DiffusionIMU: Diffusion-Based Inertial Navigation with Iterative Motion Refinement

Xiaoqiang Teng[1] , **Chenyang Li**[2] , **Shibiao Xu**[2*] , **Zhihao Hao**[1] , **Deke Guo**[3] , **Jingyuan Li**[1] , **Haisheng Li**[1*] , **Weiliang Meng**[4,5] , **Xiaopeng Zhang**[4,5] ,
[1]School of Computer and Artificial Intelligence, Beijing Technology and Business University, China
[2]School of Artificial Intelligence, Beijing University of Posts and Telecommunications, China
[3]School of Computer, Sun Yat-sen University, China
[4]State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, China
[5]School of Artificial Intelligence, University of Chinese Academy of Sciences, China
{xiaoqiangteng, lihsh, li.jingyuan.jerry}@btbu.edu.cn, {2024010495, shibiaoxu}@bupt.edu.cn,
hao.zhihao@connect.um.edu.mo, guodk@mail.sysu.edu.cn, {weiliang.meng, xiaopeng.zhang}@ia.ac.cn

Abstract

Inertial navigation enables self-contained localization using only Inertial Measurement Units (IMUs), making it widely applicable in various domains such as navigation, augmented reality, and robotics. However, existing methods suffer from drift accumulation due to the sensor noise and difficulty capturing long-range temporal dependencies, limiting their robustness and accuracy. To address these challenges, we propose DiffusionIMU, a novel diffusion-based framework for inertial navigation. DiffusionIMU enhances direct velocity regression from IMU data through an iterative generative denoising process, progressively refining motion state estimation. It integrates the noise-adaptive feature modulation for sensor variability handling, the feature alignment mechanism for representation consistency, and the diffusion-based temporal modeling to decrease accumulated drift. Experiments show that DiffusionIMU consistently outperforms existing methods, demonstrating superior generalization to unseen users while alleviating the impact of the sensor noise.

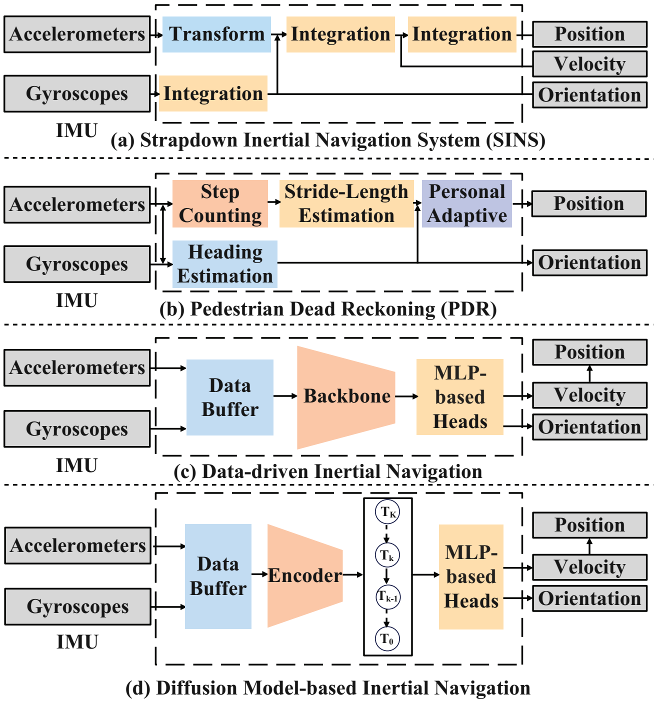


Figure 1: Main approaches in inertial navigation: (a) SINS; (b) PDR; (c) data-driven approaches; (d) proposed DiffusionIMU.

1 Introduction

Over the last decade, inertial navigation has emerged as a promising approach for achieving ubiquitous localization using only Inertial Measurement Units (IMUs). This technology has enabled diverse applications, including indoor positioning [Teng *et al.*, 2019], augmented reality [Hu *et al.*, 2023], and robotics [Campos *et al.*, 2021]. By integrating data from accelerometers and gyroscopes, inertial navigation offers a cost-effective, energy-efficient, and universally accessible solution easily embedded in modern mobile devices. Gyroscopes capture rotational motion, while accelerometers measure linear acceleration after compensating for gravitational effects. These measurements are sequentially processed to estimate velocities, which are subsequently integrated to determine positions.

Currently, inertial navigation approaches can be generally classified into three categories as depicted in Figure 1: (a) the Strapdown Inertial Navigation System (SINS), (b) the Pedestrian Dead Reckoning (PDR), and (c) the data-driven inertial navigation. SINS computes velocity and position through direct integration of IMU data [Chang *et al.*, 2016]. Al-

[*] Corresponding author.

though computationally straightforward, this method suffers from exponential drift caused by error propagation through the double integration process. Such drift makes the SINS impractical for prolonged navigation tasks, as slight sensor noise leads to significant positional inaccuracies over time [Teng *et al.*, 2020]. PDR leverages domain-specific knowledge of human walking dynamics [Wang *et al.*, 2012; Teng *et al.*, 2017; Wang *et al.*, 2024]. By integrating features like step counts, stride lengths, and heading direction, the PDR significantly mitigates integration errors compared to the SINS. However, it is still vulnerable to stride length and heading estimation inaccuracies, especially under irregular motion patterns or diverse walking environments [Teng *et al.*, 2020; Yan *et al.*, 2018]. These limitations restrict the applicability of the PDR in complex scenarios [Chen *et al.*, 2018; Rao *et al.*, 2022].

