Metrol. Meas. Syst., Vol. 32 (2025) No. 2, pp. 1–19
DOI: 10.24425/mms.2025.154345

METROLOGY AND MEASUREMENT SYSTEMS
Index 330930, ISSN 0860-8229
www.metrology.wat.edu.pl

NEURAL INERTIAL NAVIGATION METHOD FOR WHEELED ROBOTS BASED ON SELF-SUPERVISED LEARNING

Fengrong Huang[1)], Mengqi Gao[1)], Qinglin Liu[2)], Min Gao[1)]

1) School of Mechanical Engineering, Hebei University of Technology, Tianjin 300400, China
2) National Key Laboratory of Electromagnetic Space Security, Tianjin 300308, China (✉ *lql980423@163.com*)

Abstract

Low-cost Micro-Electro-Mechanical System Inertial Measurement Units (MEMS-IMUs) are plagued by large, complex, and variable errors. Traditional strap-down inertial navigation systems that utilize MEMS-IMUs are unable to meet the positioning requirements of wheeled robots. Although inertial navigation based on deep learning has been explored, it necessitates a substantial amount of carefully selected and labelled data, resulting in high costs. Consequently, this paper proposes a self-supervised neural inertial navigation method for wheeled robots that solely depends on MEMS-IMU data. Firstly, a representation learning model is established to extract general IMU features for self-supervised denoising. Subsequently, an intelligent framework employing contrastive learning is adopted to explore the latent information of the IMU and acquire the motion state of the robot. Specific motion state information is regarded as observations, and an invariant extended Kalman filter (IEKF) is applied for information fusion to enhance positioning accuracy. Experiments conducted on public datasets demonstrate that, in the absence of additional ground truth values, the Absolute Trajectory Error (ATE) and Temporal Relative Trajectory Error (T-RTE) of the proposed method are 20.23% and 30.71% lower than those of supervised learning-based methods, respectively. The proposed method offers a more cost-effective and practical solution for the development of inertial navigation technology for wheeled robots.

Keywords: MEMS-IMU, inertial navigation, self-supervised learning, wheeled robot navigation, motion state recognition.

1. Introduction

Inertial navigation utilizes gyroscopes and accelerometers to measure the angular and linear motion of a carrier in real time. It autonomously calculates the position, velocity, and attitude of the carrier without any external reference or additional sensors. Although the positioning error of high-precision inertial navigation systems is small, they are large in size, expensive, and require a long initialization process. In contrast, MEMS-IMUs have been widely used in the inertial navigation tasks of wheeled robots due to their advantages such as low cost, small size, and low power consumption [1–3].

Article history: received January 13, 2025; revised March 17, 2025; accepted April 23, 2025; available online May 15, 2025.

To address the problem that *Micro-Electro-Mechanical System-Inertial Measurement Units* MEMS-(IMUs) cannot be directly applied to navigation due to their large errors and complex error sources, the inertial navigation field usually adopts the methods of calibration and compensation based on physical/mathematical modelling to address sensor errors or the cumulative errors in the inertial navigation solution process. However, such methods not only require dedicated calibration equipment or observations from other sensors, but also the methods based on mathematical modelling, which require dedicated calibration equipment or observations from other sensors, are difficult to fully approximate the real sensor characteristics and motion states [4, 5], resulting in low efficiency and poor versatility. In recent years, with the rapid expansion of database scale and the enhancement of computer computing power, deep learning technology has flourished, and researchers have begun to explore the potential of using large amount of data to generate data-driven models. For example, the studies in [6, 7] achieved very good results by using ground truth data such as those from higher-precision IMUs or attitudes as references and calibrating the noise of inertial sensors and reducing the drift of inertial navigation systems with deep neural network models. *Tight Learned Inertial Odometry* (TLIO) [8] uses a residual network to regress the displacement increment and uncertainty of the carrier and combines it with the *Extended Kalman Filter* (EKF) to obtain accurate attitude and 3D positioning. *Lightweight Learned Inertial Odometer* (LLIO) [9] has implemented lightweight learning based on TLIO, making it more suitable for mobile devices. *Robust Inertial Navigation System on Wheels* (RINS-W) [10] and Symmetrical-Net [11] identify the special motion states of the carrier through neural networks and use the constraint information of these special motion states as observations. Through Kalman filtering for information fusion, the positioning accuracy of the navigation system is improved.

With extensive application of deep learning in various fields, deep learning has a great potential in the inertial navigation field. However, in the inertial navigation applications of wheeled robots, most deep-learning methods rely on supervised learning, and data preparation faces numerous challenges. For public datasets like KAIST [12], S3E [13], and *Fusionportablev2* [14], data collection tasks are challenging. They need to cover precise data from multiple scenarios and working conditions, which, in turn, requires a large investment of human resources, material resources, and time. Data annotation also demands complex procedures and specialized knowledge to ensure accuracy and consistency. In terms of hardware, GPS signals are prone to interference in complex scenarios, which affects the continuous collection of data. Besides, high-precision IMUs have large volumes, which is not conducive to the design of compact robots. Adding devices such as vision sensors will increase costs and complicate the system design. At the software level, different sensors have diverse data characteristics and coordinate systems, making processing and calibration complex. Furthermore, coordinate transformation involves heavy computations, and high-precision timestamp alignment is required. Minor errors can impact system performance. In summary, hardware limitations and software complexity impede the practical application of deep learning in inertial navigation. Therefore, exploring neural inertial navigation methods that do not rely on real-world data is of great significance.

Consequently, this paper, using only raw MEMS-IMU data, proposes a self-supervised neural inertial navigation method for wheeled robots. The network model of this method is based on *Transformer Bidirectional Encoder Representation* (BERT) and consists of three parts: IMU denoising, motion state recognition, and IEKF-based information fusion. In motion state recognition, pseudo-labels and contrastive learning are added to help the network extract features. By denoising IMU data and accurately identifying motion states, the accuracy of inertial navigation based on MEMS-IMU is significantly improved. The main contributions of this paper are as follows:

1. Innovation in Self-supervised Denoising: In response to the fact that supervised-learning-based IMU denoising methods rely on high-precision IMU data or other types of labelled

data, this paper proposes a *Self-supervised Learning* (SSL) model for IMU denoising. Through two key components, masked IMU modelling and next-moment IMU prediction, it realizes the denoising of IMU data based on SSL. This approach circumvents the limitations imposed by reliance on external data, enhancing the generality and autonomy of denoising.

2. Innovation in Motion State Recognition£ºIn the field of robot motion state recognition, traditional methods such as RINS-W and Symmetrical-Net use neural networks to identify special motion states but rely on a large amount of labelled data for supervised learning. The intelligent framework proposed in this paper integrates the ideas of pseudo-labels and contrastive learning. Without real data, it can accurately obtain the motion states of the robot in different time periods by exploring the latent information of the IMU through a contrastive learning algorithm.

2. System design

2.1. Coordinate frame and symbol

Inertial navigation measures the angular and linear motions of the carrier through an IMU fixed on the vehicle. Under given certain initial conditions, the attitude, velocity and position of the mobile platform relative to the starting point (R_0, v_0, p_0) are obtained through navigation solution. As shown in Fig. 1, the carrier coordinate frame b is a coordinate frame solidly attached to the vehicle, which is denoted by $(\cdot)^b$. In this application scenario, it is assumed that the carrier coordinate frame is already aligned with the IMU coordinate system, and the effects of the Earth's rotation and Coriolis acceleration are ignored. The navigation coordinate frame is the reference coordinate frame. R is the rotation matrix from the carrier frame to the navigation frame. $(\cdot)_n$ is the data corresponding to time n, and $(\hat{\cdot})$ as the estimated value. Data corresponding to the frames from the 1-st to the n-th is denoted by $(\cdot)_{1,n}$.

