Learning IMU Bias with Diffusion Model

Shenghao Zhou[1], Saimouli Katragadda[1], Guoquan Huang[1]

Abstract— **Motion sensing and tracking with IMU data is essential for spatial intelligence, which however is challenging due to the presence of time-varying stochastic bias. IMU bias is affected by various factors such as temperature and vibration, making it highly complex and difficult to model analytically. Recent data-driven approaches using deep learning have shown promise in predicting bias from IMU readings. However, these methods often treat the task as a regression problem, overlooking the stochatic nature of bias. In contrast, we model bias, conditioned on IMU readings, as a probabilistic distribution and design a conditional diffusion model to approximate this distribution. Through this approach, we achieve improved performance and make predictions that align more closely with the known behavior of bias.**

I. Introduction

3D motion tracking is essential to endow mobile devices and autonomous vehicles with spatial intelligence. Due to the recent advancements in MEMS sensing technology, 6-axis IMUs measuring angular velocity and linear acceleration have become ubiquitous and made it possible to estimate 3D motion for sensor platforms at edge with compact size, minimal weight, low power consumption and cost (SWaP-C). However, naive integration of IMU measurements to offer 3D odometry (i.e., acceleration, rotation and velocity) or dead reckoning – without aiding sources such as GPS and vision – often is not reliable and diverges in a very short period of time. Better solutions of *inertial-only odometry (IOO)* than naive inertial integration are desperately needed in practice. For example, consider hand tracking in mobile AR/VR applications, highly dynamic hands can easily move out of the tracking camera's field of view (FOV), leaving only IMU data available to keep motion tracking alive.

If IMU measurements were clean and noise free, then naive inertial integration would solve the IOO problem. The reality is much bitter, primarily due to the time-varying stochastic biases that significantly corrupt the inertial signals. As such, in order to find a better IOO solution, it is almost inevitable to better find IMU bias, which is precisely what this paper seeks to address. IMU bias represents an offset of the output from the input value and encompasses many different types of bias parameters such as in-run bias stability, turn-on bias repeatability, and bias over temperature. Many unforeseeable factors such as temperature and vibrations can affect the IMU bias, which makes it impossible to correctly model it [1], although there are simplified but useful models such as random walk widely used in practice [2], [3].

With the emerging of deep learning, there are attempts to model IMU bias in a data-driven manner with neural networks [4]. These approaches have demonstrated the possibility of regressing bias from IMU readings and subsequently integrating the IMU data to estimate motion with reasonable accuracy over short periods. In particular, one may use a differentiable integration module to integrate IMU readings with the predicted bias removed, and compare the result motion with the ground truth [5], [6]. However, it cannot guarantee the predicted correction to IMU reading is the actual bias. This is because there exists other correction to the IMU reading that can achieve the same or even better integrated motion result, but very different from the true bias. When the supervision is provided indirectly through the integrated motion, the network can learn to make these spurious predictions instead of the real bias. This may not generalize to new data, because the learned correction is not an intrinsic property of IMU, as bias does. Alternatively, one can directly use ground truth bias for supervision [7]. This method currently only shows to work when integrated with camera in an optimization based VINS system. As we show in the experiment, the performance of this method is inferior compared with indirectly supervised methods. Both approaches assume a single true bias value for a given IMU reading, framing the problem as a regression task.

In this paper, we propose to model the IMU bias naturally as a probability distribution conditioned on the inertial reading, instead of a fixed value. This formulation, combined with direct supervision, allows for more accurate and faithful bias prediction. To model this complex distribution, we leverage diffusion model, which has shown promising results in capturing distributions with high uncertainity in tasks such as action planning [8] and human trajectory prediction [9]. In particular, we design a conditional diffusion model that takes feature extracted from the IMU reading as an additional condition code to approximate the underlying IMU-conditioned bias distribution. The IOO with the proposed diffusion model is shown to outperform the regression-based approaches (with both direct and indirect supervision). Additionally, our predicted bias closely resembles to the ground truth in terms of magnitude and variation patterns, showing superior accuracy and generalization.

In summary, the main contributions of this paper include:

- We, for the first time, design a lightweight diffusion model to learn IMU bias for IOO in a data-driven manner, by naturally modeling bias as a probability distribution conditioned on inertial measurements.

This work was partially supported by the University of Delaware (UD) College of Engineering, NSF (SCH-2014264, IIS-2410019), Google ARCore, and Meta Reality Labs.

[1] The authors are with the Robot Perception and Navigation Group (RPNG), University of Delaware, Newark, DE 19716, USA. Email: {shzhou, saimouli, ghuang}@udel.edu

- We experimentally validate that the proposed diffusion model achieves more accurate bias prediction, confirming that our probabilistic modeling approach is effective, outperforming regression-based methods, both with direct and indirect supervision.