The data-driven approaches leverage deep learning to derive accurate motion estimates directly from raw IMU data [Herath *et al.*, 2020]. Models such as IONet [Chen *et al.*, 2018], RIDI [Yan *et al.*, 2018], RoNIN [Herath *et al.*, 2020], and CTIN [Rao *et al.*, 2022] have showcased their capacity to predict velocities and localize trajectories effectively by learning motion dynamics from extensive datasets. These approaches are less reliant on manual feature extraction and can adapt to diverse patterns of motion [Herath *et al.*, 2020; Teng *et al.*, 2020]. Most of these models are built upon architectures such as ResNet, LSTM [Herath *et al.*, 2020], and Transformer [Rao *et al.*, 2022], which provide robust feature extraction and temporal modeling capabilities. However, these architectures face notable limitations. ResNet-based models struggle to capture long-range temporal dependencies, which are crucial for inertial navigation [Rao *et al.*, 2022]. LSTM-based models, while practical for sequential data, suffer from vanishing gradients and may fail to maintain long-term dependencies in extended sequences [Rao *et al.*, 2022]. Transformer-based methods, though powerful, can be sensitive to noisy IMU signals due to their global attention mechanisms [Rao *et al.*, 2022]. Additionally, existing models fail to exploit the spatial-temporal correlations in IMU data fully.

In response to the above observations and concerns, we propose **DiffusionIMU**, a novel diffusion-based framework for inertial navigation with iterative motion refinement. DiffusionIMU leverages a generative diffusion process to iteratively refine motion state estimation, addressing key challenges such as sensor noise, drift accumulation, and temporal dependency modeling. As shown in Figure 1(d), progressively refines motion representations over multiple steps, effectively correcting accumulated errors and modeling long-range temporal dependencies. To improve robustness and generalization, our design incorporates mechanisms that adaptively modulate feature representations and align semantic structures throughout the refinement process.

Experiments demonstrate that our DiffusionIMU outperforms the state-of-the-art models. In summary, our main contributions are as follows:

- We introduce DiffusionIMU, a novel diffusion-based framework for inertial navigation that refines motion state estimation through an iterative generative process, improving robustness against the sensor noise and drifts. To the best of our knowledge, DiffusionIMU is the first diffusion-based model for inertial navigation.
- A noise-adaptive feature modulation and a feature alignment mechanism are proposed to adjust feature representations dynamically, enhancing the model's ability to handle diverse sensor characteristics and environmental variations.
- A diffusion-based temporal modeling module is presented, which enables effective sequential refinement of velocity predictions, leading to significant improvements in trajectory accuracy. A multi-task loss is proposed to improve stability and uncertainty estimation.
- Comprehensive qualitative and quantitative comparisons with the existing baselines indicate that DiffusionIMU outperforms the state-of-the-art models.

2 Related Work

This section provides an overview of the key approaches in inertial navigation.

2.1 Strapdown Inertial Navigation System (SINS)

The SINS estimates position and orientation by integrating acceleration and angular velocity measurements. While high-precision IMUs minimize integration errors, consumer-grade IMUs suffer from significant drift due to noise and accumulation errors [Geiger *et al.*, 2012]. SINS has been combined with visual-inertial odometry to mitigate drift, leveraging visual data for correction [Campos *et al.*, 2021]. Zero-velocity updates further reduce errors by detecting stationary states and applying periodic corrections [Skog *et al.*, 2010]. However, low-cost IMUs remain prone to noise and environmental disturbances, posing challenges for accurate localization [Chen *et al.*, 2018].

2.2 Pedestrian Dead Reckoning (PDR)

The PDR estimates positions through step detection, stride length estimation, and orientation tracking [Wang *et al.*, 2012; Teng *et al.*, 2017; Laidig and Seel, 2023]. Step events are identified via accelerometer or gyroscope data using the peak or zero-crossing detection [Brajdic and Harle, 2013; de Silva *et al.*, 2018]. Distance is computed by multiplying step count by stride length, which is often modeled using walking frequency and acceleration variance [Weinberg, 2002]. Orientation estimation relies on gyroscope data to track directional changes [Madgwick *et al.*, 2011; Laidig and Seel, 2023]. PDR faces challenges such as cumulative errors, especially in orientation estimation, leading to drift and reduced accuracy [Chen *et al.*, 2021]. ZUPTs help mitigate drift by applying stationary state constraints [Skog *et al.*, 2010; Laidig and Seel, 2023]. Multi-modal sensor fusion further enhances PDR; for example, Walkie-Markie uses Wi-Fi signals for recalibration [Shen *et al.*, 2013], while Sextant integrates visual references for localization [Gao *et al.*, 2014]. However, reliance on heuristic models and sensitivity to cumulative errors remain limitations for real-world deployment.

2.3 Data-driven Inertial Navigation

Recent deep learning advancements enable motion estimation directly from raw IMU data [Yan *et al.*, 2018; Chen *et al.*, 2018; Rao *et al.*, 2022]. Neural networks capture temporal dependencies for improved adaptability. RIDI [Yan *et al.*, 2018] predicts trajectories via velocity regression, while IONet [Chen *et al.*, 2018] uses RNNs for inertial odometry. RoNIN [Herath *et al.*, 2020] enhances heading and position estimation with convolutional and recurrent layers, and CTIN [Rao *et al.*, 2022] employs Transformers for velocity regression. Hybrid models like TLIO [Liu *et al.*, 2020] and IDOL [Sun *et al.*, 2021] integrate EKFs for state estimation. However, these methods face some challenges. Velocity integration errors cause drift [Rao *et al.*, 2022], model robustness is sensitive to sensor variations, and many frameworks underutilize IMU's spatial-temporal correlations. These limitations drive research toward architectures addressing drift, sensor variability, and feature learning.

