## Master's Thesis

# Toward Improved Inertial-based Odometry via Learning-based IMU Online Calibration

**Huakun Liu**
Program of Information Science and Engineering
Graduate School of Science and Technology
Nara Institute of Science and Technology

Supervisor: Prof. Kiyoshi Kiyokawa
Cybernetics and Reality Engineering Lab (Division of Information Science)

Submitted on August 20, 2023

A Master's Thesis
submitted to Graduate School of Science and Technology,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Master of Science

Huakun Liu

Thesis Committee:

| | |
|---|---|
| Supervisor | Kiyoshi Kiyokawa<br>(Professor, Division of Information Science) |
| Co-supervisor | Keiichi Yasumoto<br>(Professor, Division of Information Science) |
| Co-supervisor | Hideaki Uchiyama<br>(Associate Professor, Division of Information Science) |
| Co-supervisor | Monica Perusquía-Hernández<br>(Assistant Professor, Division of Information Science) |
| Co-supervisor | Yutaro Hirao<br>(Assistant Professor, Division of Information Science) |

# Toward Improved Inertial-based Odometry via Learning-based IMU Online Calibration[*]

Huakun Liu

## Abstract

Modern navigation approaches, such as inertial odometry and visual-inertial odometry, are highly dependent on the inertial measurement unit (IMU). However, for low-cost IMU, the uncalibrated bias and noise will quickly propagate errors over time. This paper presents a deep data-driven inertial measurement unit (IMU) online calibration (DUET) method that can compensate for the run-time errors of the accelerometer and gyroscope to improve inertial-based odometry. We design a differential error learning strategy based on the kinematic motion model to train the sensor error compensation model. This strategy allows our method to learn IMU sensor errors, such as scale factors, axis-misalignment, and biases, solely from displacement and orientation increments given by external tracking systems. Then during the odometry computation, the trained model leverages the past inertial data to mitigate the sensor errors and thus reduces the integration errors to reflect the odometry state. The experiments were conducted on two public visual-inertial datasets. The results show that our method can reduce the errors of the accelerometer and gyroscope by 52% and 87%, respectively. These result in an average of 20% improvement in the orientation estimation accuracy compared with state-of-the-art learning-based methods, an average of 77% improvement in the velocity estimation accuracy compared with those from raw accelerations, and an average of 20% improvement in the position estimation accuracy of visual-inertial odometry, which is comparable to existing learning-based methods with lower operational complexity.

**Keywords:** inertial sensors, calibration, localization, deep learning

[*] Master's Thesis, Graduate School of Information Science,
Nara Institute of Science and Technology, August 20, 2023.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Odometry is a process to track the position and orientation of the tracking target relative to its starting pose in a 3D space. It is a fundamental element for various applications in health care [1,2], robotics [3], and virtual/augmented reality [4]. Recent odometry methods are mainly inertial-based, such as visual-inertial odometry (VIO) [5] and inertial-only odometry (IO) [6]. The essential component of inertial-based odometry is the microelectromechanical system (MEMS) inertial measurement unit (IMU). It consists of an accelerometer and a gyroscope that measure accelerations and angular velocities to provide continuous and high-frequency tracking.

VIO uses a lightweight camera to capture visual data and a low-cost and small-size MEMS IMU to measure inertial data. By fusing the visual data and inertial data, it has achieved centimeter-level accuracy in 6 degrees of freedom (DoF) odometry [7]. However, in challenging scenes, such as extreme brightness and dynamic environments, its accuracy is degraded to meter-level when visual odometry fails, and the odometry relies solely on IMU measurement integration [8]. IO is seen as the most compact method because it utilizes only a small MEMS IMU and has lower computational complexity and higher robustness to surrounding changes. However, it suffers from low accuracy due to the rapid accumulation of various IMU errors during the integration [6]. The MEMS IMU errors include scale factors, axis-misalignment, zero-biases, and noise [9]. During the integration process, these errors explode exponentially with time. As a result, IO is unreliable even for a few seconds.

IMU calibration is a method to diminish these problems. It is a process to measure and compensate for gyroscope and accelerometer errors before or during odometry computation to mitigate the negative impact of IMU errors on inertial-based odometry results. Traditional offline calibration methods, such as the multi-position method, can identify the unknown error parameters by optimization algorithms with external reference information [10–13]. However, they have strict requirements for complicated procedures and expensive experimental equipment, such as using a large-scale turntable and moving the device in specified patterns. Furthermore, they are not applicable to handling high non-linear and time-varying errors [9]. Online calibration methods, such as sensor fusion and filtering-based methods [14], can estimate and compensate for run-time IMU errors by fusing multi-sensor and using additional information. It is capable of handling poor offline calibration and time-varying sensor errors, thus improving the odometry results at run-time. However, the performance is highly dependent on the fused sensors' reliability. For example, in visually challenging scenarios such as dark environment, VIO no longer applies to error compensation [8]. The dependencies on the additional sensors also prevent them from being used in IO.

Recent data-driven online calibration methods, such as Denoising IMU Gyro (DIG) [15] and Temporal Convolutional Network Denoising IMU Gyro (TCN-DIG) [16], use supervised learning to train a calibration model and then directly output correction terms, that is, biases and noise, by inputting the run-time inertial data. These learning-based standalone systems are capable of calibrating complex run-time sensor errors from only IMU sequences and have demonstrated superior performance over traditional filtering-based approaches [17].

In inertial-based odometry scenarios, it is feasible to learn an IMU calibration model in the training process using the dynamic positions and orientations captured by high-precision tracking systems such as the laser tracker and the optical tracker [18–21]. However, current methods require additional sensors and assumption-based processed data such as derived accelerations based on the constant-velocity assumption in addition to the tracking system. This increases the operational complexity and limits the scenarios in which these data-driven methods can be applied. Currently, there is still a lack of a simple yet universal data-driven calibration model and learning strategy for robust inertial-based odometry.

The main challenge arises from the module that calibrates the accelerometers. On the one hand, an accelerometer senses not only its linear acceleration but also the local gravity. As a result, the presence of a standard gravity value leads to larger noise densities in accelerometers compared to gyroscopes. This increases the difficulty of predicting its correction term, i.e., bias and noise [22]. On the other hand, tracking systems such as laser and optical trackers only capture high-precision positions and orientations. Training the gyroscope calibration model from the ground truth orientations is feasible because the orientation can be obtained by directly integrating the angular velocity. However, training the accelerometer calibration model based on ground truth positions is not straightforwardly feasible unless ground truth velocities or accelerations are also provided [20].

Most data-driven IMU calibration studies have focused on learning the calibration model using extra sensors and assumption-based processed data, in addition to high-precision tracking systems. However, in this work, we propose to learn sensor errors solely from high-precision orientations and positions without additional sensors, which allows data-driven calibration to be applied to a broader range of scenarios. This obtained performance is attributed to a loss function designed based on the kinematic motion model. Moreover, we conducted the first comprehensive experiments on datasets covering multiple scenarios using different metrics to delve into the impact of data-driven calibration methods on IO and VIO. Our main contributions are as follows:

- A deep data-driven IMU calibration method based on two connected dilated convolution networks for calibrating dynamic gyroscope and accelerometer data simultaneously. With this, we overcome the limitation of using separate models to calibrate gyroscope and accelerometer data, respectively.

- A loss function based on the kinematic motion model for learning the errors using solely ground truth positions and orientations. This simplifies the learning process and improves the applicability of data-driven IMU calibration methods.

- A first thorough analysis through extensive experiments on sensor readings, velocity, orientation, and position estimates to evaluate the impact of the data-driven IMU calibration method on the 6-DoF inertial odometry and visual-inertial odometry. The results reveal the effectiveness of data-driven calibration on reducing the error accumulation rate of IO and improving the robustness of VIO.

The rest of the paper is organized as follows. In Section 2, we introduce the related work, which includes traditional and state-of-the-art data-driven gyroscope and accelerometer calibration methods. In Section 3, we introduce our deep IMU calibration method. Finally, Section 4 provides the experimental results, and the conclusions and future directions are presented in Section 5 and Section 6.

# 2 Related works

With the introduction of deep learning methods, MEMS IMU calibrations can be divided into two categories: traditional calibration and data-driven calibration.

## 2.1 Traditional Calibration

### 2.1.1 Traditional Offline Calibration

Traditional offline calibration methods use parameter state estimation algorithms such as least squares [23] and maximum likelihood estimation [24] to solve the error coefficient by fusing the IMU readings and external reference information. Traditional calibration methods are further divided into two categories: non-autonomous and autonomous calibration, based on the source of reference information [9].

The reference information of non-autonomous calibration is obtained from high-precision equipment such as high-precision turntables [25]. The controllable turntables make specific and accurate motion. Then the reference motion is used to solve the IMU error parameters. They are the most accurate calibration methods. However, due to the complex procedures and the need for expensive equipment, non-autonomous calibration methods are only suitable for use in large companies and in scenarios with specialized needs. Autonomous calibrations, on the contrary, do not dependent on high-precision equipment. They rely on external reference excitation, such as the local gravity, magnetic fields, and the Earth' s rotation rate [23]. These methods based on the characteristics of the gyroscope and accelerometer, that is, the static accelerometer measures only the gravity and the static gyroscope measures the Earth's rotation rate. In such cases, we can use these natural reference value to optimize and correct the error parameters. In addition, for 9-axis IMU, the magnetic fields measured by the magnetometer also provide a reference for correcting the gyroscope. Although these methods are simpler and more efficient than non-autonomous calibration methods, they may fail for low-cost MEMS IMUs due to the inability to sense smaller natural references and external environmental disturbances. For example, low-cost gyroscopes cannot accurately sense the Earth's rotation, and indoor geomagnetic fields cannot be accurately measured due to disturbances. Furthermore, considering the time-varying characteristics of sensor errors is also a challenge for traditional offline methods For details about traditional offline calibration, a survey on MEMS inertial sensor calibration is a reference [9].

### 2.1.2 Traditional Online Calibration

Traditional online calibration, utilizing filtering or optimization-based methods, performs error estimation by fusing multiple sensors such as magnetometer and camera. They are capable of estimating the varying sensor errors in run-time, thereby maintaining accurate calibrations. IMU-only online calibration, such as Madgwick orientation filter [26] and complementary filter [27], enables computing the gyroscope errors from the accelerometer and magnetometer. These methods are advantageous due to their efficiency and low computational load, making them an ideal choice for resource-limited systems. Visual-based online filtering-based methods such as VIO [28, 29], are more advanced methods that combine visual data from a camera and inertial data from an IMU to estimate the IMU error in run-time. By integrating visual data, VIO can correct drift errors typically encountered in IMU-only systems for long-term running. VIO is a highly effective approach in GPS-denied environments and is commonly used in autonomous vehicle navigation and augmented reality applications. However, the computational load and complexity of VIO are much higher compared to the IMU-only optimization-based methods, and its performance is dependent on good quality visual data and robust feature tracking [8].

## 2.2 Data-driven Calibration

Deep learning has provided new possibilities for unimodal position estimation from IMU. For instance, it is used to extract latent features from IMU signals to estimate the velocity [6], orientation [17,30], and displacement [31–34]. These methods have seen great success in 6-DoF IO, owing to avoiding the error accumulation caused by the integration process. Additionally, using data-driven deep learning to calibrate run-time gyroscopes and accelerometers has become popular [35]. This is because introducing deep learning decreases human intervention, facilitates the realization of autonomous online calibration systems, and is well coupled with inertial-based odometry [9].

### 2.2.1 Data-driven Gyroscope Calibration

In [36], the first long short-term memory (LSTM)-recurrent neural network (RNN)-based denoising method was proposed to denoise IMU gyroscope signals. Compared to autoregressive and moving average models, the standard deviation of denoised signals decreased by up to 42.4% with deep learning. In [17], learning-based OriNet was proposed to estimate the 3D orientation with a genetic bias calibration algorithm. The orientation estimation in real scenarios was improved by 72% compared to the complementary filter and 89% compared to Madgwick filter. Consequently, there is a growing interest in applying learning-based methods to improve inertial-based odometry. Brossard *et al.* [15] proposed a convolutional neural network (CNN) to predict the run-time gyroscope correction term, i.e., zero bias and noise, and to find the optimal coefficients of scale factor and axis-misalignment during training from measured accelerometer and gyroscope readings. The experimental results on two public VIO datasets demonstrated the comparative orientation estimation accuracy from IMU only with VIO. Then, Huang *et al.* [16] used a temporal convolutional network to further improve the performance of the gyroscope online calibration. They showed that the orientation estimated from the calibrated gyroscope data could be used to improve the accuracy of the VIO position estimation. The absolute translation error of 3D position was reduced by 33%, i.e., from 61.48 cm to 39.52 cm, through the online learning-based calibration. To solve the low generalizability problem of data-driven denoising models, Yao [37] proposed a few-shot domain adaptation gyroscope calibration method that consists of an embedding module, restructor module, and generator module.

### 2.2.2 Data-driven Accelerometer Calibration

Engelsman and Klein [22] implemented three learning algorithms and one machine learning method, including unidirectional bi-layer LSTM, bi-directional one-layer RNN, bi-directional one-layer gated recurrent unit (GRU), and k-nearest neighbor, to calibrate the accelerometer. The evaluation of a simulated dataset and static accelerometer data showed that data-driven accelerometer calibration achieves a 60% noise reduction and 20% improvement in stationary course alignment compared to traditional methods.

Our basic network structure is similar to the aforementioned data-driven gyroscope or accelerometer calibration methods, which are based on convolutional neural networks. However, instead of calibrating only one component of an IMU, we simultaneously calibrate the gyroscope and accelerometer.

### 2.2.3 Data-driven IMU Calibration

Chen *et al.* [18] used a convolutional neural network to reduce errors from both the accelerometer and gyroscope in a laboratory environment. As a reference, the ground truth acceleration and angular velocity data were generated from a well-designed linear motion stage and a rotary motor. Zhang *et al.* [19] trained an RNN to calibrate the run-time gyroscope and accelerometer. Additional high-quality sensors and sensor fusion algorithms provide ground truth orientations, velocities, and positions to train the calibration model. Similarly, Steinbrener *et al.* [20] proposed LSTM-based and Transformer-based methods to output calibrated IMU measurements for real-time 6-DoF pose estimation. The ground truth velocities are computed from positions based on the assumption that velocities are constant between two consecutive frames. Recently, Buchanan *et al.* [21] proposed to train both LSTM and Transformer from the optimized ground truth biases of target IMU to learn and compensate for the bias. They showed that fusing the data-driven bias compensation model with VIO reduced the drift rate by an average of 15%.

