iMoT: Inertial Motion Transformer for Inertial Navigation

Son Minh Nguyen, Duc Viet Le, Paul Havinga

Department of Computer Science, University of Twente, Enschede, The Netherlands
{m.s.nguyen; v.d.le; p.j.m.havinga}@utwente.nl

Abstract

We propose iMoT, an innovative Transformer-based inertial odometry method that retrieves cross-modal information from motion and rotation modalities for accurate positional estimation. Unlike prior work, during the encoding of the motion context, we introduce Progressive Series Decoupler at the beginning of each encoder layer to stand out critical motion events inherent in acceleration and angular velocity signals. To better aggregate cross-modal interactions, we present Adaptive Positional Encoding, which dynamically modifies positional embeddings for temporal discrepancies between different modalities. During decoding, we introduce a small set of learnable query motion particles as priors to model motion uncertainties within velocity segments. Each query motion particle is intended to draw cross-modal features dedicated to a specific motion mode, all taken together allowing the model to refine its understanding of motion dynamics effectively. Lastly, we design a dynamic scoring mechanism to stabilize iMoT's optimization by considering all aligned motion particles at the final decoding step, ensuring robust and accurate velocity segment estimation. Extensive evaluations on various inertial datasets demonstrate that iMoT significantly outperforms state-of-the-art methods in delivering superior robustness and accuracy in trajectory reconstruction.

Code — https://github.com/Minh-Son-Nguyen/iMoT

Introduction

Inertial odometry systems are intended to estimate the three-dimensional trajectories of a moving body by jointly analyzing motion, rotation, and geographically magnetic interactions from Inertial Measurement Unit (IMU) signals. Given privileges typified by energy efficiency, privacy preservation, and environmental robustness over other modalities, these approaches become more essential for tracking and tracing missions, including virtual and augmented reality, biodiversity monitoring, and rescue operations, especially in harsh environments, such as areas shrouded in smoke, or mist where common modalities (e.g., radio, acoustic, and visual signals) are no longer reliable for tracking purposes.

Contemporary approaches to inertial navigation are commonly classified into three distinct research paradigms: physics-based methods, heuristic priors-based methods, and data-driven priors-based methods. Since quadratic errors are leaked through the double integral of noisy IMU measurements whilst computing displacements, classical physics-based methods, exemplified by strap-down inertial navigation systems (SINS) (Titterton and Weston 2004) result in huge drift accumulation in a very short time. Heuristic priors-based methods, represented by pedestrian dead reckoning (PDR) approaches (Jimenez et al. 2009; Tian et al. 2015), decompose trajectory estimation into discrete components: step detection, step-length estimation, and heading estimation, all operating under assumptions of regular human gait patterns. However, these methods often face limitations in adaptability, with step detection and step-length estimation being confined to specific scenarios and occasionally relying on fixed parameters. Moreover, heading estimation may suffer from inaccuracies induced by gravitational and magnetic disturbances. In an advanced manner, data-driven priors-based methods (Chen et al. 2018a; Herath, Yan, and Furukawa 2020; Liu et al. 2020) utilize end-to-end deep learning architectures to directly estimate velocity segments from IMU sequences, significantly reducing drifting errors in trajectory reconstruction. Despite their progress, these methods fail to consider modality distinctions between acceleration and angular rates, as well as motion uncertainties among individuals, resulting in compromised accuracy.

In this paper, we present a multimodal **i**nertial **Mo**tion **T**ransformer (iMoT), which maximizes the utilization of complementary features between motion and rotation inputs to dynamically represent motion uncertainties for different instantaneous velocity segments. Firstly, we address the practical challenges of input tokenization in iMoT. Conventional tokens captured at a single time step with multiple variates (i.e., channels) often struggle to convey a complete chain of motion events due to excessively local receptive fields and time-unaligned events represented by concurrent time points (Zhang and Yan 2023). In contrast, we consider the entire time series of each variate as an input token.

Secondly, given the multivariate time series nature of IMU sequences, we extend the time series decomposition (Hyndman and Athanasopoulos 2018), a widely adopted pre-processing technique, into Progressive Series Decoupler (PSD), a trainable module that can be seamlessly integrated within encoder layers. Specifically, PSD enhances unimodal

features of acceleration and angular velocity signals and aids the absorption of such information by progressively decomposing their intricate temporal patterns into more interpretable components, which better highlight critical motion events such as half-turns, U-turns, and periods of stillness. To reflect modality differences, we propose Adaptive Positioning Encoding (APE), which associates acceleration and angular tokens with appropriate positional embeddings based on their temporal contents. These fully embedded tokens are then processed through a self-attention module to capture multivariate correlations. Additionally, we introduce Adaptive Spatial Sync (ASC) at every residual connection to retain fine-grained details across channels.

Thirdly, we recognize that motion modes can vary between individuals and over time, affecting both the direction and magnitude of instantaneous velocity segments. Then, each velocity segment can be defined by the most likely combination of these motion modes. Inspired by Particle Filter techniques, we model motion uncertainties for a given velocity segment by manipulating a set of learnable query motion particles during decoding. Specifically, each query motion particle represents the velocity of a specific motion mode, functioning as a learnable positional embedding to probe related cross-modal features. This probing initiates a cyclic process, where the cross-modal features extracted in each decoding step are continuously used to inform and refine the query particles in successive steps. Finally, we propose a Dynamic Scoring Mechanism (DSM) to optimize the query particle set during both training and testing phases.

Our main contributions are four-fold: (1) We propose both novel inertial Transformer encoder and decoder networks (iMoT) that harness the complementary effects of motion and rotation modalities for uncertainty modeling to enhance positional accuracy. (2) In encoding context features, we first introduce Progressive Series Decoupler (PSD) to highlight motion events within each modality, followed by Adaptive Positional Encoding (APE) to include distinctions between modalities in position encoding. Additionally, we implement Adaptive Spatial Sync (ASC) at every residual connection to ensure the seamless integration of spatial details. (3) In decoding, we introduce a novel concept of query motion particles that utilize cross-modal information retrieval from context features to learn all possible motion modes accounting for uncertainties in motion. At the final decoding step, these adjusted particles are collectively considered to determine the desired velocity segments through a unique Dynamic Scoring Mechanism (DSM). (4) We rigorously validate the contribution of each proposed module and demonstrate the overall effectiveness of iMoT against state-of-the-art (SoTA) odometry methods over four benchmark inertial datasets.