II. Related Work

Many IOO methods exist and can be categorized into model-free and model-based approaches, depending on whether or not the IMU bias is explicitly modeled.

A. Model-free Method

Early work explores to leverage motion pattern, with primary applications in Pedestrian Dead Reckoning (PDR) scenarios. Heuristic algorithms such as step counting algorithm with step length estimation [10] and stationary period detection with zero velocity update (ZUPT) [11] are explored. A system combine multiple heuristic algorithms working on mobile phone is presented in [12]. In recent years, there is attempt to use deep learning neural network, to learn to regress the motion from IMU reading end-to-end [13], [14], [15], [16], [17], [18], [19]. These methods show promising results on PDR scenarios, suppressing the classical method. Positional displacement and velocity are explored as the target for network prediction. Some work leverages the equivariance in the IMU reading, as a way to enable self-supervised learning [20] or boost the performance [21], further pushing the limit of this method. However, these methods still implicitly rely on motion pattern. Essentially, these methods use deep learning to capture motion pattern in a data-driven fashion. Noticeably, [19] shows such end-to-end learning can work in drone-racing scenario, though it only works when training and testing is on the same trajectory. In this case, the high-speed drone motion for a particular trajectory becomes a complex motion pattern. This shows deep learning can learn non-trivial motion pattern. Yet, it still can't break the theoretical limitation of the reliance on the patterned motion. In this work, we consider general scenario without patterned motion assumption. In this scenario, model-free method shows inferior performance because it struggles to find motion pattern.

B. Model-based Method

Model-based method aims to estimate the bias from IMU readings, then remove the bias, and use integration to get motion estimation. Early analysis of IMU bias shows many factors such as temperature, vibration and impacts, all affect IMU bias [1]. However, the compound effect is hard to model with analytical model. Popular in the system with IMU and other sensors, random walk model [22] is a simplified choice for bias modeling. It models the bias evolution as a Brownian noise process. However, such model has limited accuracy, and it can't be used without other sensors. Also, it typically requires collecting long period stationary IMU readings for offline calibration to get model parameters.

Recent deep learning methods offer new way to model bias. Since the end goal is to remove the bias, this approach is also referred as denoising approach. Since bias is not directly available as data, some approaches use indirect supervision from integrated motion, leveraging a differentiable integration process. The first work [23] estimates gyro bias only, with integrated rotation as training data. [5] proposes to use supervision from integrated pre-integration terms to regress bias. Recent work [6] uses integrated motion to regress both bias and its uncertainty. It achieves state of the art result on a few datasets. However, indirect supervision has a misalignment between their training target and the network output. Since multiple IMU readings can produce the same integrated result, supervising with integrated result can't guarantee the network can learn the actual bias instead of predicting other signals. Our experiment shows methods trained with indirect supervision will make spurious prediction that is very different from actual bias. This may hurt the generalization ability, since other signals might not generalize to new data even for the same IMU.

Close to our work, [7] proposes to use direct supervision from bias for training. However, it only demonstrates the performance when fusing the bias prediction with vision in a joint factor graph system. As our experiment shows, such direct supervision under regression setting will have limited accuracy. Our method follow the deep learning approach for bias modeling using direct bias supervision. However, different from all the work mentioned above, we deviates from the regression formulation, and treats the bias given IMU reading as a conditional probability distribution.

III. Inertial-Only Odometry

While inertial navigation systems (INS) aided by different exteroceptive sensors (such as vision and GPS) have been widely studied in the literature (e.g., see [24]), IOO requires further investigation as aiding sensors can easily degrade or fail in practice. In this section, we will revisit the IOO problem from an INS perspective while focusing on the bias modeling challenges.

A. Inertial Navigation

IOO shares the same IMU kinematics as INS to estimate motion (i.e., position, rotation and velocity) using IMU (accelerometer and gyroscope) measurements. Each accelerometer measures proper acceleration on only one axis, and are therefore usually found in groups of three orthogonal devices on a single low cost MEMS chip. However, low-cost accelerometer measurements are far from ideal and are corrupted by noise and bias:

$$\mathbf{a}_m(t) = \mathbf{C}({}^I_G\bar{\mathbf{q}}(t))\left({}^G\mathbf{a}(t) - {}^G\mathbf{g}\right) + \mathbf{b}_a(t) + \mathbf{n}_a(t) \qquad (1)$$

where ${}^I_G\bar{\mathbf{q}}$ is the unit quaternion that represents the rotation from the global frame of reference $\{G\}$ to the IMU frame $\{I\}$ (i.e., corresponding to the rotation matrix $\mathbf{C}({}^I_G\bar{\mathbf{q}})$), ${}^G\mathbf{a}$ is the true acceleration of the IMU in the global frame $\{G\}$, ${}^G\mathbf{g}$ is the gravitational acceleration expressed in $\{G\}$, and $\mathbf{n}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{N}_a)$ is zero-mean, white Gaussian noise, and $\mathbf{b}_a$ is the bias changing over time. Like the accelerometer, gyroscope measures angular velocity of the sensor and suffers from noise and bias, and sometimes, misalignment and scale errors.

