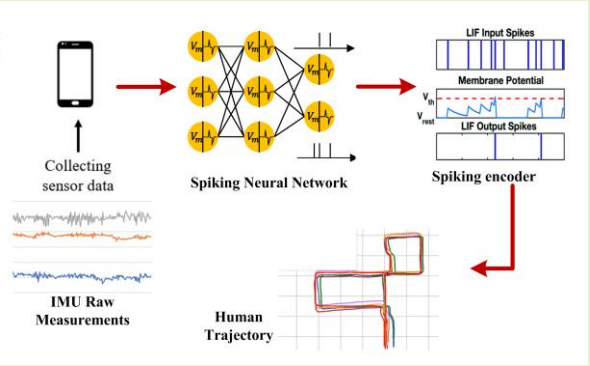
SpikePR: Position Regression with Deep Spiking Neural Network

Zhao Huang, Yifeng Zeng, Stefan Poslad, and Fuqiang Gu*

***Abstract*—Data-driven human localization technology has been on the rise with advancements in end-to-end Artificial Neural Networks (ANNs) in recent years. Different from the traditional Pedestrian Dead Reckoning (PDR) algorithms, the data-driven method can significantly reduce cumulative error over time arising from integration and improve the accuracy and efficiency of localization. However, the computation complexity of ANNs imposes high requirements on hardware conditions and heavily hinders its application on mobile devices. Targeting the above challenges, we design a Position Regression algorithm with a Deep Spiking Neural Network (called SpikePR)-an architecture inspired by biological neurons-to regress the user's position when collecting a sequence of raw IMU measurements from mobile devices. This architecture integrates ANNs and the Spiking Neural Network (SNN) with a Leaky Integrate-and-Fire (LIF) mechanism due to its low-power computation with binary spikes and capability to model the temporal dynamics in time series data. We conduct extensive experiments on four open-source datasets with the proposed SpikePR algorithm. The experiment results demonstrate that compared to the state-of-the-art driven-based position regression algorithms, the proposed SpikePR can save more than 90% energy consumption while achieving similar location errors.**



***Index Terms*—Human Localization Technology, Artificial Neural Networks, Deep Spiking Neural Network, IMU**

I. Introduction

With the development of smart wearable and mobile devices, the Inertial Measurement Unit (IMU) installed in these smart devices has become increasingly popular for various applications, including activity recognition [1], health monitoring [2], and human tracking [3], etc. IMU-based positioning has become an important research area in recent years. IMUs typically consist of accelerometers, gyroscopes, and magnetometers. These sensors are utilized to collect data on various aspects of motion. This data is then consolidated using sensor fusion algorithms, facilitating estimation of the user's position and orientation in 3 Dimensional (3D) space without needing an external infrastructure or communication with other devices. IMU-based localization methods [4-5] are not only capable of operating continuously for a long time but also readily available in each smartphone, which makes them easily accessible.

Existing IMU-based localization approaches have been demonstrated with great properties (e.g., low cost, high accuracy). For example, in [6], IMUs are affixed to the user's foot to improve localization accuracy, where the Zero Velocity Update (ZUPT) is used to revise the localization error according to the principle that the user's speed becomes 0 each time when a foot makes contact with the ground. A fuzzy logic-based energy-adaptive localization scheme is introduced by [7], to revise the drift error of the heading estimates of Pedestrian Dead Reckoning (PDR) using WiFi finger localization and achieve a trade-off between localization accuracy and energy consumption. A lightweight and highly precise localization system is constructed by [8], which fuses the magnetic positioning and PDR methods with the extended Kalman filter to improve localization accuracy. In [9], the pedestrian's position in an indoor environment is accurately estimated by combining PDR with an online sequential extreme learning machine. The adaptive feedback extended Kalman filter (AFEKF) is designed in [10], where the position, the orientation, and the range measurement from BLE are used to correct position estimation rather than only range measurements.

Even though these localization algorithms achieve better localization accuracy and efficiency, they still suffer from shortcomings such as cumulative errors and drift due to the integration computation. Therefore, data-driven-based localization approaches are proposed, they use end-to-end Artificial Neural Networks (ANNs) to predict human positions and mapping evaluation [11-12]. In [13], a novel deep learning

This work was supported by Youth Science Fund Project (Grant: 42301520) (Corresponding author: Fuqiang Gu)

Zhao Huang and Yifeng Zeng are with the Computer and Information Science, Northumbria University, Newcastle, Great London, U.K, NE1 8ST. (e-mail: zhao.huang@northumbria.ac.uk; yifeng.zeng@northumbria.ac.uk).

Stefan Poslad is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, Great London, U.K, E1 4NS. (e-mail: stefan.poslad@qmul.ac.uk).

Fuqiang Gu is with the College of Computer Science of Chongqing University, Chongqing, China, 400044. (e-mail: gufq@cqu.edu.cn).

architecture based on graph convolutional networks (GCNs) leverages the inertial sensors data to accurately forecast a sequence of human lower limb joint position. In [14], six IMUs and head-mounted sensor data are used to estimate human position using a hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network. In [15], a single hand-held camera except for IMUs attached to the body limb is utilized to regress precise 3D poses in unconstrained environments. All the data-driven-based approaches are efficient in regressing the user's position when inputting the IMU data into the end-to-end ANNs. However, they show huge computation complexity and energy consumption to hardware devices, which indicates that these algorithms are not easy to transform to edge devices due to the limited computing resources and battery life on these devices.

To overcome the above challenges, we first developed an edge-device-friendly Deep Spiking Neural Network (SNN) to process and fuse the raw IMU measurement data, then, regress human position (called SpikePR). Due to the binary spike (either 0 or 1) construction, SNNs [16-20] enable highly sparse computation, significantly reducing energy consumption and facilitating implementation on edge devices. Simultaneously, the membrane potential of spiking neurons changes over time. If the membrane potential exceeds a pre-defined threshold, the neuron fires a spike in the current time step. This mechanism is well-suited for modeling time series data, showing great potential for human position estimation. Overall, this work has four key contributions:

- To the best of our knowledge, we first introduce SNN to regress people's position according to the raw IMU measurements from mobile or wearable devices, this method takes advantage of binary spikes (either 0 or 1) and benefits from highly sparse, and multiplication-free computation, which greatly reduces the energy consumption required to process time-series data. It can be widely used in smart homes and extend the lifespan of batteries.
- Different from traditional ANN-based position regression algorithms, the proposed lightweight SpikePR inherently models the temporal dynamics in time series data by setting a threshold of the membrane potential, which will build the time relation between different timesteps and improve or keep the similar accuracy of position regression.
- Compared with vision-based and Wi-Fi-based methods, the proposed SpikePR model regresses the user's position by only utilizing IMU measurements, which has no privacy-invasion problems people are concerned about. Additionally, IMU sensors often are embedded in many mobile devices, which are easy to carry for users.
- Comprehensive qualitative and quantitative comparisons with the existing baselines on four benchmarks demonstrate the sparsity and energy consumption of our proposed method outperforms the existing state-of-the-art methods, our model achieves better, or similar performance while saving more than 90% energy consumption and increases more than 2x sparsity.

