Boost event-driven tactile learning with location spiking neurons

Peng Kang[1]*, Srutarshi Banerjee[2], Henry Chopp[2],
Aggelos Katsaggelos[2] and Oliver Cossairt[1]

[1]Department of Computer Science, Northwestern University, Evanston, IL, United States, [2]Department of
Electrical and Computer Engineering, Northwestern University, Evanston, IL, United States

Tactile sensing is essential for a variety of daily tasks. Inspired by the event-driven nature and sparse spiking communication of the biological systems, recent advances in event-driven tactile sensors and Spiking Neural Networks (SNNs) spur the research in related fields. However, SNN-enabled event-driven tactile learning is still in its infancy due to the limited representation abilities of existing spiking neurons and high spatio-temporal complexity in the event-driven tactile data. In this paper, to improve the representation capability of existing spiking neurons, we propose a novel neuron model called "location spiking neuron," which enables us to extract features of event-based data in a novel way. Specifically, based on the classical Time Spike Response Model (TSRM), we develop the Location Spike Response Model (LSRM). In addition, based on the most commonly-used Time Leaky Integrate-and-Fire (TLIF) model, we develop the Location Leaky Integrate-and-Fire (LLIF) model. Moreover, to demonstrate the representation effectiveness of our proposed neurons and capture the complex spatio-temporal dependencies in the event-driven tactile data, we exploit the location spiking neurons to propose two hybrid models for event-driven tactile learning. Specifically, the first hybrid model combines a fully-connected SNN with TSRM neurons and a fully-connected SNN with LSRM neurons. And the second hybrid model fuses the spatial spiking graph neural network with TLIF neurons and the temporal spiking graph neural network with LLIF neurons. Extensive experiments demonstrate the significant improvements of our models over the state-of-the-art methods on event-driven tactile learning, including event-driven tactile object recognition and event-driven slip detection. Moreover, compared to the counterpart artificial neural networks (ANNs), our SNN models are 10× to 100× energy-efficient, which shows the superior energy efficiency of our models and may bring new opportunities to the spike-based learning community and neuromorphic engineering. Finally, we thoroughly examine the advantages and limitations of various spiking neurons and discuss the broad applicability and potential impact of this work on other spike-based learning applications.

1. Introduction

With the prevalence of artificial intelligence, computers today have demonstrated extraordinary abilities in visual and auditory perceptions. Although these perceptions are essential sensory modalities, they may fail to complete tasks in certain situations where tactile perception can help. For example, the visual sensory modality can fail to distinguish objects with similar visual features in less-favorable environments, such as dim-lit or in the presence of occlusions. In such cases, tactile sensing can provide meaningful information like texture, pressure, roughness, or friction and maintain performance. Overall, tactile perception is a vital sensing modality that enables humans to gain perceptual judgment on the surrounding environment and conduct stable movements (Taunyazov et al., 2020).

With the recent advances in material science and Artificial Neural Networks (ANNs), research on tactile perception has begun to soar, including tactile object recognition (Soh and Demiris, 2014; Kappassov et al., 2015; Sanchez et al., 2018), slip detection (Calandra et al., 2018), and texture recognition (Baishya and Bäuml, 2016; Taunyazov et al., 2019). Unfortunately, although ANNs demonstrate promising performance on the tactile learning tasks, they are usually power-hungry compared to human brains that require far less energy to perform the tactile perception robustly (Li et al., 2016; Strubell et al., 2019).