Moreover, gyroscope measurements are also influenced by acceleration (i.e. g-sensitivity), whose magnitude is negligible if it is within the range of the additive white noise $\mathbf{n}_g$, while in some low-cost MEMS hardware, it can be more significant:

$$\boldsymbol{\omega}_m(t) = \mathbf{T}_g {}^I\boldsymbol{\omega}(t) + \mathbf{T}_s {}^I\mathbf{a} + \mathbf{b}_g(t) + \mathbf{n}_g(t) \tag{2}$$

where $\mathbf{T}_g$ is the shape matrix causing both misalignment and scale errors in the gyro measurements, $\mathbf{T}_s$ is the g-sensitivity coefficient, $\mathbf{n}_g \sim \mathcal{N}(\mathbf{0}, \mathbf{N}_g)$ is zero-mean white Gaussian noise, and the bias $\mathbf{b}_g$ is time-varying and random.

The INS kinematic model is given by [22]:

$$ {}^I_G\dot{\bar{\mathbf{q}}}(t) = \frac{1}{2}\boldsymbol{\Omega}\left({}^I\boldsymbol{\omega}(t)\right) {}^I_G\bar{\mathbf{q}}(t) \tag{3}$$

$$ {}^G\dot{\mathbf{p}}(t) = {}^G\mathbf{v}(t) \tag{4}$$

$$ {}^G\dot{\mathbf{v}}(t) = {}^G\mathbf{a}(t) \tag{5}$$

where ${}^I\boldsymbol{\omega} = \begin{bmatrix}\omega_1 & \omega_2 & \omega_3\end{bmatrix}^T$ is the true rotational velocity of the IMU, and $\boldsymbol{\Omega}(\boldsymbol{\omega})$ is defined by:

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \quad \lfloor\boldsymbol{\omega}\times\rfloor = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

Using the IMU measurements and assuming *known* bias models (e.g., random walk), 3D motion estimates can be obtained by integrating the above continuous-time kinematics. Clearly, the quality of IMU data (affected by noise and bias) determines the motion accuracy.

Because of the (aided) INS observability properties [25], any method that tries to bypass bias modeling and directly predict global position or velocity has the fundamental limitation on the target motion pattern. As we focus on general scenario without prior motion pattern assumption, we limit to estimate motion increment (i.e., odometry), while only assuming known initials if absolute motion is needed.

B. Modeling Bias

As evident, it is critical to find biases for IOO from IMU measurements in order to be able to perform accurate inertial integration to estimate motion:

$$\begin{bmatrix}\mathbf{b_g}(t) \\ \mathbf{b_a}(t)\end{bmatrix} = f_\pi(\boldsymbol{\omega_m}(t), \mathbf{a_m}(t)) \tag{6}$$

where f_π is some estimator. However, finding such estimator is non-trivial because the bias is not deterministic. As an IMU is a physical electronic sensor, factors such as temperature, impacts, vibration, and quantization noise all affect it [1]. These compound effects are complex and difficult to model. Moreover, many of these factors are time-varying, giving the bias a stochastic nature. Not only does the bias change as the IMU operates, but also after the power cycles further complicating model development. As such, it is almost impossible to analytically model the IMU bias.

While building an exact model is challenging, analysis on bias as a black-box signal using Power Spectral Density (PSD) [26] and Allan variance analysis [27] reveal certain bias characteristic, such as angle/velocity random walk, bias instability and rate ramp. These characteristics become standard in the industry for IMU sensors [28], [29]. However, utilizing all these characteristics to build an estimator is difficult because some of them, like bias instability are defined only in frequency domain, without state-space equivalent.