The rest of the paper is organized as follows. Section II briefly reviews related work in pedestrian localization. Subsequently, we present the technical details of the proposed method and a summary of baseline models in Section III. The experimental setting and results are shown in Section IV. Finally, Section V concludes this article.

II. RELATED WORK

A. Dead Reckoning

Dead reckoning plays a key role in localization for indoor pedestrians. Judd et al. [21] designed a drift-free localization algorithm by installing a compass on their foot to provide their heading and a pedometer to count steps automatically, but there exist huge measurement errors over time. To reduce the error, Randell et al. [22] proposed fusing the pedometers and accelerometers to improve step counting accuracy, and rate gyroscopes are utilized to revise people's heading errors from compasses, but the cumulative errors and drift still exist. Thus, an Ultra-wideband (UWB) and PDR fusion model based on a robust adaptive square root cubature Kalman filter (RA-SRCKF) is developed by Li et al. [23], where the UWB can reduce the cumulative error and improve localization accuracy, but many outliers under non-line-of-sight (NLOS) conditions are produced. Guo et al. [24] propose an improved UWB and PDR fusion scheme, where a variable noise variance Kalman filter combines gait and step information to overcome outliers problems, but it has jumping phenomena and is unstable due to the effect of moving objects. Li et al. [25] developed a Wi-Fi and PDR fusion system, in this system, the computational complexity of Wi-Fi fingerprint matching is significantly reduced by the affinity propagation clustering algorithm. To further improve localization accuracy, Chen et al. [26] introduce landmarks to Wi-Fi and PDR fusion schemes, where the landmark is recognized by human's human-specific sensor patterns, but someplace wifi is unavailable. Therefore, a stable PDR system is shown by Ma et al. [27], where the adaptive zero-velocity detection model is utilized to count steps by detecting the zero-velocity point, the limitation is that sensors have to attach on foot, which is inconvenient for users.

Recently, Artificial Neural Networks based Dead Reckoning has been proposed to estimate step length, orientation, speed or landmarks, etc. so as to reduce cumulative errors and drift. Cortés et al. [28] designed a strap-down inertial navigation model based on speed estimation using a deep CNN, which greatly improves navigation accuracy, but different devices are not tested for the generalization. Luo et al. [29] developed a Pedestrian Dead Reckoning system based on motion pattern recognition using the combined model of trained CNN and LSTM, however, the time resolution of the motion pattern information and computation complexity are the challenges. Asraf et al. [30] introduced a PDRNet, where ResNet is used to map from raw IMU measurements to the change in distance and heading, but two models have to be trained to estimate heading and distance respectively rather than together. Kawaguchi et al. [31] developed an end-to-end walking speed estimation method using Dual CNN-LSTM for smartphone PDR systems, this

algorithm achieves high localization accuracy, and similar to the above approaches, the computation complexity problem is still to be solved. Gu et al. [32] adopt a step-length estimation model based on Stacked Autoencoders to help localize pedestrians, this method has no spatial constraints and outperforms the existing approaches.

B. Position Regression

Position regression algorithms, which help avoid cumulative errors unlike Dead Reckoning, have been proposed by various researchers. Cardarelli et al. [33] introduced an innovative method to accurately estimate a person's position using an IMU placed on the pelvis. This method employs the meta-heuristically optimized Weighted Fused Least-squares (WFLC), demonstrating high localization accuracy and repeatability across different trials. However, it is susceptible to external disturbances, such as sudden movements. Brossard et al. [34] developed an Open-Loop Attitude Estimation algorithm that removes gyroscope noise through a learning method and adjusts the gyroscope readings to enhance estimation accuracy. Despite this, using only gyroscopes has proven sufficient for achieving reliable localization accuracy. Herath et al. [35] estimated people's positions and orientations using Artificial Neural Networks (ANNs) to extract features from raw IMU data collected during movement. This method resulted in lower localization errors, and the dataset has been made public. Zeinali et al. [36] devised the IMUNet framework, a technique that uses raw IMU readings from edge devices to estimate people's orientations. This method has notably reduced localization errors. Additionally, their dataset is publicly accessible. On the other hand, Herath et al. [37] proposed a concept known as Neural Inertial Localization (NILoc), which focuses on indoor localization. Unlike the previously mentioned methods, this approach uses a transformer-based neural architecture to estimate a user's location from a sequence of velocities, instead of IMU measurements. Zhang et al. [38] employ deep neural networks (DNNs) to regress the observable IMU motion terms. Instead of utilizing IMU measurements to predict relative IMU position and orientation, they process IMU data with a unique network design, loss functions, and training strategies. However, these algorithms encounter a common issue when implemented on mobile devices, which is energy consumption.

III. METHOD

In this section, we will introduce the SpikePR in detail, including the framework of this system, the structure of the Spiking Neural Network, and the integrated network.

A. The framework of the proposed method

The framework of the SpikePR system is shown in Fig. 1, which presents the process of human position regression. Firstly, the raw IMU measurements with timestep from the smartphone are extracted and preprocessed. Then, the processed samples with a fixed batch size are fed into the input layer. Following this, the SpikeRP model is adopted to transfer these data into 0-1 binary format and the new data is used to train different human regression models. Finally, human location is predicted according to the previous IMU measurements.

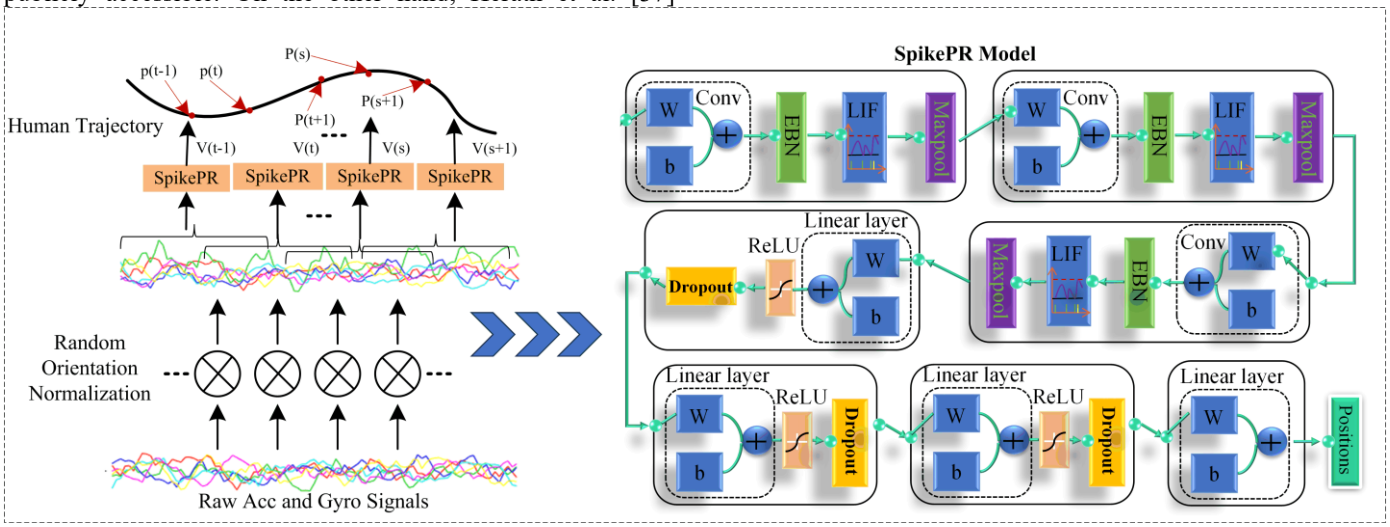


Fig. 1. SpikePR system diagram. From IMU measurements to estimate position. Given inertial sensor data (left), our approach (right) uses a spiking Neural network with LIF as an activation function to train a position regression model and then, regress the human's position according to the IMU measurements.