Related Work

Physics-based methods (no priors). A strap-down inertial navigation system (SINS)(Savage 1998) first rotates acceleration measurements from the body frame to the navigation frame using a rotation matrix derived from integrating angular velocity measurements in order to subtract the Earth's gravity. Locations are then acquired by double-integrating the linear acceleration(Shen, Gowda, and Roy Choudhury 2018). Due to noisy sensing compounded by multiple integrations, these strap-downs incur quadratic error propagation heavily, drifting far away from desired locations.

Heuristic priors. To alleviate drifting errors accrued in SINS, step-based pedestrian dead reckoning (PDR) approaches (Jimenez et al. 2009; Tian et al. 2015) leverage human motion regularities by separately detecting steps, estimating step length and heading before updating the location with each step. These approaches yield impressive results under controlled environments where the assumption remains valid. Some other methods (Janardhanan, Dutta, and Tripuraneni 2014; Kourogi and Kurata 2014) that incorporate principal component analysis and frequency domain analysis have been developed to enhance rotational accuracy by determining body motion directions. However, these heuristic-based methods fall short of robustness, as their inner components are interdependent on gravitational and magnetic factors, and confined to specific scenarios.

Data-Driven priors. Significant efforts have recently been directed toward developing deep learning networks to extract useful features from IMU measurements for enhanced position estimation. RIDI (Yan, Shan, and Furukawa 2018) introduces a two-stage system that first regresses low-frequency corrections to the acceleration before double-integrating it into displacement. To bypass the noisy double integration, IoNeT (Chen et al. 2018a) employs LSTMs to directly regress polar velocity segments and changes in heading rates, which are then cumulatively computed to approximate translations. Similarly, RoNIN framework (Herath, Yan, and Furukawa 2020) explores three types of neural networks to assess their distinct contributions to inertial navigation. Building on RoNIN, TLIO (Liu et al. 2020) integrates displacement estimates with raw IMU measurements using a stochastic-cloning Extended Kalman Filter (EKF) to derive position, orientation, and sensor biases. More recently, CTIN (Rao et al. 2022) adapts ResNet blocks into a Transformer architecture, which incorporates motion uncertainties by optimizing covariance of velocity segments.

Transformer. Transformers have found extensive applications in natural language processing (Vaswani et al. 2017; Devlin et al. 2018) and computer vision (Dosovitskiy et al. 2020; Carion et al. 2020). Our approach is partially inspired by DETR (Carion et al. 2020) that utilizes query object tokens during decoding to extract object-related information for detection. Based on this idea, we propose a novel concept of learnable query motion pairs, yet consisting of query motion particles and their associated content features, for modeling motion uncertainties. In an advanced manner, these query pairs, especially the query motion particle set, are not only externally updated in the form of learnable positional embeddings but also undergo internal motion refinement within transformer decoder layers. This dual updating process enables more precise modeling of motion dynamics.

Proposed Method

Inertial odometry methods aim to reconstruct a traveled trajectory from corresponding IMU sequences, denoted as $A = \{A_a, A_g\} \forall A \in \mathbb{R}^{2D \times T}$, where $A_a \in \mathbb{R}^{D \times T}$ and

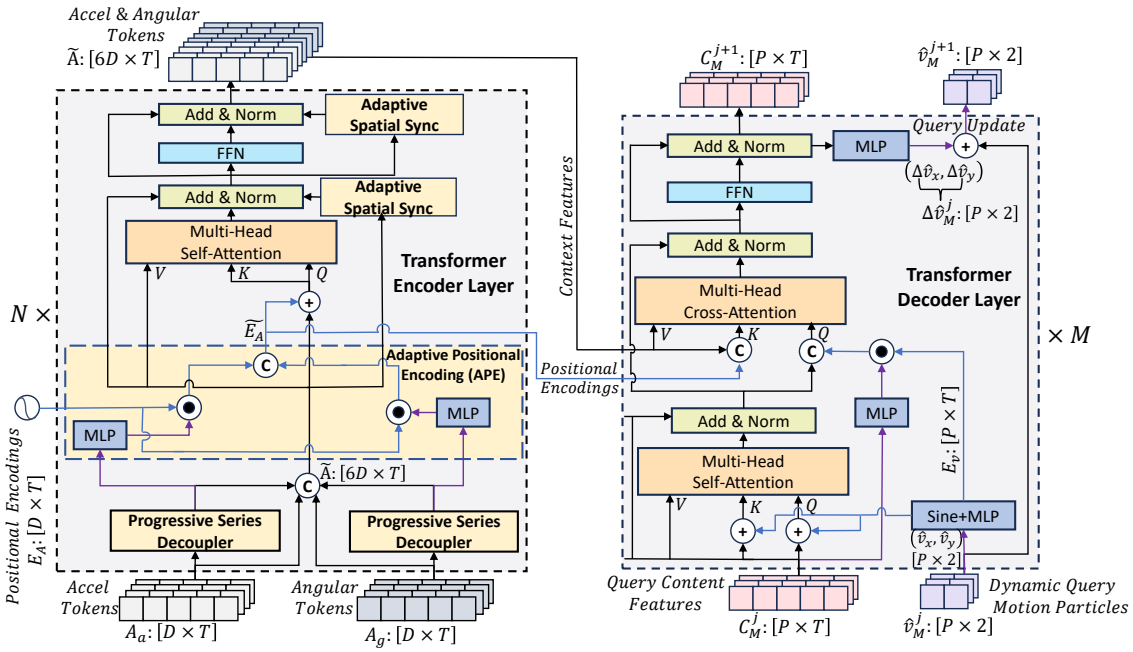

Figure 1: iMoT architecture: (1) The Encoder synthesizes motion context features from Acceleration and Angular Velocity tokens, incorporating three key innovations: Progressive Series Decoupler for enhanced information absorption, Adaptive Positional Encoding to handle modality differences, and Adaptive Spatial Sync to maintain cross-channel interactions. (2) The Decoder manipulates query motion particles to capture motion variability within velocity segments through cross-modal information retrieval. Two signal types are highlighted: the *blue flow* for base signals like sinusoidal encoding and the *magenta flow* for controlling signals, including tokens and velocity particles, making static modules or operations responsive to changes.