Our objective is similar to these studies to build the calibration model from high-precision dynamic motion data and then mitigate the sensor errors at runtime. Nevertheless, these methods are forced to rely on external sensors and complex setups to learn the model. In contrast, our proposed general learning strategy keeps us from relying on these.

# 3 Methods

In this section, we first present an IMU sensor error model and a kinematic motion model used in inertial-based odometry, followed by our deep IMU calibration network structure and the loss function.

## 3.1 Preliminaries

**Sensor error model** A MEMS IMU consists of an accelerometer and a gyroscope that measures the acceleration $\boldsymbol{a}$ and the angular velocity $\boldsymbol{\omega}$ of the carrier. However, the measurement contains not only the true $\boldsymbol{a}$ and $\boldsymbol{\omega}$, but also other error terms, as shown in Figure 3.1. There errors are mainly caused by imperfect assembling procedures and the influence of temperature on the MEMS IMU's silicon [9].

Given a three-axis strapdown accelerometer and gyroscope, a commonly used error model is established as follows [9]:

$$\tilde{\boldsymbol{u}} = (\boldsymbol{S} + \boldsymbol{N})\boldsymbol{u} + \boldsymbol{b} + \boldsymbol{n}, \tag{3.1}$$

where $\tilde{\boldsymbol{u}} \in \mathbb{R}^3$ and $\boldsymbol{u} \in \mathbb{R}^3$ denote the measurement output of an IMU and the true acceleration or angular velocity, respectively.

$$\boldsymbol{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \tag{3.2}$$

$\boldsymbol{S} \in \mathbb{R}^{3\times 3}$ is the scale factor that refers to the ratio between the output quantity and the input quantity. This ratio is caused mainly by sensitivity of the circuit

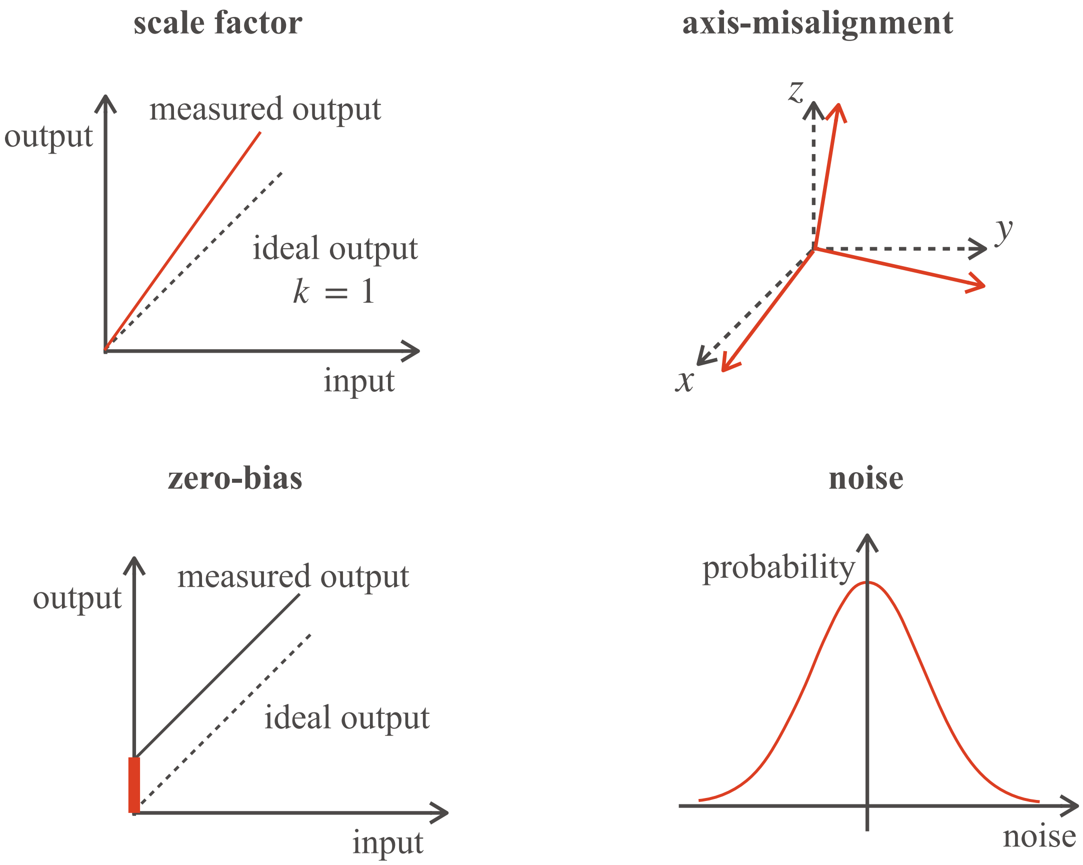


Figure 3.1: Illustration of IMU errors.

on each axis. It is a 3 × 3 identity matrix in the ideal condition.

$$\boldsymbol{N} = \begin{bmatrix} 0 & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & 0 & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & 0 \end{bmatrix} \tag{3.3}$$

$\boldsymbol{N} \in \mathbb{R}^{3\times3}$ denotes the axis-misalignment error results from the non-orthogonality between each axis, which is caused by the manufacturing technique limitations.

$$\boldsymbol{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \tag{3.4}$$

$\boldsymbol{b} \in \mathbb{R}^3$ is the so-called zero-bias. It is the output value of the accelerometer or gyroscope when the measured physical quantity is equal to zero.

$$\boldsymbol{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \tag{3.5}$$

$\boldsymbol{n} \in \mathbb{R}^3$ is commonly assumed to be the high-frequency random sensor white noise that follows the zero-mean Gaussian distribution.

**Kinematic motion model** Inertial and visual-inertial odometry calculate the trajectory from the acceleration and angular velocity obtained from an IMU using the kinematic motion model. The core processing is to rotate the acceleration from the IMU frame to a fixed global frame and then accumulate it to compute the velocity and the moving distance.

Given the angular velocity $\boldsymbol{\omega}$ obtained from a gyroscope, the special orthogonal rotation matrix in 3D space $\boldsymbol{R} \in SO(3)$ that maps from IMU frame to global frame at time step $i$ can be expressed as follows:

$$\boldsymbol{R}_i = \boldsymbol{R}_{i-1} \exp(\boldsymbol{\omega}_{i-1} \Delta t), \tag{3.6}$$

where $\exp(\cdot)$ is the $SO(3)$ exponential map and $\Delta t$ is the time interval of two consecutive frames. The velocity $\boldsymbol{v}$ in the global frame is then calculated by rotating the measured acceleration $\boldsymbol{a}$ and removing the local gravity $\boldsymbol{g}$,

$$\boldsymbol{v}_i = \boldsymbol{v}_{i-1} + (\boldsymbol{R}_{i-1}\boldsymbol{a}_{i-1} + \boldsymbol{g})\Delta t, \quad \boldsymbol{g} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}. \tag{3.7}$$

Finally, the position in the fixed global frame is calculated as follows:

$$\boldsymbol{p}_i = \boldsymbol{p}_{i-1} + \boldsymbol{v}_{i-1}\Delta t + \frac{1}{2}(\boldsymbol{R}_{i-1}\boldsymbol{a}_{i-1} + \boldsymbol{g})\Delta t^2. \tag{3.8}$$

## 3.2 Problem Modeling

According to the sensor error model in (3.1), the value $\boldsymbol{u}_i$ of an IMU accelerometer or gyroscope at time step $i$ can be expressed as follows:

$$\boldsymbol{u}_i = (\boldsymbol{S} + \boldsymbol{N})^{-1} (\tilde{\boldsymbol{u}}_i - (\boldsymbol{b}_i + \boldsymbol{n}_i)). \tag{3.9}$$

To simplify the problem, we denote $(\boldsymbol{S}+\boldsymbol{N})^{-1}$ and $(\boldsymbol{b}_i+\boldsymbol{n}_i)$ in (3.9) by $\boldsymbol{C} \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{\varepsilon}_i \in \mathbb{R}^3$, respectively. $\boldsymbol{C}$ contains both the scale factor and axis-misalignment. $\boldsymbol{\varepsilon}_i$ is the correction term that contains zero bias and noise. They are expressed as follows:

$$\boldsymbol{C} = \begin{bmatrix} s_x & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & s_y & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & s_z \end{bmatrix}^{-1}, \tag{3.10}$$

$$\boldsymbol{\varepsilon}_i = \begin{bmatrix} b_x + n_x \\ b_y + n_y \\ b_z + n_z \end{bmatrix}. \tag{3.11}$$

Then, we have

$$\boldsymbol{u}_i = \boldsymbol{C}(\tilde{\boldsymbol{u}}_i - \boldsymbol{\varepsilon}_i). \tag{3.12}$$

In this paper, we aim to improve the IMU data reliability by learning, predicting, and compensating for $\boldsymbol{C}$ and $\boldsymbol{\varepsilon}_i$ using a deep neural network.

## 3.3 Network Structure

Our network structure is based on the dilated convolutional neural network [38]. A dilated CNN is a type of CNN that uses dilated convolutions to exponentially expand the receptive field with few memory consumption and efficient computation. The dilated convolutions enable the network to maintain the temporal property of the data and capture long-range contextual information, making it suitable for time-series data. This network is widely used for data-driven calibration because it is lightweight and has no loss of accuracy compared to recurrent neural networks [15, 16].

As shown in Figure 3.2, we define the network as predicting the correction term $\hat{\boldsymbol{\varepsilon}}_{a,i}$ and $\hat{\boldsymbol{\varepsilon}}_{g,i}$ for the accelerometer and the gyroscope from uncalibrated IMU data. In particular, we leverage the current frame and past $N$ frames of uncalibrated 3-axis accelerations $(\tilde{\boldsymbol{a}}_{i-N}, \cdots, \tilde{\boldsymbol{a}}_i)$ and 3-axis angular velocities $(\tilde{\boldsymbol{\omega}}_{i-N}, \cdots, \tilde{\boldsymbol{\omega}}_i)$ to predict the correction term $\hat{\boldsymbol{\varepsilon}}_{g,i}$. In addition, we optimize the multiplier $\boldsymbol{C}_g$ for the angular velocity $\boldsymbol{\omega}_i$ during training. The neural network that predicts $\hat{\boldsymbol{\varepsilon}}_{g,i}$ at

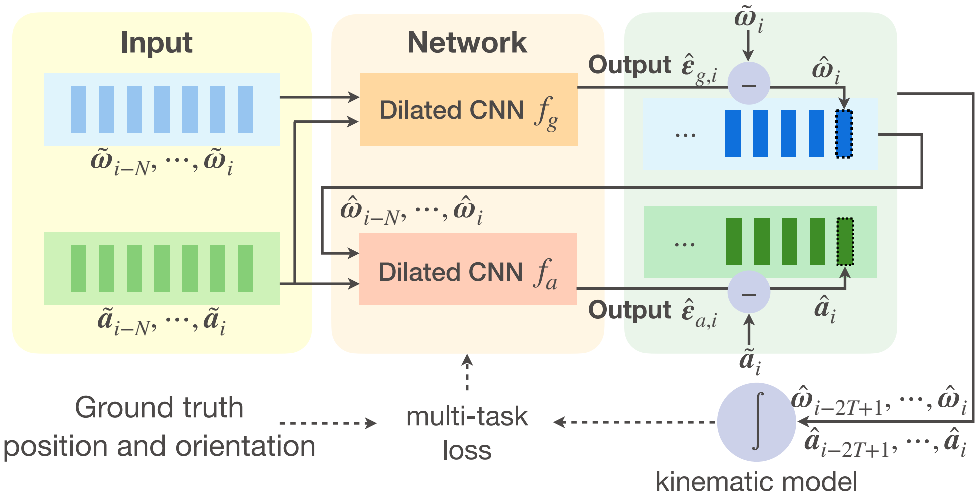

Input
$\tilde{\omega}_{i-N}, \cdots, \tilde{\omega}_i$
$\tilde{a}_{i-N}, \cdots, \tilde{a}_i$
Network
Dilated CNN $f_g$
$\hat{\omega}_{i-N}, \cdots, \hat{\omega}_i$
Dilated CNN $f_a$
$\tilde{\omega}_i$
Output $\hat{\varepsilon}_{g,i}$
$\hat{\omega}_i$
Output $\hat{\varepsilon}_{a,i}$
$\hat{a}_i$
$\tilde{a}_i$
$\hat{\omega}_{i-2T+1}, \cdots, \hat{\omega}_i$
$\hat{a}_{i-2T+1}, \cdots, \hat{a}_i$
Ground truth position and orientation
multi-task loss
kinematic model

Figure 3.2: The flowchart of building the calibration model and the network architecture for calibrating run-time gyroscope and accelerometer measurements. The first dilated convolutional network inputs $N + 1$ frames raw IMU measurements and outputs current frame calibrated angular velocity. Then the second dilated convolutional network inputs $N + 1$ frames calibrated angular velocities and raw accelerations then outputs the current frame calibrated acceleration. $N = 340$ is determined by the network architecture Figure 3.3. When training the calibration model, the ground truth orientations and positions are used to compute the loss with inferred values from calibrated angular velocities and accelerations.