B. Spiking Neural Network

In recent decades, there has been a significant interest in developing third-generation neural networks called Spiking Neural Networks (SNN) that are biologically inspired by the communication scheme used by neurons in the brain [39]. SNNs offer several advantages over their predecessors, including the ability to process temporal data natively, increased computational efficiency, and a high degree of biological realism. The neurons in SNNs communicate by means of spikes, which are discrete events that take place at points in time. This allows SNNs to process information in a way that is much closer to how real neurons work. A spike is produced when the membrane potential of a neuron reaches a

threshold, and the temporal pattern of spikes carries information about external stimuli and ongoing computations [39].

There are several types of neuron models used in SNNs, The Leaky Integrate-and-Fire (LIF) is one of the most commonly used neuron models, which constantly receives inputs and outputs spikes over time, is used in our spiking neurons [41], as shown in Fig. 2. Formally, the LIF neuron maintains the membrane potential v over time, and assuming that the membrane potential obtains the pre-synaptic input charge c(t) at the t-*th* time step ($1 \le t \le T$), given by:

$$p^{(t+1),pre} = \tau p^{(t)} + i^{(t)} \quad (1)$$

$$i^{(t)} = Ws^{(t)} \quad (2)$$

This equation controls the decay factor of the membrane potential over time and determines the correlation between time steps. if τ=0, which indicates that the LIF neuron degenerates to binary activation lacking temporal dynamics. if τ=1, the correlation will reach the highest, the LIF neuron will transform into a ReLU neuron with sufficient T. The product of the weights W and the spike s(t+1) from the last layer generates the pre-synaptic input charge c(t+1). After receiving the input charge If the membrane potential exceeds a certain threshold, the LIF neuron will fire a spike [42].

$$s^{(t+1)} = \begin{cases} 1, p^{(t+1),pre} > U_T \\ 0, otherwise \end{cases} \quad (3)$$

In (3), U_T is the LIF firing threshold, and the spike will propagate to the next layer, where we omit the layer index for simplicity.

The membrane potential will be adjusted through either a soft reset or a hard reset if the spike is fired by the LIF neurons, indicated by:

$$p^{(t+1)} = p^{(t+1),pre} \cdot (1 - s^{(t+1)}) \quad \text{\#Hard-Reset} \quad (4)$$

$$p^{(t+1)} = p^{(t+1),pre} - s^{(t+1)} \cdot U_T \quad \text{\#Soft-Reset} \quad (5)$$

where hard-reset sets to 0, while soft-reset subtracts by U_T. The LIF neurons are selected because $s^{(t+1)}$ are binary and dependent on input in previous time steps.

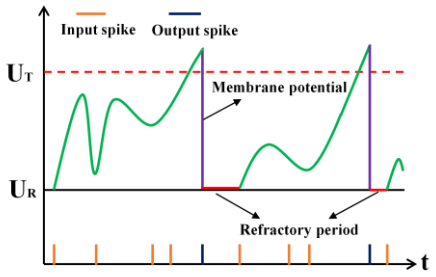


Fig. 2. The dynamics of LIF spiking neurons. Orange and blue colors indicate the input spike and output spike respectively. When the membrane potential surpasses the threshold U_T, it will fire a spike, and then, the membrane goes back to U_R.

C. Baseline Methods

In order to make a comparison between previous algorithms and the proposed algorithm, in this section, three baseline methods and the proposed SpikePR will be presented in detail, including ResNet [43], MobileNetV2 [40], and IMUNet [36].

1) ResNet: Introduced in 2016 by He et al. [43], the aim is to address the problem of vanishing gradients in deep networks and improve the efficiency of network training, where the use of skip connections is used to enable the network to learn residual functions and makes it easier for the network to learn useful features, even in deeper architectures. The ResNet18 architecture consists of 18 layers, including a convolutional layer, a max pooling layer, and four groups of residual blocks. Each residual block contains two convolutional layers and a shortcut connection that adds the input to the output of the second convolutional layer. The convolutional layers have filter sizes of 3x3 or 1x1, and the shortcut connections either perform identity mappings or use 1x1 convolutional layers to match the dimensions of the input and output. The final layer is a fully-connected layer that maps the output of the last residual block to the number of classes in the dataset.

2) MobileNetV2: introduced in 2018 by Mark et al. [40] and designed for mobile and embedded devices that have limited computational resources, which is an improved version of MobileNet [44]. MobileNetV2 consists of Depth-wise Separable Convolution, Inverted Residual Block, Linear Bottleneck Block, Global Average Pooling, and Fully-Connected Layer. The Depth-wise Separable Convolution technique is utilized in MobileNetV2 to lower the computational cost of traditional convolutions by decomposing them into a depth-wise convolution followed by a point-wise convolution. The Inverted Residual Blocks implement a shortcut connection between layers to enhance the gradient flow and mitigate vanishing gradient issues, thereby improving the overall stability and convergence of the model. The Linear Bottleneck Blocks are incorporated into the architecture to further reduce the computational overhead while retaining the model's expressive capacity.

3) IMUNet: designed by Zeinali et al. [36], which is a novel architecture that has been specifically designed for edge device implementation, utilizing IMU measurements to perform efficient and precise inertial navigation. One of the key features of IMUNet is its ability to operate with significantly reduced energy consumption while maintaining high accuracy in position and orientation estimation. Meanwhile, IMU can capture and analyze the temporal and spatial dependencies in the IMU data, allowing it to accurately predict the device's position and orientation over time. Moreover, IMUNet has been trained on large datasets of IMU measurements and ground truth data and optimized using a regression loss function. This ensures that the architecture is robust and capable of delivering accurate and efficient results even in challenging real-world scenarios.

4) MnasNet [45]: Designing a CNN model manually for specific purposes can be challenging due to the difficulty in balancing trade-offs between conflicting goals, especially when numerous neural architectures are possible. To tackle this

problem, Mnasnet was developed using Neural Architecture Search, unlike manually designed Mobilenet and MobilenetV2. The authors presented a method to balance the model's accuracy and inference latency by running the inference on a real device. They introduced a factorized hierarchical search space, which allows for diversity across different networks.

