

# StarIO: A Lightweight Inertial Odometry for Nonlinear Motion

Shanshan Zhang<sup>1,2</sup>, Siyue Wang<sup>1</sup>, Qi Zhang<sup>1</sup>, Liqin Wu<sup>1</sup>, Tianshui Wen<sup>1</sup>, Ziheng Zhou<sup>1</sup>,  
 Xuemin Hong<sup>1</sup>, Ao Peng<sup>1</sup>, Lingxiang Zheng<sup>1,\*</sup>, Yu Yang<sup>2,\*</sup>

**Abstract**—Inertial odometry (IO) is an attractive approach for consumer-grade localization. However, existing data-driven IO methods often suffer from significant drift under complex nonlinear motion patterns (e.g., turns), as they struggle to capture the nonlinear relationships between Inertial Measurement Unit (IMU) signals and motion states. To address this issue, we propose a lightweight IO model, StarIO. Specifically, we first apply the Star Operation to project IMU signals into a high-dimensional implicit nonlinear feature space, enabling effective extraction of the complex nonlinear motion characteristics that typically cause drift. We then capture contextual dependencies across both the temporal and channel dimensions to enhance trajectory estimation over long sequences. In addition, we introduce a multi-scale gated unit that fuses fine-grained local motion dynamics with contextual information to achieve a comprehensive representation of motion. Extensive experiments on six representative open-source datasets demonstrate that StarIO achieves a superior trade-off between model lightweightness and localization accuracy. For example, on the RoNIN dataset, our approach reduces the ATE by 5.21% compared to R-ResNet while using only 2.762M parameters.

## I. INTRODUCTION

An inertial measurement unit (IMU) typically comprises a gyroscope and an accelerometer, providing high-rate measurements of acceleration and angular velocity. Inertial odometry (IO) estimates platform motion directly from IMU signals and is a key technology enabling the widespread adoption of consumer-grade positioning systems [1]. Due to its advantages such as low cost, independence from external infrastructure, and strong data privacy, IO is indispensable for applications such as augmented reality, navigation for people with disabilities, and privacy-sensitive localization services [2]–[4].

Traditional IO derives motion estimates by integrating IMU signals based on Newtonian mechanics, but localization accuracy deteriorates rapidly due to error accumulation [5]. Incorporating additional physical priors (e.g., motion constraints or dynamic models) can partially reduce drift, but such priors often restrict the range of applicable scenarios and motion patterns [6]–[9]. In recent years, data-driven IO has substantially improved localization accuracy and broadened the scope of feasible applications [10]. However, these

This work is supported by Science and Technology Major Program of Fujian Province (No. 2022HZ026007) and partly supported by the Science and Technology Planning Project of Fujian Province (2022I0001).

<sup>1</sup>Department of Information and Communication Engineering, National and Local Joint Engineering Research Center of Navigation and Location-Based Services, Xiamen University, Xiamen 361005, China.

<sup>2</sup>Department of Electronic Science, State Key Laboratory of Physical Chemistry of Solid Surfaces, Xiamen University, Xiamen 361005, China.

\*Corresponding authors: Lingxiang Zheng (e-mail: lxzheng@xmu.edu.cn); Yu Yang (e-mail: yuyang15@xmu.edu.cn).

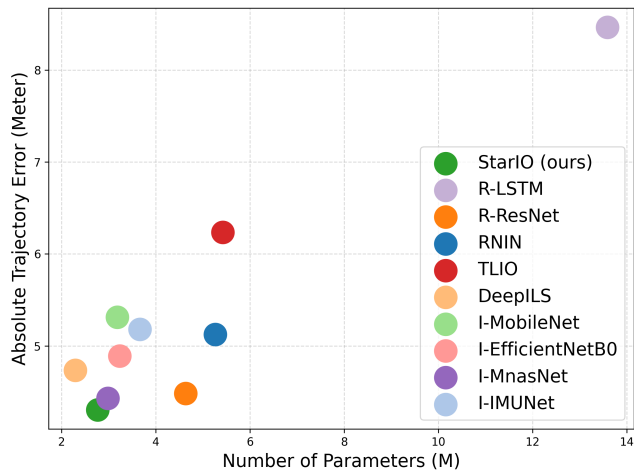


Fig. 1: Comparison of Absolute Trajectory Error (ATE) for different models on the RoNIN dataset. Points nearer the bottom-left indicate lower trajectory error and fewer parameters, reflecting a superior overall performance-efficiency trade-off.

methods commonly exhibit significant drift in the presence of complex nonlinear motion patterns such as turns [11]. We attribute this limitation to the typical deployment of IO on edge devices, where constrained computational resources and storage capacity necessitate lightweight models for IO [12]. Such models typically project IMU signals into a low-dimensional feature space with limited nonlinear representational capacity, making it difficult to model complex motion patterns. Therefore, a central challenge is to design a lightweight IO model that can efficiently project IMU signals—rich in motion information—into a high-dimensional nonlinear feature space so as to accurately capture complex nonlinear motion characteristics.

To address these challenges, this paper proposes StarIO, a lightweight IO model that incorporates the Star Operation [13]. Specifically, the Star Operation applies element-wise self-products on extracted features, thereby expanding the feature space and introducing nonlinear representations without increasing network depth. This projection facilitates subsequent extraction of rich contextual features and fusion of multi-scale motion characteristics. The main contributions of this work are as follows:

- We propose using the Star Operation to efficiently project IMU signals into an implicit high-dimensional nonlinear feature space, substantially enhancing the capacity to model complex nonlinear motion characteristics.
- We introduce a lightweight dual-wing attention mechanism that models contextual motion information along both the channel and temporal dimensions within the implicit high-

dimensional feature space.