$\mathrm{A_g} \in \mathbb{R}^{D\times T}$ represent acceleration and angular velocity of $D \times T$ instances recorded over $D = 3$ channels along x-, y-, z-axes within 1 second, respectively. In this work, we propose **i**nertial **Mo**tion **T**ransformer (iMoT), a novel transformer encoder-decoder structure designed to iteratively refine query motion particles to learn all possible motion modes for every single instantaneous velocity segment. This approach allows iMoT to effectively adapt to variations in velocity segments, thereby enhancing trajectory synthesis. The overall structure is illustrated in Fig.1.

Encoder

In practice, all movements are inherently reflected by fluctuations in the motion and rotation modalities within IMU sequences. To seamlessly encode such motion context, we directly reshape raw IMU sequences into input tokens compatible with the transformer encoder. However, existing spatial tokens formed by a single time step $t \in T$ along D channels often struggle to express contextual information due to excessively local receptive fields and time-unaligned events. In this way, series variations are significantly influenced by the sequence order, which might undermine the attention mechanism when improperly applied to the temporal dimension. As a result, the proposed model may be compromised in its ability to capture essential series representations and portray multivariate correlations, limiting its capacity and generalizability on diverse IMU data. Therefore, we first tokenize

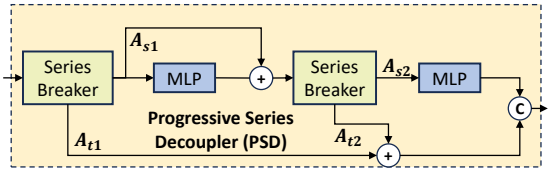

Figure 2: Progressive Series Decoupler.

the sequence into temporal variate tokens A_a and A_g of T-dimensions, which offer a broad receptive field to capture important events along the temporal axis. Following this, we introduce a series of specialized modules incorporated in encoder layers to enhance the exploitation of these tokens.

Progressive Series Decoupler To promote the capture of necessary motion events during encoding, Progressive Series Decoupler (PSD) is introduced at the facade of each encoder layer, breaking down complex temporal patterns of motion and rotation series into two more interpretable components, namely seasonal signals and trend-cycle signals. That is, seasonal signals carry prominent information about symmetric human motions (e.g., leg movements, hip swiveling, and arm swinging), which is particularly useful for tracing repeated sequence patterns (e.g., walking, strolling, and running). On the other hand, trend-cycle signals exhibit longer-term movements, revealing sudden motion changes (e.g., sudden stopping and turning). Unlike the prior work

(Wu et al. 2021), we design a centered moving average to smooth out periodic fluctuations and eliminate some of the randomness in the data, leaving a smooth trend-cycle component. The seasonal signal is then computed as the residual:

$$A_t = AvgPool_{k_2 \times k_1}\left(Padding\left(A\right)\right) ; A_s = A - A_t \tag{1}$$

Here, $A_s \in \mathbb{R}^{2D \times T}$, and $A_t \in \mathbb{R}^{2D \times T}$ denote the seasonal and the extracted trend-cycle parts, respectively. The $Padding$ operation is used to keep the series length unchanged. The $AvgPool_{k_2 \times k_1}(\cdot)$ operation depicts the application of a moving average of order k_1 followed by another moving average of order k_2. Note that k_2 must be less than k_1, and both should be either odd or even numbers to ensure the symmetry of the weighted average applied to the observations from both the inner and outer sides. As depicted in Fig.2, we denote $A_s, A_t = SeriesBreaker(A)$, an internal operation embedded within PSD module that is progressively learned over layers. After undergoing PSD, sequence tokens A are associated with their decoupled components A_t and A_s together as standard input tokens $\tilde{A} \in \mathbb{R}^{6D \times T}$ to the subsequent blocks. This approach helps isolate and interpret different motion cues with much greater ease, enhancing the model's ability to handle complex IMU data effectively.

Adaptive Positional Encoding Positional Embeddings play a crucial role in extracting, storing, and pooling context features in both encoding and decoding phases. As presented in Fig.1, we initialize Adaptive Positional Encoding (APE) with common base sinusoidal positional embeddings $E_A \in \mathbb{R}^{D \times T}$. From these base embeddings, we start to learn temporal scaling factors according to content discrepancies between modalities, resulting in adaptive positional embeddings $\widetilde{E_A} \in \mathbb{R}^{6D \times T}$:

$$\widetilde{E_A} = \left[\mathrm{MLP}\left(\tilde{A_a}\right) \cdot E_A, \mathrm{MLP}\left(\tilde{A_g}\right) \cdot E_A\right] \tag{2}$$

where $\tilde{A}_a \in \mathbb{R}^{3D \times T}$, and $\tilde{A}_g \in \mathbb{R}^{3D \times T}$ are fully integrated versions of acceleration and angular tokens, respectively. $\mathrm{MLP}(\cdot)$ refers to a multilayer perceptron network, and $[\cdot]$ denotes a concatenation operation. Subsequently, the attention module of j-th transformer encoder layer is applied to capture multivariate correlations between temporal tokens, which is formulated as below:

$$\tilde{A}^j = \mathrm{SelfAttn}\left(\mathrm{Q} = \tilde{\mathrm{A}}^{\mathrm{j}-1} + \widetilde{\mathrm{E}_{\mathrm{A}}}^{\mathrm{j}-1}, \mathrm{K} = \tilde{\mathrm{A}}^{\mathrm{j}-1} + \widetilde{\mathrm{E}_{\mathrm{A}}}^{\mathrm{j}-1}, \mathrm{V} = \tilde{\mathrm{A}}^{\mathrm{j}-1}\right) \tag{3}$$

In this equation, $\tilde{A}^j$ represents the updated tokens at layer j, while $\tilde{A}^{j-1} = [\tilde{A_a}^{j-1}, \tilde{A_g}^{j-1}]$ and $\widetilde{E_A}^{j-1}$ are the tokens and positional embeddings from the previous layer, respectively. By applying adaptive scaling factors to the common base positional encodings E_A, iMoT becomes permutation-invariant to token orders, thus allowing it to tailor the positional encoding to specific characteristics of motion and rotation data. Staying in the flow of these benefits, the self-attention module, denoted as $\mathrm{SelfAttn}$ can better encode unique cross-modal interactions into contextual features.