D. The proposed SpikePR

To save energy consumption and enhance position regression capability, spiking neurons with LIF are integrated into the deep neural networks as the activation functions with LIF rather than utilizing the non-linear activation function, such as ReLU. Specifically, IMU measurements have a time dimension, so the pre-synaptic potential charge has been integrated along this dimension. For example, if $\alpha \in \mathbb{R}^{n\times c\times T}$ is a pre-activation tensor, where indicates the parameters: batch size, channel number, and total time steps separately. We set the charge in each time step for LIF as the pre-activation in the corresponding time step. For instance, $c^{(t)} = a_{:,:,t.}$ Afterward, we stack the output spikes along the time dimension, in order to calculate the pre-activation in the next layer [41].

Although LIF neurons are able to model temporal features and generate binary spikes, the firing function described in equation (2) is discrete, resulting in gradients that are almost zero everywhere. This, in turn, makes it difficult to optimize weights using gradient-based methods. Specifically, the gradient of the loss function (denoted by L) concerning the weights can be computed using the chain rule:

$$\frac{\partial L}{\partial W} = \sum_{t=1}^{T} \frac{\partial L}{\partial s^{(t)}} \frac{\partial s^{(t)}}{\partial p^{(t),pre}} \cdot \left(\frac{\partial p^{(t),pre}}{\partial c^{(t)}} \frac{\partial c^{(t)}}{\partial W} + H \right) \tag{6}$$

$$H = \sum_{t'=1}^{t-1} \frac{\partial p^{(t),pre}}{\partial p^{(t)}} \frac{\partial p^{(t)}}{\partial p^{(t'),pre}} \frac{\partial p^{(t'),pre}}{\partial c^{(t')}} \frac{\partial c^{(t')}}{\partial W} \tag{7}$$

All other terms in the equation above can be differentiated, with the exception of $\frac{\partial s^{(t)}}{\partial v^{(t),pre}}$, which results in zero-but-all gradients. To address this issue, we utilize the surrogate gradient method [46]. Specifically, we use the triangle surrogate gradient, which is given by:

$$\frac{\partial s^{(t)}}{\partial p^{(t),pre}} = \max\left(0, 1 - \left| \frac{p^{(t),pre}}{V_{th}} - 1 \right| \right) \tag{8}$$

Therefore, the SNNs can be improved with stochastic gradient descent algorithms.

When the SNN with LIF is chosen as the activation function, we also design the structure of the proposed algorithm, which contains the feature extraction block and the fully-connected. This block consists of three small convolutional blocks, one small convolutional block includes one convolution layer, one Event-based Batch Normalization (EBN) layer, one LIF layer, and a Maxpool layer. This convolution layer is used to extract features for the training stage. In order to mitigate the issues of vanishing and exploding gradients in our proposed method, we have incorporated the EBN method. This approach normalizes the pre-synaptic inputs across the channel dimension, the inputs of each neuron are adjusted into the interval ranging from 0 to U_T , where U_T represents pre-defined the spiking firing threshold, and the EBN is denoted as the below equations.

$$F'^{l}_{i} = \frac{U_T (F_i^l - E[F_i^l])}{\sqrt{Var[F_i^l] + \Psi}} \tag{9}$$

$$Y_i^l = \gamma_i F'^{l}_{i} + \lambda_i \tag{10}$$

Where F'^{l}_{i} denotes the i-th channel feature map of F_i^l . Here, $E[F_i^l]$ and $Var[F_i^l]$ represent the expectation and variance of F_i^l over the mini-batch, respectively. The Ψ is a small constant introduced to prevent division by zero errors. Additionally, γ_i and λ_i are two parameters that can be learned. During the inference stage, we are unable to directly compute the expectation and variance of F_i^l as the batch is not applicable. As a result, we estimate the expectation $E[F_i^l]$ and variance $Var[F_i^l]$ over the entire dataset. This estimation can be achieved using a moving average solution during the training stage. Following that, the LIF activation function is adopted to transfer the features into binary format. Finally, the Maxpool layer is used to sparsity the features. In the fully-connected layer, we have three cells, every cell includes one liner layer, one ReLU activation function, and one Dropout layer. The spiking features are fed into the three cells, and after that, one linear layer is employed to decode the network output.

IV. EXPERIMENTS AND RESULTS

To validate the effectiveness and efficiency of the proposed algorithm, quite a few experiments have been implemented. We evaluate the model on three popular IMU datasets. We applied all four datasets to the data-driven method with the mentioned state-of-the-art deep learning models as well as our proposed SpikePR architecture.

A. Implementation Details and Datasets

(1) Hardware**:** we implement the proposed SpikePR architecture using Pytorch and train all models on the GPU: Nvidia GeForce GTX 3060 with 12 GB memory and utilizing the CPU: 12th Gen Intel(R) Core (TM) i7-12700@2.10 GHz for testing the trained models.

(2) Parameters: we obtain one training/validation sample, composed of 200 frames, at every 10-frame interval. For each epoch, training samples are shuffled randomly. Employing a batch size of 128, an initial learning rate of 0.0001, and the ADAM optimizer, we adjust the learning rate by reducing it to 0.3 of its value if there is no decrease in validation loss within 5 epochs. The decay factor and firing threshold are 0.5 and 0.5 respectively.

(3) Evaluation Metrics: Two metrics used in [47] are adopted to measure the performance of the baseline and proposed methods: Absolute Trajectory Error (ATE) and Relative Trajectory Error (RTE).

- Absolute Trajectory Error (ATE): which represents the global accuracy of the estimated trajectory. It is calculated as the

average root mean squared error (RMSE) between the estimated and ground-truth trajectories globally. The calculation equation of ATE is shown below:

$$ATE = \sqrt{\frac{\sum_{t=1}^{N}\left\|(x_t - x_t')\right\|^2}{N}} \quad (11)$$

• Relative Trajectory Error (RTE): which is the RMSE of average errors over a time-interval window span t_i (t_i =60 s in our case). It measures the local consistency of the estimated and ground truth path. The estimation error is defined formally as Equation (12).

$$RTE = \sqrt{\frac{\sum_{t=1}^{N}\left\|(x_{t+t_i} - x_t) - (x'_{t+t_i} - x'_t)\right\|^2}{N}} \quad (12)$$

(4) Datasets: all selected datasets (including RONIN, OXIOD dataset [48], IMUNet, and RIDI dataset [49],) with rich motion contexts (e.g., handheld, pocket) have been collected by multiple subjects as shown in Table I. RIDI and IMUNet datasets are collected by Lenovo Phab2 Pro while the OXIOD dataset and RoNIN dataset are collected by iPhone 5/6 Plus, Nexus 5 and Galaxy S9, Pixel 2 XL respectively. Meanwhile, the ground truth of these datasets is generated by Google Tango phone, Vicon, Asus Zenfone AR and ARCore API separately.

TABLE I
THE DETAILED INFORMATION OF THE PUBLIC DATASETS USED IN OUR EXPERIMENTS

Dataset	Year	IMU Carrier	Sample Frequency	Number of Subjects	Number of Sequences	Ground Truth	Device placements
RIDI	2017	Lenovo Phab2 Pro	200Hz	10	98	Google Tango phone	Hand, pocket, bag, body
OXIOD	2018	iPhone 5/6 Plus, Nexus 5	100Hz	5	158	Vicon	Hand, pocket, bag, trolley
RoNIN	2019	Galaxy S9, Pixel 2 XL	200Hz	100	276	Asus Zenfone AR	unrestricted
IMUNet	2022	Lenovo Phab2 Pro	60Hz	/	51	ARCore API	unrestricted

B. Evaluation Results

In order to evaluate the efficacy of our proposed algorithms, we have performed several experiments. These experiments utilize four existing algorithms as well as the newly proposed SpikePR algorithm. The performance evaluation includes two parts: Localization Error Evaluation and energy consumption.