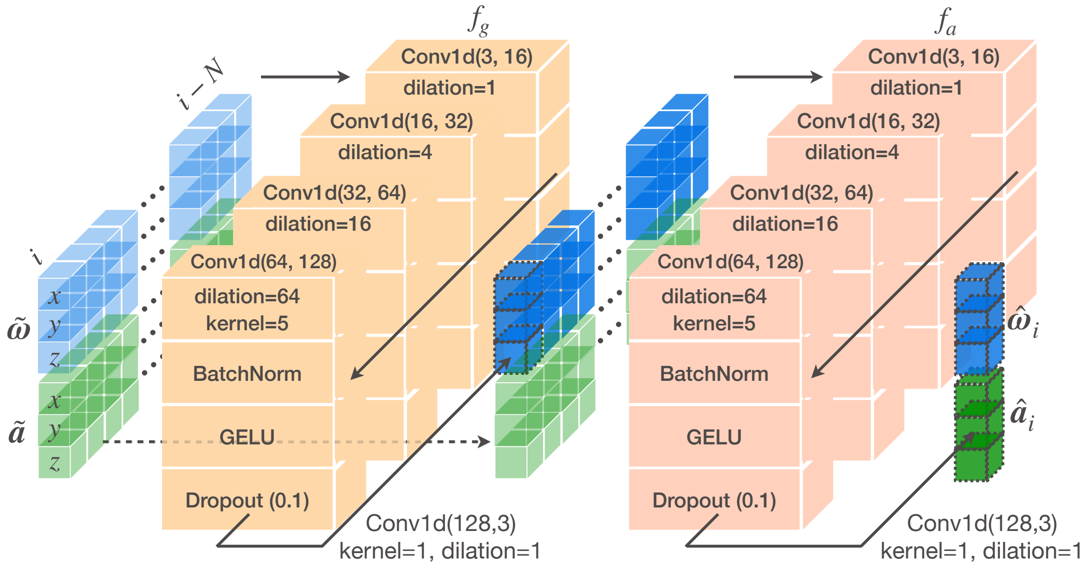


Figure 3.3: Illustration of the network architecture. $f_g$ is a four-layer dilated convolutional neural network. Each layer of the neural network consists of four components, i.e., a 1-d convolutional neural network with kernel size equals 5, batch normalization, GELU activation function, and dropout layer. The input channels of each layer are 3, 16, 32, and 64. The out channels of each layer are 16, 32, 64, and 128. A 1 × 1 convolutional neural network with 128 input channels equals 128 and 3 output channels is followed by the four-layer neural network and outputs the calibrated angular velocity. $f_a$ has the same structure as $f_g$. However, the input of $f_a$ is composed of the output of $f_g$ and raw accelerations.

time step $i$ can be expressed as follows:

$$\hat{\boldsymbol{\varepsilon}}_{g,i} = f_g((\tilde{\boldsymbol{a}}_{i-N}, \tilde{\boldsymbol{\omega}}_{i-N}), \cdots, (\tilde{\boldsymbol{a}}_i, \tilde{\boldsymbol{\omega}}_i)), \tag{3.13}$$

where $f_g(\cdot)$ is the function defined by a dilated convolutional neural network. Then, the calibrated angular velocity $\hat{\boldsymbol{\omega}}_i$ is computed by

$$\hat{\boldsymbol{\omega}}_i = \hat{\boldsymbol{C}}_g(\tilde{\boldsymbol{\omega}}_i - \hat{\boldsymbol{\varepsilon}}_{g,i}). \tag{3.14}$$

After obtaining calibrated angular velocities, we pass these values with uncalibrated accelerations into the second dilated neural network to optimize the multiplier $\boldsymbol{C}_a$ during training and predict the acceleration correction term $\hat{\boldsymbol{\varepsilon}}_{a,i}$, which is defined as follows:

$$\hat{\boldsymbol{\varepsilon}}_{a,i} = f_a((\tilde{\boldsymbol{a}}_{i-N}, \hat{\boldsymbol{\omega}}_{i-N}), \cdots, (\tilde{\boldsymbol{a}}_i, \hat{\boldsymbol{\omega}}_i)). \tag{3.15}$$

Similarly, the calibrated acceleration $\hat{\boldsymbol{a}}_i$ is computed by

$$\hat{\boldsymbol{a}}_i = \hat{\boldsymbol{C}}_a(\tilde{\boldsymbol{a}}_i - \hat{\boldsymbol{\varepsilon}}_{a,i}). \tag{3.16}$$

The network architecture is presented in Figure 3.3. The neural network $f_g(\cdot)$ is structured with four dilated convolutional layers. Every layer in this network has a sequence of four components: a one-dimensional convolutional neural network with a kernel size of 5, batch normalization, a GELU activation function, and a dropout layer. The input channel counts across these layers are sequentially 3, 16, 32, and 64, while their corresponding output channels are 16, 32, 64, and 128. Subsequent to these four layers, there's a $1 \times 1$ convolutional layer with 128 input channels and 3 output channels, which produces the calibrated angular velocity. The structure of $f_a(\cdot)$ mirrors that of $f_g(\cdot)$. Note that the input of $f_a(\cdot)$ consists of uncalibrated accelerations and calibrated angular velocities, which are used to connect $f_a(\cdot)$ and $f_g(\cdot)$. Furthermore, from (3.8), the accurate inferred position depends on both angular velocity and acceleration. Thus, it is reasonable to learn the acceleration error from calibrated angular velocities, which also mitigates the impact of gyroscope errors.

## 3.4 Loss Function

The most straightforward way to train the model is to minimize the loss between the predicted and target ground truth acceleration and angular velocity. However,

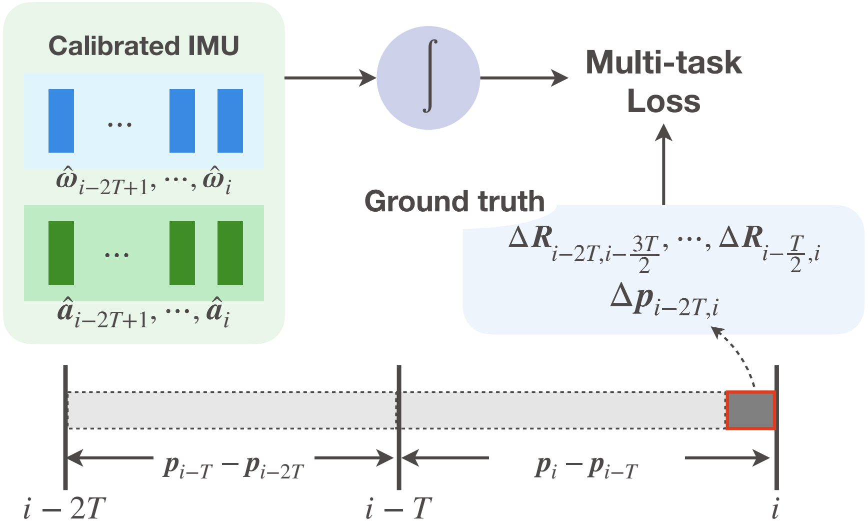

Calibrated IMU
$\hat{\omega}_{i-2T+1}, \cdots, \hat{\omega}_i$
$\hat{a}_{i-2T+1}, \cdots, \hat{a}_i$
$\int$
Multi-task Loss
Ground truth
$\Delta R_{i-2T,i-\frac{3T}{2}}, \cdots, \Delta R_{i-\frac{T}{2},i}$
$\Delta p_{i-2T,i}$
$p_{i-T} - p_{i-2T}$
$p_i - p_{i-T}$
$i-2T$
$i-T$
$i$

Figure 3.4: Schematic diagram of the proposed loss function. The target is to minimize: 1) the residuals between the ground truth displacement difference and the calibrated displacement difference with the condition of $T$ window size for each displacement; 2) the difference between the accumulated orientation from calibrated angular velocity and the ground truth orientation over $\frac{T}{2}$ windows.

given high-precision positions and orientations, acquiring the dynamic ground truth IMU data at a high IMU frequency (200 Hz or more) is not feasible in practice. This is because the derivative process of high-precision positions suffers from a jitter problem [20] (see Section 4.4.2 for more details). Therefore, solely using ground truth positions and orientations for learning is a preferred solution.

We first consider learning accelerometer errors from positions. From the position inference function (3.8), the displacement between two consecutive frames, i.e., with the condition that the window size equals to 1, can be expressed as:

$$\boldsymbol{p}_i - \boldsymbol{p}_{i-1} = \boldsymbol{v}_{i-1}\Delta t + \frac{1}{2}(\boldsymbol{R}_{i-1}\boldsymbol{a}_{i-1} + \boldsymbol{g})\Delta t^2 \tag{3.17}$$

$$= \boldsymbol{v}_{i-1}\Delta t + \frac{1}{2}\boldsymbol{a}^W_{i-1}\Delta t^2, \tag{3.18}$$

where

$$\boldsymbol{a}^W_{i-1} = \boldsymbol{R}_{i-1}\boldsymbol{a}_{i-1} + \boldsymbol{g} \tag{3.19}$$

is the linear acceleration in the fixed world frame at time step $i-1$. However, the initial velocity $\boldsymbol{v}_{i-1}$ in (3.17) is not available in practice, especially when the tracked object does not move from stationary. Thus, to remove the effect of the uncertain $\boldsymbol{v}_{i-1}$, we consider a consecutive displacement between $\boldsymbol{p}_{i-2}$ and $\boldsymbol{p}_{i-1}$ as

$$\boldsymbol{p}_{i-1} - \boldsymbol{p}_{i-2} = \boldsymbol{v}_{i-2}\Delta t + \frac{1}{2}\boldsymbol{a}_{i-2}^{W}\Delta t^2. \tag{3.20}$$

Then, according to the velocity inference function (3.7), equation (3.17) can be further formulated as:

$$\begin{aligned}\boldsymbol{p}_i - \boldsymbol{p}_{i-1} &= (\boldsymbol{v}_{i-2} + \boldsymbol{a}_{i-2}^{W}\Delta t)\Delta t + \frac{1}{2}\boldsymbol{a}_{i-1}^{W}\Delta t^2 \\ &= \boldsymbol{v}_{i-2}\Delta t + \boldsymbol{a}_{i-2}^{W}\Delta t^2 + \frac{1}{2}\boldsymbol{a}_{i-1}^{W}\Delta t^2. \end{aligned} \tag{3.21}$$

We define $\Delta\boldsymbol{p}_{i-2,i}$ as the difference of displacement $\boldsymbol{p}_i - \boldsymbol{p}_{i-1}$ and $\boldsymbol{p}_{i-1} - \boldsymbol{p}_{i-2}$, that is,

$$\begin{aligned}\Delta\boldsymbol{p}_{i-2,i} &= (\boldsymbol{p}_i - \boldsymbol{p}_{i-1}) - (\boldsymbol{p}_{i-1} - \boldsymbol{p}_{i-2}) \\ &= \frac{1}{2}\boldsymbol{a}_{i-2}^{W}\Delta t^2 + \frac{1}{2}\boldsymbol{a}_{i-1}^{W}\Delta t^2. \end{aligned} \tag{3.22}$$

Thus, the displacement difference can be expressed as a function of the linear acceleration in the fixed world frame. However, using (3.22) still suffers from high-frequency position jitters. We thus expand the window size of each displacement to $T$ and calculate the cumulative difference over $2T$ windows. This allows for attenuating position jitter effects in the calculation by enlarging the displacement difference. Similar to the calculation of $\Delta\boldsymbol{p}_{i-2,i}$, we derive that $\Delta\boldsymbol{p}_{i-2T,i}$ can be expressed as only related to the linear acceleration within the corresponding period as:

$$\begin{aligned}\Delta\boldsymbol{p}_{i-2T,i} &= (\boldsymbol{p}_i - \boldsymbol{p}_{i-T}) - (\boldsymbol{p}_{i-T} - \boldsymbol{p}_{i-2T}) \\ &= \sum_{j=1}^{T} j\boldsymbol{a}_{i-2T+j-1}^{W}\Delta t^2 + \sum_{j=1}^{T-1}(T-j)\boldsymbol{a}_{i-T+j-1}^{W}\Delta t^2 \\ &\quad + \frac{1}{2}\sum_{j=0}^{T-1}(\boldsymbol{a}_{i-T+j}^{W} - \boldsymbol{a}_{i-2T+j}^{W})\Delta t^2, \end{aligned} \tag{3.23}$$

where $T$ is the window size of each displacement. $\boldsymbol{a}_{i+j}^{W}$ is the linear acceleration in the fixed world frame at time step $i+j$ and computed as:

$$\boldsymbol{a}_{i+j}^{W} = \boldsymbol{R}_{i+j}\boldsymbol{a}_{i+j} + \boldsymbol{g}. \tag{3.24}$$

Figure 3.4 represents the schematic diagram of the loss function. We define the loss function based on (3.23) and (3.24) as:

$$\mathcal{L}_1\Big((\boldsymbol{p}, \boldsymbol{R}), \hat{\boldsymbol{a}}\Big) = \sum_i \rho(\Delta\boldsymbol{p}_{i-2T,i} - \Delta\hat{\boldsymbol{p}}_{i-2T,i}), \tag{3.25}$$

where $\Delta\boldsymbol{p}_{i-2T,i}$ is computed from ground truth positions, $\Delta\hat{\boldsymbol{p}}_{i-2T,i}$ is calculated from the computed acceleration $(\hat{\boldsymbol{a}}_{i-2T}, \cdots, \hat{\boldsymbol{a}}_i)$ through (3.16), (3.23), and (3.24), and $\rho(\cdot)$ is the Huber loss function [39].

In order to train the model for gyroscope calibration, we minimize the accumulated orientation error within every $T/2$ windows as:

$$\mathcal{L}_2(\boldsymbol{R}, \hat{\boldsymbol{\omega}}) = \sum_i \rho\left(\log_{SO3}\left(\Delta\boldsymbol{R}_{i,i+\frac{T}{2}}\Delta\hat{\boldsymbol{R}}_{i,i+\frac{T}{2}}^{-1}\right)\right) \tag{3.26}$$

$$\Delta\boldsymbol{R}_{i,i+\frac{T}{2}} = \boldsymbol{R}_i^{-1}\boldsymbol{R}_{i+\frac{T}{2}} \tag{3.27}$$

$$\Delta\hat{\boldsymbol{R}}_{i,i+\frac{T}{2}} = \prod_{j=i}^{i+\frac{T}{2}-1} \exp_{SO3}(\hat{\boldsymbol{\omega}}_j\Delta t) \tag{3.28}$$

where $\log_{SO3}(\cdot)$ is the $SO(3)$ logarithm map. Note that the output of the first layer network is also used as the input of the second layer network. Thus, we set a larger error calculation frequency of gyroscope than that of accelerometer, as every $\frac{T}{2}$ windows against every $2T$ windows. This allows for a faster convergence of the first layer network for calibrating gyroscope and further aided in the training of the second layer network used to calibrate the accelerometer.