In this paper, we introduce DiffusionIMU, which leverages diffusion models [Yang *et al.*, 2024] to enhance IMU-based localization. By integrating diffusion-based probabilistic modeling, DiffusionIMU improves robustness against sensor noise, mitigates trajectory drift, and better captures complex motion dynamics.

3 Preliminaries

IMU Modeling. Inertial navigation systems rely on Newtonian mechanics to determine an object's position and orientation using initial pose information and IMU data [Chen *et al.*, 2018]. An IMU consists of a three-axis accelerometer and a three-axis gyroscope, which respectively measure acceleration ($\hat{\alpha}$) and angular velocity ($\hat{\omega}$). However, these measurements are affected by sensor biases and noise, leading to inaccuracies. Mathematically, the outputs of the accelerometer and gyroscope can be expressed as follows.

$$\hat{\alpha}_t = \alpha_t + b_t^\alpha + n_t^\alpha, \tag{1}$$

$$\hat{\omega}_t = \omega_t + b_t^\omega + n_t^\omega, \tag{2}$$

where α_t and ω_t represent the true acceleration and angular velocity at timestamp t, as measured by the accelerometer and gyroscope, respectively. The terms b_t^α and b_t^ω denote the inherent biases in the accelerometer and gyroscope readings at timestamp t. Additionally, n_t^α and n_t^ω correspond to measurement noise, which typically follows a zero-mean Gaussian distribution [Yan *et al.*, 2018].

Inertial Navigation. In inertial navigation, the input consists of IMU readings, while the output includes position (P), velocity (V), and orientation (represented by the rotation matrix R) within a reference frame (n), as defined below [Rao *et al.*, 2022].

$$R_b^n(t) = R_b^n(t-1) \cdot \exp(\frac{dt}{2}\hat{\omega}(t-1)), \tag{3}$$

$$V^n(t) = V^n(t-1) + (R_b^n(t-1) \odot \hat{\alpha}(t-1) - g^n)dt, \tag{4}$$

$$P^n(t) = P^n(t-1) + V^n(t-1)dt, \tag{5}$$

where, $R_b^n(t)$ denotes the rotation matrix transforming coordinates from the body (b) frame to the navigation (n) frame at timestamp t. It is updated using the angular velocity $\hat{\omega}(t-1)$ in the body frame, along with the previous rotation matrix $R_b^n(t-1)$. The navigation frame is typically defined such that its Z-axis aligns with Earth's gravity, while the other two axes correspond to the initial orientation of the body frame. Velocity $V^n(t)$ is updated based on its temporal change $\Delta(t)$, which is computed by rotating the acceleration reading $\hat{\alpha}(t-1)$ into the navigation frame using $R_b^n(t-1)$ and subtracting the gravitational force g^n. Finally, position $P^n(t)$ is obtained by integrating velocity $V^n(t)$.

Diffusion Model. Diffusion models are generative models that learn data distributions by gradually perturbing and denoising samples [Yang *et al.*, 2024]. They operate through a forward process, which adds noise to the data, and a reverse process, which learns to reconstruct the original distribution [Ho *et al.*, 2020]. Given a data sample $x_0 \sim q(x)$, the forward process is defined as a Markovian sequence:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{6}$$

where β_t is a predefined noise schedule. The noised sample at step t can be directly obtained as:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{7}$$

where $\bar{\alpha}_t = \prod_{s=1}^t (1-\beta_s)$ and $\epsilon \sim \mathcal{N}(0, I)$. The reverse process seeks to approximate the true denoising distribution $q(x_{t-1}|x_t)$ by learning a parameterized model $p_\theta(x_{t-1}|x_t)$:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{8}$$

To train the model, a noise prediction network estimates ϵ in a variational framework, minimizing the objective [Yang *et al.*, 2024]:

$$L = E_{x_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]. \tag{9}$$

4 Methodology

4.1 Problem Definition

The insights of data-driven inertial navigation provide a paradigm shift from purely integrative methods to modeling inertial navigation as a **sequence-to-sequence prediction problem** as follows [Rao *et al.*, 2022].

$$V_t^n = f([R_b^n, \hat{\alpha}, \hat{\omega}]^{t-m:t}), \tag{10}$$

where $f(\cdot)$ denotes a latent neural function that maps a sequence of inertial measurements to velocities in the navigation frame. The input consists of a sliding window of rotation matrices R_b^n, accelerometer readings $\hat{\alpha}$, and gyroscope readings $\hat{\omega}$ from time $t-m$ to t. The position trajectory is then obtained by numerically integrating the predicted velocities over time, as defined in Eq. (5).