1) Localization Error Evaluation

Experiments were conducted across four separate datasets, enabling a comprehensive analysis of both Absolute Trajectory Error (ATE) and Relative Trajectory Error (RTE). It's worth noting that the RIDI and RoNIN datasets consist of both seen and unseen parts, with the unseen parts representing the subjects not included in the training stage. In contrast, the OXOID and IMUNet datasets are comprised exclusively of seen parts. We have presented several qualitative result visualizations implemented by the proposed SpikePR and the comparative methods in Fig. 3.

1) Results on the RIDI dataset: We separated the seen and unseen parts of the dataset to conduct our experiments. The experimental results are shown in Fig. 3(a) and (b), where the two figures represent the predicted trajectories from the five algorithms and the ground truth (represented by the red curve). These are chosen from handheld devices (handheld pattern) and devices attached to the front leg (leg_front pattern). Fig. 3(a) is from the seen dataset, where our method exhibits the lowest ATE (0.81m) and RTE (1.34m). RoNIN-Res and IMUNet demonstrate the next best ATE values (0.84m and 0.86m respectively), while MobileNetV2 presents the next best RTE (1.37m). Conversely, MnasNet has the highest ATE (0.9m) and RTE (1.45m). In Fig. 3(b), our proposed algorithm yields the smallest ATE and RTE, which are 0.46m and 0.79m respectively, while MobileNetV2 shows the highest ATE and RTE (1.14m and 1.38m).

2) Results on the IMUNet dataset: The IMUNet dataset was utilized to test the trained model. The ground truth and predicted trajectory for the behnam_8 and behnam_11 patterns are depicted in Fig. (c), represented by six color-coded curves. MobileNetV2 registers the lowest RTE (1.18m), while the proposed SpikePR achieves the lowest ATE (0.95m). Conversely, RONIN-Res performs less effectively than the others, with ATE and RTE values of 2.06m and 2.03m, respectively.

3) Results on the RONIN dataset: The entire test dataset (including seen and unseen parts) was utilized to test the trained model. Sample trajectories are depicted in Fig. 3 (d) and (e), where we utilize six color-coded curves to present the corresponding trajectories. As can be observed from Fig. 3 (d), the proposed method slightly reduces the ATE, from 2.44m (IMUNet) to 2.16m, and the RTE also decreases from 2.45m to 2.27m. In Fig. 3(e), our proposed algorithm achieves the lowest ATE of 2.32m, compared to IMUNet's 2.50m. Meanwhile, RONIN-Res records the smallest RTE (2.45 m) while MnasNet registers the highest RTE at 3.84 m.

4) Results on the OXIOD dataset: To evaluate the performance of the proposed algorithms on the OXIOD dataset, we extracted only the seen part of this dataset to train a position regression model. The result is depicted in Fig. 3 (f), where we utilize six color-coded curves to represent the corresponding trajectories with these algorithms. Our proposed algorithm and the MobileNetV2 algorithm record the lowest ATE of 0.6 meters, while the IMUNet algorithm registers the highest ATE, which is 0.74 meters. In regard to RTE, the IMUNet algorithm achieves the smallest value of 0.49m, while our method registers a slightly higher value of 0.52m.

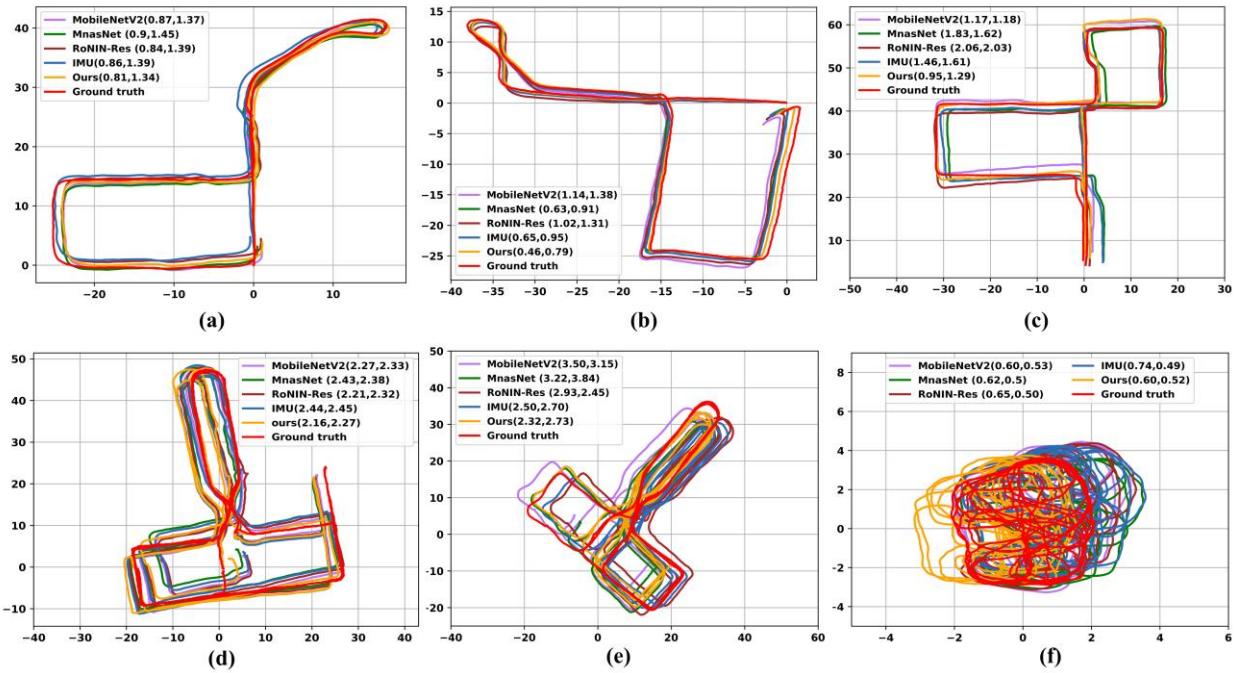


Fig. 3 depicts the prediction trajectories of the proposed algorithm (SpikePR) and four baseline methods on the four datasets. Each color curve corresponds to the predicted result of a specific algorithm, with the red curve representing the ground truth. The numbers associated with each algorithm indicate the ATE and RTE of the respective position predictions. (a) and (b) illustrate the trajectories of the seen and unseen parts on the RIDI dataset. (c) represents the trajectory of the seen part on the IMUNet dataset, while (d) and (e) showcase trajectories from the RONIN dataset. (f) signifies the trajectory of the OXIOD dataset.