Finally, we use the sum of $\mathcal{L}_1$ and $\mathcal{L}_2$ to jointly train the network:

$$\mathcal{L} = w_1\mathcal{L}_1\Big((\boldsymbol{p}, \boldsymbol{R}), \hat{\boldsymbol{a}}\Big) + w_2\mathcal{L}_2(\boldsymbol{R}, \hat{\boldsymbol{\omega}}). \tag{3.29}$$

The computational complexity of $\mathcal{L}$ is $O(n)$, however, in practice, the computation time depends more on the window size $T$ because of the matrix multiplications when calculating the accumulated orientation within each window. In our experiments, we analyzed the impact of $T$ on the training time, see Section 4.6 for details. Furthermore, for balancing the two loss terms, we adopt multi-task training strategy [40] to adapt the weights as

$$\begin{aligned}\mathcal{L} = \frac{1}{2c_1}\mathcal{L}_1\Big((\boldsymbol{p}, \boldsymbol{R}), \hat{\boldsymbol{a}}\Big) + \ln(1 + c_1^2)\\ + \frac{1}{2c_2}\mathcal{L}_2(\boldsymbol{R}, \hat{\boldsymbol{\omega}}) + \ln(1 + c_2^2),\end{aligned} \tag{3.30}$$

where $c_1$ and $c_2$ are parameters optimized during training. The objective of (3.30) is to train the calibration model by minimizing the adaptive weighted sum of the accumulated displacement error and orientation error.

# 4 Experiments

In this section, we present two experiments to evaluate the proposed method. In the first experiment, we analyze the direct output of our model, i.e., the predicted sensor error, and compare it with the raw sensor errors. In the second experiment, we evaluate the proposed method in terms of orientation, velocity, and position estimation by comparing it with existing methods. Additionally, we offer two discussions on the generalization issue of the data-driven method and the hyperparameter $T$. In the first discussion, we explore the impact of bias changes on the model performance by analyzing the correlation between biases and evaluation results. Then, we performed further experimental validation to clarify these findings. In the second discussion, we analyze the impact of the $T$ on the model performance and training time. Then, we emphasize a few important points for setting a proper $T$.

## 4.1 Datasets

### 4.1.1 EuRoC

EuRoC [41] is one of the most used visual-inertial datasets. The dataset contains various sequences collected by a micro aerial vehicle (MAV). The angular velocity and specific force were measured using an uncalibrated ADIS16448 IMU at 200 Hz. Ground truth positions were recorded at 20 Hz using a Leica Nova MS50 laser tracker. For sequences collected in a Vicon room, the 6D pose of the MAV was recorded by the Vicon motion capture system at a rate of 100 Hz. All collected data were precisely time-space aligned with the IMU measurements. Then, a classic maximum likelihood state estimator incorporated all ground truth and IMU measurements to estimate final ground truth orientations, positions, velocities, and the biases of the gyroscope and accelerometer.

This dataset provides estimated high-precision velocity and IMU sensor biases. Therefore, it is well suited for experimenting with our method. It enables us to test sensor error prediction and position estimates. The dataset was split into training set and test set following the same splitting strategy as in related works [15, 16], as shown in Table 4.1. The ground truth data was aligned to the IMU timestamps by interpolation. Additionally, as [42] reported, the ground truth on the V1_01 sequence of EuRoC is not accurate in its orientation estimate. Thus, this sequence was not involved in our experiments.

### 4.1.2 TUM-VI

TUM-VI [8] is a more recent visual-inertial dataset. It was collected by a handheld device with a non-static start. The IMU data was logged by the Bosch BMI160 IMU at 200 Hz. The accurate 6D pose ground truth was collected using a MoCap OptiTrack Flex13 at 120 Hz. All data were accurately aligned with the IMU measurements.

TUM-VI differs from EuRoC in two points. First, this dataset only provides ground truth positions and orientations thus it is not applicable to methods that require the use of accurate accelerations, angular velocities, and velocities. Second, the BMI160 IMU was strictly calibrated using the global optimization method. Accordingly, the calibrated parameters of the IMU were provided and the IMU data was calibrated. This allows us to simulate uncalibrated IMU data based on the given parameters to increase the diversity of the dataset.

We synthesized the uncalibrated IMU data, i.e., $\boldsymbol{a}_{\text{uncali}}$ and $\boldsymbol{\omega}_{\text{uncali}}$, of TUM-VI based on the given parameters and the formulas in [8] as follows:

$$\boldsymbol{a}_{\text{uncali}} = \boldsymbol{M}_{\boldsymbol{a}}(\boldsymbol{a}_{\text{cali}} + \boldsymbol{b}_{\boldsymbol{a}}), \tag{4.1}$$

$$\boldsymbol{\omega}_{\text{uncali}} = \boldsymbol{M}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_{\text{cali}} + \boldsymbol{b}_{\boldsymbol{\omega}}), \tag{4.2}$$

where $\boldsymbol{M}_{\boldsymbol{a}}$ and $\boldsymbol{M}_{\boldsymbol{\omega}}$ contain scale factors and axis-misalignment, $\boldsymbol{b}_{\boldsymbol{a}}$ and $\boldsymbol{b}_{\boldsymbol{\omega}}$ are zero biases, $\boldsymbol{a}_{\text{cali}}$ and $\boldsymbol{\omega}_{\text{cali}}$ are the calibrated IMU data provided by the TUM-VI. The dataset containing the synthesized IMU data is referred to as TUM-VI (Uncali) in our experiments. Correspondingly, the dataset containing the calibrated IMU data is dubbed TUM-VI (Cali). We took the six-room sequences of TUM-VI (Uncali) and TUM-VI (Cali) as the ground truth 6D poses are available over the whole trajectory. Similarly, the ground truth data was aligned to the IMU timestamps by interpolation and the dataset was split into a training set and test set, as shown in Table 4.1.

Table 4.1: Train and test sequences.

| Dataset | Train sequences [length] | (No.) Test sequences [length] |
|---|---|---|
| EuRoC | MH_01_easy [182 s]<br>MH_03_medium [150 s]<br>MH_05_difficult [111 s]<br>V1_02_medium [83.5 s]<br>V2_01_easy [112 s]<br>V2_03_difficult [115 s] | (1) MH_02_easy [150 s]<br>(2) MH_04_difficult [99 s]<br>(3) V1_03_difficult [105 s]<br>(4) V2_02_medium [115 s] |
| TUM-VI<br>(Uncali) | room1 [140 s]<br>room3 [140 s]<br>room5 [141 s] | (5) room2 [143 s]<br>(6) room4 [111 s]<br>(7) room6 [130 s] |
| TUM-VI<br>(Cali) | room1 [140 s]<br>room3 [140 s]<br>room5 [141 s] | (8) room2 [143 s]<br>(9) room4 [111 s]<br>(10) room6 [130 s] |

## 4.2 Method Implementation

Our method was implemented based on PyTorch 1.13. The training process used an Adam optimizer [43] with 0.01 initial learning rate and 0.1 weight decay, which was adjusted using a cosine annealing schedule [44]. We empirically set the window size in loss function $T = 64$ with the IMU frequency of 200 Hz (see Section 4.6 for details). Setting $T$ to a power of 2 allows a faster computation of the loss function by concatenating the multiplications [15]. We trained the model for 1500 epochs using an NVIDIA TITAN RTX GPU, which took about 1 min with 8 min training data. In the test, predicting the calibrated IMU data takes 0.24 $\mu s$ per IMU measurement. We ran our method ten times with the same setting and then took the mean results.

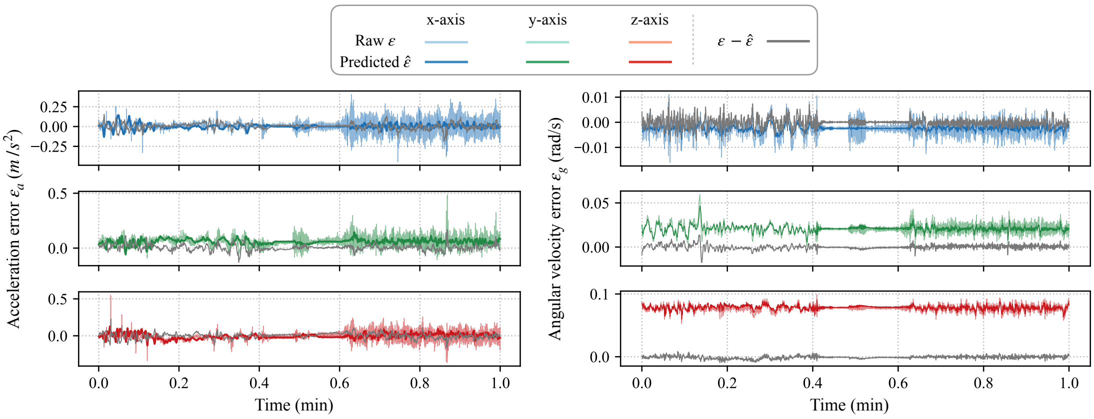


Figure 4.1: Predicted error of accelerometer (left) and gyroscope (right) of test sequence EuRoC-MH_02_easy. All data were averaged every ten samples. The predicted error has the same trend as the raw error. The compensated signal errors were shifted to around zero. However, the noise remains.

## 4.3 Evaluation of Sensor Error Reduction

Our model aims to predict and compensate for the IMU sensor error. Thus, we compared the raw sensor error $\boldsymbol{\varepsilon}_a$, $\boldsymbol{\varepsilon}_g$ with the compensated sensor error $\boldsymbol{\varepsilon}_a - \hat{\boldsymbol{\varepsilon}}_a$ and $\boldsymbol{\varepsilon}_g - \hat{\boldsymbol{\varepsilon}}_g$ to evaluate whether our method correctly predicted and reduced the errors. Note that we only compared the residuals between the before and after calibration sensor data and ground truth inertial data. This is because the output of the data-driven calibration model does not explicitly distinguish between various types of sensor errors and is not directly related to the underlying IMU physical characteristics [35].

According to the sensor error model (3.12), we computed the raw error at time step $i$ as

$$\begin{aligned} \boldsymbol{\varepsilon}_{a,i} &= \tilde{\boldsymbol{a}}_i - \hat{\boldsymbol{C}}_a^{-1} \boldsymbol{a}_i, \\ \boldsymbol{\varepsilon}_{g,i} &= \tilde{\boldsymbol{\omega}}_i - \hat{\boldsymbol{C}}_g^{-1} \boldsymbol{\omega}_i, \end{aligned} \tag{4.3}$$

where $\tilde{\boldsymbol{a}}_i$, $\tilde{\boldsymbol{\omega}}_i$ are the raw acceleration and angular velocity, respectively. $\hat{\boldsymbol{C}}_a^{-1}$ and $\hat{\boldsymbol{C}}_g^{-1}$ are the optimized scale factor and axis-misalignment for aligning the IMU

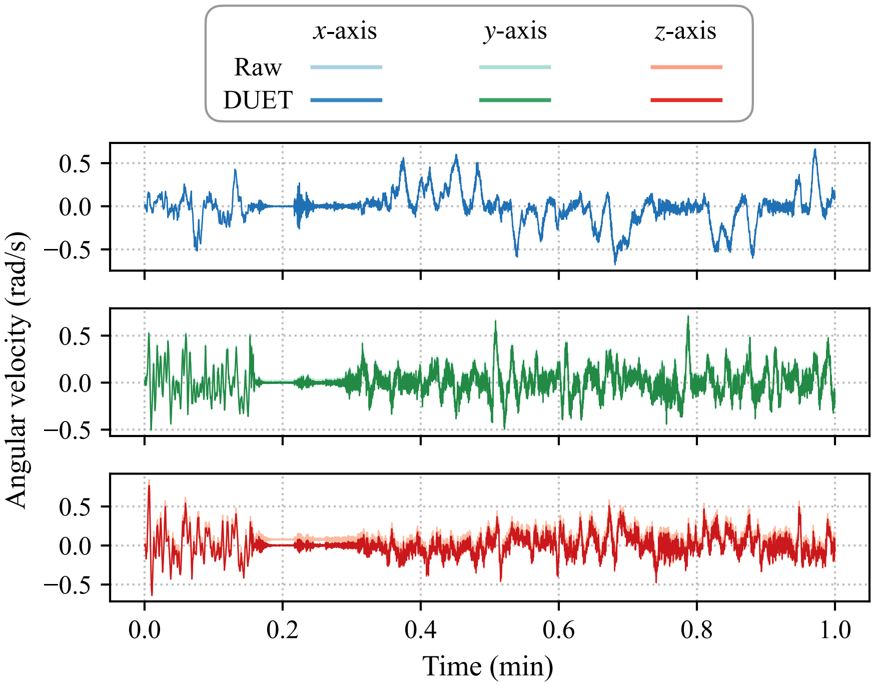


Figure 4.2: Raw and deep calibrated angular velocity of test sequence EuRoC-MH_04_difficult. Our method removed the slight drift from raw angular velocity without changing the overall magnitude.

data. $\boldsymbol{a}_i$ and $\boldsymbol{\omega}_i$ are the ideal IMU data derived from the interpolated ground truth velocity and orientation as

$$\boldsymbol{a}_i = \boldsymbol{R}_i^T \left( \frac{(\boldsymbol{v}_{i+1} - \boldsymbol{v}_i)}{\Delta t} - \boldsymbol{g} \right), \tag{4.4}$$

$$\boldsymbol{\omega}_i = \log_{SO3}(\boldsymbol{R}_i^T \boldsymbol{R}_{i+1})/\Delta t. \tag{4.5}$$

We computed the compensated sensor error $\boldsymbol{\varepsilon}_a - \hat{\boldsymbol{\varepsilon}}_a$ and $\boldsymbol{\varepsilon}_g - \hat{\boldsymbol{\varepsilon}}_g$ by removing the predicted error from the raw error. Then the remains were the sensor errors of our calibrated IMU data. We conducted this experiment only on EuRoC dataset as the high-precision velocities are available.