As approximation, in practice, a simplified model leveraging rate random walk is commonly used as the bias model [2], [3]. Specifically, it assumes the bias dynamic model as:

$$\begin{bmatrix}\dot{\mathbf{b}}_g(t) \\ \dot{\mathbf{b}}_a(t)\end{bmatrix} = \begin{bmatrix}\eta_g(t) \\ \eta_a(t)\end{bmatrix} \tag{7}$$

To fit parameters η_g, η_a, a common approach is to collect a sequence of stationary IMU readings and fit them with using Allan variance analysis, e.g., as demonstrated in Kalibr [2]:

$$\begin{bmatrix}\eta_g(t) \\ \eta_a(t)\end{bmatrix} = \text{calibration}(\boldsymbol{\omega}_{\text{m\_static}}, \mathbf{a}_{\text{m\_static}}) \tag{8}$$

The initial values $\mathbf{b_g}(0), \mathbf{b_a}(0)$ require extra heuristics to estimate, such as taking the average of stationary IMU reading and subtract. Although this simple model captures the slow variations in bias, its accuracy is limited and typically requires external sensors to aid inertial navigation. Additionally, the process involves two steps: first, estimating dynamic parameters using specific IMU readings, and second estimating bias based on the dynamic model. The first step calibration is not only time-consuming but also restrictive, as it demands an extended period of stationary IMU readings.

IV. Learning Bias for IOO

In this section, we thus design a deep neural network to represent the modeling function f_π (6), which can be trained end-to-end to predict the IMU bias. This is in contrast to the classical random walk model, which uses a hand-craft two-step pipeline and assumes a long period static IMU reading available. These models are shown to be able to generalize to unseen readings with good accuracy. The success motivates us to take the approach of deep learning based bias modeling.

$$\begin{bmatrix}\mathbf{b_g(t)} \\ \mathbf{b_a(t)}\end{bmatrix} = \text{network}(\boldsymbol{\omega_m}, \mathbf{a_m}) \tag{9}$$

However, different from the literature, we do not treat it as a regression problem, assuming $\mathbf{b_g}, \mathbf{b_a}$ are fixed value. Instead, we model them as probability distribution, as $p(\mathbf{b_g}, \mathbf{b_a} | \boldsymbol{\omega}, \mathbf{a})$. This probability distribution can be very complex, thus deep learning model is a good fit to estimate them.

A. Diffusion Model

Diffusion models [30] are generative models that aims to represent data $\mathbf{x}_0$ using a series of latent codes $\mathbf{x}_1, \ldots, \mathbf{x}_T$ through a forward and reverse diffusion process. The forward process gradually adds noise to the data, encoding it into a structured latent space, while the reverse process decodes the latent code back into the original data. Once trained, the model allows us to sample a latent code $\mathbf{x}_T$ from a simple distribution and generate the corresponding data $\mathbf{x}_0$ by running the reverse diffusion process. The key strength of the diffusion model lies in its ability to model highly complex distribution $\mathbf{x}_0 \sim q(x)$, by leveraging the multiple latent representations between $\mathbf{x}_1$ and $\mathbf{x}_T$. That is why we want to use diffusion model to learn conditional bias distribution. The latent space is structured in such a way that

$\mathbf{x}_T$ follows a simple Gaussian distribution, making sampling straightforward.

The encoding process between two latent codes $\mathbf{x}_{t-1}, \mathbf{x}_t$ is performed by adding Gaussian noise.

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}, \epsilon_{t-1} \sim \mathcal{N}(0,1) \tag{10}$$

where β_t is the hyperparameter that controls the amount of noise added at each step t, and T is the total number of diffusion steps. As t increases, the latent variable $\mathbf{x}_t$ transitions towards pure Gaussian noise.

At the core of the diffusion model is the denoiser network, which is described in Sec. IV-B, aiming to estimate the noise added at each step in the forward process. Given corrupted data $\mathbf{x_t}$ and the step t, the network predicts the noise ϵ_{t-1} added at the previous step:

$$\hat{\epsilon}_{t-1} = \epsilon_\theta(\mathbf{x}_t, t)$$

The denoiser network is trained using the Mean Squared Error (MSE) loss on the noise:

$$\|\epsilon_{t-1} - \hat{\epsilon}_{t-1}\|_2 \tag{11}$$

This simple training loss function is equivalent to minimizing the evidence lower bound (ELBO) from variational inference perspective, which allows the model to approxiamte the underlying distribution of data $\mathbf{x}_0$. To generate a sample from the diffusion model, we first sample a latent code $\mathbf{x}_T$, and decode it back to original data $\mathbf{x}_0$. $\mathbf{x}_T$ follows a Gaussian distribution, as conceptually it is the result of adding Gaussian noise for T steps in the forward process. The sampling process begins with:

$$\mathbf{x}_T \sim \mathcal{N}(0, I) \tag{12}$$

Next, we use denoiser network to iteratively decode $\mathbf{x}_t$ back to $\mathbf{x}_{t-1}$ as follows:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}(\mathbf{x}_t - \gamma_t\epsilon_\theta(\mathbf{x}_t, t)) + \sigma_t z, z \sim \mathcal{N}(0, I) \tag{13}$$

where $\beta_t, \sigma_t, \gamma_t$ are fixed value. The parameter β_t is the same noise variable used in the forward process, while both β_t, σ_t are hyperparameters of the diffusion model, controlling the noise schedule. The parameter γ_t is a fixed function of β_t.