2.2. System overview

The neural inertial navigation system for wheeled robots uses raw IMU data as input. Relying on SSL, it outputs position, attitude, and velocity estimations. Comprising three key parts, as shown in Fig. 2, it starts with IMU denoising. Here, the network predicts masked IMU data and next-sequence values to clean up random noise in the original data.

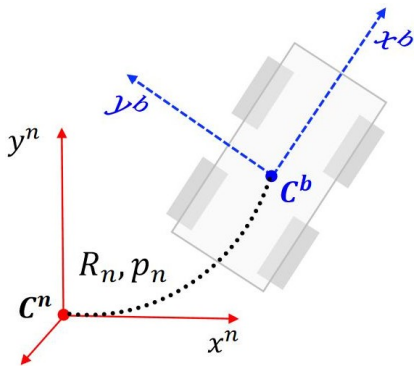


Fig. 1. Coordinate frame definition.

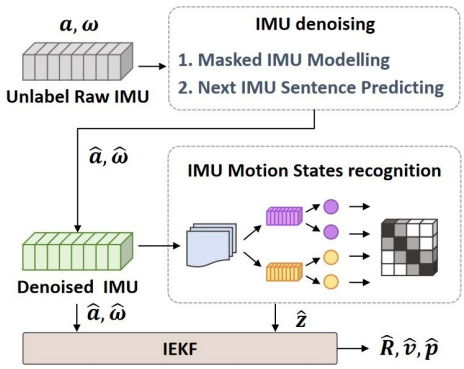


Fig. 2. Proposed system structure block diagram.

Next, the motion state recognition step focuses on distinguishing the categories of motion states. By segmenting denoised IMU data, labelling with pseudo-labels, and constructing sample sets, it trains the network to distinguish different motion states.

Finally, the IEKF part obtains more accurate positioning information of the wheeled robot by fusing the denoised IMU data and the motion state categories.

3. Network model based on SSL

3.1. IMU Denoising

3.1.1. Masked IMU modelling

BERT is an effective SSL model in natural language processing. It uses the bidirectional transformer model and can better understand the context in the continuous measurements of IMU. BERT has two pre-training tasks: *Masked Language Modelling* (MLM) and *Next Sentence Prediction* (NSP). This paper adapts them as *Masked IMU Modelling* (MIM) and *Next IMU Prediction* (NIP), with corresponding BERT models BERT_{MIM} and BERT_{NIP} and network parameters α and β. By masking, reconstructing, and predicting data, the network improves its understanding of IMU noise and feature representation learning, enhancing its denoising ability. As shown in Fig. 3, the BERT network's main architecture has 4 stacked transformer encoders and an additional decoder. Unlabelled IMU data is first contaminated with Gaussian noise, then normalized, and finally 15% of its values are randomly masked. The processed data needs to undergo data transformation via the projection function and then undergo layer normalization again to further restrict the statistical variance. Before the data enters the encoder, it is also necessary to perform positional encoding on the sequence data so that the model can capture long-distance dependencies based on positional information.

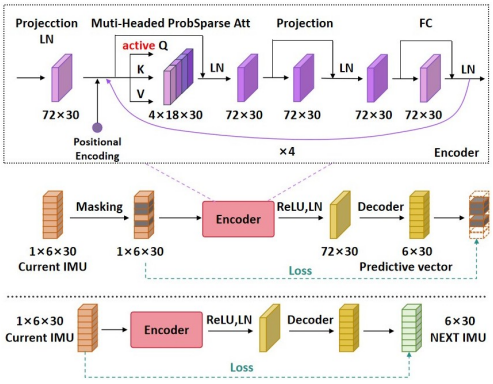


Fig. 3. Network structure diagram in IMU denoising.

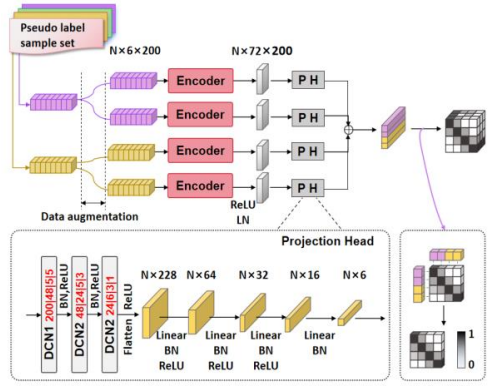


Fig. 4. Construction method of motion state recognition network structure.

The encoder consists of multi-head self-attention layers and feed-forward neural networks, with each sub-layer followed by residual connection and layer normalization. A multi-head prob-sparse self-attention mechanism [15] is used to reduce the time and space complexity of traditional transformers and speed up model training. It probabilistically samples only part of keys K and queries Q for calculation, multiplying attention weights by a normalization factor to keep the sum

at 1. The decoder is a linear layer with input and output dimensions of (72, 6). In the masked IMU modelling task, 15% of IMU sequence values are randomly masked. After the decoder generates the prediction vector, the values at masked positions are obtained, and the associated loss is calculated

$$\ell_{\mathrm{MIM}} = \mathrm{Loss}(f(\mathrm{BERT}_{\mathrm{MIM}}(U\alpha)mask\_position)U^{M}), \tag{1}$$

where $U \in R^{L\times 6}$ denotes an IMU sequence of sequence length L, U^M represents the IMU measurements at the masked position, *mask_position* represents the mask position, and $f(\cdot)$ is a function extracting the corresponding value of the masking position from the prediction vector.

3.1.2. Next IMU sequence prediction

The original NSP task of BERT is to determine whether the second sequence is the subsequent sentence of the first sequence when two sequences are provided. Thus, the original NSP task of BERT is modified. When the IMU sequence of the current time period is given, the IMU sequence of the next time period is directly predicted to enhance the generation ability of the model. The next moment IMU sequence corresponding to U is denoted by U^F, and $U_L = U_1^F$, meaning that the last frame of the current IMU sequence is equal to the first frame of the future IMU sequence. As the training progresses, the last frame of the reconstructed IMU sequence of $\mathrm{BERT}_{\mathrm{MIM}}$ should be equal to the first frame of the future IMU sequence generated by $\mathrm{BERT}_{\mathrm{NIP}}$. Furthermore, the estimated values corresponding to (2) and (3) are equal:

$$\hat{U}_L^{\mathrm{MIM}} = \mathrm{BERT}_{\mathrm{MIM}}(U, \hat{\alpha})_L, \tag{2}$$

$$\hat{U}_L^{\mathrm{NIP}} = \mathrm{BERT}_{\mathrm{NIP}}(U, \hat{\beta})_1, \hat{U}_L^{NIP+MIM} = \mathrm{BERT}_{\mathrm{NIP}}(\mathrm{BERT}_{\mathrm{MIM}}(U, \hat{\alpha}), \hat{\beta})_1. \tag{3}$$

Equation (2) is the IMU estimation equation for the operation at the actual denoising time, and U_L^{gt} represents the ground truth of IMU measurements, then the systematic error at the actual denoising time is:

$$\mathrm{Error}_{(1)} = U_L^{gt} - \hat{U}_L^{\mathrm{MIM}} = (U_L^{gt} - U_L) + (U_L - \hat{U}_L^{\mathrm{MIM}}), \tag{4}$$

where $(U_L^{gt} - U_L)$ is the error between the IMU outputs and the ground truth, and $(U_L - \hat{U}_L^{\mathrm{MIM}})$ is the $\mathrm{BERT}_{\mathrm{MIM}}$'s estimation error. Similarly, the error $\mathrm{Error}_{(2)}$ corresponding to (3) is:

$$\mathrm{Error}_{(2)} = (U_L^{gt} - U_L) + (U_L - \hat{U}_L^{\mathrm{NIP}}). \tag{5}$$