Table 4.2 shows the sensor error comparison results. For each sequence, we averaged the error on each axis to eliminate the noise influence. We show that our method removed the accelerometer and gyroscope errors by an average of 52% and 86%, respectively. Figure 4.1 shows the comparison of raw errors and

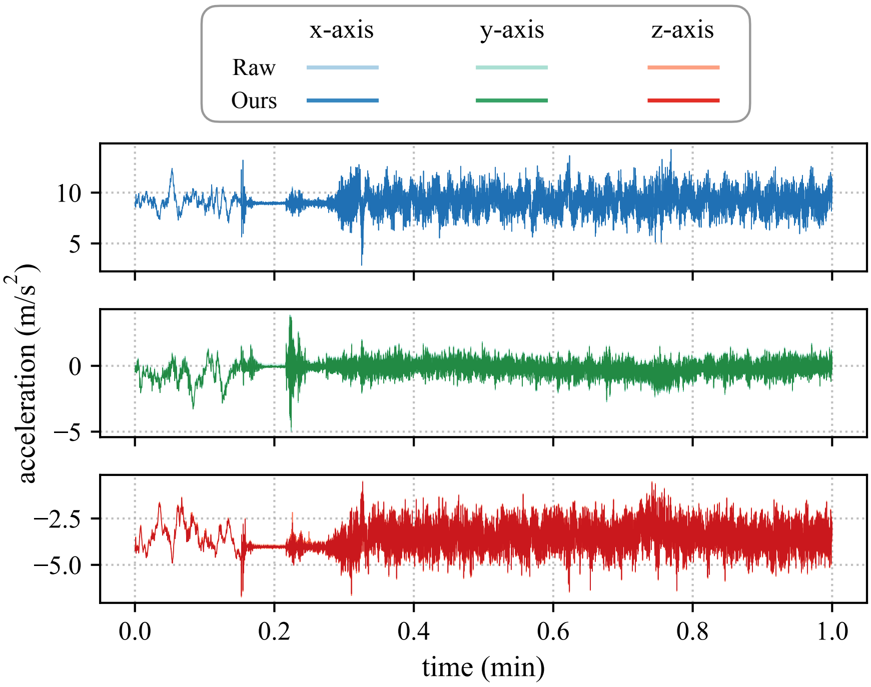


Figure 4.3: Raw and deep calibrated acceleration of test sequence EuRoC-MH_04_difficult. Our method removed the slight drift from the raw acceleration, which is difficult to visually distinguish from the overall signals. However, The integrated velocity errors were reduced by a mean of 77%.

our predicted errors on one test sequence. For the accelerometers, we were unable to predict high-frequency large noise errors. However, our predicted errors had the same trend as the raw errors, which indicates that our method eliminates the major low-frequency errors. Similarly, for the gyroscope with lower noise, low-frequency errors were predicted while high-frequency noise errors remained. Figure 4.2 and Figure 4.3 illustrate the comparison of raw and our calibrated angular velocities and accelerations for one test sequence, respectively. Our method shifted the signal slightly, which is visually hard to notice. However, the errors were reduced by more than 60%.

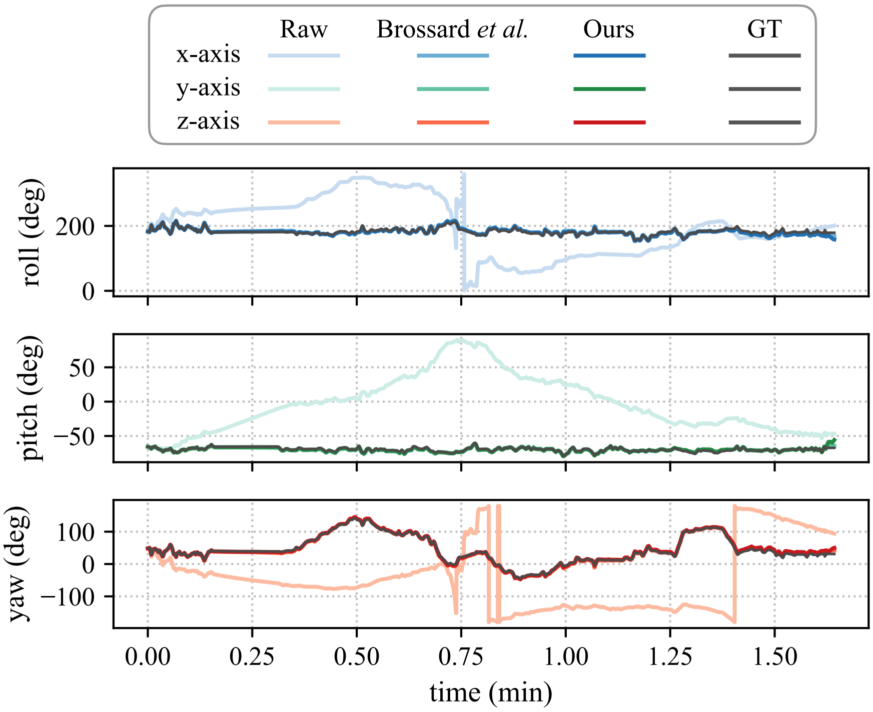


Figure 4.4: Raw and learning-based calibrated orientation of test sequence EuRoC-MH_04_difficult. The orientations estimated from raw angular velocities drift quickly within a few seconds and are unreliable. However, the orientations estimated from the calibrated angular velocities are almost the same as the ground-truth orientations.

## 4.4 Evaluation of Calibration Effect for Inertial-based Odometry

The evaluation of the velocity, orientation, and position that are calculated from calibrated IMU data allows an intuitive perception of the change caused by the calibration. Thus, we inferred and evaluated these data from raw IMU data and calibrated IMU data using the kinematic motion model (3.6), (3.7), and (3.8).

### 4.4.1 Comparison of Orientation Estimates

We evaluated the orientation estimates using the following metrics:

- *Absolute Orientation Error (AOE)*: the root mean square error (RMSE) between the ground truth and inferred orientation for a whole trajectory sequence as

$$\mathrm{AOE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} || \log_{SO3}(\boldsymbol{R}_i^T \hat{\boldsymbol{R}}_i)||_2^2}, \tag{4.6}$$

  where $\boldsymbol{R}_i$ and $\hat{\boldsymbol{R}}_i$ denote the ground truth and inferred rotation matrices from the angular velocity at time step $i$ through (3.6). $\log_{SO(3)}(\cdot)$ is the $SO(3)$ logarithm map.

- *Absolute Yaw Error (AYE)*: the RMSE between the ground truth and inferred yaw error for a whole trajectory sequence as

$$\mathrm{AYE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\gamma_i - \hat{\gamma}_i||_2^2}, \tag{4.7}$$

  where $\gamma_i$ and $\hat{\gamma}_i$ denote the ground truth and inferred yaw at time step $i$. We measure AYE for comparison with the competing method Huang *et al.* [16] as it only reported this metric.

In terms of the orientation estimates, the closest works to ours are Brossard *et al.* [15], Huang *et al.* [16], and Buchanan *et al.* [21]. For the EuRoC dataset, we ran Brossard *et al.* ten times with the same setting using the source code they provided, then took the average results as the baseline for the comparison. Huang *et al.* and Buchanan *et al.* also used the results of Brossard *et al.* as baselines. However, the baselines were evaluated in different environments and metrics. Thus, we used the results provided in their papers directly and show relative improvement for comparison. Note that Huang *et al.* only reported AYE while Buchanan *et al.* reported AOE. For better comparison, we reported both. In addition, we calculated the AOE and AYE resulting from the Madgwick orientation filter [26] for comparison with traditional filter-based methods. For TUM-VI (Uncali) and TUM-VI (Cali), we compared with Madgwick *et al.* and Brossard *et al.* that used the same evaluation strategy as on EuRoC.

Table 4.3 summarizes the results on EuRoC. Overall, errors of learning-based methods are smaller than that of the traditional IMU-only filtering-based method, which is consistent with the findings in OriNet [17]. In terms of absolute yaw error, we slightly outperformed Huang *et al.*, even though they use a more sophisticated TCN network structure. However, in the case of AOE, Buchanan *et al.* is better than ours. There are two reasons for this. First, they directly use ground truth bias for learning and prediction. This allows them to learn pure biases of the gyroscope without any external impact. However, we use orientation for model learning, which is more practical but is affected by noise and dynamic motion. Second, the results may have been caused by network differences because they use a more sophisticated Transformer network, while we use the basic CNN network. Note that in this work, instead of using complex networks and optimizing network parameters to improve performance, we focus on a more general and practical framework for deep IMU calibration from dynamic high-precision motion data for inertial-based odometry. Figure 4.4 illustrates the comparison of orientation estimates.

Table 4.4 shows the AOE on uncalibrated and calibrated TUM-VI dataset. For the synthesized uncalibrated sequences, we achieved an improvement of over 90% compared to the raw data and outperformed the Brossard *et al.* by 10% to 20%. This indicates that our method can remove the gyroscope sensor error and improve the accuracy of orientation estimates. Even for calibrated sequences, our method achieved an improvement of more than 50% in orientation estimates and was comparable to that of Brossard *et al.*

### 4.4.2 Comparison of Velocity Estimates

We evaluated the performance of calibrated acceleration by comparing the inferred velocity with that of raw acceleration using the absolute velocity error (AVE), which is defined as:

$$\mathrm{AVE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{v}_i - \hat{\boldsymbol{v}}_i||_2^2}, \tag{4.8}$$

where $\boldsymbol{v}_i$ denotes the ground truth velocity at time step $i$, and $\hat{\boldsymbol{v}}_i$ is the velocity computed through (3.7) from the acceleration. To eliminate the impact of orientation errors, we rotated the acceleration using the ground truth orientation. We only measured the AVE on EuRoC dataset as they provided the accurate veloc-

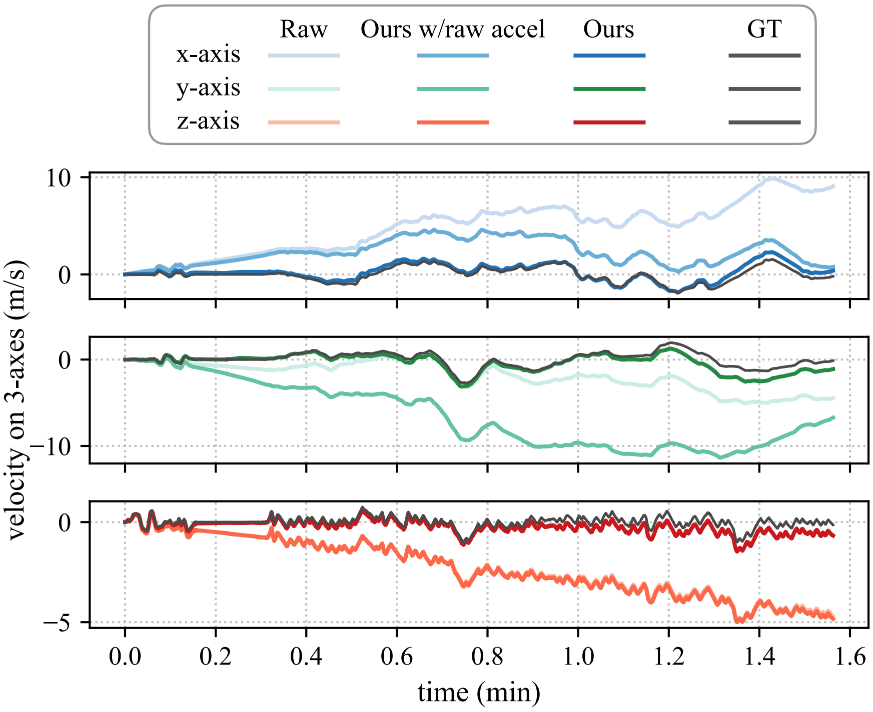


Figure 4.5: Raw and deep calibrated velocity of test sequence EuRoC-MH_04_difficult. The velocities estimated from the raw accelerations drift linearly with time. The absolute velocity error was achieved at 15 $m/s$ along three axes. However, our method calibrated the acceleration correctly and reduced the absolute velocity error to approximate 1 $m/s$.

ity ground truth. In addition, we replaced our loss function using mean square error (MSE) and trained using the ideal acceleration that was derived from the interpolated ground truth velocity and orientation through (4.4). This follows the same idea as in [22].

Table 4.5 summarizes the AVE on EuRoC test sequences. Compared to the results from the raw acceleration, we improved the accuracy by more than 50% on all test sequences. Besides, we reduced the error to nearly 1 m/s on three of the test sequences. Compared to the results based on the MSE loss function, we achieved comparable results that were more robust. This demonstrates that our proposed position-based loss function not only achieves a similar effect as based on ground truth acceleration, but also effectively avoids network overfitting and being affected by anomalous acceleration. Figure 4.5 illustrates the comparison of velocity estimates.

Although accelerometer calibration can be achieved on the basis of ground truth acceleration, in practice it is difficult to obtain relatively accurate acceleration using only high-precision pose tracking devices. Current studies [20] typically extrapolate this information solely from ground truth positions and orientations as

$$\boldsymbol{a}_i = \frac{\boldsymbol{p}_{i+1} - 2\boldsymbol{p}_i + \boldsymbol{p}_{i-1}}{\Delta t^2}. \tag{4.9}$$

However, this derivation is impaired in practice by the position jitter of the tracking system. As shown in Figure 4.6, the derived acceleration changed abruptly and had higher noise than the raw acceleration, due to sharp fluctuations in the position ground truth. Therefore, our proposed position-based method is more flexible and robust than methods based on accelerations and angular velocities.