- We design a multi-scale gated unit that effectively aggregates fine-grained local motion dynamics with contextual information to form a comprehensive motion representation.
- Extensive experiments on multiple open-source datasets demonstrate that StarIO achieves an excellent trade-off between localization accuracy and model size, as illustrated in Fig. 1.

## II. RELATED WORK

### A. Newtonian mechanics-based IO

Traditional IO relies on Newtonian kinematic principles to directly compute motion information [10]. These methods do not require large-scale IMU datasets or significant computational resources, but they are often limited in accuracy and robustness. For example, strapdown inertial navigation systems (SINS) use high-precision and costly IMUs to compute displacement via double integration [5], [14], [15]; however, persistent error accumulation renders long-range localization unreliable. To mitigate error accumulation, researchers have introduced physical priors to correct such errors. Pedestrian dead reckoning (PDR) assumes a regular walking pattern [7]. The zero-velocity update (ZUPT) technique is applied by mounting the IMU on the foot and enforcing zero-velocity constraints during stance phases to reduce system errors [6]. Coriolis-based heading estimation (CHE) builds on ZUPT by incorporating a magnetometer and exploiting the Coriolis effect for heading estimation to correct residual errors [8]. However, the introduction of physical priors often relies on strong assumptions about motion patterns, which limits the generalization and robustness of IO methods in complex motion scenarios [16].

### B. Data-driven IO

The recent introduction of machine learning approaches has further improved IO localization accuracy and expanded its applicability [3], [17]–[20]. RIDI [21] and PDRNet [22] first identify IMU placement and then apply dedicated network architectures for position inference. IONet [23] and RoNIN [16] are agnostic to IMU placement and instead use unified models to infer motion information directly from IMU signals, demonstrating a degree of robustness.

Building on the ResNet architecture proposed in RoNIN, TLIO [24] and LIDR [25] introduce the State-Cloning Extended Kalman Filter (SCEKF) and the Left-Invariant Extended Kalman Filter (LIEKF), respectively, to further reduce localization error. Similarly, RNIN [26], SCHNN [27], SSHNN [28], and CTIN [29] combine convolutional neural networks with LSTMs or attention mechanisms to capture contextual motion information. RIO [1] and EqNIO [30] design modular network components to address orientation bias in IMU signals.

To facilitate mobile deployment, IMUNet [12] adopts depthwise separable convolutions to create a lightweight IO, and DeepILS [31] further incorporates CBAM [32] to enhance contextual motion awareness. Because IO is often

deployed on mobile devices, data-driven IO methods tend to favor shallow architectures with small parameter counts. However, shallow networks typically struggle to map IMU signals into a high-dimensional nonlinear feature space capable of modeling complex nonlinear characteristics. Real-world IMU data are often highly intricate, which further challenges the design of effective models. Therefore, designing lightweight IO architectures that efficiently project IMU signals into high-dimensional nonlinear feature spaces and accurately model their complex nonlinear characteristics remains an important research challenge.

## III. PROPOSED METHOD

In this section, we present the overall pipeline of StarIO and then describe its core component—the Lightweight Transform Block (LTB), which comprises two submodules: the Dual-Wing Star Block (DWSB) and the Multi-Scale Gated Convolution Unit (MSGCU).

### A. Overall Architecture

As illustrated in Fig. 2, StarIO employs a Transformer-based encoder. Let  $\mathbf{X} \in \mathbb{R}^{C \times L}$  denote IMU signals collected over a temporal window, where  $C = 6$  corresponds to the three-axis angular velocities from the gyroscope and the three-axis accelerations from the accelerometer, and  $L$  is the number of samples in the window.

We first apply a one-dimensional convolutional layer with kernel size 3 along the temporal axis to adjust the channel dimension of  $\mathbf{X}$ . Within the main backbone, the network comprises four stages indexed by  $i \in \{1, 2, 3, 4\}$ . Stage  $i$  stacks  $N_i$  LTBs to progressively capture complex, global motion dynamics. Downsampling between stages is performed using one-dimensional convolutions with stride of 3.

Within each LTB, the DWSB applies the Star Operation [13] to project IMU signals into a high-dimensional, nonlinear feature space, enhancing representations of complex motion patterns and capturing global dynamics across both the channel and temporal dimensions. The output of the DWSB is then fed into the MSGCU, which fuses local fine-grained motion features with cross-channel dependencies, producing richer and more diverse motion representations.

Finally, StarIO maps features to the average velocity within the window via a multi-layer perceptron (MLP) and obtains displacement by temporal integration. We train the model with the mean squared error (MSE) loss.

### B. Lightweight Transform Block

The Lightweight Transform Block (LTB) is the core module of StarIO and consists of two submodules: the DWSB and the MSGCU. To avoid the computationally expensive standard attention mechanism, we employ an efficient nonlinear feature-space mapping to achieve lightweight IO.