In our bias modeling, $\mathbf{x}_0$ corresponds to the original bias $(\mathbf{b_g}, \mathbf{b_a})$. The bias is the only required training data.

As we want to model the conditional probability distribution of bias given IMU readings, we introduce an additional feature vector $\mathbf{c}$ extracted from the IMU readings. This feature $\mathbf{c}$ serves as conditional code to the denoiser network at each step t of the denoising process, so that we can model the conditional distribution:

$$\hat{\epsilon}_{t-1} = \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) \tag{14}$$

The training and sampling steps remain the same.

B. Model Design

As shown in Fig. 1, we design two models to implement ϵ_θ in equation 14: the IMU encoder and the denoiser network of the diffusion model. The IMU encoder extracts feature code $\mathbf{c}$ and pass it to the denoiser network, as the implementation of $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$.

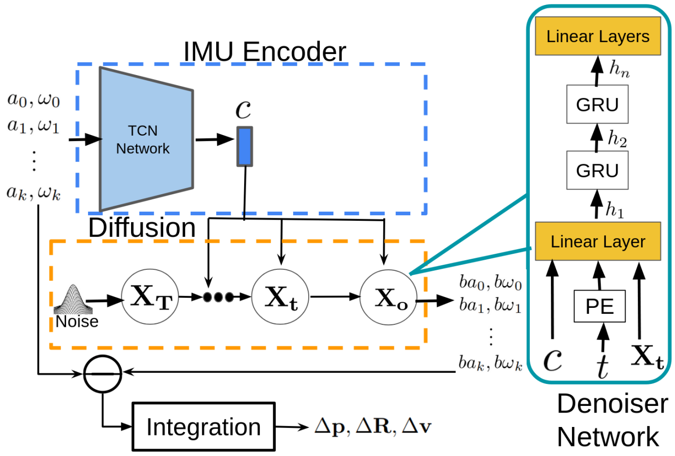

Fig. 1: System overview: our model consists of IMU encoder and denoiser network of the diffusion model. Conditional code $\mathbf{c}$ extracted by IMU encoder from IMU readings is passed to the denoiser network, to generates the bias with multiple diffusion steps. Bias is used to correct the IMU readings for integration, to get the motion estimation.

We choose Temporal Convolutional Network (TCN) as the IMU encoder, because it effectively captures the temporal relation in sequential data. It is easy to train and deploy because it mainly consists of convolutional layers [31]. Previous deep learning based IOO work [16], [17] shows it can extract useful information in IMU reading sequence.

The second component is the denoiser network for the diffusion model. It needs to fuse the conditional code $\mathbf{c}$ from IMU encoder with diffusion model latent code x_t and step number t, and then process the fused code with its backbone to make prediction, and optimize for the loss function in equation 11. The internal structure is illustrated on the right of Fig. 1.

The fusion is done with one linear layer, as a simple design. Since in diffusion models, each denoising step corresponds to a specific noise level, the timestep information is critical. Thus, we add sinusoidal positional embedding to the step number t to provide a smooth, continuous representation of time, inspired by the design of transformer [32].

Deviating from U-Net [33], a popular design for diffusion models, we design a lightweight RNN-based network, because U-Net is computationally expensive with large number of paramaters, making it less suitable for real-time applications where efficiency is the key. Our backbone consists of only a stacked GRU [34] with two cells, followed by a linear layer. Despite the simple architecture, as demonstrated in Table I, our network outperforms U-Net, offering better performance with a significantly smaller architecture.

C. Implementation Details

In practice, we process a window of IMU readings at once, instead of one-by-one, because the network needs context information from IMU readings. However, the window can't be too long either, because the drift inevitably becomes larger as the window is larger, even with correction from predicted bias. We choose one-second window as the window size, inspired the choice of [17] and [7], striking a balance between capturing sufficient temporal information and maintaining the system online performance.

For network training, we allow overlapping between IMU windows taken from the full IMU reading sequence, so the network can see more IMU reading patterns. However, too much overlapping provides very similar data, slowing down the training without clear benefit. In practice, we find 50% overlapping to be a good choice.