Since the model $\mathrm{BERT}_{\mathrm{MIM}}$ is utilized to denoise the original IMU data, it is necessary to minimize $\mathrm{Error}_{(1)}$. However, when the ground truth U_L^{gt} of the IMU are unknown, training using (2) can only reduce $(U_L - \hat{U}_L^{\mathrm{MIM}})$ and cannot eliminate the measurement noise of the IMU. Therefore, in order to offset the impact of the unknown U_L^{gt} on the training process, The following transformation will be made to obtain:

$$\min(\mathrm{Error}_{(1)}) = \min(\mathrm{Error}_{(1)} - \mathrm{Error}_{(2)} + \mathrm{Error}_{(2)}). \tag{6}$$

Further simplification of the formula gives:

$$\min(\mathrm{Error}_{(1)}) = \min([\hat{U}_L^{\mathrm{NIP}} - (\hat{U}_L^{\mathrm{MIM}} - U_L^{gt}) - (U_L]) + \min((U_L - \hat{U}_L^{\mathrm{NIP}})), \tag{7}$$

$$\min((U_L - \hat{U}_L^{\mathrm{NIP}}) \rightarrow \ell_{\mathrm{NIP}} = \mathrm{Loss}(\mathrm{BERT}_{\mathrm{NIP}}(U, \beta), U^F). \tag{8}$$

By observing (7), it is known that during network training, when the loss function is set in the manner of (8), the second half part of (7) can be minimized. As for the former part, a relationship between $\mathrm{BERT}_{\mathrm{MIM}}$, $\mathrm{BERT}_{\mathrm{NIP}}$, and U_L needs to be established. The loss function is designed as follows to endow the network model with the ability to denoise the IMU:

$$\langle \hat{\alpha}, \hat{\beta} \rangle = \ell_{\mathrm{MIM}} + \ell_{\mathrm{NIP}} + \mathrm{Loss}(\mathrm{BERT}_{\mathrm{NIP}}(\mathrm{BERT}_{\mathrm{MIM}}(U\alpha), \beta), U^F). \tag{9}$$

3.2. Motion State Recognition of Wheeled Robots

The well-trained network in Section 3.1 is utilized to denoise the original IMU data, and the measurement of the denoised accelerometer $\hat{a}$ and gyroscope $\hat{\omega}$ are obtained. The denoised IMU data will be employed to train the motion state recognition network. In this section, the four specific motion states of the wheeled robot and the design principle of the motion state recognition method will be introduced in detail.

3.2.1. Specific motion states

Four distinct specific motion states are considered, and their validity is encoded in the following binary vector z_n, where 1 represents the emergence of a corresponding state of motion:

$$z_n = \left(z_n^{\mathrm{VEL}}, z_n^{\mathrm{ANG}}, z_n^{\mathrm{LAT}}, z_n^{\mathrm{UP}} \right) \in \{0, 1\}^4, \tag{10}$$

where z_n is the motion state, z_n^{VEL} is the zero-velocity state, z_n^{ANG} is the zero angular velocity state, and z_n^{LAT} and z_n^{UP} represent the zero lateral and vertical velocity states, respectively. The latter two assumptions effectively ensure the long-term estimation accuracy. The lateral and vertical velocities should be expressed in the carrier coordinate frame. Table 1 shows the output characteristics of IMU data corresponding to typical motion states. It is worth noting that when the wheels stop, zero-speed does not imply zero angular velocity, and the two must be distinguished [10]. As shown in Fig. 5, in the lateral zero-speed state, the lateral acceleration stabilizes at a value close to zero (not always zero) for a certain period of time, making it a recognition feature for lateral zero speed.

Table 1. Typical motion states of wheeled robots.

motion state	IMU features	
z_n^{VEL}	$z_n^{\mathrm{VEL}} = 1 \Rightarrow \begin{cases} v_n \approx 0 \\ R_n a_n + g \approx 0 \end{cases}$	(11)
	$z_n^{\mathrm{VEL}} = 1 \Rightarrow \left\| a_{1,n} \right\|_2 \approx 9.8$	(12)
z_n^{ANG}	$z_n^{\mathrm{ANG}} = 1 \Rightarrow \omega_n \approx 0$	(13)
z_n^{LAT}	$z_n^{\mathrm{LAT}} = 1 \Rightarrow v_n^{\mathrm{LAT}} \approx 0$	(14)
	$z_n^{\mathrm{LAT}} = 1 \Rightarrow \left\| a_{2,n}^{\mathrm{LAT}} - a_{1,n-1}^{\mathrm{LAT}} \right\|_1 \approx 0$	(15)
z_n^{UP}	$z_n^{\mathrm{UP}} = 1 \Rightarrow v_n^{\mathrm{UP}} \approx 0$	(16)
	$z_n^{\mathrm{UP}} = 1 \Rightarrow a_{1,n}^{\mathrm{UP}} \approx 0$	(17)

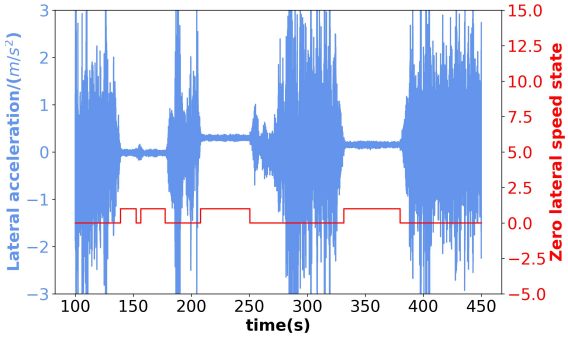


Fig. 5. Lateral acceleration characteristics when wheeled robots are the state of zero lateral speed.

3.2.2. Motion state recognition network structure

The motion state recognition network structure still employs the BERT network structure, but with certain adjustments. As depicted in Fig. 4, "$*|*|*|*$" represents the input dimension, output dimension, convolution kernel and dilation coefficient of the one-dimensional convolutional neural network, respectively. The decoder is modified to a projection head, which functions as the motion state recognizer. It is worth noting that motion state recognition requires highly distinguishable features. However, during the process of IMU denoising, the network, in order to minimize reconstruction losses and reconstruct sequences, tends to extract features with high coincidence. Therefore, the parameters trained in the denoising process cannot be utilized in the motion recognition step. Moreover, as the feature dimensions output by the encoder are more similar to the size of the generated sequence, it is necessary to further compress and extract the features generated by the encoder. First, a three-layer dilated convolutional network is used to fuse the features, and the range of the receptive field of the network is changed by adjusting the size of the convolution kernel and the dilation coefficient. Finally, the final category features are generated through four fully connected layers.

The absence of precise motion state labels in this article makes it infeasible to construct a loss function by minimizing the loss between the true and predicted labels. To address this issue, the concept of contrastive learning is adopted, where the similarity of the same category is higher. Contrastive loss is then employed for training the model. The specific approach is stated as follows: with sample pairs as the unit, we aim to maximize the similarity of pairs that belong to the same category and minimize the similarity of pairs from different categories. As depicted in Fig. 6a, when the zero lateral velocity is selected as a positive sample, all other motion states that are distinct from the zero lateral velocity are regarded as negative samples.

Before entering the network, data augmentation is carried out on both positive and negative samples, generating two new sets of sequences. The neural network converts the IMU time series into corresponding category probability distributions. Determining whether a sample belongs to the same category can be regarded as a binary classification problem. *Binary Cross Entropy* (BCE) is used as the loss function for optimization:

$$L_{ij} = \mathrm{BCE}(r_{ij}, s_{ij}) = -r_{ij} \cdot \log(s_{ij}) - (1 - r_{ij}) \cdot \log(1 - s_{ij}), \tag{18}$$

where r_{ij} represents the true value of the label for the binary classification problem, and back-propagation is utilized to optimize the parameters of the entire model.