Adaptive Spatial Sync Since contextual features are typically encoded in a temporal manner where cross-channel interactions at each time step are not considered, we propose an Adaptive Spatial Sync (ASC) module to compensate for the shortage of such spatial information. Placed at every residual connection, the ASC module can infuse fine-grained channel details directly along with two other information flows. As shown in Fig.3, this infusion involves three stages: (1) an 1×3 convolution operated along 3D direction to extract cross-channel features from temporally concatenated tokens $\widetilde{A}_T \in \mathbb{R}^{3D \times 2T}$ for each time step; (2) a parallel branch with Global Average Pooling (GAP), 1×1 convolution, and sigmoid function to learn a channel weight vector from all sensor channels along the temporal axis $W = [w_1, \cdots, w_{2T}]$; and finally, (3) a Reshape operation followed by an 1×1 convolution with GELU to incorporate temporal interactions and then project the tokens back to their space.

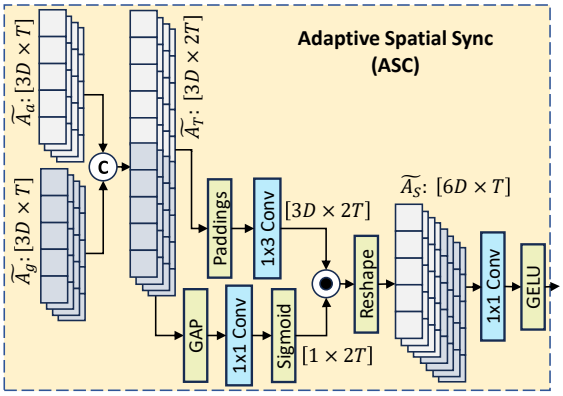

Figure 3: Adaptive Spatial Sync.

Decoder

A given instantaneous velocity segment varies across individuals and is influenced by their unique gaits, giving rise to uncertainties in motion. To address this, we introduce P motion pairs of query motion particles $\hat{v} \in \mathbb{R}^{P \times 2}$ and their query content features $C \in \mathbb{R}^{P \times T}$ into decoder layers, as illustrated in Fig.1. Each pair represents a specific motion mode, collectively modeling motion uncertainties. By iteratively refining the query motion particles based on their updated content features for individual velocity segments, the model can quickly adapt to motion variability and effectively approximate the desired velocity segments.

Query motion Particles The query motion particles, which characterize respective velocity, are represented as positional embeddings and dynamically updated according to the predicted velocity within each decoder layer as follows:

$$E_{\hat{v}}^j = \mathrm{MLP}(\mathrm{PE}(\hat{\mathrm{v}}^{\mathrm{j}-1})) \tag{4}$$

where PE denotes the sinusoidal positional encoding operation, which is conditioned on velocity of the query particles $\hat{v}^{j-1}$ in the previous layer $j-1$. The resulting learnable positional embeddings $E_{\hat{v}}^j$ are added to the content features C_j (i.e., C_0 initialized with a zero matrix) and then passed into

the $\mathrm{SelfAttn}$ module of the decoding layer j:

$$C_{sa}^{j} = \mathrm{SelfAttn}\left(\mathrm{Q} = C^{j-1} + E_{\widehat{v}}^{j},\, \mathrm{K} = C^{j-1} + E_{\widehat{v}}^{j},\, \mathrm{V} = C^{j-1}\right) \tag{5}$$

Here, $C_{sa}^{j} \in \mathbb{R}^{P\times T}$ represents the updated query content. To enable the pooling of cross-modal information from encoded context features, we concatenate positional embeddings with the updated content information as queries in the cross-attention module. This also allows us to decouple the contributions of content and position to the attention weights. To align with positional embeddings from the encoder, we learn an MLP on the content information to generate a scaling vector for the positional embeddings in the decoder. Specifically, two cross-attention modules are employed singly for retrieving features regarding specific motion modes from both motion $\widetilde{A}_a$ and rotation tokens $\widetilde{A}_g$:

$$\begin{aligned} C_{a}^{j} &= \mathrm{SelfAttn}\left(\begin{array}{l}\mathrm{Q} = \left[C_{sa}^{j}, \mathrm{MLP}\left(C^{j-1}\right)\cdot E_{\widehat{v}}^{j}\right],\, \mathrm{K} = \left[\widetilde{A}_a, \widetilde{E}_a\right], \\ \mathrm{V} = \widetilde{A}_a\end{array}\right) \\ C_{g}^{j} &= \mathrm{SelfAttn}\left(\begin{array}{l}\mathrm{Q} = \left[C_{sa}^{j}, \mathrm{MLP}\left(C^{j-1}\right)\cdot E_{\widehat{v}}^{j}\right],\, \mathrm{K} = \left[\widetilde{A}_g, \widetilde{E}_g\right], \\ \mathrm{V} = \widetilde{A}_g\end{array}\right) \\ C^{j} &= \mathrm{MLP}\left(\left[C_{a}^{j}, C_{g}^{j}\right]\right) \end{aligned} \tag{6}$$

where $C_{a}^{j} \in \mathbb{R}^{P\times T}$, $C_{g}^{j} \in \mathbb{R}^{P\times T}$, and $C^{j} \in \mathbb{R}^{P\times T}$ denote the acceleration-augmented content features, angular velocity-augmented content features, and the updated query content features for each query motion particle, respectively. Additionally, $\widetilde{E}_a \in \mathbb{R}^{3D\times T}$, $\widetilde{E}_g \in \mathbb{R}^{3D\times T}$ represent positional embeddings of motion and rotation tokens in $\widetilde{E}_A$.