As can be observed from Table II, the numbers denote the error scores across four datasets, respectively. For the RIDI dataset, the proposed SpikePR records the smallest ATE of 1.53m on the seen dataset, followed by MobileNetV2 and IMUNet with ATEs of 1.55m and 1.56m, respectively. IMUNet records the lowest RTE of 1.83m, while SpikePR and RoNIN-Res register RTEs of 1.93m and 1.91m, respectively. On the unseen dataset, MobileNetV2 achieves the smallest ATE of 1.28m, while SpikePR records an RTE of 1.3m. For the RoNIN dataset, our algorithm registers an ATE of 3.58m and an RTE of 2.67m for the seen data, which is slightly higher in ATE compared to IMUNet (3.52m) but lower than RoNIN-Res (3.63m), while the RTE is the smallest compared to the other two top values of 2.71m and 2.75m. For the unseen data, the overall performance is lower than the seen data, with the top three ATEs being 5.19m, 5.65m, and 5.68m, and the top three RTEs being 4.49m, 4.53m, and 4.54m. Furthermore, the IMUNet and OXIOD datasets were also evaluated. Our proposed algorithm records the smallest ATE (2.33m) and RTE (2.16m) on the OXIOD dataset and also achieves the smallest ATE (2.53m) and one of the three lowest RTEs (3.01m) on the IMUNet dataset. In summary, the proposed SpikePR demonstrates equal or superior performance compared to state-of-the-art baseline methods in terms of ATE and RTE performance.

TABLE II
THE COMPARISON OF ATE AND RTE USING THE FOUR ALGORITHMS AND THE PROPOSAL ALGORITHM

Methods	Year/	Metric	Datasets					
			RIDI		RoNIN		OXIOD	IMUNet
			seen	unseen	seen	unseen	seen	seen
MobileNetV2	CVPR 2018	ATE	1.55	1.28	3.83	6.17	3.38	3.03
		RTE	1.97	1.78	2.85	4.69	2.89	3.55
MnasNet	CVPR 2019	ATE	1.71	1.46	3.78	5.19	3.08	2.75
		RTE	2.10	1.94	2.75	4.54	2.64	3.19
RoNIN-Res	ICRA 2020	ATE	1.63	1.67	3.63	5.65	3.14	2.73
		RTE	1.91	1.62	2.76	4.57	2.66	3.03
IMUNet	TMC 2022	ATE	1.56	1.33	3.52	5.68	2.88	2.59
		RTE	1.83	1.83	2.71	4.49	2.58	2.97
Ours (SpikePR)		ATE	1.53	1.35	3.58	5.72	2.33	2.53
		RTE	1.93	1.88	2.67	4.53	2.16	3.01

2) Energy Consumption

To compare the energy consumption, our focus is on two key metrics: activation sparsity and actual energy consumption.

Greater sparsity can sidestep additional computations with weights in hardware designed to handle sparse computation. We assess sparsity either in SpikePR with Rectified Linear Unit (ReLU) as the activation function, denoted as SpikePR ANN-based, or in SpikePR with Leaky Integrate-and-Fire (LIF) as the activation function. These are visualized in Fig. 4 (a). Commonly, ReLU in an Artificial Neural Network (ANN) exhibits around 20% sparsity, an intuitive result given that the mean of activation is typically around zero. LIF neurons, on the other hand, demonstrate over twice the sparsity, likely due to the firing threshold exceeding zero. Thus, SpikePR holds greater potential for reducing operations during inference. The second metric we consider is energy consumption. We estimate this by evaluating the proposed SpikePR model alongside four ReLU-based ANN baselines using the energy simulator proposed in [50]. Specifically, we estimate the energy reduction ratio on the hardware accelerator, with results displayed in Fig. 4 (b). The SNN-based SpikePR shows up to 90% less energy consumption compared to its ANN-based counterpart, potentially significantly extending battery life in smart devices. In conclusion, SpikePR, through the use of LIF neurons, achieves superior task performance and, thanks to binary representation with high sparsity, increased energy efficiency.

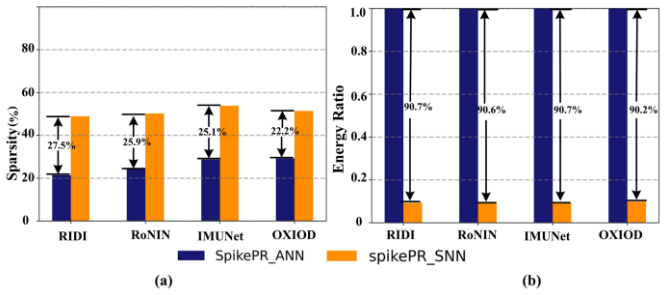


Fig. 4: Hardware performance comparison of SpikePR ANN-based and SpikePR SNN-based (SpikePR), the latter being our proposed model. (a) illustrates the disparity in sparsity between the two algorithms. (b) depicts the relative energy consumption of both algorithms, with the energy usage of the ANN-based model normalized to 1 for comparison.

C. Ablation Studies

To element the three factors (Decay factor, Firing threshold and Reset mechanism) to our experiments, the ablation studies on seen part data of the four datasets are carried out.

1) *Decay Factor.*

We examine five set decay factors, specifically 0.0, 0.25, 0.5, 0.75, and 1.0. As previously discussed, a τ value of 0 implies no correlation between consecutive time steps, rendering SpikePR equivalent to Binary Activation Networks (BAN), while a τ value of 1 signifies total correlation. All results are presented in Table III.

It's evident that τ significantly influences the final test performance. For the RIDI dataset, the ATE and RTE for τ=0 are 1.72m and 2.19m, respectively, while these metrics for τ=0.5 are 1.53m and 1.93m, respectively. Additionally, when comparing cases where $0 < \tau < 1$ with τ=0, it's clear that τ=0 invariably results in a substantial deficit. This suggests that considering the temporal correlation with $\tau > 0$ is essential for time series tasks. Furthermore, for the RoNIN dataset, the errors for τ=1 are 3.76m and 2.77m for ATE and RTE, respectively, whereas the τ=0.75 case yields 3.58m and 2.76m. The OXIOD and IMUNet datasets demonstrate better results for τ=0.75 and τ=0.25, respectively, with the errors being 2.33m and 2.53m for ATE, and 2.16m and 3.01m for RTE.

TABLE III
ABLATION STUDY ON THE DECAY FACTOR T

Dataset	Metric	Decay Factor τ				
		0.0	0.25	0.5	0.75	1.0
RoNIN	ATE	3.68	3.72	3.79	**3.58**	3.76
	RTE	2.76	2.79	2.73	**2.67**	2.77
RIDI	ATE	1.72	1.69	**1.53**	1.67	1.73
	RTE	2.19	2.08	**1.93**	2.06	2.11
OXIOD	ATE	2.46	2.37	2.41	**2.33**	2.47
	RTE	2.23	2.32	2.29	**2.16**	2.31
IMUNet	ATE	2.68	**2.53**	2.67	2.76	2.73
	RTE	3.16	**3.01**	3.16	3.25	3.19

2) *Firing Threshold*

We explore the impact of the firing threshold. Typically, this threshold corresponds to how easily a neuron can fire a spike. For this study, thresholds are adjusted at 0.25, 0.5, 0.75, and 1.0, with experiments replicated from the earlier ablation study. Insights gleaned from Table IV suggest a consistent pattern regarding the firing threshold. Optimal performance for SpikePR is attained at a threshold of 0.5, a finding that aligns with expectations. This midpoint between 0 and 1 appears to minimize the error in the firing function. Additionally, it's notable that the variations in Average Total Error (ATE) and Relative Total Error (RTE) influenced by the firing threshold are less pronounced compared to those affected by the decay factor. For example, the most significant disparity in ATE and RTE for the IMUNet dataset, when altering the threshold (from 0.25 to 0.5), is from 2.60m to 2.53m and from 3.12m to 3.01m, respectively. In contrast, adjusting the decay factor (from 0.25 to 0.5) leads to shifts in ATE and RTE from 2.35m to 2.67m and from 3.01m to 3.16m, respectively. Consequently, SpikePR exhibits greater sensitivity to changes in the decay factor than to adjustments in the firing threshold.