Let  $\mathbf{X}_i$  denote the input features to the  $i$ -th LTB. The processing within an LTB is expressed as

$$\mathbf{X}'_i = \mathbf{X}_i + \text{DWSB}(\mathbf{X}_i), \quad (1)$$

$$\mathbf{X}_{i+1} = \mathbf{X}'_i + \text{MSGCU}(\mathbf{X}'_i). \quad (2)$$

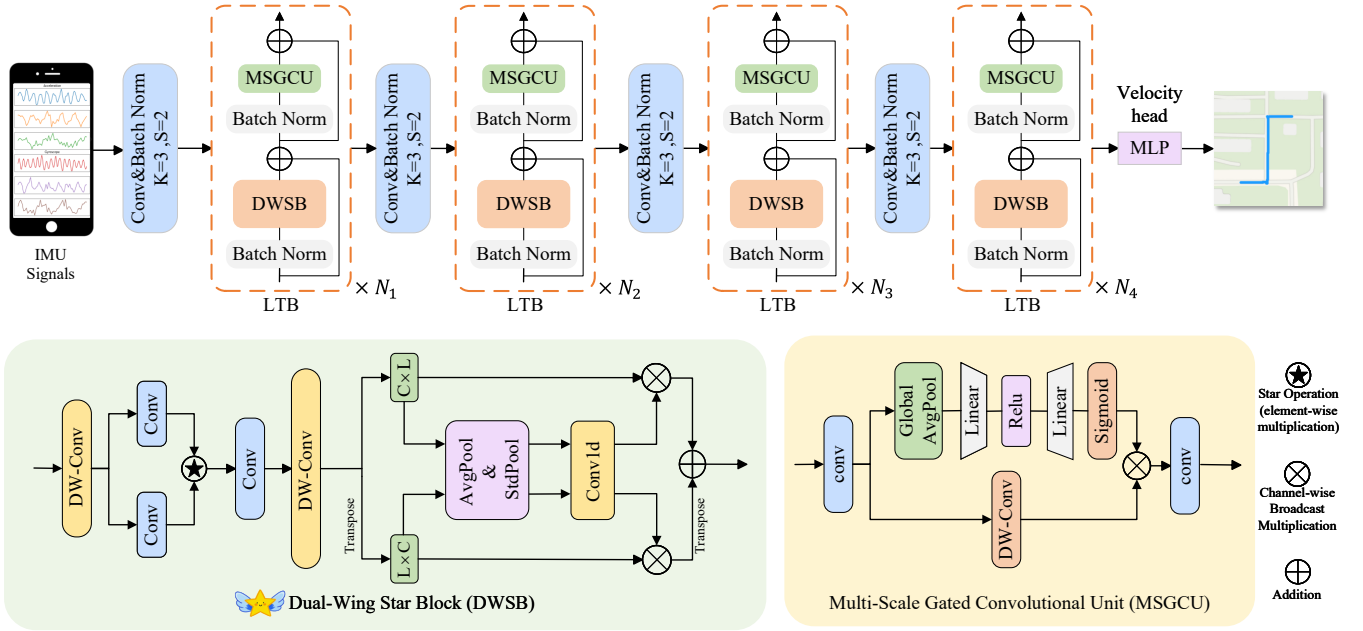


Fig. 2: The overall architecture of the proposed StarIO, which primarily consists of the Lightweight Transform Block (LTB) featuring the Dual-Wing Star Block (DWSB) and Multi-Scale Gated Convolution Unit (MSGCU).

Here,  $\mathbf{X}'_i$  and  $\mathbf{X}_{i+1}$  are the outputs of the DWSB and MSGCU modules, respectively.

**Dual-Wing Star Block (DWSB).** In IO methods, deep neural networks typically rely on convolutional or fully connected layers for feature transformation, coupled with non-linear activation functions to enhance model expressiveness. However, such approaches exhibit limitations in modeling high-dimensional nonlinear representations. To address these limitations, we propose the Dual-Wing Star Block (DWSB), which combines the high-dimensional nonlinear mapping capacity of the Star Operation with a lightweight dual-wing attention mechanism.

1) *High-Dimensional Nonlinear Mapping.* IMU signals are inherently nonlinear and dynamic, requiring strong nonlinear representational capacity from the model. To effectively extract and model complex motion features, we first project the IMU signals into a high-dimensional implicit nonlinear feature space using the Star Operation. To simplify the derivation, we consider the IMU signal at a single time step as  $\mathbf{x}_t \in \mathbb{R}^{C \times 1}$ . Its affine transformation can be formulated as:

$$\mathbf{W}'_1 \mathbf{x}'_t = \mathbf{W}_1 \mathbf{x}_t + \mathbf{B}_1 \quad (3)$$

where  $\mathbf{W}_1 \in \mathbb{R}^{M \times C}$  is the weight matrix ( $M$  denotes the output dimension),  $\mathbf{B}_1 \in \mathbb{R}^{M \times 1}$  is the bias,  $\mathbf{W}'_1 = [\mathbf{W}_1, \mathbf{B}_1] \in \mathbb{R}^{M \times (C+1)}$ , and  $\mathbf{x}'_t = [\mathbf{x}_t; 1] \in \mathbb{R}^{(C+1) \times 1}$ . Although this transformation increases the dimensionality when  $M > C$ , the nonlinear capacity still largely depends on the subsequent activation functions.

To enhance nonlinear representational capacity, we introduce the Star Operation, which facilitates feature interactions via element-wise multiplication. This enables an efficient mapping to a high-dimensional implicit nonlinear space, which is equivalent to implicit feature dimensionality expansion.