Particle Refinement In addition to external updates of positional embeddings, employing motion particles as learnable queries also enables internal, layer-by-layer updates, allowing the particle set to rapidly adapt to motion variability. For each decoding layer j, we utilize an MLP to predict relative velocity adjustments $\Delta\widehat{v}^{j} = [\Delta v_x^j, \Delta v_y^j]$ based on the updated content features. However, these MLPs share the same parameters across layers to ensure consistent updates throughout the model.

$$\Delta\widehat{v}^{j} = \mathrm{MLP}\left(\left[C^{j}\right]\right) \quad \widehat{v}^{j+1} = \widehat{v}^{j} + \Delta\widehat{v}^{j} \tag{7}$$

Dynamic Scoring Mechanism At the final layer, we start with the calculation of Euclidean distances between neighboring particles $\hat{v}$ and the ground-truth velocity particle $v_{GT} \in \mathbb{R}^{1\times 2}$ to establish an inverse score list $S = [S_0, \cdots, S_{P-1}] \in \mathbb{R}^{1\times P}$ that assigns higher scores to particles closer to v_{GT}. To optimize the particle set, we focus on making significant adjustments to distant particles while giving less attention to the closer ones when computing the mean particle $v_m \in \mathbb{R}^{1\times 2}$. This strategy encourages a more compact distribution of particles around the desired velocity particle v_{GT}. The calculation is formalized as follows:

$$\begin{aligned} S_p = \frac{e^{-d(\hat{v}_p, v_{GT})}}{\sum_{i=0}^{P-1} e^{-d(\hat{v}_i, v_{GT})}}\ ; v_m = \sum_{i=0}^{P-1}\left(1 - S_p\right)^{\gamma}\widehat{v}_p \\ J_{vel} = \frac{1}{B}\sum_{b=0}^{B-1}\left\|v_{GT}^{b} - v_{m}^{b}\right\|^2 \end{aligned} \tag{8}$$

where γ is the weighting factor largely controlling the influence of distant particles on the mean particle v_m, and $d(\cdot)$ represents the Euclidean distance operation. J_{vel} is the mean square error loss between mean particles and ground-truth velocity particles over a batch of B samples. To maintain stability during both training and testing, we further introduce an entropy loss function to maximize the entropy of the particle score list S:

$$J_{ent} = \frac{1}{BP}\sum_{b=0}^{B-1}\sum_{p=0}^{P-1} -S_p^b \log\left(S_p^b + \varepsilon\right) \tag{9}$$

Here, J_{ent} denotes the entropy loss function and ϵ is a small constant of $1e-10$ to stabilize the loss. This loss encourages a more uniform distribution of estimated particles around the ground-truth particles, allowing the use of average pooling of the estimated particles to approximate the desired velocities once training is complete. While the combined constraints of J_{vel} and J_{ent} could steer the proposed model to significantly minimize velocity discrepancies and adjust the particle distribution, we empirically find some instabilities during testing. Specifically, the absence of ground-truth particles for generating the score list necessitates the use of average pooling to approximate these particles, leading to insufficient fine-grained velocity segments and thereby degrading the quality of trajectory reconstruction. To address this issue, we employ an MLP that attends to all aligned particles at the last layer, considering both x-, and y-directions, to directly learn a dynamic two-dimensional score list $S^d = [S_1^d, \cdots, S_{P-1}^d] \in \mathbb{R}^{2\times P}$:

$$S^d = \mathrm{MLP}\left(\widehat{v}^{\mathrm{T}}\right)\ ; v_m = S^d \cdot \widehat{v}_p \tag{10}$$

where $\widehat{v}^{\mathrm{T}}$ denotes transposed motion particles. This approach adaptively transforms the particle distribution and pools the mean particle accordingly during both training and testing, thereby eliminating the need for the entropy loss J_{ent}. As a result, iMoT can be efficiently optimized using only the velocity loss J_{vel}.

Experiment

In this section, we empirically verify the effectiveness of our method in controllable and dynamic scenarios.

Dataset Four popular benchmark datasets are used for evaluation: RIDI (Yan, Shan, and Furukawa 2018), RoNIN (Herath, Yan, and Furukawa 2020), OxIOD (Chen et al. 2018b), and IDOL (Sun, Melamed, and Kitani 2021). For controlled scenarios, RIDI and OxIOD are configured with predefined attachments, such as pocket, handheld, bag, body, and trolley, which are specified separately for each sequence. To provide end-users with greater freedom of movement in practice, IDOL and RoNIN are designed with dynamic motion contexts, where devices are naturally placed across all recorded sequences. Notably, RoNIN is the largest dataset, containing more than 40 hours of IMU data from 100 human subjects performing natural human motions.

Evaluation Metric Four types of metrics are used for the quantitative trajectory evaluation:

- **Absolute Trajectory Error** (ATE) (m) is calculated as the average Root Mean Squared Error (RMSE) between the estimated and ground-truth trajectories as a whole.

- **Distance-Relative Trajectory Error** (D-RTE) (m), is calculated as the average RMSE between the estimated and the ground-truth over a fixed distance d_r (*i.e.*, 1 m).
- **Time-Relative Trajectory Error** (T-RTE) (m) is the average RMSE over a regular period t_r (*i.e.*, 1 minute).
- **Position Drift Error** (PDE) (%) measures the drifting error at the final position relative to the traveled distance.

Implementation Details To initialize PSD module in the encoder, we implement $AvgPool_{k_2 \times k_1}$ with k_1 and k_2 set to 9, 3, respectively. For decoding, we empirically find that using a set of $P = 128$ query motion particles representing 128 motion modes is sufficient. Depending on the sampling rate of each dataset, the token dimension is set to 100 for IMU sequences recorded at 100 Hz and to 200 for sequences recorded at 200 Hz. The network, consisting of $N = 2$ encoder layers and $M = 2$ decoder layers, is trained with the learning rate of $1e-4$ and batch size of $B = 128$ using Adam optimization. The training is performed with PyTorch version 2.4.0 on an H100 GPU with 80 GB of memory.

Ablation Study As presented in Tab.1, we develop 14 configurations on the largest RoNIN dataset, where all the proposed modules are progressively incorporated or removed, to meticulously examine both stand-alone and complementary contributions of the proposed modules.