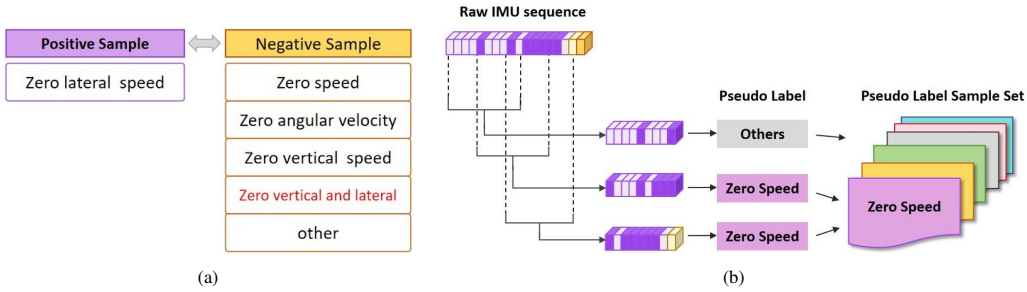


Fig. 6. a) Method of dividing positive and negative samples; b) method for constructing pseudo-labelled sample sets.

3.2.3. Construction of pseudo-sample set

Wheeled robots' motion states far exceed the four above, with numerous hard-to-classify categories. Thus, determining whether IMU sequences are positive or negative samples without labels is crucial for the algorithm. Section 3.2.1 details typical IMU output features in different motion states. Using these, most time series can be classified and pseudo-labels can be created. Though unreliable pseudo-labels misclassify some data into the "other" category, they ensure that different feature data belong to different categories, thus enabling contrastive learning. Despite misjudgements caused by using them, the "other" category in training enhances network discrimination, as will be demonstrated in subsequent experimental sections.

Figure 6b shows the pseudo-label sample set construction. In practice, as the four states may overlap and disrupt feature extraction, single-characteristic time periods should be extracted. For example, pure zero-velocity states without zero angular velocity. A separate category is made for concurrent zero lateral and vertical speeds to avoid overlap. Only when no typical features exist is it the "other" category. So, there are six categories in training. After training, the overlapping parts are re-assigned in category statistics.

3.3. Network training details

The method needs three independent neural nets. As IMU sensor readings vary in distribution, which affects model performance, appropriate normalization is needed before inputting sensor data as well as processing only accelerometer values because gyroscope readings are small. In motion state recognition, IMU data requires normalization and data augmentation (crucial for contrastive learning). Data augmentation methods include:

1. Data translation: Shifting the time series on the time axis by a fixed interval while keeping the same pseudo-label category.
2. Add noise: Adding random noise to the time series.
3. Data replacement: Replacing the original time series with a time series of the same pseudo-label category but different source.

The model was implemented with *PyTorch Lightning* on an RTX3060 GPU. Training used batch size 512, ADAM optimizer with initial learning rate 10^{-3}, and cosine annealing warm restart strategy (first restart after 500 iterations, subsequent restarts with more than 10 times previous iterations, both λ_1 and λ_2 set to 1). For denoising, the IMU sequence length was 30 with a 15% masking rate; for motion state recognition, it was set to 200.

Table 2. Modifying the dynamic model based on observed values.

Dynamic model (19)	**Propagation Step** (20)	**Update Step** (21)
$R_{n+1} = R_n \exp([\omega_n \mathrm{d}t]_\times)$ $v_{n+1} = v_n + (R_n a_n + g) \times \mathrm{d}t$ $p_{n+1} = p_n + v_n \times \mathrm{d}t$	$\hat{z}_n^{\mathrm{VEL}} = 1 \Rightarrow \begin{cases} v_{n+1} = v_n \\ p_{n+1} = p_n \end{cases}$ $\hat{z}_n^{\mathrm{ANG}} = 1 \Rightarrow R_{n+1} = R_n$	$\hat{z}_n^{\mathrm{VEL}} = 1 \Rightarrow \begin{bmatrix} R_{n+1}^T v_{n+1} \\ b_{n+1}^a - R_{n+1}^T g \end{bmatrix} = \begin{bmatrix} 0 \\ a_n \end{bmatrix}$ $\hat{z}_n^{\mathrm{ANG}} = 1 \Rightarrow b_{n+1}^{\omega} = \omega_n$ $\hat{z}_n^{\mathrm{LAT}} = 1 \Rightarrow v_n^{\mathrm{LAT}} = 0$ $\hat{z}_n^{\mathrm{UP}} = 1 \Rightarrow v_n^{\mathrm{UP}} = 0$

4. IEKF information fusion

The Extended Kalman Filter (EKF) has been widely utilized in information fusion in the field of inertial navigation [16]. The original EKF often lacked rigorous convergence proof and suffers from system divergence and inconsistency issues. In contrast, the IEKF has ameliorated these issues [17]. Therefore, the denoised IMU measurements are integrated into its dynamic model, and the constraint information of the detected motion state is used as observations. The IEKF is then employed to fuse this information to refine its estimates. The system state X_n is defined as:

$$X_n = [R_n, v_n, p_n, b_n^{\omega}, b_n^a], \tag{22}$$

where v_n and p_n represent the velocity and position under the navigation coordinate frame, respectively, $\mathrm{d}t$ is the time interval between two samplings, and the operator $[\cdot]_\times$ denotes a 3×3 skew-symmetric matrix. $b_n = [b_n^{\omega}, b_n^a]^T$ represents the biases of the gyroscope and accelerometer. Table 2 illustrates how the IEKF exploits the observed motion states during the propagation and update stages to modify the corresponding dynamic model of wheeled robots. For a more detailed description of the parameter setting methods and the iterative process, please refer to [18].

5. Experiments

In this section, to evaluate the effectiveness of the proposed method, experimental analyses were conducted on both the publicly available KAIST dataset and a self-collected dataset. Three main evaluation objectives were set to demonstrate that the proposed method generally approximates the accuracy of IMU motion recognition methods based on supervised learning:

1. Verifying that the masked IMU modelling and next IMU sequence prediction tasks can denoise the IMU.
2. Verifying that the motion states recognition method can accurately identify the specific motion states of the IMU.
3. Validating the accuracy of the final position estimate.

5.1. Data sources

The KAIST URBAN dataset is vehicle data collected in complex urban environments. For more detailed information about the dataset, please refer to [12]. The data was divided into a training set (*urban6-12*) and a test set (*urban13-17*). The original KAIST dataset provides medium-precision consumer-grade IMU data, which has higher accuracy compared to the more

cost-effective MEMS-IMU. Therefore, a certain amount of noise and bias was added to the original dataset to simulate the characteristics of MEMS-IMU. Specifically, Gaussian noise $N(0, 10^{-3})$ with random bias $B(0.015, 0.025)$ was added to the gyroscope data, and Gaussian noise $N(0, 10^{-2})$ with random bias $B(0.45, 0.55)$ was added to the accelerometer.

5.2. Baselines and metrics definitions

Due to the current lack of new research on SSL-based neural inertial navigation methods in the industry, to demonstrate the performance of the system, it is compared with three typical supervised learning-based neural inertial navigation methods applied in the field of wheeled robots, namely RINS-W [10], AI-IMU [18], and the method proposed by Guo [20]. Each of these methods encompasses some key components such as *Inertial Measurement Unit* (IMU) denoising, motion state recognition, and information fusion. Given the special application scenario of the method proposed in this paper, under the harsh condition of having no reference data at all, as long as it can be proven that the positioning error of this method has no significant difference compared with that of the supervised learning-based methods, the effectiveness of the method proposed in this paper can be fully demonstrated.