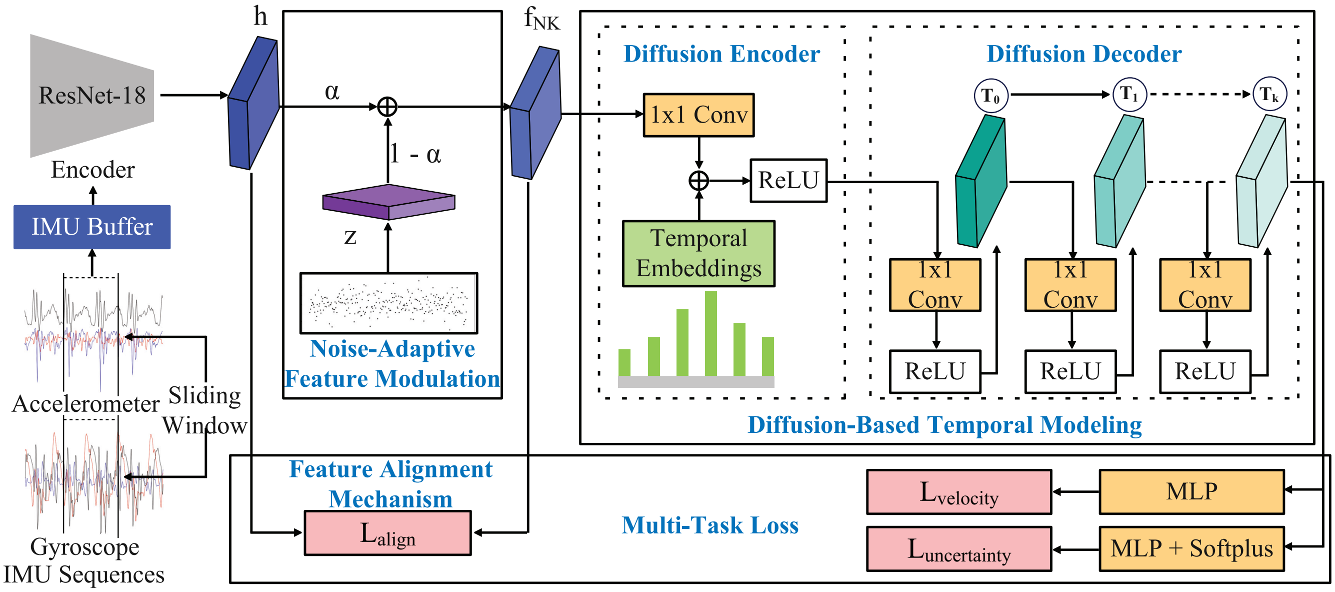


Figure 2: Main architecture of DiffusionIMU framework. The system consists of five key components: (1) Encoder, which utilizes a ResNet-18 backbone to extract spatial features from IMU sequences; (2) Noise-Adaptive Feature Modulation, where a learnable noise kernel adjusts the contribution of injected noise z through a weighting mechanism α; (3) Feature Alignment Mechanism, ensuring consistency between encoder outputs and the diffusion pipeline; (4) Diffusion-Based Temporal Modeling, where a diffusion encoder applies temporal embeddings followed by a diffusion decoder that iteratively refines motion representations; and (5) Multi-Task Loss, which predicts velocity and uncertainty through MLP layers and a Softplus activation function.

4.2 Overview

The primary challenge in inertial navigation lies in mitigating the sensor noise and effectively modeling temporal dependencies. Existing methods primarily rely on ResNet [Herath *et al.*, 2020], LSTM [Herath *et al.*, 2020], or Transformer architectures [Rao *et al.*, 2022] in isolation, limiting their ability to address these challenges comprehensively. To address this, we propose DiffusionIMU, a novel diffusion-based framework designed for robust inertial navigation, as illustrated in Figure 2. It consists of five interconnected modules: (1) **Encoder**: a ResNet-based feature extractor to capture spatial representations from IMU signals [He *et al.*, 2015], (2) **Noise-Adaptive Feature Modulation**: a learnable Noise Kernel to adaptively model sensor noise variations, (3) **Feature Alignment Mechanism**: a lightweight feature alignment mechanism that ensures consistency between the encoder and the diffusion pipeline, (4) **Diffusion-Based Temporal Modeling**: a diffusion-based encoder-decoder to refine motion state estimation through iterative denoising, and (5) **Multi-Task Loss**: a multi-task output module for simultaneous velocity prediction and uncertainty estimation.

The overall process is formulated as follows:

$$\mathbf{y}, \mathbf{u}, L_{\text{align}} = \text{DiffDe}(\text{DiffEn}(\text{NK}(\text{En}(\mathbf{x}), \mathbf{z}), \mathbf{t})), \quad (11)$$

where $\mathbf{x}$ is the input IMU data, $\mathbf{z}$ represents injected noise, and $\mathbf{t}$ denotes temporal embeddings, which encode timestep information to guide the diffusion process across different denoising steps. $\mathbf{y}$ and $\mathbf{u}$ are velocity and uncertainty predictions, and $\mathcal{L}_{\text{align}}$ is the alignment loss. DiffDe and DiffEn are the diffusion-based decoder and encoder, respectively. NK is the Noise Kernel, and EN is the Encoder.

4.3 Encoder

The encoder module extracts spatial features from IMU data while preserving motion dependencies. It processes an input sequence $\mathbf{x}_{1:m}$ into latent representations $\mathbf{h} = (h_1, \dots, h_m)$. The modified ResNet-18 bottleneck block [He *et al.*, 2015] is used by replacing spatial convolutions with local self-attention and adding a global self-attention module before the final 1×1 downsampling convolution. The modified bottleneck layer is repeated multiple times to form the encoder, with the output of one block being the input of the next.

4.4 Noise-Adaptive Feature Modulation

IMU data is inherently noisy due to environmental disturbances, sensor drift, and device variations. To address this, we introduce a *learnable noise kernel* that dynamically adjusts the contribution of injected noise in feature processing. The noise integration is defined as:

$$\mathbf{f}_{NK} = \alpha \mathbf{h} + (1 - \alpha)\mathbf{z}, \quad (12)$$

where $\mathbf{h}$ and $\mathbf{f}_{NK}$ are the extracted and noise-modulated features, respectively. α is a learnable parameter initialized. The injected noise $\mathbf{z}$ follows a Gaussian distribution to introduce controlled randomness, improving model robustness against sensor variability, i.e., $\mathbf{z} \sim \mathcal{N}(0, 1)$.