Specifically, we first apply a depthwise convolution to extract local features for each channel. Let the output of the depthwise convolution at a specific time step  $t$  be  $\mathbf{x}_t \in \mathbb{R}^{C \times 1}$ , where  $C$  is the number of channels. Then, two parallel pointwise convolution branches are employed to capture inter-channel contextual information. Their outputs are linearly transformed as follows:

$$\mathbf{W}'_2 \mathbf{x}'_t = \mathbf{W}_2 \mathbf{x}_t + \mathbf{B}_2 \quad (4)$$

$$\mathbf{W}'_3 \mathbf{x}'_t = \mathbf{W}_3 \mathbf{x}_t + \mathbf{B}_3 \quad (5)$$

where  $\mathbf{x}'_t = [\mathbf{x}_t; 1] \in \mathbb{R}^{(C+1) \times 1}$  is the augmented feature vector. Here,  $\mathbf{W}_2, \mathbf{W}_3 \in \mathbb{R}^{M \times C}$  and  $\mathbf{B}_2, \mathbf{B}_3 \in \mathbb{R}^{M \times 1}$  denote the weight matrices and bias terms, respectively. We define the augmented weight matrices as  $\mathbf{W}'_2 = [\mathbf{W}_2, \mathbf{B}_2]$  and  $\mathbf{W}'_3 = [\mathbf{W}_3, \mathbf{B}_3]$ , where  $\mathbf{W}'_2, \mathbf{W}'_3 \in \mathbb{R}^{M \times (C+1)}$ .

Instead of simple addition, we apply the Star Operation (element-wise multiplication) [13] to project the features into a high-dimensional implicit nonlinear space:

$$\begin{aligned} & (\mathbf{W}'_2 \mathbf{x}'_t) * (\mathbf{W}'_3 \mathbf{x}'_t) \\ &= \left( \sum_{i=1}^M w'_{2,i} x'_i \right) * \left( \sum_{j=1}^M w'_{3,j} x'_j \right) \\ &= \sum_{i=1}^M \sum_{j=1}^M w'_{2,i} w'_{3,j} x'_i x'_j \\ &= \underbrace{\beta_{(1,1)} x'_1 x'_1 + \dots + \beta_{(4,5)} x'_4 x'_5 + \dots + \beta_{(M,M)} x'_M x'_M}_{M(M+1)/2 \text{ terms}} \end{aligned} \quad (6)$$

This operation implicitly maps the measurements into a nonlinear feature space with an approximate dimensionality of  $\frac{M(M+1)}{2} \approx \left(\frac{M}{\sqrt{2}}\right)^2$ , which holds when  $M \gg 2$ . Among these components, only  $M$  terms are linearly related to the input features  $\mathbf{x}_t$ , while the remaining terms represent

TABLE I: Trajectory Prediction Accuracy of StarIO and Baseline Models. The best results are highlighted in bold.

Dataset	Metric	R-ResNet	R-LSTM	RNIN	I-IMUNet	I-MobileNet	I-MnasNet	I-EfficientNetB0	DeepILS	TLIO	StarIO	StarIO Improvement over R-ResNet
		Performance (meters)										
RoNIN	ATE	4.165	8.103	4.781	4.755	4.883	4.077	4.629	4.374	5.805	<b>3.948</b>	5.210%
	RTE	3.724	4.836	3.705	3.736	3.713	3.572	3.681	3.665	4.276	<b>3.554</b>	4.565%
RIDI	ATE	2.080	3.839	2.103	2.486	2.500	2.290	2.379	2.047	2.563	<b>2.033</b>	2.260%
	RTE	2.383	3.589	2.445	2.644	2.584	2.257	2.406	2.395	2.929	<b>2.274</b>	4.574%
RNIN	ATE	1.844	3.182	1.605	2.174	3.119	3.142	2.648	2.478	2.428	<b>1.455</b>	21.095%
	RTE	1.836	3.726	1.593	2.114	1.568	1.842	1.613	1.499	1.979	<b>1.370</b>	25.381%
IMUNet	ATE	6.428	8.108	6.445	9.130	6.846	8.593	6.479	6.918	10.240	<b>4.964</b>	22.775%
	RTE	4.802	6.381	4.463	6.404	4.809	5.609	4.458	5.638	5.735	<b>3.815</b>	20.554%
TLIO	ATE	1.333	2.145	1.300	1.354	1.369	1.373	1.724	1.258	1.832	<b>1.108</b>	16.879%
	RTE	3.705	3.685	3.627	4.060	0.842	<b>0.806</b>	0.959	0.818	5.162	3.145	15.115%
OxIOD	ATE	4.336	7.282	2.780	2.325	2.381	2.163	1.982	1.696	3.855	<b>1.484</b>	65.775%
	RTE	1.595	2.301	1.082	1.135	1.142	1.103	0.979	1.100	1.270	<b>0.856</b>	46.332%
Parameters (M)		4.635	13.589	5.259	3.659	3.179	2.979	3.232	<b>2.291</b>	5.424	2.762	40.410%
FLOPs (M)		38.252	57.473	72.032	18.838	34.559	25.256	28.831	<b>15.287</b>	39.436	25.122	34.325%
Peak Memory (MB)		22.835	70.755	25.490	38.274	67.257	<b>13.073</b>	25.631	17.746	60.026	20.458	10.409%

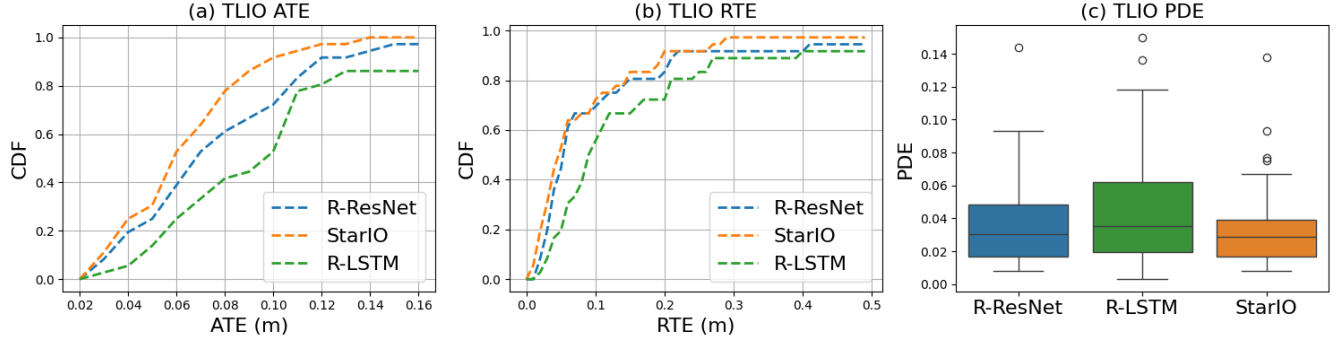


Fig. 3: Performance comparison of StarIO, R-ResNet, and R-LSTM evaluated on the TLIO dataset. (a) and (b) show the CDF of ATE and RTE, respectively, while subfigure (c) presents a box plot of PDE.