Secondly, as the method proposed in this paper consists of three modules, the performance of each module is evaluated separately. First, the performance of the IMU denoising module is assessed using the absolute attitude error. To evaluate the recognition performance of the motion state recognition network, the commonly used measurement $F_{\beta=0.5}$ indices for binary classification are utilized, with *precision* as precision, *recall* as recall, and β representing the relative weight of precision and recall. Finally, the overall positioning performance of the proposed method is judged using the *Absolute Trajectory Error* (ATE,m) and *Temporal Relative Trajectory Error* (T-RTE,m):

$$F_\beta = (1+\beta^2)\cdot precision \cdot recall / ((\beta^2 \cdot precision) + recall), \tag{23}$$

$$\mathrm{ATE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left\|p_n - \hat{p}_n\right\|^2}, \quad \mathrm{T-RTE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left\|p_{n+\Delta t} - p_n - (\hat{p}_{n+\Delta t} - \hat{p}_n)\right\|^2}. \tag{24}$$

5.3. IMU denoising performance analysis

Among the baseline methods, only Guo's method incorporates IMU denoising. Therefore, we compare the attitude angle errors obtained from the original gyroscope inertial navigation solution and those obtained with Guo's method respectively. As can be seen from Fig. 7 and Table 3, with reference information, this method reduces the average attitude estimation error by 67.17%. In contrast, without any additional external reference information, the method proposed in this paper reduces the average error by 52.06%. This is sufficient to demonstrate the effectiveness of the IMU denoising module.

5.4. Analysis of Motion State Recognition Performance

The RINS-W method includes a motion state recognition module, so the motion state recognition module of this paper is compared with that of the RINS-W method. Table 4 shows the accuracy rate of motion state recognition of the method in this paper. It can be seen that the motion recognition method based on SSL proposed in this paper is close to the accuracy rate of supervised learning. This experiment explored whether adding the "other-motion-states" data benefits the motion state recognition network's training. By using t-SNE for dimensionality reduction, we visualized the

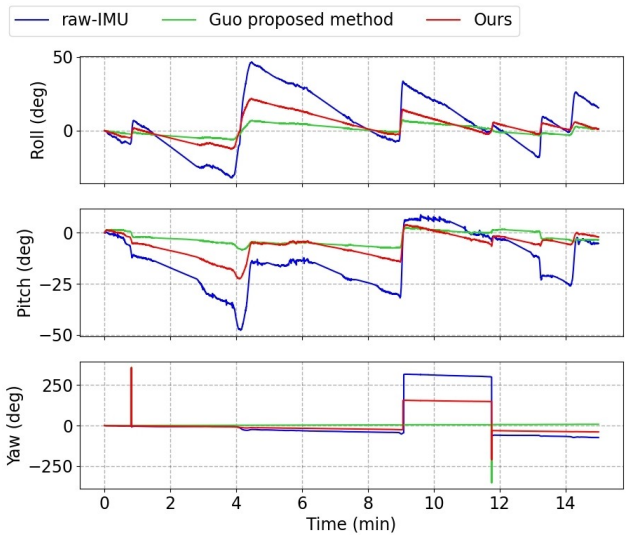


Fig. 7. AAE of *urban16*.

Table 3. AAE of different methods on the test set.

Method	Raw IMU	Guo	Ours
urban13	2.51	1.55	1.84
urban14	23.54	2.55	10.91
urban15	0.76	0.16	0.21
urban16	6.89	0.36	2.97
urban17	1.44	0.94	0.71
average	7.03	1.11	3.33

features in 2D (Fig. 8). Without the "other motion states" in training, the visualization result showed low discrimination among category features. The actual accuracy was only 40%-60%, which was far from the high accuracy rate shown in Table 4. Focusing only on four zero-velocity types, gyroscope and accelerometer data had small numerical differences. Pseudo-labels limited the amount of training data, and self-supervised denoising could not fully remove noise, making it difficult for the network to distinguish data. In contrast, the "other-motion-states" data has distinct numerical characteristics. Since contrastive learning amplifies data differences, including this data in the training process boosted the network's recognition accuracy and overall performance.

Table 4. F_β of motion state recognition, before and after ("|") applying the methods proposed in this paper and RINS-W, respectively.

Seq.	Zero Speed	Zero Angular	Zero Lateral	Zero Vertical	Difference
13	0.72 | 0.95	0.98 | 0.99	0.88 | 0.94	0.73 | 0.95	0.13
14	0 | 0	0.98 | 0.99	0.93 | 0.96	0.92 | 0.97	0.04
15	0.85 | 0.97	0.98 | 0.99	0 | 0	0.90 | 0.98	0.07
16	0 | 0	0.98 | 0.99	0.94 | 0.97	0.95 | 0.97	0.02
17	0 | 0	0.99 | 0.99	0 | 0	0.95 | 0.98	0.02

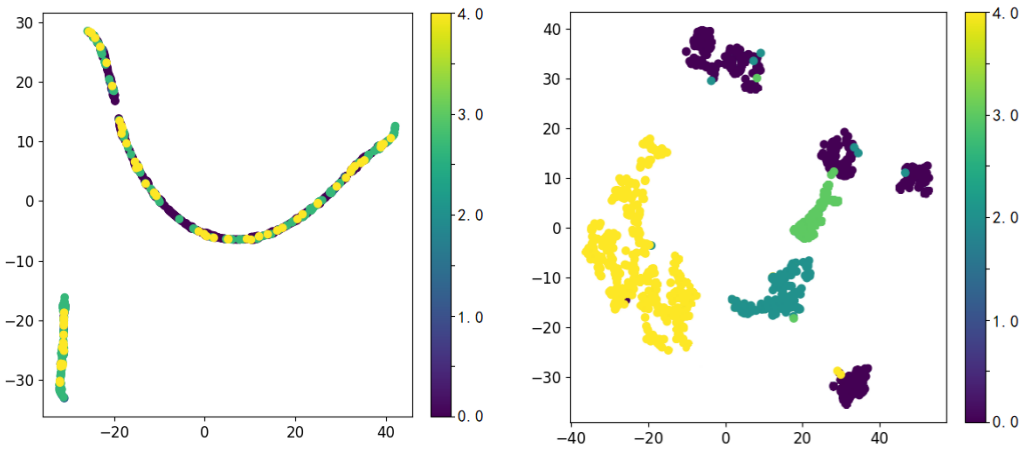


Fig. 8. Visualization analysis of t-SNE clustering effect: a) the network training does not include the category of "other"; b) the category of 'other' included during network training.

5.5. Analysis of position

Table 5, Table 6, Fig. 9 and Fig. 10 compare the proposed method with the baseline method, with ME representing maximum error. The proposed method is essentially on a par with RINS-W in terms of motion state recognition accuracy. Meanwhile, its additional denoising step further enhances the accuracy of the IMU. This advantage enables the proposed method to exhibit excellent comprehensive performance in most driving scenarios, showing a remarkable improvement compared with AI-IMU and RINS-W.

Table 5. Position errors of different methods on KAIST test set for urban13-16.

Method	**Supervised learning**				**Ours**	**Supervised learning**				**Ours**
	AI-IMU	**RINS-W**	**Guo**	**average**		**AI-IMU**	**RINS-W**	**Guo**	**average**	
	urban13					***urban14***				
ATE	296.76	98.40	256.25	217.14	152.54	767.26	275.77	238.00	427.01	123.31
T-RTE	1.86	0.50	0.87	1.08	0.61	1.64	1.04	1.04	1.04	0.54
RMSE	616.27	102.00	325.28	347.85	164.68	785.75	273.08	470.25	509.69	126.45
STD	262.60	27.95	49.10	113.21	30.35	425.65	116.25	252.39	264.76	54.15
ME	1265.77	143.42	416.54	608.58	151.20	1429.91	581.66	825.38	945.65	263.92
	urban15					***urban16***				
ATE	421.87	386.52	170.68	326.36	174.24	2131.74	1960.23	538.25	1543.41	768.77
T-RTE	1.43	2.01	0.69	1.38	0.96	2.00	1.59	0.66	1.42	0.97
RMSE	430.94	390.04	276.44	365.81	173.57	2145.22	2150.20	1202.33	1832.58	968.89
STD	190.85	275.80	117.28	194.64	120.56	1102.31	1086.23	636.97	941.84	488.40
ME	637.49	718.43	404.11	586.67	316.58	3695.45	3866.49	2067.72	3209.89	1736.25

Metrol. Meas. Syst., Vol. 32 (2025), No. 2, pp. 1–19
DOI: 10.24425/mms.2025.154345

Table 6. Position errors of different methods on KAIST test set for urban17 and all average.