Progressive Series Decoupler. To assess the isolated impact of PSD, we compare the performance between the baseline model (i), where all proposed modules are excluded, and model $(viii)$ with only PSD enabled. The findings reveal considerable improvements across all error metrics when PSD is activated, especially a 13.89% reduction in ATE. Furthermore, the effectiveness of PSD is consistently observed even when jointly incorporated with other modules. For instance, there is an 8.16% ATE reduction between model (iv) with PSD and model (v) without PSD, and an 8.29% reduction between the full version (xv) and model (xiv), which is identical except for the absence of PSD. These consistent improvements across different configurations firmly verify the necessity of the additional information provided by PSD.

Adaptive Positional Encoding. To verify the stand-alone influence of APE, we collate the difference in performance between the baseline (i) and model (x) with only APE enabled. The improvements, particularly $\sim$ 11.11% in ATE reduction underscores APE's significant role in producing adaptive positional embeddings accounting for modal distinctions. Furthermore, consistent performance gains across various configurations pairs, *e.g.*, (iii) vs. (ii), (vi) vs. (v), (vii) vs. (iv), and $(xiii)$ vs. (xv) further validates the effectiveness of the proposed module with higher confidence.

Adaptive Spatial Sync. The ASC module is designed to compensate for the absence of cross-channel interactions. Although its contribution margin may appear smaller than others, particularly a 7.41% reduction in ATE when upgrading the baseline model (i) to the configuration (ix) with the inclusion of ASC, it has consistently delivered improvements both in standalone and complementary scenarios. Typically, the performance gains from the configuration (vi) to the full version (xv) further confirm the effectiveness of fusing spatial interactions, as evidenced by a reduction in ATE from 5.39 m to 5.31 m.

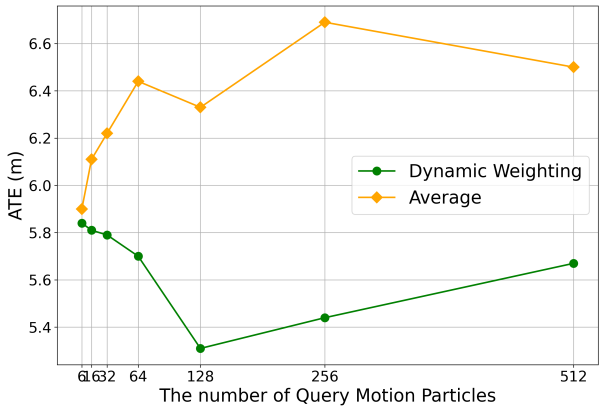

Figure 4: Ablation Study on the number of query motion particles on RoNIN dataset.

Query Motion Particles & Dynamic Scoring Mechanism. We first examine the bare influences of manipulating query motion particles using the combined constraints of J_{vel} and J_{ent} to describe uncertainties in motion by comparing the baseline (i) with model (ii). Unlike other cases, this results in a deterioration of 0.62 m in ATE. Further investigation reveals that directly adding query particles to existing models, such as model $(viii)$ with PSD (which already outperformed the baseline (i) by 13.89 %), leads to worse performance. For instance, the upgraded model (xii) performs 1.45 m worse than model $(viii)$ and 0.55 m worse than the baseline (i). This can be attributed to the inefficacy of the above constraints in optimizing the particle set, which becomes severe as extra sources are introduced by PSD. A similar phenomenon is observed in configurations (x) and (iii), where introducing the particles exacerbates existing performance.

However, these issues are significantly alleviated, and performance gains are achieved when the DSM module is introduced. Specifically, the configuration (iv) with DSM surpasses all its previous versions, including (ii) and (i). Similar enhancements are witnessed in comparisons of (iii) vs. (vii) and (xii) vs. (v). Besides, an intriguing observation is that $(viii)$ and (x), which integrate only a single module (*e.g.*, PSD or APE), outperform their counterparts (v) and (vii), which further incorporate both the query particles and DSM. This raises questions about the efficacy of these two components when combined. To investigate, an additional experiment was conducted, comparing the first three basic modules combined in (xi) with the full version (xv). Notably, the full version, which incorporates query particles and DSM, achieves a further 3.99 % improvement in ATE over the combined model (xi). These findings underscore the critical importance of the query particle set in fully exploiting basic components, while also highlighting the necessity of a flexible optimization method like DSM to maximize their effectiveness.

Particle Number. The number of query motion particles

Config	PSD	ASC	APE	Particles	DSM	Unseen		
						ATE (m) ↓	T-RTE (m) ↓	D-TRE (m) ↓
(i)	-	-	-	-	-	6.48	5.53	0.41
(ii)	-	-	-	✓	-	7.10	4.61	0.42
(iii)	-	-	✓	✓	-	6.33	4.65	0.41
(iv)	-	-	-	✓	✓	6.13	4.61	0.38
(v)	✓	-	-	✓	✓	5.63	4.42	0.37
(vi)	✓	-	✓	✓	✓	5.39	4.48	0.35
(vii)	-	-	✓	✓	✓	6.05	4.69	0.38
(viii)	✓	-	-	-	-	5.58	4.41	0.37
(ix)	-	✓	-	-	-	6.00	4.70	0.39
(x)	-	-	✓	-	-	5.76	4.64	0.36
(xi)	✓	✓	✓	-	-	5.54	4.58	0.36
(xii)	✓	-	-	✓	-	7.03	4.35	0.43
(xiii)	✓	✓	-	✓	✓	5.41	4.53	0.37
(xiv)	-	✓	✓	✓	✓	5.79	4.52	0.37
(xv)	✓	✓	✓	✓	✓	**5.31**	**4.39**	**0.36**

Table 1: Ablation studies on the proposed module on RoNIN dataset.