higher-order nonlinear interactions—effectively analogous to a kernel function mapping. When stacking  $n$  such layers, the resulting feature dimensionality increases to  $\left(\frac{M}{\sqrt{2}}\right)^{2n}$ , enabling expressive nonlinear modeling even with shallow networks. Therefore, the Star Operation introduces a complementary nonlinearity by modeling feature interactions in a higher-dimensional space, augmenting the effect of conventional activation functions.

This process can be easily extended to handle measurement data over the entire time window  $\mathbf{X} \in \mathbb{R}^{C \times L}$ , yielding the output  $\mathbf{X}_{\text{star}} \in \mathbb{R}^{M \times L}$ .

2) *Dual-Wing Attention Mechanism.* Attention mechanisms offer superior capabilities for modeling global relationships. However, standard Transformer self-attention suffers from quadratic computational complexity with respect to measurement sequence length, making it inefficient for long sequences [33]. To address this issue, we propose a Dual-Wing Attention mechanism that captures global motion dynamics along both the temporal and channel dimensions in the high-dimensional implicit nonlinear feature space, while maintaining low computational overhead [34].

Focusing on the channel dimension, given high-dimensional features  $\mathbf{X}_{\text{star}}$ , we first apply adaptive average pooling (AvgPool) and adaptive standard deviation pooling (StdPool) along the temporal scale to extract contextual statistics along the channel dimension. These are combined using learnable parameters  $\mu$  and  $\xi$ :

$$\mathbf{X}'_c = \mu \cdot \mathbf{X}_{\text{star}}^{\text{Avg}} + \xi \cdot \mathbf{X}_{\text{star}}^{\text{Std}} \in \mathbb{R}^{M \times 1} \quad (7)$$

where  $\mathbf{X}_{\text{star}}^{\text{Avg}}$  and  $\mathbf{X}_{\text{star}}^{\text{Std}}$  represent the features obtained via AvgPool and StdPool, respectively. Subsequently, a 1D con-

volution is applied to  $\mathbf{X}'_c$  to generate the adaptive channel-wise weights  $W_c \in \mathbb{R}^{M \times 1}$ .

Similarly, by transposing  $\mathbf{X}_{\text{star}}$ , the same process is repeated along the temporal dimension to obtain  $W_t \in \mathbb{R}^{L \times 1}$ .

Finally, we integrate the attention weights from both dimensions to produce the Dual-Wing representation:

$$\mathbf{X}_{\text{DWS}} = W_c \otimes \mathbf{X}_{\text{star}} + (W_t \otimes \mathbf{X}_{\text{star}}^T)^T \in \mathbb{R}^{M \times L} \quad (8)$$

where  $\otimes$  represents channel-wise broadcasting multiplication. This mechanism effectively captures bidirectional contextual motion information in IMU signals at minimal computational overhead, resulting in substantial improvements in localization accuracy compared to baseline methods.

**Multi-Scale Gated Convolution Unit (MSGCU).** In standard Transformer architectures, the attention module is typically followed by a MLP consisting of linear layers and nonlinear activations to enhance the model’s representational capacity [35]. However, in our architecture the nonlinear representational capacity has already been effectively realized by DWSB. We therefore focus on multi-scale context modeling to aggregate rich contextual information across diverse receptive fields.

We propose the Multi-Scale Gated Channel Unit (MSGCU), which injects contextual motion information into the value branch for inter-channel dependency modeling and incorporates a local dynamic gating mechanism to capture fine-grained feature variations. The MSGCU fuses local and global cues within a unified channel-attention module and comprises two principal branches: the **value branch** and the **gating branch**.

1) *Value branch.* A depthwise 1-D convolution with kernel size 3 aggregates local motion-variation features. This branch