TABLE IV
ABLATION STUDY ON THE FIRING THRESHOLD V

Dataset	Metric	Firing Threshold V			
		0.25	0.5	0.75	1.0
RoNIN	ATE	3.62	**3.58**	3.69	3.66
	RTE	2.77	**2.67**	2.72	2.74
RIDI	ATE	1.64	**1.53**	1.71	1.68
	RTE	2.03	**1.93**	2.01	2.08
OXIOD	ATE	2.44	**2.33**	2.38	2.46
	RTE	2.23	**2.16**	2.24	2.19
IMUNet	ATE	2.60	**2.53**	2.59	2.56
	RTE	3.08	**3.01**	3.07	3.06

3) *Reset Mechanism.*

We examine the reset mechanisms for SpikePR, specifically soft-reset and hard-reset. The outcomes of this study are also cataloged in Table IV. Across all scenarios, the soft-reset mechanism outperforms the hard-reset. We hypothesize that

this occurs because the hard reset directly reduces the membrane potential to 0, thus severing the correlation between two consecutive timesteps. Conversely, the soft reset maintains a portion of the information about the membrane potential post-firing.

TABLE V

ABLATION STUDY ON THE RESET MECHANISM

Reset	Metric	Dataset			
		RoNIN	OXIOD	RIDI	IMUNet
Hard	ATE	3.63	2.36	1.64	2.62
	RTE	2.72	2.22	2.01	3.11
Soft	ATE	**3.58**	**2.33**	**1.53**	**2.53**
	RTE	**2.67**	**2.16**	**1.93**	**3.01**

To evaluate the localization accuracy while using different numbers of fully-connected layers (with ReLU), we performed four group experiments, on the four datasets (seen part) where the fully-connected layers are 1, 2, 3, and 4 respectively, and the localization errors are shown in Table VI. As can be seen, when three and four fully-connected layers are used, the SpikePR achieves similar accuracy, but the four layers model will increase the number of parameters and computation complexity. Therefore, three layers are utilized.

TABLE VI

ABLATION STUDY ON FULLY-CONNECTED LAYERS

Layers	Metrics	Datasets			
		RIDI	RoNIN	OXIOD	IMUNet
1	ATE	8.66	11.2	9.84	9.23
	RTE	8.93	9.76	8.32	9.71
2	ATE	3.36	6.21	5.42	5.24
	RTE	3.88	4.96	4.46	5.49
3	ATE	1.53	3.58	2.33	2.53
	RTE	1.93	2.67	2.16	3.01
4	ATE	1.49	3.52	2.42	2.47
	RTE	2.06	2.73	2.09	2.98

V. CONCLUSION

To the best of our knowledge, we are the first to study the problem associated with Spiking Neural Networks and position regression. In this paper, we have designed the Position Regression with Spiking Neural Networks (SpikePR) to perform human position regression tasks. In contrast to Artificial Neural Networks (ANNs) based position regression, SpikePR not only makes use of their LIF neurons to generate spikes through time but also avoids cumulative errors from double integration. Meanwhile, this SpikePR shows greater energy efficiency and temporally correlated non-linearity. Extensive experiment results demonstrate that SpikePR SNN-based approaches can achieve smaller or similar ATE and RTE than the four baseline methods with more than 90% reduction of energy consumption and more than 2x activation sparsity.

Despite the significant advancements and conveniences attributed to Spiking Neural Networks (SNNs), they are not without their challenges. One key limitation is the difficulty associated with training an SNN model. The inherent complexity of SNNs and the high computational cost make this aspect particularly challenging. To elaborate, the spike-based, event-driven nature of SNNs necessitates specialized training algorithms and increased computing resources. As the complexity of a network increases, so does the difficulty of training the model. Traditional approaches often yield suboptimal results, and more advanced methods can become computationally expensive, potentially creating barriers to broader implementation. Furthermore, defining appropriate parameters and settings for training SNNs can be a nuanced process requiring careful consideration. To optimize network performance, variables such as the neuron model, learning rule, and network architecture need to be accurately set, adding to the difficulty of training these models.

References

[1] Huang, Z., Poslad, S., et al. “DeepWE: A Deep Bayesian Active Learning Waypoint Estimator for Indoor walkers”. IEEE Internet of Things Journal, 2023.

[2] Petropoulos, A., Sikeridis, D., et al. “Wearable smart health advisors: An IMU-enabled posture monitor”. IEEE Consumer Electronics Magazine, 9(5), 20-27, 2020.

[3] Gu, Fuqiang, et al. "Robust and accurate smartphone-based step counting for indoor localization." IEEE Sensors Journal 17.11 (2017): 3453-3460.

[4] Zhou, Baoding, et al. "ALIMC: Activity landmark-based indoor mapping via crowdsourcing." IEEE Transactions on Intelligent Transportation Systems 16.5 (2015): 2774-2785.

[5] Zhou, Baoding, et al. "A robust crowdsourcing-based indoor localization system." Sensors 17.4 (2017): 864.

[6] Jimenez, A. R, Seco F., et al. “A comparison of pedestrian dead-reckoning algorithms using a lowcost mems imu”. In 2009 IEEE International Symposium on Intelligent Signal Processing, pages 37–42. IEEE, 2009.

[7] Yang, Y., Huang, B., et al. “A Fuzzy Logic-Based Energy-Adaptive Localization Scheme by Fusing WiFi and PDR”. Wireless Communications and Mobile Computing, 2023.

[8] Sun, M., Wang, Y., et al. “Indoor localization using mind evolutionary algorithm-based geomagnetic positioning and smartphone IMU sensors”. IEEE Sensors Journal, 22(7), 7130-7141.

[9] Zhang, M., Wen, Y., et al. “Pedestrian dead-reckoning indoor localization based on OS-ELM”. IEEE Access, 6, 6116-6129, 2018.

[10] Kong, X., Wu, C., et al. “Hybrid Indoor Positioning Method of BLE and PDR Based on Adaptive Feedback EKF With Low BLE Deployment Density”. IEEE Transactions on Instrumentation and Measurement, 72, 1-12, 2022.