Method	Supervised learning					Supervised learning				Ours
	AI-IMU	RINS-W	Guo	average	Ours	AI-IMU	RINS-W	Guo	average	
	urban17					All averages				
ATE	764.35	1286.63	201.45	750.81	583.53	876.40	801.51	280.93	451.91	360.48
T-RTE	1.91	3.74	0.59	2.08	1.76	1.73	1.82	0.79	1.40	0.97
RMSE	747.76	1301.57	775.68	941.67	560.36	945.19	843.38	609.99	673.47	398.79
STD	413.29	810.50	233.94	485.91	346.19	478.94	463.35	257.94	400.07	207.93
ME	1369.81	2565.12	775.68	1570.20	1108.78	1679.69	1575.02	897.89	1155.76	715.35

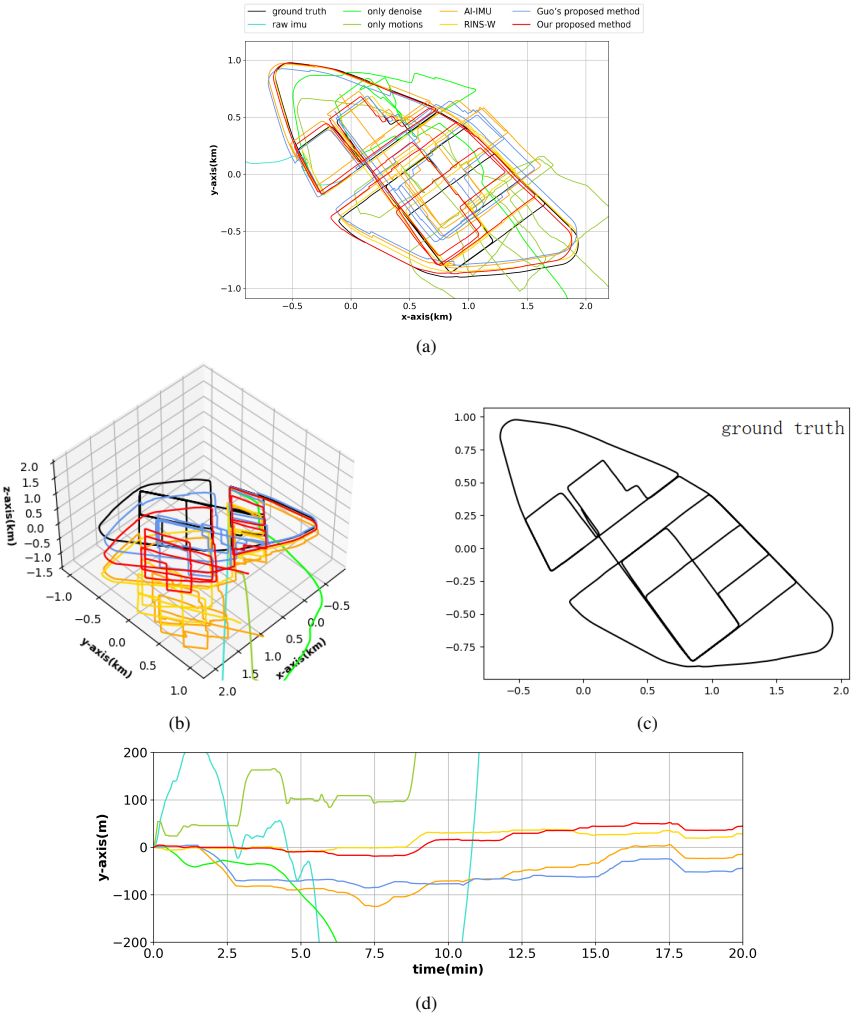


Fig. 9. Position comparison for different methods: a) the 2D trajectory map, b) 3D trajectory map, c) thumbnails of the ground truth, and d) position error maps of the x-axis for *urban16*.

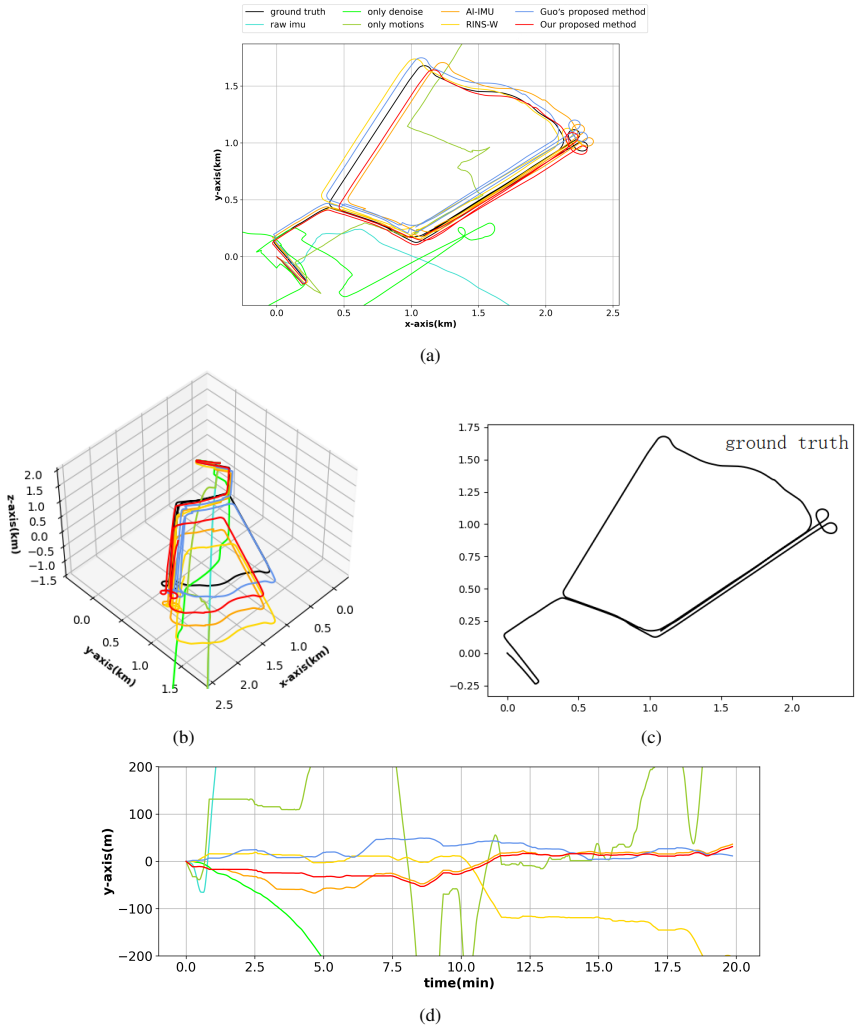


Fig. 10. Position comparison for different methods: a) the 2D trajectory map, b) 3D trajectory map, c) thumbnails of the ground truth, and d) position error maps of the x-axis for *urban17*.

Given that this study does not utilize any additional ground truth and the denoising ability of the IMU is limited, among the five groups of data, the performance of three data groups is superior to or close to that of Guo's method.