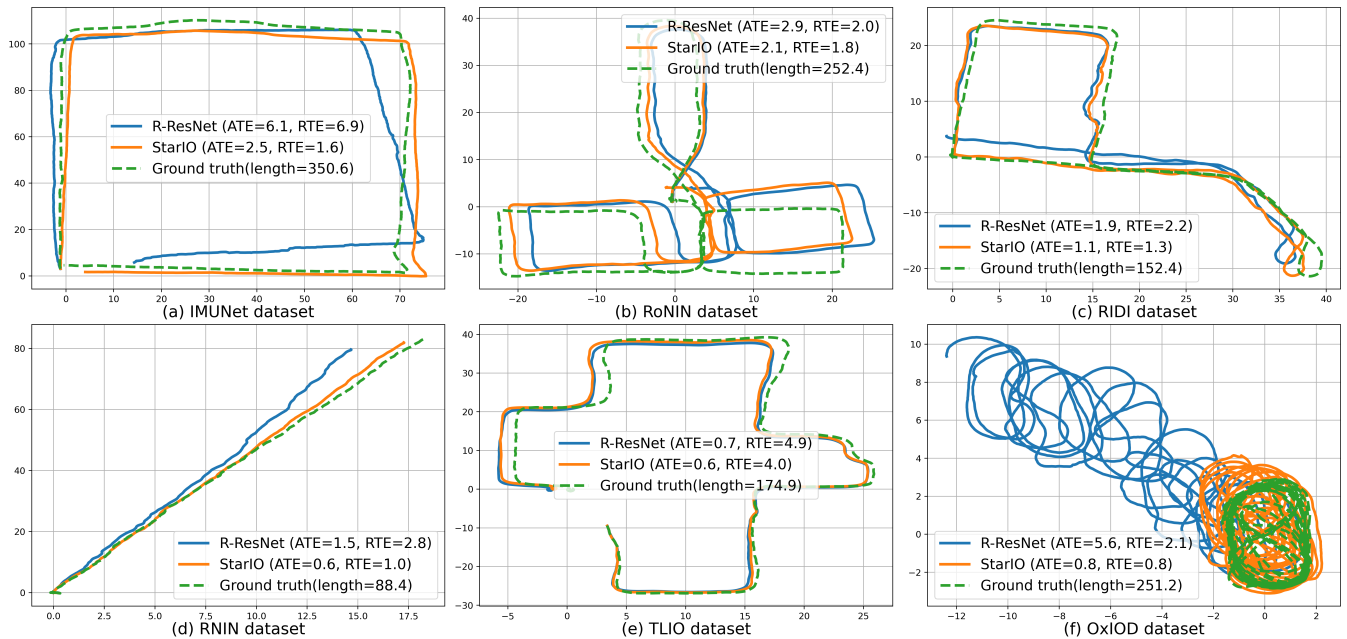


Fig. 4: Sample trajectories from six test datasets. The results compare the performance of the proposed StarIO model against the R-ResNet baseline. Values in parentheses indicate the ATE, RTE, and trajectory length, all measured in meters.

captures subtle variations across neighboring channels and enables flexible modulation based on local dependencies [36]. Such a mechanism is particularly well-suited to IMU signal data, which often exhibit local continuity in motion patterns.

2) *Gating branch.* Inspired by the Squeeze-and-Excitation module [37], this branch models global inter-channel dependencies. It first applies global average pooling to the input feature and then employs a two-layer fully connected network to explicitly model channel-wise interactions. A softmax activation is used to produce adaptive selection weights that modulate the local motion-variation features.

These two branches capture channel interactions at local and global scales, respectively. The gating output serves as a gating factor to modulate the aggregated local features from the value branch, thereby achieving multi-scale gated fusion. The resulting channel-attention mechanism improves inter-channel discriminability and enhances the modeling of local motion patterns, leading to improved localization accuracy under diverse pedestrian motion conditions.

## IV. EXPERIMENTS AND ANALYSIS

### A. Experimental Settings

**Datasets.** Extensive experiments were conducted on several publicly available inertial localization datasets, including RIDI [21], OxIOD [38], RoNIN [16], RNIN [26], IMUNet [12], and TLIO [24]. All datasets were randomly split into training, validation, and testing sets at an 8:1:1 ratio.

**Baseline.** Numerous studies indicate that data-driven IO methods achieve significantly higher accuracy than Newtonian-based IO [11], [29], [39]. Therefore, we focus on comparing against data-driven IO approaches. The baseline models include R-ResNet [40] and R-LSTM [41] introduced in RoNIN [16]; the neural networks proposed in RNIN

[26] and TLIO [24]; and lightweight networks such as DeepILs [31], along with I-IMUNet [12], I-MobileNet [42], I-MnasNet [43], and I-EfficientNetB0 [44] (all proposed in IMUNet [12]).

**Metrics.** To comprehensively evaluate localization performance, we use Absolute Trajectory Error (ATE) [19], Relative Trajectory Error (RTE) [19], and Position Drift Error (PDE) [29] as the primary metrics. All of these metrics are positively correlated with localization error, where lower values indicate better performance. Trajectory visualizations are also provided to further illustrate the localization accuracy of each method.

**Implementation Details.** The network configuration is set to  $[N_1, N_2, N_3, N_4] = [2, 2, 2, 2]$ , yielding a total depth comparable to ResNet-18. All experiments were implemented using PyTorch 2.5.1 and CUDA 12.2. The models were trained using the Adam optimizer, with a maximum of 100 epochs, an initial learning rate of  $10^{-4}$ , and automatic termination triggered when the learning rate decayed to  $10^{-6}$  to prevent overfitting. All training and inference were conducted on an NVIDIA RTX 3090 (24GB) GPU. Other configurations are kept consistent with TLIO [24].

### B. Comparisons with Baselines

**Quantitative Comparison.** Table I presents the comparison of ATE and RTE performance on six benchmark datasets. Overall, StarIO outperforms the existing open-source baselines under most testing conditions. On the RNIN dataset, which includes complex motion patterns such as walking, running, stair climbing, and random shaking under high-noise conditions [26], StarIO achieves an ATE of 1.455 m, outperforming RNIN (ATE: 1.605 m). Similarly, on the IMUNet dataset, which involves challenging outdoor scenarios with significant disturbances [12], StarIO achieves

TABLE II: The ablation study results on six dataset. The best result is highlighted in bold.