### 4.4.3 Comparison of Position Estimates via Inertial-only Odometry

Position estimates are the most relevant indicator for inertial-based odometry. Thus, we inferred the position from calibrated IMU data using the kinematic motion model used in inertial-based odometry. Then we compared the results with that of raw IMU data. We also compared the results from IMU data calibrated for the gyroscope only to evaluate the effect of calibrating for both the gyroscope and the accelerometer. To evaluate the accumulated error of inertial-only odometry, we used the relative translation error (RTE), which is computed as follows:

$$\mathrm{RTE} = \frac{1}{m}\sum_{i=1}^{m}\sqrt{\frac{1}{n}\sum_{j=1}^{n}||\hat{\boldsymbol{p}}_{j,T'} - \boldsymbol{p}_{j,T'}||_2^2}, \tag{4.10}$$

where $\boldsymbol{p}$ and $\hat{\boldsymbol{p}}$ are the ground truth position and the estimated position, respectively. $n$ is the number of samples during a duration window $T'$. We randomly chose $m$ segments of length $T'$, computed the estimated position from a ground truth start, and then averaged the results. RTE is more suitable than absolute translation error for evaluating inertial-only odometry because it excludes the

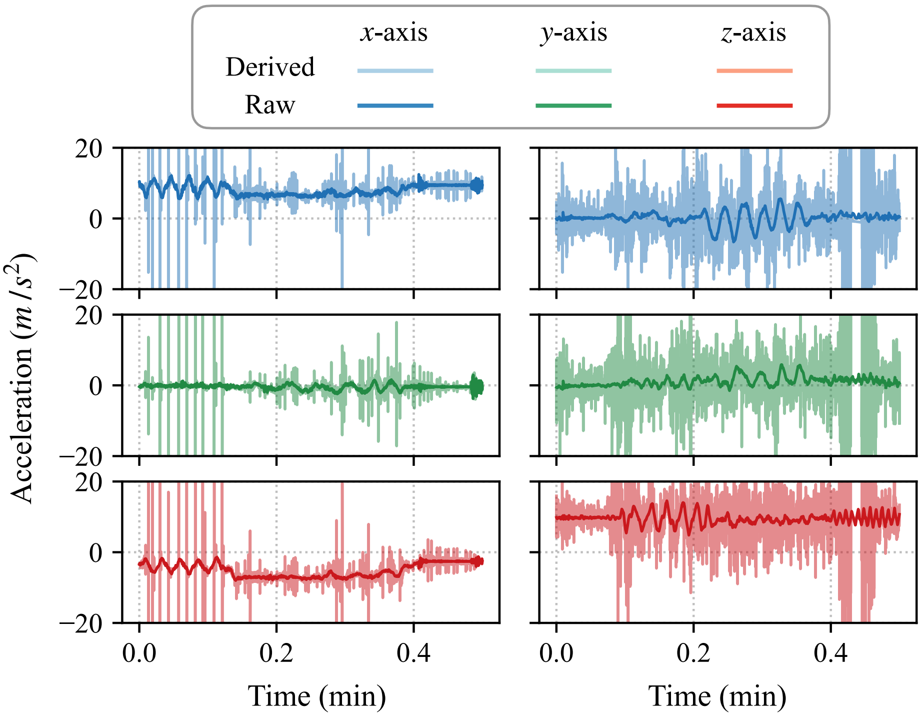


Figure 4.6: Comparison of raw acceleration and derived acceleration from ground truth positions and orientations of EuRoC MH_02_easy (left) and TUM-VI Cali Room2 (right). The derived acceleration suffered from position fluctuation.

influence of sequence length on the results and focuses only on the accumulated error over a specific duration. We set $m = 50$, $T' = 30s$ in our experiments.

As shown in Figure 4.7, the position estimates were unreliable on EuRoC and TUM-VI (Uncali) datasets, in which the IMUs were uncalibrated. In contrast, our method reduced the position error accumulated in 30 s by approximately 90%, when compared to that obtained from raw IMU data. Even compared to the position estimates from the already well-calibrated IMU, we reduced the errors by at most 30%. Moreover, our accumulated errors were reduced by at least 20% compared to the errors of calibrated gyroscope only. This indicates that calibrating the gyroscope and accelerometer further reduces the error accumulation rate than calibrating the gyroscope only.

Figure 4.8 illustrates the error accumulation with time for one test sequence

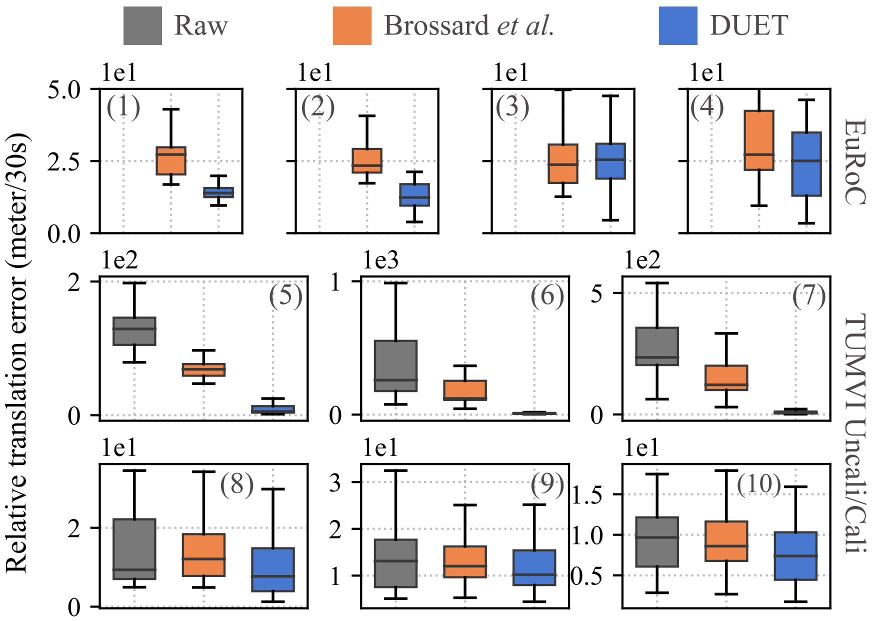


Figure 4.7: Relative Translation Error (RTE) in 30 s on the test sequences of EuRoC (top), TUM-VI Uncali (middle), and TUM-VI Cali (bottom). Our method significantly reduced the accumulated error on the sequences that use uncalibrated IMU while slightly improved the accuracy on the sequences that use calibrated IMU.

of EuRoC. The position errors from raw IMU data and IMU data measured only with accelerometer calibration increase exponentially with time, and exceed 1 km in 30 s. Our method slows down the rate of error accumulation and reduces the error to 10 m in 30 s. Compared to only gyroscope calibration, the error is reduced by more than 50%.

### 4.4.4 Comparison of Position Estimates via Visual-inertial Odometry

Data-driven IMU calibration is still being studied as a promising approach to improve the accuracy and robustness of VIO. As Zhang *et al.* [19] and Buchanan *et al.* [21] reported, the processed IMU data through data-driven methods improved the position estimate accuracy of VIO by at most 66%. In this work, we eval-

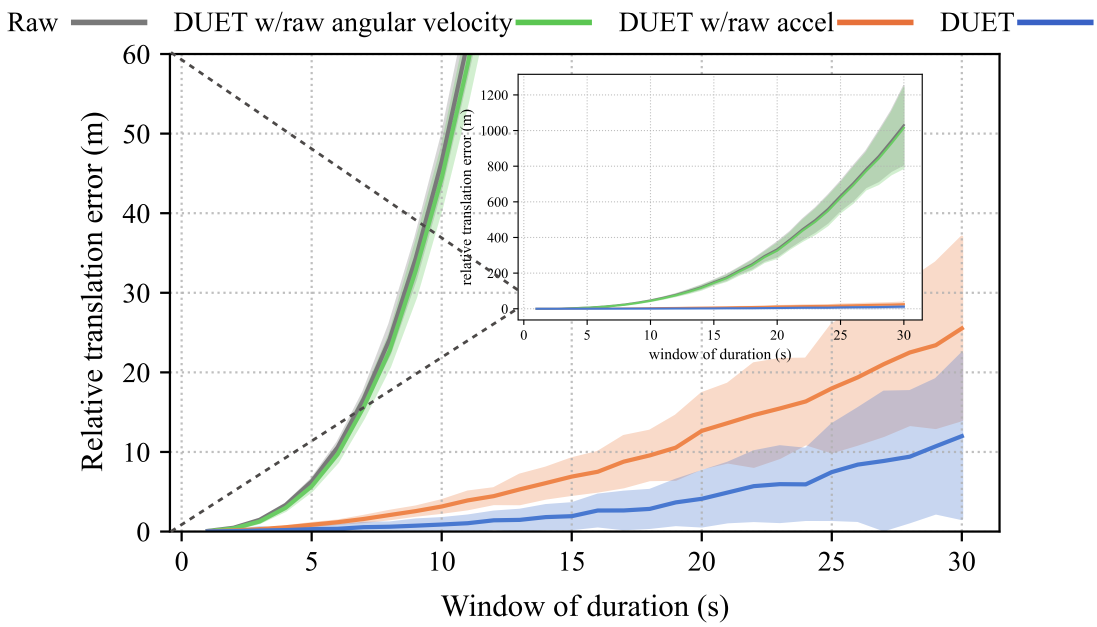


Figure 4.8: Relative Translation Error (RTE) on test sequence MH_04_difficult of EuRoC dataset. The error from raw IMU data exponentially increased with time, while our method reduced the accumulated error to 10 m in 30 s.

uated the position estimates from VINS-Mono [28] with raw IMU data and our calibrated IMU data to assess the impact of data-driven IMU calibration on VIO in different scenarios. Note that we only listed the results of competing methods but did not directly compare them because each method was based on a different VIO algorithm. We used the absolute translation error (ATE) to measure the performance, which is computed as

$$\text{ATE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\hat{\boldsymbol{p}}_i - \boldsymbol{p}_i||_2^2}, \tag{4.11}$$

where $n$ is the number of samples over a whole sequence.

Table 4.7 shows the ATE on all test sequences. For EuRoC test sequences (No.1-4), our calibrated IMU data reduced the position estimates error by an average of 20%. This improvement is limited because the good visual conditions of these sequences allowed high accuracy of VIO. For TUM-VI Uncali test sequences (No.5-7), the VINS-Mono failed with raw IMU data. As shown in Figure 4.9, a dramatic position drift appears at the beginning of VIO with raw IMU trajectory. This is because TUM-VI does not start from a stationary state and the fast rotation challenging scenes are difficult to track accurately at the beginning, resulting in the IMU not being well calibrated in real-time. In contrast, our calibrated IMU helps VIO overcome the shaky start and ensure continued odometry. This indicates that our method can improve the robustness of VIO, especially for challenging visual situations. However, for TUM-VI Cali test sequences (No.8-10), our method was not able to further improve the calibrated IMU but had a negative effect on the position estimates accuracy of VIO. This is because performing data-driven calibration on the IMU data that has already been optimized using the global optimization method impairs the data and further leads to a reduction in the position estimate accuracy. This indicates that our method is not applicable to optimized and processed IMU data.

In Section 4.3, we showed that our method successfully predicted the low-frequency sensor errors and removed them from the raw gyroscope and accelerometer measurements. In Section 4.4.1, Section 4.4.2, and Section 4.4.3, we conducted experiments in terms of IO and showed that our method reduces the orientation, velocity estimates errors as well as the error accumulation rate. We verified the feasibility of data-driven IMU calibration from high-precision tracking trajectory only, which is a promising method for calibrating IMU in VIO. To further explore the impact of our method on VIO, in Section 4.4.4, we conducted experiments on three different scenarios. The results showed that: 1) for good visual conditions and uncalibrated IMU, our method slightly improves the position estimates accuracy; however, 2) for calibrated IMU, our method impairs the position estimates; 3) once visual information is temporarily unreliable, our method is an effective way to improve the robustness and accuracy of VIO.

## 4.5 Discussion on Impact of Bias Changes

We note that in Table 4.2, the error reduction on the gyroscope is better than on the accelerometer. This is primarily due to the substantial changes in accelerometer errors, especially for the biases. This issue is also known as the generalization issue of data-driven approaches, i.e., trained on one domain and tested on another domain suffered from performance degradation [35]. Therefore, we provide experimental evaluation regarding the impact of bias changes on the data-driven model performance. We analyzed this impact by investigating provided biases of EuRoC sequences and performing further experiments through the training-test splitting of the data collected on different days.

As shown in Figure 4.10, we calculated the mean bias of each sequence in EuRoC. Overall, the biases of the acceleration vary considerably for each sequence. This has resulted in a general AVE improvement in the range of 80% to 90%, as shown in Table 4.5. Notably, the acceleration biases of the V2_02_medium sequence in the x-axis and the y-axis were completely different from others. These can be considered outlier patterns for the data-driven model, resulting in only a 51% improvement in AVE. In contrast, the changes in biases for angular velocity are more minor than those for acceleration. As such, the AOE has more than 90% improvement on all test sequences.

The low-cost MEMS IMU errors vary with time. To further validate the impact of error changes on model performance, we split the EuRoC into training and test data according to the data collection time, i.e., five MH sequences collected on the same day into training data and five V(icon) sequences into test data. Among the five test sequences, V1_02 and V1_03 were collected one day after the collection of MH sequences, and the other three test sequences were collected four months later.

A similar finding is evidenced by the results in Table 4.8. The model consistently performs above 90% in reducing the AOE where the bias changes among all sequences are not substantial. Nevertheless, the performance still degraded by about 5% on the test sequences collected four months later. As for the acceleration error mitigation, the model performance has degraded due to the large magnitude of bias changes. The average AVE improvement in the first four test sequences is 55%, ranging from 36% to 87%. On the V2_03, the completely different biases on all three axes result in a negative improvement on the AVE.

The generalization problem is always a major challenge for deep learning methods. Further experiments have shown that large differences between training and test data patterns lead to model performance degradation, which confirms the challenges summarized in the previous work [35, 45]. We now provide two points that we consider useful to solve this issue and push the model into practice in the future: 1) the current proposed method does not take into account the factors that will significantly affect the errors such as temperature, dynamic forces, etc. The integration of these variables into the data-driven model may improve the adaptability of the model to different scenarios; 2) it may prove helpful to integrate the current framework with potential online adaptation learning strategies, such as few-shot learning and test-time optimization.

## 4.6 Discussion on Choice of $T$

In our loss function, $T$ is a key hyperparameter because it affects the effectiveness of model learning and training time complexity. Thus, we emphasize a few important points to choose a proper $T$ successfully. We evaluated the effect of different $T$ on the training time and the mode performance on the EuRoC dataset, respectively.

As shown in Figure 4.11a, as $T$ increases, RTE and AVE decrease and stabilize when T reaches 64. This exhibits the negative effect of position jitter on model training when $T$ is small, e.g., $T = 4$, and verifies that increasing $T$ mitigates this effect effectively. However, AOE increases as $T$ continues to increase, e.g., from 128 to 256. This is because our loss function aims to minimize the error of accumulated values within $T$ windows, and an overly large $T$, that is, a large window size, would be detrimental to learning local error patterns and thus reduces the accuracy. This type of variation is similar to a low-pass filter in that a suitable value is chosen to reduce high-frequency noise and ensure that the necessary information is not lost.

