{ij}|$ indicates better agreement between the predicted velocity evolution and the corrected IMU excitation under the proposed finite-window approximation.

G.2 Finite-Window Work–Energy Alignment

Figure 2 compares Air-IO and PHIONet on the full Oval sequence. Both methods use the same corrected IMU excitation, so the comparison focuses on whether the predicted body-frame velocity agrees with the accumulated inertial work. PHIONet reduces the mean absolute residual from 2.133 to 1.909, while the median residual decreases from 1.926 to 1.427, corresponding to a **25.88% reduction**. This indicates a lower typical finite-window work–energy mismatch on this diagnostic sequence. The curves are not uniformly ordered, which is expected because PHSLoss is used as a soft regularizer within a multi-objective loss rather than as a hard physical constraint.

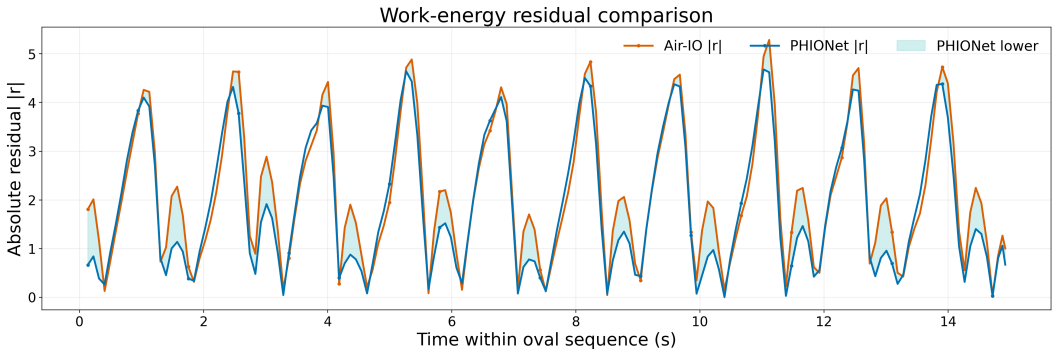

Figure 2: Work–energy residual comparison on the Blackbird Oval sequence. The curves show the absolute finite-window residual $|r_{ij}|$ over time, with shaded regions marking windows where PHIONet is lower than Air-IO.

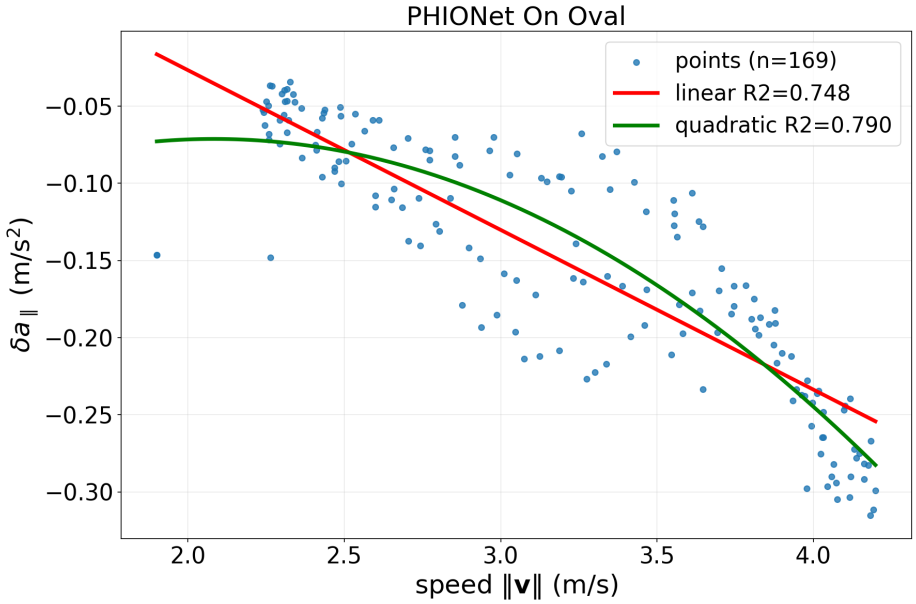

Figure 3: PHIONet case study on Blackbird `Oval`: speed $\|\mathbf{v}^B\|$ vs. the projection of the learned acceleration correction $\delta\mathbf{a}$ onto the velocity direction.

G.3 Learned Dissipative Correction

Following Eq. (15), the learned dissipative correction $\delta\mathbf{a}^B$ is intended to capture the dissipation effect in the work–energy balance. Since dissipation removes energy, we expect $(\mathbf{v}^B)^\top \delta\mathbf{a}^B \leq 0$. Accordingly, we examine the scalar projection:

$$\delta a_{\|} \triangleq (\delta\mathbf{a}^B)^\top \frac{\mathbf{v}^B}{\|\mathbf{v}^B\|}. \tag{24}$$

Furthermore, aerodynamic drag induces a speed-dependent effect rather than a speed-independent offset; thus the magnitude of this negative projection is expected to increase with $\|\mathbf{v}^B\|$, leading to an approximately linear or quadratic trend.

Figure 3 shows the results obtained by PHIONet on the `Oval` sequence of the Blackbird dataset. The projection $\delta a_{\|}$ is predominantly negative and decreases with increasing speed, matching the dissipative Port Hamiltonian expectation for a drag-like term. A simple polynomial fitting result shows that the coefficient of determination is $R^2 = 0.748$ for the linear model and $R^2 = 0.790$ for

the quadratic model. The quadratic fit is given by

$$\delta a_{\parallel} \approx -0.0473 \, \|\mathbf{v}^B\|^2 + 0.1970 \, \|\mathbf{v}^B\| - 0.2764, \tag{25}$$

where acceleration is measured in m/s^2 and velocity in m/s.

The results show that PHIONet learns a drag-like correction that does negative work along the motion direction and grows in magnitude with speed.

G.4 Interpretation

This case study is intended to make the learned physical mechanism observable. The residual time-series in Fig. 2 shows that PHIONet reduces the typical finite-window work–energy mismatch on Oval, while Fig. 3 shows that the learned correction has a velocity-aligned, drag-like dissipative pattern. Together, these results provide qualitative evidence that the proposed DCM and PHSLoss do not merely act as generic architectural modifications, but can induce physically interpretable correction behavior.