The training uses Adam optimizer [35], with learning rate of 3×10^{-5}, taking 6 hours on an NVIDIA A4500 GPU. The noise schedule is linearly spaced between $\beta_1 = 0.0001$ and $\beta_T = 0.02$, with the model trained for 1000 steps, following the default setting in [30].

For sampling, we select DDIM [36] to save sampling steps while maintaining the performance. We use only 25 sampling steps for bias generation, in contrast to the typical 1000 steps required by standard DDPM sampling [30].

D. Acquire Bias Ground Truth Data

To train the model, we require the ground truth bias at the IMU rate. Although the bias is not directly measured by the sensor since it is observable [37], it can be recovered through joint optimization of IMU data and other sensors inputs. Many VINS systems, such as OpenVINS [3], OKVIS [38] and VINS-Mono [39], provide reliable bias estimates as part of their state estimation. When additional sensors like LiDAR [40] or external motion capture system [41] are available, the bias estimation can be further refined.

Although these bias estimates are typically provided at the frame rate, we can interpolate them to match the higher frequency of the IMU. Since IMU bias tends to change slowly over time, the interpolated values offer sufficient accuracy for the use as supervision during training.

Empirically, we find that bias recovered through joint optimization and then interpolated to the IMU rate is of high quality. When the recovered bias is used to correct the IMU data, it results in better motion integration performance compared to bias predicted by deep learning models trained on integrated motion data. Therefore, the recovered bias can serve as an effective ground truth signal to guide our network towards better performance.

Moreover, we observe that the recovered bias is continuous and changes slowly, consistent with out prior understanding of IMU bias behavior. This further supports the validity of using the recovered bias as the ground truth for training the model.

V. Experimental Results

We conduct our experiments on the EuRoC dataset [41], using the same training and testing splits as prior studies [6]. For evaluation metrics, we use relative Positional RMSE (PRMSE in meters) for position and Relative Orientation Error (ROE in degrees), consistent with established conventions in [6].

We compare the performance with the following baselines:

- AirIMU [6], a recent work that predicts bias through indirect supervision using integrated motion. This method achieves state-of-the-art result on EuRoC dataset, outperforming previous work that also use indirect motion supervision [23] [5], as well as model-free methods [17]. As expected, the model-free approach, which relies on motion patterns, performs significantly worse on EuRoC dataset. By comparing our method with AirIMU, we indirectly compare it with model-free methods as well.
- Random walk modeling baseline: For this baseline, we use noise density and random walk rate parameters from offline calibrated results provided by the EuRoC dataset. Since the random walk model treats bias as a stochastic process, its actual performance is difficult to evaluate directly. We provide a strong baseline as the performance upper bound. We take the ground truth bias at the start of each IMU window, and sample multiple bias changes according to the random walk model. The final bias is the sum of the initial ground truth bias and the sampled bias changes. After removing the sampled bias from the IMU readings and integrating the result, we select the best result for each window. It should be noted that this is not a practical algorithm, as it relies on the ground truth data to choose the optimal result. In our experiments, we sample bias changes 50 times per window.
- Direct bias regression. This method follows similar approach to [7], where the network directly regresses bias values using the ground truth as supervision. For a fair comparison, we use the same network architecture as our model, with minimal changes to the output layer to match dimensions required for regression.
  We do not compare directly with the results from [7] because they only show results using predicted bias in a factor graph optimization framework with visual observations, and their code is not publicly available.

The results are presented in table I. Since our model uses a probabilistic formulation, its predictions are samples from the learned IMU-conditioned bias distribution. Thus, the metric reported in the table is averaged over 50 runs. Our model achieves improved performance in terms of position error and the second-best orientation error. Our result is better than the strong random walk baseline, demonstrating that our bias model is more accurate than commonly used random walk model in its best case. Compared with direct regression baseline, our model with almost the same network has better performance. This shows that our probabilistic model formulation can better captures the problem nature than the regression formulation, thus leading to the improved performance.

In comparison to AirIMU, our model has better position error but worse orientation error, resulting in a similar overall performance. As AirIMU is better than the RNN direct regression baseline, who has similar backbone of TCN and GRU, the indirect supervision can offer better accuracy than the direct supervision. However, as we will show in Section V-A, indirect supervision method may suffer from spurious

TABLE I: Motion estimation result for 1-second window on EuRoC dataset (PRMSE / ROE)