[11] Li, Jianping, et al. "HCTO: Optimality-aware LiDAR inertial odometry with hybrid continuous time optimization for compact wearable mapping system." ISPRS Journal of Photogrammetry and Remote Sensing 211 (2024): 228-243.

[12] Li, Jianping, et al. "WHU-Helmet: A helmet-based multi-sensor SLAM dataset for the evaluation of real-time 3D mapping in large-scale GNSS-denied environments." IEEE Transactions on Geoscience and Remote Sensing (2023).

[13] Chen, Y. L., Yang, I. J., et al. “IMU-based estimation of lower limb motion trajectory with graph convolution network. IEEE Sensors Journal, 21(21), 24549-24557, 2021.

[14] Kim, M., & Lee, S. “Fusion Poser: 3D Human Pose Estimation Using Sparse IMUs and Head Trackers in Real Time”. Sensors, 22(13), 4846, 2022.

[15] Von Marcard, T., Henschel, R., et al. “Recovering accurate 3d human pose in the wild using imus and a moving camera”. In Proceedings of the European conference on computer vision (ECCV) (pp. 601-617), 2018.

[16] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. Neural networks, 111:47–63, 2019.

[17] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. Nature, 575(7784):607–617, 2019.

[18] Youngeun Kim, Yeshwanth Venkatesha, and Priyadarshini Panda. Privatesnn: privacy-preserving spiking neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 1192–1200, 202.

[19] Youngeun Kim, Yuhang Li, Hyoungseob Park, Yeshwanth Venkatesha, Ruokai Yin, and Priyadarshini Panda. Exploring lottery ticket hypothesis in spiking neural networks. In European Conference on Computer Vision, pages 102–120. Springer, 2022.

[20] Li, Y., Deng, S., Dong, X., & Gu, S. "Converting artificial neural networks to spiking neural networks via parameter calibration." arXiv preprint arXiv:2205.10121 (2022).

[21] Judd, C. T. (1997, September). A personal dead reckoning module. In proceedings of the 10th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1997) (pp. 47-51).

[22] Randell, C., Djiallis, C., & Muller, H. (2003, October). Personal position measurement using dead reckoning. In Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings. (pp. 166-166). IEEE Computer Society.

[23] Li, Z., Zhang, L., & Tian, G. (2021, July). UWB/PDR integrated indoor positioning method based on robust adaptive square root cubature Kalman filter. In Journal of Physics: Conference Series (Vol. 1976, No. 1, p. 012021). IOP Publishing.

[24] Guo, S., Zhang, Y., et al. An improved PDR/UWB integrated system for indoor navigation applications. IEEE Sensors Journal, 20(14), 8046-8061, 2022.

[25] Li, X., Wang, J., et al. Integrated WiFi/PDR/smartphone using an adaptive system noise extended Kalman filter algorithm for indoor localization. ISPRS International Journal of Geo-Information, 5(2), 8.

[26] Chen, Z., Zou, H., et al. Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization. Sensors, 15(1), 715-732, 2015.

[27] Ma, Ming, et al. "An adaptive zero velocity detection algorithm based on multi-sensor fusion for a pedestrian navigation system." Sensors 18.10 (2018): 3261.

[28] S. Cortés, A. Solin and J. Kannala, "DEEP LEARNING BASED SPEED ESTIMATION FOR CONSTRAINING STRAPDOWN INERTIAL NAVIGATION ON SMARTPHONES," 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), Aalborg, Denmark, 2018, pp. 1-6, doi: 10.1109/MLSP.2018.8516710.

[29] Y. Luo, C. Guo, et al. "Learning-Based Complex Motion Patterns Recognition for Pedestrian Dead Reckoning," in IEEE Sensors Journal, vol. 21, no. 4, pp. 4280-4290, 15 Feb.15, 2021, doi: 10.1109/JSEN.2020.3029719.

[30] O. Asraf, F. Shama and I. Klein, "PDRNet: A Deep-Learning Pedestrian Dead Reckoning Framework," in IEEE Sensors Journal, vol. 22, no. 6, pp. 4932-4939, 15 March15, 2022, doi: 10.1109/JSEN.2021.3066840.

[31] Kawaguchi, N., Nozaki, J., et al. End-to-end walking speed estimation method for smartphone PDR using DualCNN-LSTM. In IPIN (Short Papers/Work-in-Progress Papers) (pp. 463-470).

[32] F. Gu, K. Khoshelham, et al. "Accurate Step Length Estimation for Pedestrian Dead Reckoning Localization Using Stacked Autoencoders," in IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 8, pp. 2705-2713, Aug. 2019, doi: 10.1109/TIM.2018.2871808.

[33] Cardarelli, S., Verdini, F., et al. Position estimation of an IMU placed on Pelvis through meta-heuristically optimised WFLC. In World Congress on Medical Physics and Biomedical Engineering 2018: June 3-8, 2018, Prague, Czech Republic (Vol. 2) (pp. 659-664). Springer Singapore.

[34] Brossard, Martin, Silvere Bonnabel, and Axel Barrau. "Denoising imu gyroscopes with deep learning for open-loop attitude estimation." IEEE Robotics and Automation Letters 5.3 (2020): 4796-4803.

[35] Herath, Sachini, Hang Yan, and Yasutaka Furukawa. "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods." 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020.

[36] Zeinali, Behnam, Hadi Zandizari, and J. Morris Chang. "Imunet: Efficient regression architecture for imu navigation and positioning." arXiv preprint arXiv:2208.00068 (2022).

[37] Herath, Sachini, et al. "Neural inertial localization." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[38] Zhang, Ming, et al. "IMU data processing for inertial aided navigation: A recurrent neural network based approach." 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021.

[39] Pfeiffer, Michael, and Thomas Pfeil. "Deep learning with spiking neurons: Opportunities and challenges." Frontiers in neuroscience 12 (2018): 774.

[40] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[41] Li, Y., Yin, R., et al. “Wearable-based Human Activity Recognition with Spatio-Temporal Spiking Neural Networks”. arXiv preprint arXiv:2212.02233.

[42] Koravuna, Shamini, Ulrich Rückert, and Thorsten Jungeblut. "Exploring spiking neural networks: a comprehensive analysis of mathematical models and applications." Frontiers in Computational Neuroscience 17 (2023).

[43] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[44] oward, A. Zhu, G. M. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[45] Tan, Mingxing, et al. "Mnasnet: Platform-aware neural architecture search for mobile." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[46] Li, Yuhang, et al. "Differentiable spike: Rethinking gradient-descent for training spiking neural networks." Advances in Neural Information Processing Systems 34 (2021): 23426-23439.

[47] Prokhorov, David, et al. "Measuring robustness of Visual SLAM." 2019 16th International Conference on Machine Vision Applications (MVA). IEEE, 2019.

[48] Chen, Changhao, et al. "Oxiod: The dataset for deep inertial odometry." arXiv preprint arXiv:1809.07491 (2018).

[49] Yan, Hang, Qi Shan, and Yasutaka Furukawa. "RIDI: Robust IMU double integration." Proceedings of the European conference on computer vision (ECCV). 2018.

[50] Yin, Ruokai, et al. "Sata: Sparsity-aware training accelerator for spiking neural networks." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2022).