Dataset	Test Subject	Metric	Method								
			SINS	PDR	RIDI	RoLSTM	RoTCN	RoResnet18	CTIN	TLIO	Ours
RIDI	Seen	ATE (m)	6.34	22.76	8.18	1.94	2.56	**1.64**	1.69	1.67	1.68
		T-RTE (m)	8.13	24.89	9.34	2.60	2.81	1.93	2.03	2.00	**1.91**
		D-RTE (m)	0.52	1.39	0.97	0.27	0.27	**0.21**	0.27	**0.21**	**0.21**
	Unseen	ATE (m)	4.62	20.56	8.18	2.24	2.15	1.76	2.15	1.85	**1.49**
		T-RTE (m)	4.58	31.17	10.51	2.70	1.95	1.71	1.84	1.82	**1.33**
		D-RTE (m)	0.36	1.19	1.09	0.31	0.23	0.21	0.27	0.23	**0.20**
RoNIN	Seen	ATE (m)	7.89	26.64	16.82	4.14	5.81	4.13	5.54	4.41	**3.78**
		T-RTE (m)	5.30	23.82	19.50	2.83	3.56	2.81	3.26	2.82	**2.68**
		D-RTE (m)	0.42	0.98	4.99	0.29	0.37	**0.26**	0.34	**0.26**	**0.26**
	Unseen	ATE (m)	7.62	23.49	15.75	6.90	7.19	5.95	6.89	6.77	**5.31**
		T-RTE (m)	5.12	23.07	19.13	4.46	5.15	4.53	4.95	4.69	**4.39**
		D-RTE (m)	0.43	1.00	5.37	0.42	0.47	**0.36**	0.43	0.39	**0.36**
OxIOD	Seen	ATE (m)	15.36	9.78	3.78	2.00	2.12	2.45	6.71	2.51	**1.86**
		T-RTE (m)	11.02	8.51	3.99	1.93	1.92	1.02	2.31	1.16	**0.94**
		D-RTE (m)	0.96	1.16	2.30	0.62	0.62	0.22	0.33	0.22	**0.21**
	Unseen	ATE (m)	13.90	17.72	7.16	2.03	2.00	1.06	2.34	1.03	**0.90**
		T-RTE (m)	10.51	17.21	7.65	1.40	1.35	**1.15**	1.53	1.26	1.32
		D-RTE (m)	0.89	1.10	2.62	0.62	0.61	**0.21**	0.28	0.22	0.22
IDOL	Seen	ATE (m)	21.54	18.44	9.79	3.68	4.66	2.70	3.15	2.90	**2.22**
		T-RTE (m)	14.93	14.53	7.97	3.78	5.58	2.45	3.05	2.63	**1.86**
		D-RTE (m)	1.07	1.14	0.97	0.43	0.53	0.27	0.34	0.30	**0.24**
	Unseen	ATE (m)	20.34	16.83	9.54	4.34	5.03	3.32	3.70	3.36	**3.00**
		T-RTE (m)	18.48	15.67	9.07	5.15	6.15	3.37	4.12	3.50	**2.85**
		D-RTE (m)	1.36	1.31	1.04	0.47	0.57	0.32	0.38	0.33	**0.28**

Table 2: Overall Trajectory Prediction Evaluation. The best result is shown in bold font.

plays a crucial role in extracting cross-modal interactions from context features and describing motion uncertainties thereafter. As shown in Fig.4, using J_{vel} and J_{ent} with average pooling after training cannot tolerate a higher number of particles, limiting the model's capacity. However, the introduction of DSM allows the model to effectively accommodate more particles, enhancing its ability to represent motion uncertainties. Our experiments reveal that 128 particles are optimal for modeling uncertainties across different velocity segments in RoNIN. Fewer than 128 particles may fail to cover enough motion modes for accurate velocity synthesis, while more than 128 particles could introduce redundancy, potentially diluting cross-modal information with unnecessary motion modes through their positional embeddings.

State-of-The-Art Performance The results presented in Tab. 2 exhibit a detailed evaluation of trajectory errors for various odometry methods across four benchmark datasets: RIDI, RoNIN, OxIOD, and IDOL, using three standard metrics: ATE, T-RTE, and D-RTE. Throughout the experiment, our proposed method consistently outperforms other SoTA approaches, particularly in its ability to generalize to unseen subjects. For example, in dynamic scenarios from the RoNIN dataset, where recording devices are freely held during movement, our method achieves an ATE of 5.31 m for unseen subjects, significantly outperforming RIDI (15.75 m)