Figure 4.11b illustrates the 1) training time per epoch; 2) computation time of $\mathcal{L}_1$; 3) computation time of $\mathcal{L}_2$ with respect to $T$. As $T$ increases, the total training time increases, which is mainly caused by the increased computation time of $\mathcal{L}_2$. We compute the accumulated orientation within each window in parallel. Thus, the larger $T$ is, the more rotation matrices need to be multiplied within each window. However, by computing two adjacent cumulative orientations in parallel within each window, the time complexity of $\mathcal{L}_2$ can be reduced from $O(T)$ to $O(\log(T))$ when $T$ is an exponent of 2 [15]. Therefore, as shown in Figure 4.11b, the running time does not increase noticeably when $T$ is a power of 2. We note that $T$ does not affect the computation time of $\mathcal{L}_1$. In calculating $\mathcal{L}_1$, we first rotate all accelerations into the fixed world frame, then divide them according to the window size $T$ and sum up accelerations within each window. Because $T$ has a negligible impact on the computation time of the summation and the acceleration rotation is independent of $T$, the computation time of $\mathcal{L}_1$ does not vary with $T$.

The choice of $T$ depends on the extent of the jitter in the ground truth motion data of the training set. Also, while guaranteeing the performance of the trained model, a smaller $T$ can shorten the training time.

Table 4.2: IMU Sensor Errors on EuRoC Test Sequences, Acceleration (Improvement) [$\times 10^{-2} m/s^2$ (%)]/Angular Velocity (Improvement) [$\times 10^{-2} rad/s$ (%)].

| | x-axis | |
|---|---|---|
| No. | $\varepsilon_a/\varepsilon_g$ | $(\varepsilon_a - \hat{\varepsilon}_a)/(\varepsilon_g - \hat{\varepsilon}_g)$ |
| 1 | 0.40/0.26 | 0.24 (41%)/0.025 (90%) |
| 2 | 3.52/0.22 | 1.02 (71%)/0.084 (61%) |
| 3 | 1.26/0.38 | 0.64 (49%)/0.073 (81%) |
| 4 | 2.75/0.13 | 0.38 (86%)/0.076 (42%) |
| average improvement: | | 62%/69% |
| | y-axis | |
| No. | $\varepsilon_a/\varepsilon_g$ | $(\varepsilon_a - \hat{\varepsilon}_a)/(\varepsilon_g - \hat{\varepsilon}_g)$ |
| 1 | 6.30/2.11 | 0.39 (94%)/0.033 (98%) |
| 2 | 1.69/2.10 | 0.29 (83%)/0.247 (88%) |
| 3 | 9.30/2.55 | 6.69 (28%)/0.243 (90%) |
| 4 | 3.31/2.48 | 6.50 (-96%)/0.120 (95%) |
| average improvement: | | 27%/93% |
| | z-axis | |
| No. | $\varepsilon_a/\varepsilon_g$ | $(\varepsilon_a - \hat{\varepsilon}_a)/(\varepsilon_g - \hat{\varepsilon}_g)$ |
| 1 | 0.44/7.71 | 0.13 (70%)/0.074 (99%) |
| 2 | 2.47/7.67 | 0.88 (64%)/0.233 (97%) |
| 3 | 1.04/7.70 | 0.11 (89%)/0.123 (98%) |
| 4 | 3.45/7.86 | 1.81 (47%)/0.076 (99%) |
| average improvement: | | 67%/98% |

Errors were averaged on each axis to eliminate the noise effect.
All values are shown in absolute terms.
The improvement (%) of $x$ relative to $y = 1 - \frac{x}{y}$, the computation is the same for the other parts in the paper.

Table 4.3: Absolute Orientation Error/Absolute Yaw Error (Improvement) [deg/deg (%)] on EuRoC test sequences.

| | | Huang *et al.*[1] | |
|---|---|---|---|
| No. | Madgwick *et al.* | Baseline[*] | Method |
| 1 | 115.11/44.07 | -/0.67 | -/1.35 (-101%) |
| 2 | 100.04/41.67 | -/1.02 | -/1.19 (-17%) |
| 3 | 85.25/29.46 | -/1.80 | -/1.00 (**44**%) |
| 4 | 110.74/42.20 | -/1.94 | -/1.63 (16%) |
| | | Buchanan *et al.*[2] | |
| No. | | Baseline[*] | Method |
| 1 | | 3.21/- | 2.86 (11%)/- |
| 2 | | 0.89/- | 0.76 (**15**%)/- |
| 3 | | 4.78/- | 1.87 (**61**%)/- |
| 4 | | 3.78/- | 1.31 (**65**%)/- |
| | | DUET[1] | |
| No. | | Baseline[*] | Method |
| 1 | | 6.20/5.17 | 4.78 (**23**%)/3.43 (**34**%) |
| 2 | | 1.46/1.13 | 1.96 (-34%)/1.29 (**-14**%) |
| 3 | | 1.91/1.28 | 1.73 (9%)/1.12 (13%) |
| 4 | | 3.86/3.02 | 3.66 (5%)/1.57 (**48**%) |

[1] Integration from IMU data only.

[2] Fusion with visual features.

[*] All baselines are Brossard *et al.*, but were evaluated in different ways.

Table 4.4: Absolute Orientation Error Uncalib/Calib [deg/deg] on TUM-VI test sequences.

| No. | Madgwick *et al.* | Brossard *et al.* | DUET |
|---|---|---|---|
| 5/8 | 115.44/14.45 | 15.45/2.49 | **1.77**/**1.72** |
| 6/9 | 59.11/3.55 | 18.70/**0.99** | **1.12**/1.09 |
| 7/10 | 104.80/5.37 | 20.02/2.02 | **2.05**/**1.91** |

Table 4.5: Absolute Velocity Error and Improvement [$m/s$ (%)] on EuRoC test sequences.

| No. | Raw | MSE | DUET |
|---|---|---|---|
| 1 | 11.66 | 11.26 (3%) | 0.76 (**93**%) |
| 2 | 6.85 | 2.85 (58%) | 0.99 (**86**%) |
| 3 | 4.88 | 0.62 (**87**%) | 1.02 (79%) |
| 4 | 7.20 | 3.52 (**51**%) | 3.51 (**51**%) |

Table 4.6: Absolute Translation Error (m) on EuRoC Test Sequences.

| No. | Zhang *et al.* [19] | Buchanan *et al.* [21] | Raw | DUET |
|---|---|---|---|---|
| 1 | 0.15 | 0.13 | 0.084 | **0.072** (14%) |
| 2 | 0.14 | 0.25 | 0.131 | **0.113** (14%) |
| 3 | **0.15** | 0.17 | 0.194 | 0.165 (15%) |
| 4 | 0.10 | 0.10 | 0.159 | **0.095** (40%) |

Table 4.7: Absolute Translation Error (m) on TUMVI Test Sequences.

| No. | Raw[1] | DUET | No. | Raw | DUET |
|---|---|---|---|---|---|
| 5 | - | **0.244** | 8 | **0.053** | 0.122 (-130%) |
| 6 | - | **0.211** | 9 | **0.024** | 0.084 (-250%) |
| 7 | - | **0.156** | 10 | 0.064 | **0.063** (2%) |

[1] Vins-mono with raw IMU data failed in test sequence 5, 6, and 7.

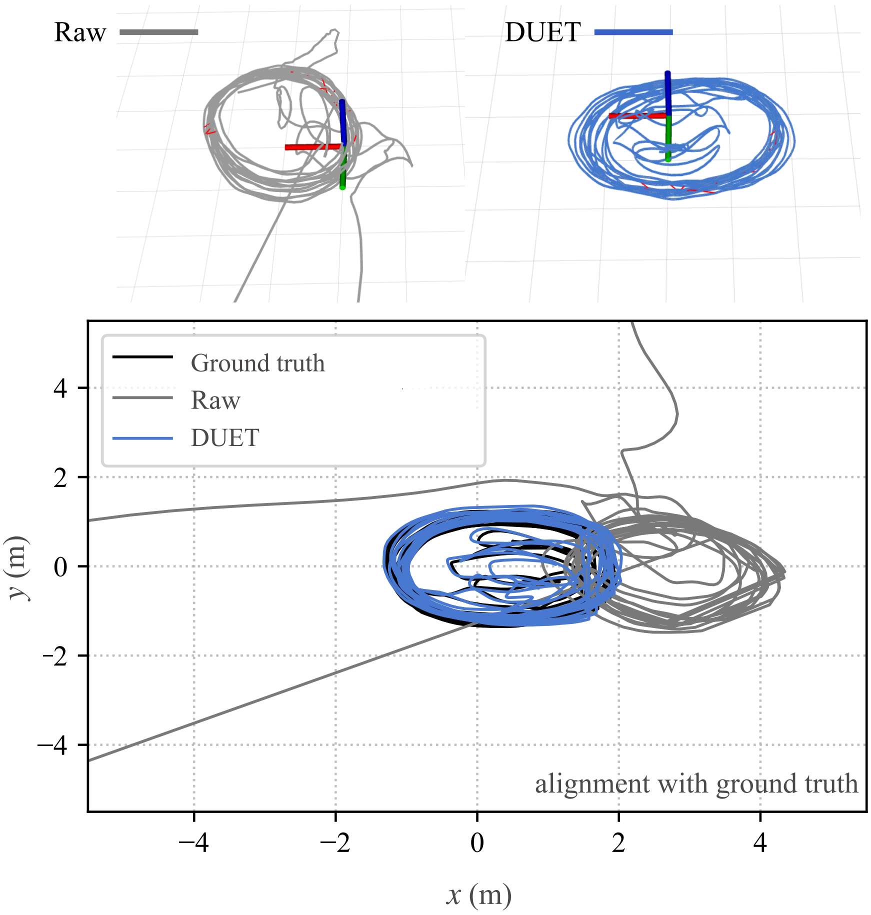


Figure 4.9: VIO 3D (top) and horizontal (bottom) trajectories of TUM-VI Uncali Room 2 sequence. VIO with raw IMU failed when visual information was unstable. Our data-driven calibrated IMU data helped VIO overcome the unstable period and ensured continued odometry.

Table 4.8: Errors and Improvement (AOE: [deg (%)], AVE: [$m/s$ (%)]) on EuRoC test sequences.

| Test seq. | AOE (Madgwick *et al.*/DUET) | AVE (Raw/DUET) |
|---|---|---|
| V1_02_medium | 76.81/1.91 (98%) | 2.56/1.64 (36%) |
| V1_03_difficult | 85.25/1.34 (98%) | 4.88/0.63 (87%) |
| V2_01_easy | 111.92/8.73 (92%) | 9.25/5.32 (42%) |
| V2_02_medium | 110.74/5.34 (95%) | 7.20/3.33 (54%) |
| V2_03_difficult | 92.92/4.59 (95%) | 5.89/7.15 (-21%) |

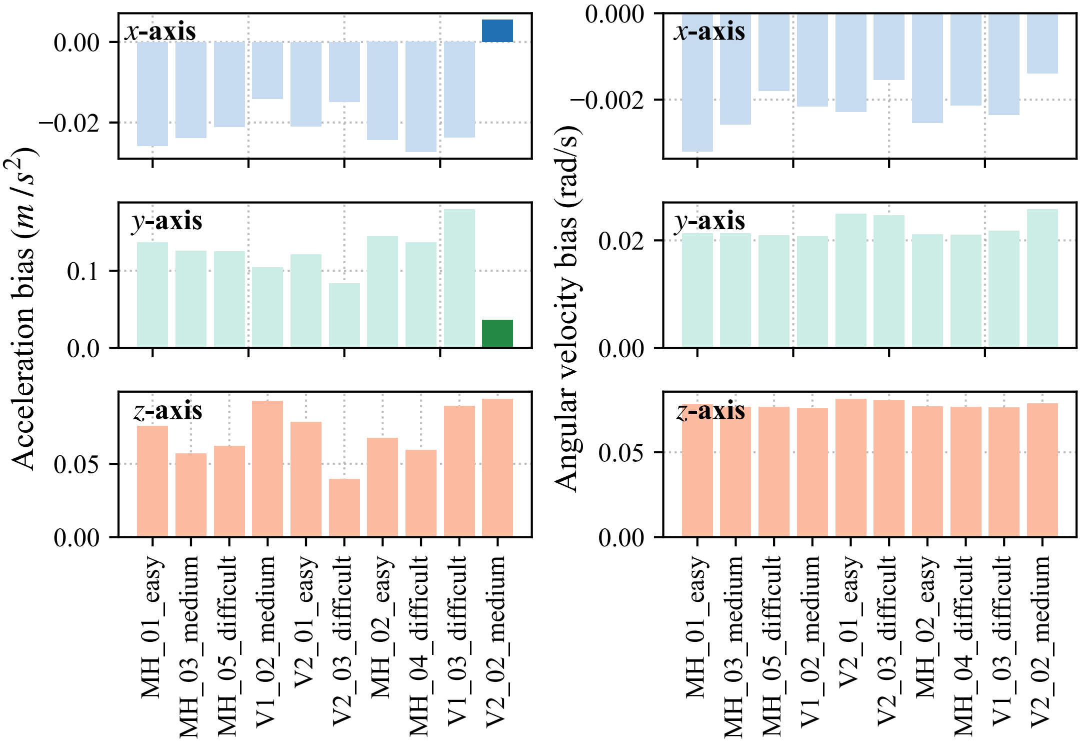
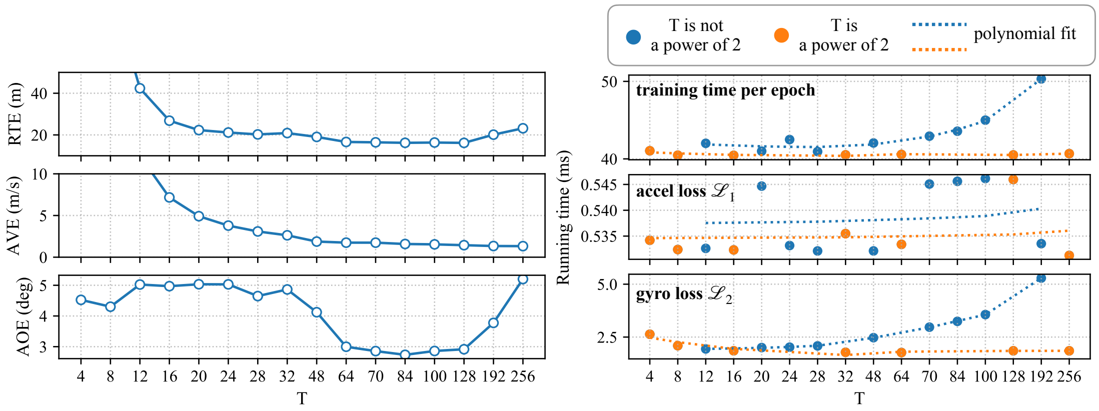


Figure 4.10: Biases of acceleration (left) and angular velocity (right) for each sequence of EuRoC. The ground truth biases were estimated by fusing IMU measurements and high-precision tracking data. We take the mean bias of each sequence and each axis.