4.5 Feature Alignment Mechanism

Introducing noise in the diffusion process can distort the spatial representations extracted by the encoder, as the added noise perturbs feature distributions, altering the learned motion dynamics and degrading the quality of velocity estimation. This perturbation can lead to inconsistencies between the encoded feature space and the denoising steps, making it difficult for the diffusion model to reconstruct accurate motion trajectories. To mitigate this issue and preserve spatial information extracted by the modified ResNet-18, we employ a lightweight feature alignment mechanism that ensures consistency between the encoder and the diffusion pipeline. This alignment is enforced via an ℓ_2 loss:

$$L_{\text{align}} = \|\mathbf{W}\mathbf{h} - \mathbf{f}_{NK}\|_2^2, \tag{13}$$

where $\mathbf{W}$ is a learnable transformation matrix. This constraint ensures that the diffusion module operates on a well-aligned latent space, improving reconstruction and velocity estimation accuracy.

4.6 Diffusion-Based Temporal Modeling

Diffusion Encoder. Our model employs a *diffusion process* to refine motion representations through an iterative motion refinement. Given an initial feature representation $\mathbf{f}_{NK}$, the diffusion encoder applies temporal embeddings and a transformation function ϕ:

$$\mathbf{f}_{\text{latent}} = \phi(\mathbf{W}_e \mathbf{f}_{NK} + \mathbf{t}), \tag{14}$$

where $\mathbf{W}_e$ is a trainable encoding matrix, $\mathbf{t}$ represents learned temporal embeddings, and ϕ is a non-linear activation function. To generate the temporal embedding $\mathbf{t}$, we first sample a discrete timestep t from a uniform distribution:

$$t \sim \text{Uniform}(0, T-1), \tag{15}$$

where $T = 100$ represents the total number of discrete diffusion steps. The embedding $\mathbf{t}$ is then obtained via a trainable embedding layer:

$$\mathbf{t} = \mathbf{E}(t), \tag{16}$$

where $\mathbf{E} \in \mathbb{R}^{T \times D}$ is the trainable temporal embedding matrix, and D is the hidden dimension. This embedding mechanism allows the model to condition its predictions on different noise levels, enabling more effective motion refinement. For example, when $t = 40$, the model uses the 40-th row of $\mathbf{E}$, i.e., $\mathbf{E}[40]$, as the temporal context vector. This vector encodes the current denoising stage and modulates the transformation accordingly.

Diffusion Decoder. The diffusion decoder iteratively refines the latent feature representation $\mathbf{f}_{\text{latent}}$ to improve velocity estimation and mitigate drift accumulation. Instead of a single-step transformation, it applies a multi-step refinement process, where each step progressively updates the feature representation.

At each step i, the decoder applies a linear transformation followed by a non-linear activation function:

$$\mathbf{f}_{\text{decoded}}^{(i)} = \phi(\mathbf{W}_{d,i} \mathbf{f}_{\text{decoded}}^{(i-1)} + \mathbf{b}_i), \tag{17}$$

where $\mathbf{W}_{d,i}$ is the transformation weight matrix, $\mathbf{b}_i$ is the bias term, and $\phi(\cdot)$ is the activation function (e.g., ReLU). The initial representation is set as follows:

$$\mathbf{f}_{\text{decoded}}^{(0)} = \mathbf{f}_{\text{latent}}. \tag{18}$$

After K steps, the final refined feature representation is obtained as:

$$\mathbf{f}_{\text{decoded}} = \mathbf{f}_{\text{decoded}}^{(K)}. \tag{19}$$

This module ensures that the motion representation is iteratively refined before being passed to the final velocity estimation module, improving the stability and accuracy of the predictions.

4.7 Multi-Task Loss

The final output module predicts velocity and uncertainty, improving robustness in real-world deployments. The predictions are formulated as follows:

$$\mathbf{y} = \mathbf{W}_y \mathbf{f}_{\text{decoded}}, \quad \mathbf{u} = \text{Softplus}(\mathbf{W}_u \mathbf{f}_{\text{decoded}}), \tag{20}$$

where $\mathbf{y}$ is the predicted velocity, $\mathbf{u}$ is the estimated uncertainty, and $\mathbf{W}_y$ and $\mathbf{W}_u$ are trainable weight matrices.

The velocity loss is formulated as a Mean Squared Error (MSE) between the predicted velocity $\hat{\mathbf{y}}$ and the ground-truth velocity $\mathbf{y}$:

$$L\text{velocity} = \frac{1}{N} \sum_{i=1}^{N} |\hat{\mathbf{y}}_i - \mathbf{y}_i|^2. \tag{21}$$

The overall loss is defined as:

$$L_{\text{total}} = L_{\text{velocity}} + L_{\text{uncertainty}} + L_{\text{align}}, \tag{22}$$

where $L_{\text{uncertainty}} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} ||\mathbf{u}_i||^2$, and L_{align} enforces feature consistency, as defined in Eq. (13).

5 Experimental Evaluation

In this section, we conducted evaluations for DiffusionIMU both qualitatively and quantitatively.

5.1 Experimental Setting

Dataset. The proposed DiffusionIMU was evaluated on the RoNIN dataset [Herath *et al.*, 2020]. This dataset contains IMU readings collected from 100 users performing natural motion, including walking and running. The IMU data was recorded at 200Hz using three smartphone models: Asus Zenfone AR, Samsung Galaxy S9, and Google Pixel 2 XL. The dataset includes 276 sequences, with ground-truth trajectories obtained from a 3D tracking system. It is one of the largest datasets for deep inertial navigation and is divided into seen and unseen test sets based on user presence in training[1].