Dataset	Metric	R-ResNet	StarIO (w/o MSGCU)	StarIO
		Performance (meters)		
RoNIN	ATE	4.165	4.471	<b>3.948</b>
	RTE	3.724	3.572	<b>3.554</b>
RIDI	ATE	2.080	2.118	<b>2.033</b>
	RTE	2.383	2.446	<b>2.274</b>
IMUNet	ATE	6.428	6.577	<b>4.964</b>
	RTE	4.802	4.601	<b>3.815</b>
TLIO	ATE	1.333	1.254	<b>1.108</b>
	RTE	3.705	3.525	<b>3.145</b>
RNIN	ATE	1.844	<b>1.250</b>	1.455
	RTE	1.836	<b>1.213</b>	1.370
OxIOD	ATE	4.336	1.502	<b>1.484</b>
	RTE	1.595	0.878	<b>0.856</b>
Parameters (M)		4.635	<b>2.252</b>	2.762
FLOPs (M)		38.252	<b>21.555</b>	25.122
Peak Memory (MB)		22.835	<b>9.800</b>	20.458

an ATE of 4.964 m, surpassing I-IMUNet (ATE: 9.130 m). These improvements are attributed to StarIO’s ability to map inertial data into a high-dimensional nonlinear feature space and to effectively model complex dynamic motions through multi-scale feature fusion.

Notably, StarIO has the second-lowest parameters (2.762M), the third-lowest peak GPU memory usage (20.458MB), and the third-lowest FLOPs (25.122M), comparable in scale to DeepILS. Yet, it achieves significantly lower ATE and RTE, and substantially outperforms existing methods in most scenarios—striking a favorable balance between accuracy and lightweight design.

Additionally, we conducted a visual analysis of the error distribution. Due to space limitations, only the results on the largest dataset (TLIO) are presented. As shown in Fig. 3, the cumulative distribution function (CDF) of trajectory errors demonstrates that StarIO outperforms R-ResNet and R-LSTM across all accuracy levels. In Fig. 3(a) and (b), the orange curve (corresponding to StarIO) consistently occupies the upper-left region of the plot, indicating that 80% of test samples achieve an ATE below 0.08 m ( $P(X < 0.08) = 0.8$ ), whereas R-ResNet and R-LSTM yield values closer to 0.1 m for the same percentage. Furthermore, the boxplot in Fig. 3(c) shows that StarIO attains the lowest median and smallest dispersion in PDE, further validating its stability and advantage in high-precision localization tasks.

**Qualitative Visualization.** To intuitively illustrate the localization performance, we visualize reconstructed trajectories for several representative test samples in Figure 4, comparing StarIO with R-ResNet. In relatively simple scenarios (e.g., Fig. 4(d)), both models accurately reconstruct the trajectory. However, in more complex cases shown in Fig. 4(a), (b), (c), (e), and (f), R-ResNet exhibits noticeable drift, particularly in sequences involving frequent turns or loops where heading errors accumulate rapidly. In contrast, StarIO more closely matches the ground-truth trajectory, demonstrating superior modeling of complex motion patterns and better capture of long-term dependencies. This further confirms its reliability and practicality for real-world applications.

### C. Ablation Studies

To assess the individual and combined contributions of each module in the proposed architecture, we designed two ablation configurations: full StarIO and StarIO without MSGCU (StarIO (w/o MSGCU)). We conducted systematic experiments across multiple datasets, and Table II summarizes the results. StarIO (w/o MSGCU) has 2.252 M parameters—slightly less than half of R-ResNet’s 4.635 M—yet achieves localization accuracy comparable to that of R-ResNet. Adding MSGCU increases StarIO’s parameter count by only 0.510 M and yields the lowest localization error in most cases. In summary, the ablation studies validate the individual effectiveness of DWSB and MSGCU as well as their complementarity.

### V. CONCLUSION

We present StarIO, a lightweight architecture that enhances the modeling of complex nonlinear motion patterns while maintaining a small parameter footprint. Extensive experiments on multiple publicly available pedestrian-inertial datasets demonstrate the effectiveness of StarIO. However, the applicability of the proposed method to non-pedestrian platforms—such as ground vehicles, unmanned aerial vehicles (UAVs), or mobile robots—remains an open question and warrants further investigation.

### REFERENCES

- [1] X. Cao, C. Zhou, D. Zeng, and Y. Wang, “Rio: Rotation-equivariance supervised learning of robust inertial odometry,” in *CVPR*, 2022, pp. 6604–6613.
- [2] D. Engelsman and I. Klein, “Information-aided inertial navigation: A review,” *IEEE Transactions on Instrumentation and Measurement*, vol. 72, no. 8507418, 2023.
- [3] R. Harle, “A survey of indoor inertial positioning systems for pedestrians,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013.
- [4] M. Freydn and B. Or, “Learning car speed using inertial sensors for dead reckoning navigation,” *IEEE Sensors Letters*, vol. 6, no. 9, pp. 1–4, 2022.
- [5] P. G. Savage, “Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 208–221, 1998.
- [6] R. P. Suresh, V. Sridhar, J. Pramod, and V. Talasila, “Zero velocity potential update (zupt) as a correction technique,” in *Proceedings of the 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2018, pp. 1–8.
- [7] A. Nayak, A. Eskandarian, Z. Doerzaph, and P. Ghorai, “Pedestrian trajectory forecasting using deep ensembles under sensing uncertainty,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 9, pp. 11317–11329, 2024.
- [8] Z. Li, Z. Deng, Z. Meng, and P. Zhang, “Coriolis-based heading estimation for pedestrian inertial localization based on mems mimu,” *IEEE Internet of Things Journal*, vol. 11, no. 16, pp. 27509–27517, 2024.
- [9] A. Brajdic and R. Harle, “Walk detection and step counting on unconstrained smartphones,” in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 225–234. [Online]. Available: <https://doi.org/10.1145/2493432.2493449>
- [10] A. K. Panja, C. Chowdhury, and S. Neogy, “Survey on inertial sensor-based ils for smartphone users,” *CCF Transactions on Pervasive Computing and Interaction*, vol. 4, no. 3, pp. 319–337, 2022.
- [11] S. M. Nguyen, L. D. Tran, D. Viet Le, and P. J. M. Havinga, “iMoT: Inertial Motion Transformer for Inertial Navigation,” *arXiv e-prints*, p. arXiv:2412.12190, Dec. 2024.