(a) Error trend with respect to $T$.

(b) Running time trend with respect to $T$.

Figure 4.11: The effect of $T$ on (a) model performance; and (b) training time. As $T$ increases, RTE and AVE decrease and stabilize until $T$ reaches 64. AOE is lower when $T$ lies between 64 and 128. The running time increases with $T$, which is mainly caused by the increased computation time of $\mathcal{L}_2$.

# 5 Conclusion

We present a deep data-driven IMU calibration method for learning and compensating for sensor errors. The learning model is designed by fully considering the sensor error and kinematic motion models. This allows our method to learn the sensor error solely from high-precision positions and orientations. Compared with similar learning-based methods, our proposed method is more generally practical and straightforward because it does not require additional sensors and data processing techniques for the captured data to learn the calibration model. By compensating for run-time sensor errors, our method reduces the IO error accumulation rate, which we demonstrated with comprehensive experiments on two public VIO datasets. We show that our method reduces the absolute translation error of the baseline VIO method by 20% on average in scenarios with good visual information and ensures consistent odometry in unstable visual conditions.

# 6 Limitations

Though learning-based methods have been shown to outperform traditional methods in some cases, there are still challenges in their deployment in real-world scenarios. For example, performance degradation in real-world applications may occur due to significant data gaps and sudden outliers. Traditional methods, on the contrary, still possess unique advantages in these situations. Future work should address how a profound integration of the strengths of both approaches would push the state of the art even further.

According to the analysis of the evaluations, improving the generalizability of the data-driven calibration method is also an interesting venue. For example, expanding the dataset and building a more robust model to better adapt to various scenarios. Considering a closer integration with inertial-based odometry, enabling the current method to estimate calibration uncertainty and to self-evolve online based on the results of other sensors are also promising directions.

# Acknowledgements

I am deeply grateful to my mentors, who have influenced my thinking throughout my graduate journey. They have provided me with the utmost support in life and research, allowing me to walk the research path with confidence and fearlessness. They also make me look forward to my upcoming doctoral studies with great anticipation. First and foremost, I would like to thank my perfect advisor who changed the trajectory of my life, Kiyoshi Kiyokawa, for allowing me to regain my self-confidence when I was doubting myself the most, for showing me the values of hard work and kindness, and for showing me what a perfect mentor and lab leader looks like. I would also like to thank my co-supervisor, Keiichi Yasumoto, for his insightful feedback and support. I would also like to thank my superstar, omniscient advisor, Hideaki Uchiyama, for his unwavering and unreserved guidance and support, for the countless times “I believe you!” he has said to me, and for imbuing me with the value of life-long learning. I am extremely grateful to Monica Perusquía-Hernández, for her guidance towards perfection in every aspect of my research, for taking me on a tour of collaborative research, and for helping me and my partner make a great qualitative improvement in our lives. I would also like to thank Naoya Isoyama and Yutaro Hirao for their valuable research guidance and support. And I would like to thank my undergraduate mentor Ruliang Xiao, for guiding me at the beginning of my research journey.

Collaborative efforts have made my short two-year research life more colorful and interesting, and I am grateful to the brilliant collaborators with whom I have worked. I would like to thank my lab mate, Riku Shinohara, for inspiring me with his brilliant work in nurse activity recognition. A big thank you to all amazing collaborators of wearable eye detection and EPiC competition, Katsutoshi Masai, Felix Dollack, and Chirag Raman. I would also like to thank my best seniors, Daiki Hagimori and Xiaodan Hu, for their guidance and advice that I will benefit for a lifetime.

My family is the strongest support for me to study in a foreign country. To my parents Haitao Liu and Cui Hua—Thank you for your support in every decision I have made, I would not be here without you both. I would also like to thank Mimi, Niuniu, Meimei, and Xiaobai, for filling my life with fun and love. Last but not the least, I am 3,000× grateful for my lifelong partner, Xin Wei.

# Bibliography

[1] S. Adhikary and A. Ghosh, “E-BMI: A gait based smart remote BMI monitoring framework implementing edge computing and incremental machine learning,” *Smart Health*, vol. 24, p. 100277, Jun. 2022.

[2] ——, “Dynamic time warping approach for optimized locomotor impairment detection using biomedical signal processing,” *Biomedical Signal Processing and Control*, vol. 72, p. 103321, Feb. 2022.

[3] J. A. Delmerico and D. Scaramuzza, “A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots,” *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2502–2509, 2018.

[4] E. Marchand, H. Uchiyama, and F. Spindler, “Pose Estimation for Augmented Reality: A Hands-On Survey,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2633–2651, Dec. 2016.

[5] T. Taketomi, H. Uchiyama, and S. Ikeda, “Visual SLAM algorithms: A survey from 2010 to 2016,” *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, Jun. 2017.

[6] S. Herath, H. Yan, and Y. Furukawa, “RoNIN: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, amp; New Methods,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 3146–3152.

[7] M. Servières, V. Renaudin, A. Dupuis, and N. Antigny, “Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking,” *Journal of Sensors*, vol. 2021, p. e2054828, Feb. 2021.

[8] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, “The TUM VI Benchmark for Evaluating Visual-Inertial Odometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 1680–1687.

[9] X. Ru, N. Gu, H. Shang, and H. Zhang, “MEMS Inertial Sensor Calibration Technology: Current Status and Future Trends,” *Micromachines*, vol. 13, no. 6, p. 879, Jun. 2022.

[10] D. Tedaldi, A. Pretto, and E. Menegatti, “A robust and easy to implement method for imu calibration without external equipments,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3042–3049.

[11] L. Ye, Y. Guo, and S. W. Su, “An Efficient Autocalibration Method for Triaxial Accelerometer,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 9, pp. 2380–2390, Sep. 2017.

[12] S.-h. P. Won and F. Golnaraghi, “A Triaxial Accelerometer Calibration Method Using a Mathematical Model,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2144–2153, Aug. 2010.

[13] I. Frosio, F. Pedersini, and N. A. Borghese, “Autocalibration of MEMS Accelerometers,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 2034–2041, Jun. 2009.

[14] F. Ghanipoor, M. Hashemi, and H. Salarieh, “Toward Calibration of Low-Precision MEMS IMU Using a Nonlinear Model and TUKF,” *IEEE Sensors Journal*, vol. 20, no. 8, pp. 4131–4138, Apr. 2020.

[15] M. Brossard, S. Bonnabel, and A. Barrau, “Denoising IMU Gyroscopes With Deep Learning for Open-Loop Attitude Estimation,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4796–4803, Jul. 2020.

[16] F. Huang, Z. Wang, L. Xing, and C. Gao, “A MEMS IMU Gyroscope Calibration Method Based on Deep Learning,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–9, 2022.

[17] M. A. Esfahani, H. Wang, K. Wu, and S. Yuan, “Orinet: Robust 3-d orientation estimation with a single particular imu,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 399–406, 2019.

[18] H. Chen, P. Aggarwal, T. M. Taha, and V. P. Chodavarapu, “Improving Inertial Sensor by Reducing Errors using Deep Learning Methodology,” in *NAECON 2018 - IEEE National Aerospace and Electronics Conference*, Jul. 2018, pp. 197–202.

[19] M. Zhang, M. Zhang, Y. Chen, and M. Li, “IMU Data Processing For Inertial Aided Navigation: A Recurrent Neural Network Based Approach,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 3992–3998.

[20] J. Steinbrener, C. Brommer, T. Jantos, A. Fornasier, and S. Weiss, “Improved State Propagation through AI-based Pre-processing and Down-sampling of High-Speed Inertial Data,” in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 6084–6090.

[21] R. Buchanan, V. Agrawal, M. Camurri, F. Dellaert, and M. Fallon, “Deep IMU Bias Inference for Robust Visual-Inertial Odometry With Factor Graphs,” *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 41–48, Jan. 2023.

[22] D. Engelsman and I. Klein, “Data-Driven Denoising of Accelerometer Signals,” *arXiv preprint arXiv:2206.05937*, Jun. 2022.

[23] I. Frosio, F. Pedersini, and N. A. Borghese, “Autocalibration of mems accelerometers,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 2034–2041, 2009.

[24] X. Lu, Z. Liu, and J. He, “Maximum likelihood approach for low-cost mems triaxial accelerometer calibration,” in *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 01, 2016, pp. 179–182.

[25] B. Fang, W. Chou, and L. Ding, "An Optimal Calibration Method for a MEMS Inertial Measurement Unit," *International Journal of Advanced Robotic Systems*, vol. 11, no. 2, p. 14, Feb. 2014.

[26] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–7.

[27] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a good attitude: A quaternion-based orientation filter for imus and margs," *Sensors*, vol. 15, no. 8, pp. 19 302–19 330, 2015. [Online]. Available: https://www.mdpi.com/1424-8220/15/8/19302

[28] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[29] D. Chen, N. Wang, R. Xu, W. Xie, H. Bao, and G. Zhang, "Rnin-vio: 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.

[30] S. Sun, D. Melamed, and K. Kitani, "IDOL: Inertial Deep Orientation-Estimation and Localization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, pp. 6128–6137, May 2021.

[31] J. P. Silva do Monte Lima, H. Uchiyama, and R.-i. Taniguchi, "End-to-End Learning Framework for IMU-Based 6-DOF Odometry," *Sensors*, vol. 19, no. 17, p. 3777, Jan. 2019.

[32] C. Chen, C. X. Lu, J. Wahlström, A. Markham, and N. Trigoni, "Deep Neural Network Based Inertial Odometry Using Low-Cost Inertial Measurement Units," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1351–1364, Apr. 2021.

[33] Q. A. Dugne-Hennequin, H. Uchiyama, and J. Paulo Silva Do Monte Lima, "Understanding the Behavior of Data-Driven Inertial Odometry With

Kinematics-Mimicking Deep Neural Network," *IEEE Access*, vol. 9, pp. 36 589–36 619, 2021.

[34] 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, Oct. 2020.

[35] C. Chen, "Deep learning for inertial positioning: A survey," *arXiv preprint arXiv:2303.03757*, 2023.

[36] C. Jiang, S. Chen, Y. Chen, B. Zhang, Z. Feng, H. Zhou, and Y. Bo, "A MEMS IMU De-Noising Method Using Long Short Term Memory Recurrent Neural Networks (LSTM-RNN)," *Sensors*, vol. 18, no. 10, p. 3470, Oct. 2018.

[37] F. Yao, Z. Wu, Z. Wei, and D. Wang, "Few-shot Domain Adaptation for IMU Denoising," *arXiv preprint arXiv:2201.01537*, Mar. 2022.

[38] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[39] P. J. Huber, "Robust estimation of a location parameter," *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518, 1992.

[40] A. Kendall, Y. Gal, and R. Cipolla, "Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7482–7491.

[41] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016.

[42] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 4666–4672.

[43] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv preprint arXiv:1412.6980*, Dec. 2014.

[44] I. Loshchilov and F. Hutter, “SGDR: Stochastic Gradient Descent with Warm Restarts,” *arXiv preprint arXiv:1608.03983*, 2016.

[45] Y. Li, R. Chen, X. Niu, Y. Zhuang, Z. Gao, X. Hu, and N. El-Sheimy, “Inertial sensing meets machine learning: Opportunity or challenge?” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 9995–10 011, 2022.

# Publication List

[1] F. Dollack, K. Kiyokawa, H. Liu, M. Perusquía-Hernández, C. Raman, H. Uchiyama, X. Wei, "Ensemble Learning to Assess Dynamics of Affective Experience Ratings and Physiological Change," *Affective Computing and Intelligent Interaction (ACII) 2023*, Cambridge, USA, 2023. (Accepted) [1]

[2] H. Liu, Xin Wei, M. Perusquía-Hernández, N. Isoyama, H.Uchiyama, and K. Kiyokawa, "DUET: Improving Inertial-based Odometry via Deep IMU Online Calibration", *IEEE Transactions on Instrumentation and Measurement*, 2023. (Accepted)

[3] X. Wei, H. Liu, M. Perusquía-Hernández, K. Masai, N. Isoyama, H.Uchiyama, and K. Kiyokawa, "Unobtrusive Refractive Power Monitoring: Using EOG to Detect Blurred Vision," *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Sydney, Australia, 2023. (To appear)

[4] H. Liu, M. Perusquía-Hernández, N. Isoyama, H. Uchiyama, and K. Kiyokawa, "What Can Data-driven Calibration Do for Visual-Inertial Odometry?," *IEICE HCG Symposium 2022*, pp. 1-6, Dec. 2022. (Special Theme Session Award)

[5] R. Shinohara, H. Liu, M. Perusquía-Hernández, N. Isoyama, H. Uchiyama, and K. Kiyokawa, "A Classification Technique based on Exploratory Data Analysis for Activity Recognition,", *The 4th International Conference on Activity and Behavior Computing (Fourth Nurse Care Activity Recognition Challenge)*, Oct. 2022. (Excellent/F1-score Award)

[6] H. Liu, M. Perusquía-Hernández, N. Isoyama, H. Uchiyama, and K. Kiyokawa, "What Can Data-driven Calibration Do for 6DoF Inertial Odometry?," *IPSJ SIGUBI 75*, pp. 1-10, Sep. 2022.

[1] Authors are listed in alphabetical order