Sequence	AirIMU	Ours (RNN)	Direct Regression (RNN)	Random Walk	Ours (UNet)	Direct Regression (UNet)
MH02	0.0234 / 0.0789	**0.0225** / **0.0604**	0.0246 / 0.1370	0.0615 / 0.1380	**0.0227** / **0.0775**	0.0343 / 0.1307
MH04	0.0415 / 0.0708	**0.0410** / **0.0636**	0.0437 / 0.1551	0.0657 / 0.1413	**0.0408** / **0.0593**	0.0462 / 0.1233
V103	0.0583 / **0.1884**	**0.0561** / 0.1931	0.0611 / 0.2369	0.0685 / **0.1762**	**0.0577** / 0.2185	0.0639 / 0.2574
V202	0.0851 / **0.2157**	**0.0703** / 0.2557	0.0777 / 0.3010	0.0813 / **0.1877**	**0.0703** / 0.2627	**0.0664** / 0.4179
Average	0.0521 / **0.1385**	**0.0475** / **0.1432**	0.0510 / 0.2075	0.0693 / 0.1608	**0.0479** / 0.1545	0.0527 / 0.2323

* the best performance
* the second best performance
* V101 not tested as its ground truth accuracy is limited, as reported in [3]

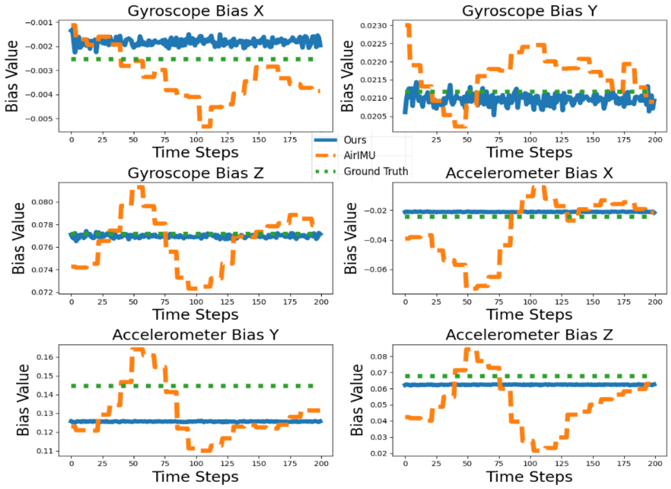

Fig. 2: Bias prediction result for our model and AirIMU in an one-second window

prediction. Our method uses the direct supervision while having similar performance to indirect supervision method, combining the best of two methods. This is thanks to our probabilistic formulation implemented with diffusion model, instead of the existing regression formulation.

A. Diffusion Model vs. Indirect Regression

In this experiment, we compare the bias predictions from our method with those from AirIMU [6]. As mentioned earlier, using indirect supervision through integrated motion can result in spurious bias predictions, as the network may predict unrealistic bias values to optimize motion integration. While this is not an issue when the end goal is accurate integrated motion, it raises concerns about its generalization ability. If the predicted correction does not correspond to the actual IMU bias, it may not capture intrinsic IMU properties and therefore may not generalize well to new data.

We validate this concern in the experiment. As an example, we randomly select one-second IMU reading window and plot the predicted bias values alongside the ground truth. In Fig. 2, our prediction match more closely to the ground truth in both magnitude and changing pattern, In contrast, AirIMU's predictions show abrupt changes, violating our prior knowledge of IMU bias.

Model	Parameters (Millions)	Inference Time (ms)
U-Net Architecture	42.8	170
RNN Architecture	2.2	145

TABLE II: Comparison of model parameters and inference time between U-Net and RNN architectures.

B. Timing on Embedded Device

We evaluate the timing of our pipeline on NVIDIA Jetson AGX Orin embedded device for both the U-Net architecture and the proposed lightweight RNN model. As shown in Table II, the U-Net model has 42.8 million parameters and requires 170 ms for inference, whereas the RNN model is significantly smaller, with 2.2 million parameters, and achieves a faster inference time of 145 ms. This demonstrates the efficiency of the RNN model in terms of both model size and speed.

Low-speed real-world applications can benefit from this inference time, such as agriculture and warehouse robots. For more demanding scenarios such as high-speed drone, further optimization is necessary.

VI. Conclusions and Future Work

In this paper, we have introduced a conditional probability distribution formulation for the IMU bias modeling. Based on this formulation, we have designed a conditional diffusion model to predict the bias from IMU reading, and used it for inertial-only odometry (IOO). Compared with classical random walk bias model and regression based neural network, our model shows better performance and more faithful prediction, which has been validated on the EuRoC dataset, showing the effectiveness of our probabilistic formulation. Although we treat the bias as IMU-conditioned probability distribution, there is more work to be done to leverage the probability distribution to make better bias prediction, rather than taking one random sample as the output. Another direction for future work is to explore how to provide uncertainty for the prediction. In all, we believe our new probabilistic formulation for IMU bias modeling opens up new opportunity to capture IMU bias and benefits the field of IOO.