Metrics. Two commonly used metrics were adopted: the Absolute Trajectory Error (ATE) and the Relative Trajectory Error (RTE) [Herath *et al.*, 2020]:

- The ATE quantifies the overall discrepancy between the estimated and ground-truth trajectories. It is calculated as the Root Mean Squared Error (RMSE) over the entire trajectory.
- The RTE measures the RMSE within a fixed time interval, set to 1 minute in this paper. For sequences shorter than 1 minute, the positional error at the final frame is computed and scaled proportionally.

[1] https://ronin.cs.sfu.ca/README.txt

Test Subject	Metric	PDR	RIDI	RoNIN-LSTM	RoNIN-TCN	RoNIN-ResNet	CTIN*	DiffusionIMU (ours)	Perf. Improvement over CTIN
Seen	ATE	28.10	16.90	4.83	5.78	3.86	4.62	**3.64**	21.21%
	RTE	20.60	17.80	2.81	3.68	2.75	2.81	**2.72**	3.20%
Unseen	ATE	26.17	15.88	7.46	6.73	5.76	5.61	**5.27**	6.06%
	RTE	20.70	18.13	4.46	4.33	4.45	4.48	**4.31**	3.79%

* CTIN results are cited from [Rao *et al.*, 2022] for fair comparison.

Table 1: Performance comparison across different methods on the RoNIN dataset. The best results in each row are bolded.

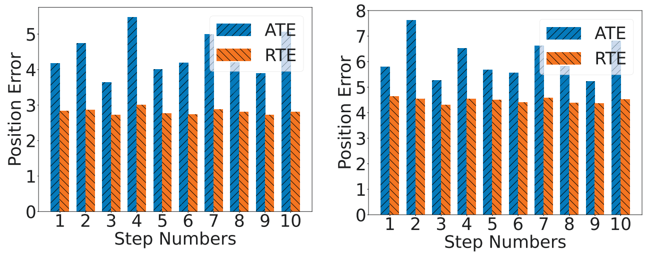
Baselines. We compared our method with four baselines:

- **PDR**: The approach serves as a classical baseline in inertial navigation. It comprises three primary components: step counting [Teng *et al.*, 2020], stride estimation [Weinberg, 2002], and orientation tracking [Madgwick *et al.*, 2011].
- **RIDI**: The approach uses a data-driven approach to estimate velocity vectors from linear accelerations and angular velocities [Yan *et al.*, 2018]. The model applies a correction for low-frequency biases in acceleration signals before performing double integration to compute positions [Yan *et al.*, 2018].
- **RoNIN**: RoNIN employs a neural network-based approach to enhance position and heading estimation [Herath *et al.*, 2020]. Three variants of RoNIN were considered: RoNIN-LSTM, RoNIN-ResNet, and RoNIN-TCN, each leveraging distinct neural architectures to capture temporal and spatial features from IMU data.
- **CTIN**: CTIN introduces a Transformer-based architecture for velocity prediction, leveraging its ability to capture long-range dependencies within sequential IMU data [Rao *et al.*, 2022]. This method demonstrates the potential of attention mechanisms to enhance motion estimation accuracy.

The proposed DiffusionIMU model was implemented in PyTorch 1.7.1 [Paszke *et al.*, 2019] and optimized using the Adam optimizer [Kingma and Ba, 2015]. The training process employed a batch size of 128 and an initial learning rate of 0.0003. A dropout rate of 0.2 was applied to mitigate overfitting. The maximum diffusion steps were set to 3, and the hidden dimensionality of the model was configured to 128. The model was trained on a single NVIDIA A100 GPU, leveraging its high computational efficiency for forward and backward passes. The total training time for the model was approximately 10 hours, with convergence typically achieved within this duration.

5.2 Overall Performance

It can be seen from Table 1 that DiffusionIMU consistently outperforms existing state-of-the-art models across all evaluated scenarios on the RoNIN dataset. Specifically, DiffusionIMU achieves the lowest ATE and RTE in both seen and unseen test sets. Compared to the baseline CTIN, DiffusionIMU demonstrates an overall performance improvement of 21.21% in the seen test set and 6.06% in the unseen test set for ATE, indicating its superior generalization capability. These results validate the effectiveness of the proposed diffusion-based iterative motion refinement framework. The iterative denoising process progressively corrects accumulated errors and significantly reduces trajectory drift, especially in challenging unseen test sets. Furthermore, the introduced noise-adaptive feature modulation and feature alignment mechanism enhance the robustness and adaptability of the model under varying sensor noise and motion patterns. This leads to more reliable motion estimation than previous models, which often suffer from degraded performance in unseen conditions due to limited generalization ability.



(a): Seen Test Set

(b): Unseen Test Set

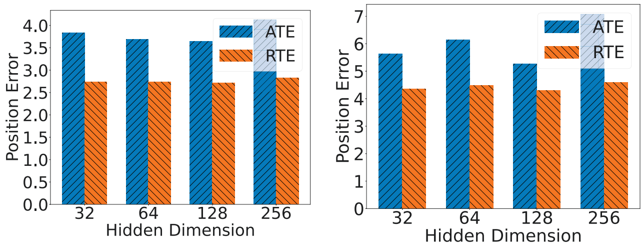
Figure 3: The impact of diffusion denoising steps on inertial navigation accuracy in the RoNIN dataset. (a) Results on the seen test set. (b) Results on the unseen test set.

Step Number Selection in the Diffusion Decoder. Figure 3 presents the impact of diffusion denoising steps on inertial navigation accuracy using the RoNIN dataset, considering both seen and unseen test sets. The x-axis represents the number of diffusion steps (K), while the y-axis denotes the ATE and RTE metrics. The results indicate that a 3-step denoising process ($K = 3$) in the diffusion decoder achieves the lowest ATE and RTE values across both test conditions. Using fewer than 3 steps results in inadequate noise suppression, whereas increasing the step beyond this threshold leads to numerical artifacts and over-smoothing, diminishing the model's ability to retain fine-grained motion details. This consistency across seen and unseen sets demonstrates the model's ability to generalize denoising behavior to novel motion patterns.