- [12] B. Zeinali, H. Zanddizari, and M. J. Chang, "Imunet: Efficient regression architecture for inertial imu navigation and positioning," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, no. 2516213, 2024.
- [13] X. Ma, X. Dai, Y. Bai, Y. Wang, and Y. Fu, "Rewrite the stars," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 5694–5703.
- [14] M. Yanling, X. Zhi, X. Li, L. Jianye, and Y. Dequan, "An improved method of sins/cns integrated navigation algorithm based on dual quaternions with gyro error corrected online," in *2017 29th Chinese Control And Decision Conference (CCDC)*, 2017, pp. 179–184.
- [15] Q. Liang and M. Liu, "An automatic site survey approach for indoor localization using a smartphone," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 191–206, 2020.
- [16] S. Herath, H. Yan, and Y. Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3146–3152.
- [17] Y. Qiu, C. Xu, Y. Chen, S. Zhao, J. Geng, and S. Scherer, "Airio: Learning inertial odometry with enhanced imu feature observability," *IEEE Robotics and Automation Letters*, pp. 1–8, 2025.
- [18] S. Zhao, S. Zhou, R. Blanchard, Y. Qiu, W. Wang, and S. Scherer, "Tartan imu: A light foundation model for inertial positioning in robotics," *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22 520–22 529, 2025. [Online]. Available: <https://api.semanticscholar.org/CorpusID:280089012>
- [19] C. L. Gentil, F. Tschopp, I. Alzugaray, T. Vidal-Calleja, R. Siegwart, and J. Nieto, "Idol: A framework for imu-dvs odometry using lines," 2020, arXiv:2008.05749.
- [20] X. Li, K. Li, J. Liu, and R. Gao, "Smartphone-based indoor pedestrian tracking via transformer," in *Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2024, pp. 1280–1285.
- [21] H. Yan, Q. Shan, and Y. Furukawa, "Ridi: Robust imu double integration," in *ECCV*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 641–656.
- [22] O. Asraf, F. Shama, and I. Klein, "Pdrnet: A deep-learning pedestrian dead reckoning framework," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 4932–4939, 2022.
- [23] C. Chen, X. Lu, A. Markham, and N. Trigoni, "Ionet: Learning to cure the curse of drift in inertial odometry," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [24] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, "Tlio: Tight learned inertial odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5653–5660, 2020.
- [25] D. Yang, H. Liu, X. Jin, J. Chen, C. Wang, X. Ding, and K. Xu, "Enhancing vio robustness under sudden lighting variation: A learning-based imu dead-reckoning for uav localization," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4535–4542, 2024.
- [26] D. Chen, N. Wang, R. Xu, W. Xie, H. Bao, and G. Zhang, "Rninvio: Robust neural inertial navigation aided visual-inertial odometry in challenging scenes," in *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2021, pp. 275–283.
- [27] Y. Wang, H. Cheng, and M. Q.-H. Meng, "Spatiotemporal co-attention hybrid neural network for pedestrian localization based on 6d imu," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 1, pp. 636–648, 2023.
- [28] Y. Wang, H. Cheng, A. Zhang, and M. Q.-H. Meng, "From imu measurement sequence to velocity estimate sequence: An effective and efficient data-driven inertial odometry approach," *IEEE Sensors Journal*, vol. 23, no. 15, pp. 17 117–17 126, 2023.
- [29] B. Rao, E. Kazemi, Y. Ding, D. M. Shila, F. M. Tucker, and L. Wang, "Ctin: Robust contextual transformer network for inertial navigation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 5, pp. 5413–5421, Jun. 2022. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/20479>
- [30] R. K. Jayanth, Y. Xu, Z. Wang, E. Chatzipantazis, K. Daniilidis, and D. Gehrig, "EqNIO: Subequivariant neural inertial odometry," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: <https://openreview.net/forum?id=C8jXEugWkq>
- [31] O. Tariq, B. Dastagir, M. Bilal, and D. Han, "Deepils: Towards accurate domain invariant aiot-enabled inertial localization system," *IEEE Internet of Things Journal*, pp. 1–1, 2025.
- [32] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [33] X. Chen, H. Li, M. Li, and J. Pan, "Learning a sparse transformer network for effective image deraining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 5896–5905.
- [34] Y. Yu, Y. Zhang, Z. Cheng, Z. Song, and C. Tang, "Mca: Multidimensional collaborative attention in deep convolutional neural networks for image recognition," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107079, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0952197623012630>
- [35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv e-prints*, p. arXiv:2010.11929, Oct. 2020.
- [36] D. Shi, "Transnext: Robust foveal visual perception for vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 17 773–17 783.
- [37] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:140309863>
- [38] C. Chen, P. Zhao, C. X. Lu, W. Wang, A. Markham, and N. Trigoni, "Oxiod: The dataset for deep inertial odometry," *arXiv preprint arXiv:1809.07491*, 2018.
- [39] S. Herath, D. Caruso, C. Liu, Y. Chen, and Y. Furukawa, "Neural inertial localization," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6594–6603.
- [40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [42] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv e-prints*, p. arXiv:1704.04861, Apr. 2017.
- [43] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2820–2828.
- [44] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.