References

[1] D. Titterton and J. L. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.

[2] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.

[3] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, “OpenVINS: a research platform for visual-inertial estimation,” in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020. [Online]. Available: https://github.com/rpng/open_vins

[4] N. Cohen and I. Klein, “Inertial navigation meets deep learning: A survey of current trends and future directions,” *Results in Engineering*, p. 103565, 2024.

[5] 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)*, 2021, pp. 3992–3998.

[6] Y. Qiu, C. Wang, C. Xu, Y. Chen, X. Zhou, Y. Xia, and S. Scherer, “Airimu: Learning uncertainty propagation for inertial odometry,” *arXiv preprint arXiv:2310.04874*, 2023.

[7] 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.

[8] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics Research*, 2024.

[9] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, “Stochastic trajectory prediction via motion indeterminacy diffusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 113–17 122.

[10] R. H. A. Brajdi, “Walk detection and step counting on unconstrained smartphones,” in *ACM international joint conference on Pervasive and ubiquitous computing*, 2011.

[11] E. Foxlin, “Pedestrian tracking with shoe-mounted inertial sensors,” *IEEE Computer graphics and applications*, vol. 25, no. 6, pp. 38–46, 2005.

[12] A. Solin, S. Cortes, E. Rahtu, and J. Kannala, “Inertial Odometry on Handheld Smartphones,” in *2018 21st International Conference on Information Fusion (FUSION)*. Cambridge, United Kingdom: IEEE, July 2018, pp. 1–5.

[13] H. Yan, Q. Shan, and Y. Furukawa, “Ridi: Robust imu double integration,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 621–636.

[14] C. Chen, X. Lu, A. Markham, and N. Trigoni, “Ionet: Learning to cure the curse of drift in inertial odometry,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[15] S. Sun, D. Melamed, and K. Kitani, “Idol: Inertial deep orientation-estimation and localization,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, 2021, pp. 6128–6137.

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

[17] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. R. Kumar, and J. J. Engel, “Tlio: Tight learned inertial odometry,” *IEEE Robotics and Automation Letters*, vol. 5, pp. 5653–5660, 2020.

[18] S. Herath, D. Caruso, C. Liu, Y. Chen, and Y. Furukawa, “Neural inertial localization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6604–6613.

[19] G. Cioffi, L. Bauersfeld, E. Kaufmann, and D. Scaramuzza, “Learned inertial odometry for autonomous drone racing,” *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2684–2691, 2023.

[20] X. Cao, C. Zhou, D. Zeng, and Y. Wang, “Rio: Rotation-equivariance supervised learning of robust inertial odometry,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6614–6623.

[21] R. K. Jayanth, Y. Xu, Z. Wang, E. Chatzipantazis, D. Gehrig, and K. Daniilidis, “Eqnio: Subequivariant neural inertial odometry,” *arXiv preprint arXiv:2408.06321*, 2024.

[22] N. Trawny and S. I. Roumeliotis, “Indirect kalman filter for 3d attitude estimation,” *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.

[23] S. Brossard, Martin ands 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, 2020.

[24] G. Huang, “Visual-inertial navigation: A concise review,” in *Proc. International Conference on Robotics and Automation*, Montreal, Canada, May 2019.

[25] Y. Yang and G. Huang, “Observability analysis of aided ins with heterogeneous features of points, lines and planes,” *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 399–1418, Dec. 2019.

[26] T. Witt, “Low-frequency spectral analysis of dc nanovoltmeters and voltage reference standards,” *IEEE Transactions on Instrumentation and Measurement*, vol. 46, no. 2, pp. 318–321, 1997.

[27] D. W. Allan, “Should the classical variance be used as a basic measure in standards metrology?” *IEEE Transactions on Instrumentation and Measurement*, vol. IM-36, no. 2, pp. 646–654, 1987.

[28] “Ieee standard specification format guide and test procedure for coriolis vibratory gyros,” *IEEE Std 1431-2004*, pp. 1–78, 2004.

[29] “Ieee standard specification format guide and test procedure for linear single-axis, nongyroscopic accelerometers,” *IEEE Std 1293-2018 (Revision of IEEE Std 1293-1998)*, pp. 1–271, 2019.

[30] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[31] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv:1803.01271*, 2018.

[32] A. Vaswani, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.

[33] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[34] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[36] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” *arXiv preprint arXiv:2010.02502*, 2020.

[37] A. Martinelli, “Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.

[38] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual–inertial odometry using nonlinear optimization,” *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[39] 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, 2018.

[40] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, “The newer college dataset: Handheld lidar, inertial and vision with ground truth,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4353–4360.

[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*, 2016.