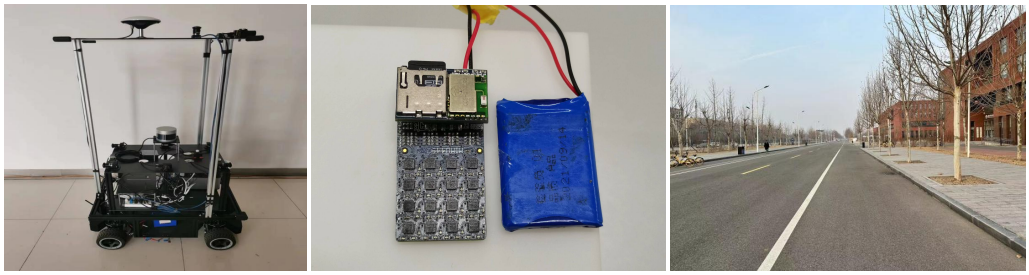
From the perspective of comprehensive evaluation results, compared with the average performance of supervised learning-based methods, the proposed method demonstrates significant superiority in the overall average performance of various trajectory error metrics (ATE, T-RTE, RMSE, STD, ME). Specifically, the errors in these metrics are reduced by 20.23%, 30.71%, 40.78%, 48.03%, and 38.11%, respectively. This outcome far exceeds the pre-set expectations. It fully verifies that the proposed method has outstanding advantages in terms of the accuracy, stability of trajectory estimation and error control, and can rival supervised learning-based IMU neural inertial navigation methods.

5.6. Real Scenario Evaluations

The method in this paper was also tested using self-made dynamic and static datasets. As shown in Fig. 11, the dynamic data was collected by wheeled robots in a real campus environment. The model of the MEMS-IMU is EPSON's M-G370, and the zero-bias instability of the gyroscope is 0.8°/h, with a sampling frequency of 150 Hz. The sampling frequencies of the *Global Navigation Satellite System* (GNSS) receiver and the camera were 1 Hz and 15 Hz respectively. The timestamps among different sensors were synchronized by software.

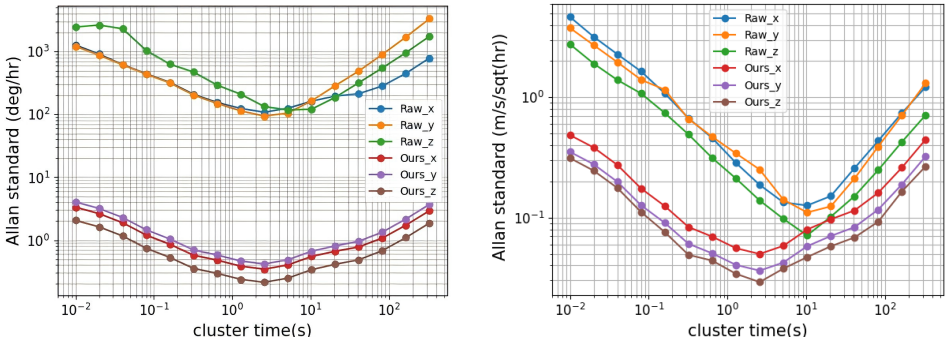
When conducting static data collection work, a low-cost multi-IMU array module of the MIMU48XC model produced by GT SILICON PVT LTD is selected. This module integrates 32 MEMS-IMUs of the ICM20948 model. The purpose of collecting the static data was to evaluate the denoising effect of the IMU through Allan variance analysis. Both the dynamic data and the static data were collected in 25 sets respectively. Among them, 20 sets were used for training and the remaining 5 sets were reserved for testing.

The Allan variance method is a commonly used error analysis approach in the field of inertial navigation [21]. Intuitively, the lower the curve is located, the smaller the error will be. Figure 12 and Table 7 demonstrate that the random errors of the gyroscope and accelerometer have been significantly improved before and after applying the method proposed in this paper.



(a) (b) (c)

Fig. 11. Partial real scene diagrams: (a) is the wheeled robot used in the experiment, (b) is the MEMS-IMU module for collecting static data and (c) is the outdoor collection scene.



(a) (b)

Fig. 12. Double logarithmic curve of Allan variance before and after denoising: (a) is gyroscope data and (b) is accelerometer data.

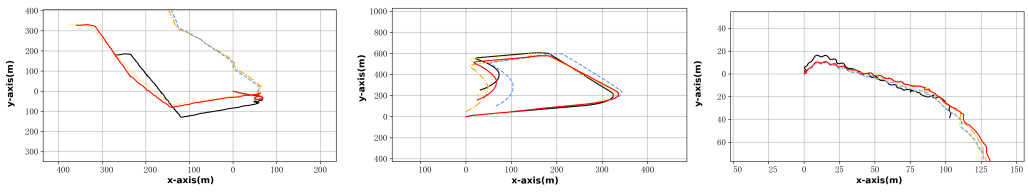
Table 8 and Fig. 13 display the comparison of the position estimation performance between the proposed method and the baseline methods on the self-made dynamic dataset. It is worth noting that even when the small wheeled robot and the cars used in the public datasets are traveling on the same flat road, the small robot will be more significantly affected by vibrations. Therefore, methods equipped with motion state recognition components will perform better in this scenario. Thanks to the crucial role of the motion state components, among the 5 sets of test data, for four of them, the position errors of the SSL-based method proposed in this paper are even smaller than the average value of the supervised learning-based methods, which demonstrates the effectiveness of the method proposed in this paper.

Table 7. Allan variance of output data before and after IMU denoising.

Noise type	**Raw IMU**			**Proposed method**		
	***x*-axis**	***y*-axis**	***z*-axis**	***x*-axis**	***y*-axis**	***z*-axis**
Angle random walk $(^{\circ}/\sqrt{hr})$	1.980	1.919	4.461	0.005	0.006	0.003
Rate random walk $(^{\circ}/\sqrt{hr^3}))$	4543.728	>5000	>5000	16.854	21.274	10.722
Gyro bias instabilities $(^{\circ}/hr)$	163.298	141.562	177.411	0.517	0.626	0.319
Velocity random walk $(m/s/\sqrt{hr})$	0.006	0.006	0.005	<0.001	<0.001	<0.001
Acceleration random walk $(m/s/\sqrt{hr}^3)$	7.001	7.54	4.096	2.566	1.859	1.535
Acceleration bias instabilities $(m/s/hr)$	0.190	0.166	0.107	0.075	0.0546	0.0438

Table 8. Position errors of different methods on self-made datasets.

Method		**Supervised learning**				**Ours**
		AI-IMU	**RINS-W**	**Guo**	**average**	
denoise | motion | IEKF		×|×|√	×|√|√	√|×|√		√|√|√
Test1	ATE	920.91	343.61	969.19	744.57	313.91
	T-RTE	1.50	0.54	1.59	1.21	0.50
Test2	ATE	1039.09	777.37	1023.05	946.50	779.09
	T-RTE	1.64	1.01	1.62	1.42	1.00
Test3	ATE	46.87	49.82	49.64	48.78	47.38
	T-RTE	0.41	0.42	0.41	0.41	0.41
Test4	ATE	175.16	111.46	169.58	152.07	119.32
	T-RTE	0.65	0.34	0.63	0.54	0.37
Test5	ATE	185.06	179.66	172.75	179.16	184.73
	T-RTE	1.21	1.21	1.16	1.19	1.17



(a) (b) (c)

Fig. 13. (a), (b) and (c) are trajectory visualizations of the real-world.