Hidden Dimension Selection. Figure 4 presents the impact of the hidden dimension D on inertial navigation accuracy using the RoNIN dataset, considering both seen and unseen test sets. The x-axis represents the hidden dimension values ($D = 32, 64, 128, 256$), while the y-axis denotes the ATE

Noise-Adaptive Feature Modulation	Feature Alignment Mechanism	Diffusion-Based Temporal Modeling	Multi-Task Loss	Seen		Unseen	
				ATE	RTE	ATE	RTE
✗	✗	✗	✗	3.86	2.75	5.76	4.45
✗	✓	✓	✓	3.90	2.75	5.29	4.40
✓	✗	✓	✓	4.75	2.93	5.94	4.54
✓	✓	✓	✗	3.75	2.75	5.55	4.50
✓	✓	✓	✓	**3.64**	**2.72**	**5.27**	**4.31**

Table 2: Ablation on each component in the DiffusionIMU. The evaluation is on the RoNIN dataset.



(a): Seen Test Set

(b): Unseen Test Set

Figure 4: The impact of hidden dimension on inertial navigation accuracy in the RoNIN dataset. (a) Results on the seen test set. (b) Results on the unseen test set.

and RTE metrics. The results indicate that $D = 128$ achieves the lowest ATE and RTE values across both test conditions. Smaller hidden dimensions limit the model's capacity to capture complex motion dynamics, resulting in suboptimal performance. Conversely, increasing D beyond 128 provides no further gains while increasing computational overhead. This suggests that $D = 128$ offers the best trade-off between feature expressiveness for inertial navigation.

5.3 Ablation Study

Table 2 presents the results of an ablation study on the RoNIN dataset, which is the largest and most diverse dataset used in our experiments. The baseline model corresponds to RoNIN-ResNet, which is the case where all four modules are disabled. The complete DiffusionIMU model, which integrates all four components, achieves the lowest ATE and RTE across all settings, demonstrating the complementary benefits of each module. Among the individual components, the noise-adaptive feature modulation and the diffusion-based temporal modeling have the most substantial impact on ATE. In contrast, the feature alignment mechanism and the multi-task loss contribute to reducing trajectory drift. The most significant performance gains are observed in unseen cases, highlighting the role of these modules in improving generalization.

Noise-Adaptive Feature Modulation. As shown in Table 2, the results indicate that enabling this module increases the ATE from 3.86 to 3.90 in the seen test cases and from 5.27 to 5.29 in the unseen cases. The RTE also shows a minor increase. These results indicate that noise-adaptive feature modulation is crucial in refining motion representations by dynamically adjusting feature embeddings to account for sensor noise.

Feature Alignment Mechanism. As observed in Table 2, disabling this module increases ATE from 3.64 to 4.75 in seen cases and from 5.27 to 5.94 in unseen cases, with a slight increase in RTE from 2.72 to 2.93 (seen) and from 4.31 to 4.54 (unseen). The results demonstrate that the feature alignment mechanism helps maintain consistency between extracted features and the diffusion pipeline. It aligns the high-level representations with the expected noise-aware latent space, ensuring that the denoising steps remain semantically meaningful across stages. This consistency proves especially important when generalizing to motion patterns not seen during training.

Diffusion-Based Temporal Modeling. As shown in Table 2, removing this component leads to an increase in ATE from 3.64 to 3.86 in seen cases and from 5.27 to 5.76 in unseen cases. RTE also increases from 2.72 to 2.75 (seen) and from 4.31 to 4.45 (unseen). These findings highlight the significance of diffusion-based temporal modeling in reducing trajectory drift and improving temporal consistency. The multi-step structure helps preserve local motion smoothness while maintaining long-term accuracy.

Multi-Task Loss. As reported in Table 2, removing this module increases ATE from 3.64 to 3.75 in seen cases and from 5.27 to 5.55 in unseen cases. RTE also rises from 2.72 to 2.75 (seen) and from 4.31 to 4.50 (unseen). The results indicate that the multi-task loss helps stabilize trajectory estimation by jointly optimizing velocity and uncertainty.

6 Conclusion

In this paper, we propose DiffusionIMU, a novel diffusion-based framework for inertial navigation that iteratively refines motion state estimation through a generative denoising process. To achieve this, the noise-adaptive feature modulation dynamically adjusts feature representations to handle sensor variability, the feature alignment mechanism ensures consistency between learned representations and the diffusion process, and the diffusion-based temporal modeling refines velocity predictions over multiple iterations to improve long-term accuracy. Additionally, the multi-task loss optimizes trajectory estimation and uncertainty quantification, enhancing overall robustness. Extensive experiments demonstrate that DiffusionIMU significantly outperforms state-of-the-art models. Our results highlight the effectiveness of diffusion-based modeling for inertial navigation, and we hope that this work will inspire future research in inertial navigation.

Acknowledgments

This work was supported by the Beijing Natural Science Foundation (Nos. JQ23014, JQ23014, L231013, and L233026), the National Natural Science Foundation of China (Nos. 62302052, 62271074, 62302052, 62071157, U21A20515, 62376271, U22B2034, 62277001, and 62365014), and the Henan Province Key Research and Development Project (No. 241111211900).

References

[Brajdic and Harle, 2013] Agata Brajdic and Robert Harle. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 225–234, 2013.