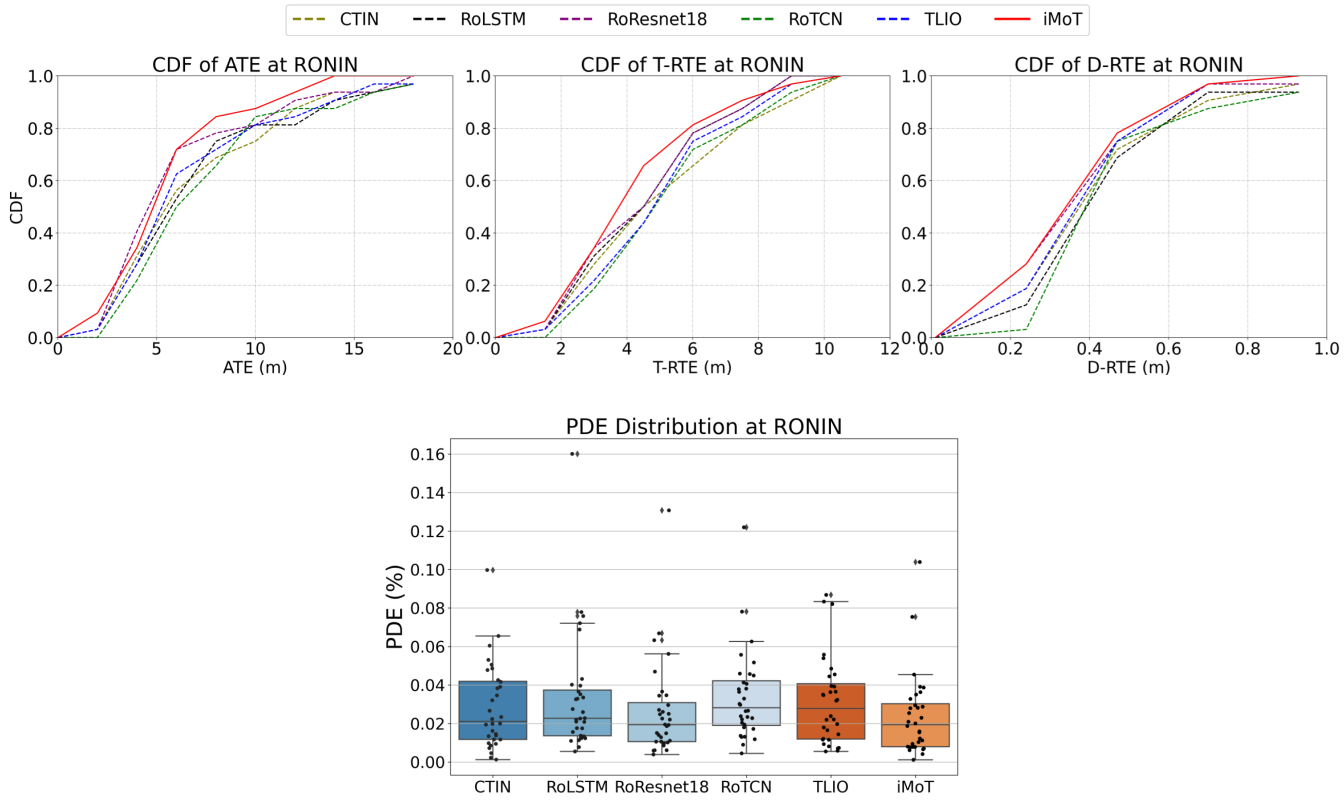

Figure 5: Cumulative Error Distributions (CDF) with three types of metric types, and boxplot of PDE on RoNIN dataset.

by 66.29%, and other robust methods that account for motion uncertainties, such as CTIN (6.89 m) and TLIO (6.77 m), by 22.93% and 21.57%, respectively. This trend of superior performance is consistent across all evaluated datasets in terms of D-RTE and T-RTE as well. For instance, on the IDOL dataset, our model demonstrated a 15.43% improvement in T-RTE and a 12.50% improvement in D-RTE compared to the second-best model, RoResnet18. These results can be attributed to our method's distinct advantage in effectively modeling motion uncertainties with learnable query motion particles. While other methods show a significant performance decline in unseen dynamic scenarios, the minimal error increase between seen and unseen subjects in our method highlights its generalization capability.

For stability verification, we further visualize the detailed metrics over the entire RoNIN dataset. As illustrated in Fig. 5, the cumulative distribution function (CDF) of the trajectory errors clearly shows that the proposed method consistently outperforms the others, as indicated by the uppermost positions of the red curve. Specifically, in 80% of the cases predicted by our method, the ATE remains below approximately 6.5 m, denoted as $P(X < 6.5) = 0.8$. In contrast, other methods only achieve an ATE upper bound of around 10 m for 80% of the examples. Furthermore, the network also exhibits the lowest position drift error (PDE %) with the highest confidence and fewest outliers, further highlighting its robustness and precision in trajectory prediction.

Conclusion

This paper introduces iMoT, an innovative transformer architecture designed to capture and model motion uncertainties within instantaneous velocity segments. Extensive experimental results demonstrated that PSD greatly enhances the encoding of complex temporal patterns, while the manipulation of the query particle set effectively learns and represents various motion modes, accounting for motion variability. Our approach not only improves trajectory reconstruction quality but also exhibits robust generalization across a wide range of dynamic scenarios, setting a new SoTA standard in handling motion uncertainty for odometry tasks.

Acknowledgements

This publication is part of the project MOSAIC: enhancement of MicrOfluidic Sensing with deep symbolic Artificial IntelligenCe with file number 19985 of the research programme Open Technology Programme which is (partly) financed by the Dutch Research Council (NWO).

This work made use of the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-6216.

References

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection

with transformers. In *European conference on computer vision*, 213–229. Springer.

Chen, C.; Lu, X.; Markham, A.; and Trigoni, N. 2018a. Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Chen, C.; Zhao, P.; Lu, C. X.; Wang, W.; Markham, A.; and Trigoni, N. 2018b. Oxiod: The dataset for deep inertial odometry. *arXiv preprint arXiv:1809.07491*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Herath, S.; Yan, H.; and Furukawa, Y. 2020. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In *2020 IEEE international conference on robotics and automation (ICRA)*, 3146–3152. IEEE.

Hyndman, R. J.; and Athanasopoulos, G. 2018. *Forecasting: principles and practice*. OTexts.

Janardhanan, J.; Dutta, G.; and Tripuraneni, V. 2014. Attitude estimation for pedestrian navigation using low cost mems accelerometer in mobile applications, and processing methods, apparatus and systems. US Patent 8,694,251.

Jimenez, A. R.; Seco, F.; Prieto, C.; and Guevara, J. 2009. A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. In *2009 IEEE International Symposium on Intelligent Signal Processing*, 37–42. IEEE.

Kourogi, M.; and Kurata, T. 2014. A method of pedestrian dead reckoning for smartphones using frequency domain analysis on patterns of acceleration and angular velocity. In *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014*, 164–168. IEEE.

Liu, W.; Caruso, D.; Ilg, E.; Dong, J.; Mourikis, A. I.; Daniilidis, K.; Kumar, V.; and Engel, J. 2020. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4): 5653–5660.

Rao, B.; Kazemi, E.; Ding, Y.; Shila, D. M.; Tucker, F. M.; and Wang, L. 2022. Ctin: Robust contextual transformer network for inertial navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 5413–5421.

Savage, P. G. 1998. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control, and dynamics*, 21(2): 208–221.

Shen, S.; Gowda, M.; and Roy Choudhury, R. 2018. Closing the gaps in inertial motion tracking. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 429–444.

Sun, S.; Melamed, D.; and Kitani, K. 2021. IDOL: Inertial deep orientation-estimation and localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 6128–6137.

Tian, Q.; Salcic, Z.; Kevin, I.; Wang, K.; and Pan, Y. 2015. An enhanced pedestrian dead reckoning approach for pedestrian tracking using smartphones. In *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 1–6. IEEE.

Titterton, D.; and Weston, J. L. 2004. *Strapdown inertial navigation technology*, volume 17. IET.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34: 22419–22430.

Yan, H.; Shan, Q.; and Furukawa, Y. 2018. RIDI: Robust IMU double integration. In *Proceedings of the European conference on computer vision (ECCV)*, 621–636.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.