6. Conclusion

In this paper, a neural inertial navigation method is proposed for low-cost wheeled robots that can achieve fully autonomous pose estimation without relying on external information. The feature extraction capability of the BERT network is utilized to denoise the IMU. At the same time, based on self-supervised contrastive learning and the method relying on unreliable pseudo-labels, the motion state labels corresponding to the IMU sequences in different time periods are obtained, and IEKF is used to further fuse the motion state information and IMU information to enhance the reliability and accuracy of the system. The proposed method was verified on the public dataset and self-collected dataset in real scenes. Experiments conducted on public datasets demonstrate that, in the absence of additional ground truth values, the *Absolute Trajectory Error* (ATE) and *Temporal Relative Trajectory Error* (T-RTE) of the proposed method are respectively 20.23% and 30.71% lower than those of supervised learning-based methods. The experimental results show that our position estimation can be comparable to the existing supervised learning-based inertial navigation methods for wheeled robots in both local and global accuracy.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 61973333.

References

[1] Chen, Y., Xu, H., Yang, W., Yang, C., & Xu, K. (2021). Rat robot motion state identification based on a wearable inertial sensor. *Metrology and Measurement Systems*, Vol. 28 (2021), No. 2, 255–268. https://doi.org/10.24425/mms.2021.136605

[2] Chen, C., & Pan, X. (2024). Deep Learning for Inertial Positioning: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, *25*(9), 10506–10523. https://doi.org/10.1109/tits.2024.3381161

[3] Rong, H., Wu, X., Wang, H., Jin, T., & Zou, L. (2024). Attitude estimation based on multi-scale grouped spatio-temporal attention neural networks. *Metrology and Measurement Systems*, Vol. 31 (2024), No. 1, 195–211. https://doi.org/10.24425/mms.2024.148542

[4] Lan, J., Wang, K., Song, S., Li, K., Liu, C., He, X., Hou, Y., & Tang, S. (2024). Method for measuring non-stationary motion attitude based on MEMS-IMU array data fusion and adaptive filtering. *Measurement Science and Technology*, *35*(8), 086304. https://doi.org/10.1088/1361-6501/ad44c8

[5] Wang, Z., & Cheng, X. (2021). Adaptive optimization online IMU self-calibration method for visual-inertial navigation systems. *Measurement*, *180*, 109478. https://doi.org/10.1016/j.measurement.2021.109478

[6] Gao, Y., Shi, D., Li, R., Liu, Z., & Sun, W. (2022). Gyro-NeT: IMU Gyroscopes Random Errors Compensation Method Based on Deep Learning. *IEEE Robotics and Automation Letters*, *8*(3), 1471–1478. https://doi.org/10.1109/lra.2022.3230594

[7] Brossard, M., Bonnabel, S., & Barrau, A. (2020b). Denoising IMU Gyroscopes with Deep Learning for Open-Loop Attitude Estimation. *IEEE Robotics and Automation Letters*, 1. https://doi.org/10.1109/lra.2020.3003256

[8] Liu, W., Caruso, D., Ilg, E., Dong, J., Mourikis, A. I., Daniilidis, K., Kumar, V., & Engel, J. (2020). TLIO: Tight Learned Inertial Odometry. *IEEE Robotics and Automation Letters*, *5*(4), 5653–5660. https://doi.org/10.1109/lra.2020.3007421

[9] Wang, Y., Kuang, J., Niu, X., & Liu, J. (2022). LLIO: Lightweight Learned Inertial Odometer. *IEEE Internet of Things Journal*, *10*(3), 2508–2518. https://doi.org/10.1109/jiot.2022.3214087

[10] Brossard, M., Barrau, A., & Bonnabel, S. (2019). RINS-W: Robust Inertial Navigation System on wheels. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. https://doi.org/10.1109/iros40897.2019.8968593

[11] Yang, M., Zhu, R., Xiao, Z., & Yan, B. (2021). Symmetrical-Net: Adaptive zero velocity detection for ZUPT-Aided pedestrian navigation system. *IEEE Sensors Journal*, *22*(6), 5075–5085. https://doi.org/10.1109/jsen.2021.3094301

[12] Jeong, J., Cho, Y., Shin, Y., Roh, H., & Kim, A. (2018). Complex Urban LiDAR Data set. In *2018 IEEE International Conference on Robotics and Automation (ICRA),* 6344-6351. https://doi.org/10.1109/icra.2018.8460834

[13] Feng, D., Qi, Y., Zhong, S., Chen, Z., Chen, Q., Chen, H., Wu, J., & Ma, J. (2024). S3E: a Multi-Robot Multimodal dataset for collaborative SLAM. *IEEE Robotics and Automation Letters*, *9*(12), 11401–11408. https://doi.org/10.1109/lra.2024.3490402

[14] Wei, H., Jiao, J., Hu, X., Yu, J., Xie, X., Wu, J., Zhu, Y., Liu, Y., Wang, L., & Liu, M. (2024). FusionPortableV2: A unified multi-sensor dataset for generalized SLAM across diverse platforms and scalable environments. *The International Journal of Robotics Research*. https://doi.org/10.1177/02783649241303525

[15] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence time-series Forecasting. *In Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325

[16] Wang, Y., Sun, X., Cui, D., Wang, X., Jia, Z., & Zhang, Z. (2024). An adaptive estimation of ground vehicle state with unknown measurement noise. *Metrology and Measurement Systems*, Vol. 31 (2024) No. 2, 389–399. https://doi.org/10.24425/mms.2024.149705

[17] Barrau, A., & Bonnabel, S. (2016). The invariant extended Kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, *62*(4), 1797–1812. https://doi.org/10.1109/tac.2016.2594085

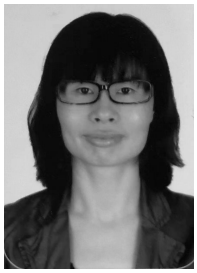
[18] Brossard, M., Barrau, A., & Bonnabel, S. (2020). AI-IMU Dead-Reckoning. *IEEE Transactions on Intelligent Vehicles*, *5*(4), 585–595. https://doi.org/10.1109/tiv.2020.2980758

[19] Jeong, J., Cho, Y., Shin, Y., Roh, H., & Kim, A. (2018). Complex Urban LiDAR Data set. In *In 2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6344-6351. https://doi.org/10.1109/icra.2018.8460834

[20] Guo, F., Yang, H., Wu, X., Dong, H., Wu, Q., & Li, Z. (2023). Model-Based deep Learning for Low-Cost IMU dead reckoning of wheeled mobile robot. *IEEE Transactions on Industrial Electronics*, *71*(7), 7531–7541. https://doi.org/10.1109/tie.2023.3301531

[21] El-Sheimy, N., Hou, H., & Niu, X. (2007). Analysis and modeling of inertial sensors using Allan Variance. *IEEE Transactions on Instrumentation and Measurement*, *57*(1), 140–149. https://doi.org/10.1109/tim.2007.908635

Metrol. Meas. Syst., Vol. 32 (2025), No. 2, pp. 1–19
DOI: 10.24425/mms.2025.154345



Fengrong Huang received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2004. She is currently an Associate Professor with the School of Mechanical Engineering, Hebei University of Technology, Tianjin, China. Her field of interest is navigation and information fusion.

Qinglin Liu received the M.Sc. degree from the School of Mechanical Engineering, Hebei University of Technology, Tianjin, China, in 2023. He is currently an engineer at the National Key Laboratory of Electromagnetic Space Security, Tianjin, China. His current research interests include deep learning, electro-optical countermeasure and inertial navigation.

Mengqi Gao obtained a Bachelor's degree from the School of Engineering, Beijing Forestry University, China in 2016. Afterwards, she worked as a designer at a mechanical design company. She is currently pursuing an M.Sc. at the Hebei University of Technology in Tianjin, China. Her current research interests include deep learning and inertial navigation.

Min Gao received the B.Sc. degree from the School of Mechanical Engineering, Hebei University of Technology, Tianjin, China, in 2020. She is currently pursuing an M.Sc. at the Hebei University of Technology, Tianjin, China. Her current research interests include deep learning and inertial navigation.