[Campos *et al.*, 2021] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Transactiosn on Robotics*, 37(6):1874–1890, 2021.

[Chang *et al.*, 2016] Lubin Chang, Jingshu Li, and Kailong Li. Optimization-based alignment for strapdown inertial navigation system: comparison and extension. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1697–1713, 2016.

[Chen *et al.*, 2018] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. IONet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 6468–6476, 2018.

[Chen *et al.*, 2021] Qijin Chen, Quan Zhang, and Xiaoji Niu. Estimate the pitch and heading mounting angles of the IMU for land vehicular GNSS/INS integrated system. *IEEE Transactions on Intelligent Transportation Systems*, 22(10):6503–6515, 2021.

[de Silva *et al.*, 2018] Rajitha de Silva, Jehan Perera, Chamli Priyashan Abeysingha, and Nimsiri Abhayasinghe. A gyroscopic data based pedometer algorithm with adaptive orientation. In *Proceedings of the 14th IEEE International Conference on Control and Automation*, pages 953–956, 2018.

[Gao *et al.*, 2014] Ruipeng Gao, Yang Tian, Fan Ye, Luo Guojie, Kaigui Bian, Yizhou Wang, Tao Wang, and Xiaoming Li. Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment. *IEEE Transactions on Mobile Computing*, 13(9):1–14, 2014.

[Geiger *et al.*, 2012] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proc. of 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2015.

[Herath *et al.*, 2020] Sachini Herath, Hang Yan, and Yasutaka Furukawa. RoNIN: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. *Proc. of 2020 IEEE International Conference on Robotics and Automation*, 3146–3152, 2020.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. of Annual Conference on Neural Information Processing Systems*, 2020.

[Hu *et al.*, 2023] Jiaxin Hu, Kefei Ren, Xiaoyu Xu, Lipu Zhou, Xiaoming Lang, Yinian Mao, and Guoquan Huang. Efficient visual-inertial navigation with point-plane map. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 10659–10665, 2023.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of the 3rd International Conference on Learning Representations*, 2015.

[Laidig and Seel, 2023] Daniel Laidig and Thomas Seel. VQF: Highly accurate IMU orientation estimation with bias estimation and magnetic disturbance rejection. *Infomation Fusion*, 91:187–204, 2023.

[Liu *et al.*, 2020] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I. Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. TLIO: Tight learned inertial odometry. *CoRR*, abs/2007.01867, 2020.

[Madgwick *et al.*, 2011] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *Proc. of IEEE International Conference on Rehabilitation Robotics*, 2011.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, unjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of Annual Conference on Neural Information Processing Systems*, pages 8024–8035, 2019.

[Rao *et al.*, 2022] Bingbing Rao, Ehsan Kazemi, Yifan Ding, Devu M. Shila, Frank M. Tucker, and Liqiang Wang. CTIN: robust contextual transformer network for inertial navigation. In *Proc. of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, pages 5413–5421, 2022.

[Shen *et al.*, 2013] Guobin Shen, Zhuo Chen, Peichao Zhang, Thomas Moscibroda, and Yongguang Zhang. Walkie-Markie: Indoor pathway mapping made easy. In *Proc. of the 10th USENIX Symposium on Networked Systems Design and Implementation*, pages 85–98, 2013.

[Skog *et al.*, 2010] Isaac Skog, Peter Händel, John-Olof Nilsson, and Jouni Rantakokko. Zero-velocity detection - an algorithm evaluation. *IEEE Transactions on Biomedical Engineering*, 57(11):2657–2666, 2010.

[Sun *et al.*, 2021] Scott Sun, Dennis Melamed, and Kris Kitani. IDOL: Inertial deep orientation-estimation and localization. In *Proc. of Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 6128–6137, 2021.

[Teng *et al.*, 2017] Xiaoqiang Teng, Deke Guo, Yulan Guo, Xiaolei Zhou, Zeliu Ding, and Zhong Liu. IONavi: An indoor-outdoor navigation service via mobile crowdsensing. *ACM Transactions on Sensor Networks*, 13(2):12:1–12:28, 2017.

[Teng *et al.*, 2019] Xiaoqiang Teng, Deke Guo, Yulan Guo, Xiaolei Zhou, and Zhong Liu. CloudNavi: Toward ubiquitous indoor navigation service with 3d point clouds. *ACM Transactions on Sensor Networks*, 15(1):1:1–1:28, 2019.

[Teng *et al.*, 2020] Xiaoqiang Teng, Pengfei Xu, Deke Guo, Yulan Guo, Runbo Hu, and Hua Chai. ARPDR: An accurate and robust pedestrian dead reckoning system for indoor localization on handheld smartphones. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10888–10893, 2020.

[Wang *et al.*, 2012] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, pages 197–210, 2012.

[Wang *et al.*, 2024] Jiale Wang, Chuang Shi, Ming Xia, Fu Zheng, Tuan Li, Yunfeng Shan, Guifei Jing, Wu Chen, and T. C. Hsia. Seamless indoor-outdoor foot-mounted inertial pedestrian positioning system enhanced by smartphone PPP/3-D map/barometer. *IEEE Internet of Things Journal*, 11(7):13051–13069, 2024.

[Weinberg, 2002] Harvey Weinberg. Using the ADXL202 in pedometer and personal navigation applications. In *Application Notes American Devices; Analog Devices, Inc.*, 2002.

[Yan *et al.*, 2018] Hang Yan, Qi Shan, and Yasutaka Furukawa. RIDI: Robust IMU double integration. In *Proc. of the 15th European Conference on Computer Vision*, pages 641–656, 2018.

[Yang *et al.*, 2024] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 4(56):105:1–105:39, 2024.