Inspired by biological systems, research on event-driven perception has started to gain momentum, and several asynchronous event-based sensors have been proposed, including event cameras (Gallego et al., 2020) and event-based tactile sensors (Taunyazov et al., 2020). In contrast to standard synchronous sensors, such event-based sensors can achieve higher energy efficiency, better scalability, and lower latency. However, due to the high sparsity and complexity of event-driven data, learning with these sensors is still in its infancy (Pfeiffer and Pfeil, 2018). Recently, several works (Gu et al., 2020; Taunyazov et al., 2020; Taunyazov et al., 2020) utilized Spiking Neural Networks [SNNs; Pfeiffer and Pfeil (2018); Shrestha and Orchard (2018); Xu et al. (2021)] to tackle event-driven tactile learning. Unlike ANNs, which require expensive transformations from asynchronous discrete events to synchronous real-valued frames, SNNs can process event-based sensor data directly. Moreover, unlike ANNs that employ artificial neurons (Maas et al., 2013; Clevert et al., 2015; Xu et al., 2015) and conduct real-valued computations, SNNs adopt spiking neurons (Gerstner, 1995; Abbott, 1999; Gerstner and Kistler, 2002) and utilize binary 0–1 spikes to process information. This difference reduces the mathematical dot-product operations in ANNs to less computationally expensive summation operations in SNNs (Roy et al., 2019). Due to the advantages of SNNs, these works are always energy-efficient and suitable for power-constrained devices. However, due to the limited representative abilities of existing spiking neuron models and high spatio-temporal complexity in the event-based tactile data (Taunyazov et al., 2020), these works still cannot sufficiently capture spatio-temporal dependencies and thus hinder the performance of event-driven tactile learning.

In this paper, to address the problems mentioned above, we make several contributions that boost event-driven tactile learning, including event-driven tactile object recognition and event-driven slip detection. We summarize a list of acronyms and notations in Table 1. Please refer to it during the reading.

First, to enable richer representative abilities of existing spiking neurons, we propose a novel neuron model called "location spiking neuron." Unlike existing spiking neuron models that update their membrane potentials based on time steps (Roy et al., 2019), location spiking neurons update their membrane potentials based on locations. Specifically, based on the Time Spike Response Model [TSRM; Gerstner (1995)], we develop the "Location Spike Response Model (LSRM)." Moreover, to make the location spiking neurons more applicable to a wide range of applications, we develop the "Location Leaky Integrate-and-Fire (LLIF)" model based on the most commonly-used Time Leaky Integrate-and-Fire (TLIF) model (Abbott, 1999). Please note that TSRM and TLIF are the classical Spike Response Model (SRM) and Leaky Integrate-and-Fire (LIF) in the literature. We add the character "T (Time)" to highlight their differences from LSRM and LLIF. These location spiking neurons enable the extraction of feature representations of event-based data in a novel way. Previously, SNNs adopted temporal recurrent neuronal dynamics to extract features from the event-based data. With location spiking neurons, we can build SNNs that employ spatial recurrent neuronal dynamics to extract features from the event-based data. We believe location spiking neuron models can have a broad impact on the SNN community and spur the research on spike-based learning from event sensors like NeuTouch (Taunyazov et al., 2020), Dynamic Audio Sensors (Anumula et al., 2018), or Dynamic Vision Sensors (Gallego et al., 2020).

Next, we investigate the representation effectiveness of location spiking neurons and propose two models for event-driven tactile learning. Specifically, to capture the complex spatio-temporal dependencies in the event-driven tactile data, the first model combines a fully-connected (FC) SNN with TSRM neurons and a fully-connected (FC) SNN with LSRM neurons, henceforth referred to as the Hybrid_SRM_FC. To capture more spatio-temporal topology knowledge in the event-driven tactile data, the second model fuses the spatial spiking graph neural network (GNN) with TLIF neurons and temporal spiking graph neural network (GNN) with LLIF neurons, henceforth referred to as the Hybrid_LIF_GNN. To be more specific, the Hybrid_LIF_GNN first constructs tactile spatial graphs and tactile temporal graphs based on taxel locations and event time sequences, respectively. Then, it utilizes the spatial spiking graph neural network with TLIF neurons and the temporal spiking graph neural network with LLIF neurons to extract features of these graphs. Finally, it fuses the spiking tactile features from the two networks and provides the final tactile learning prediction. Besides the novel model construction, we also specify the location orders to enable the spatial recurrent neuronal dynamics of location spiking neurons in event-driven tactile learning. In addition, we explore the robustness of location orders on event-driven tactile learning. Moreover, we design new loss functions involved with locations and utilize the backpropagation methods to optimize the proposed models. Furthermore, we develop the timestep-wise inference algorithms

TABLE 1 List of acronyms and notations in the paper.

TSRM	Time Spike Response Model
LSRM	Location Spike Response Model
TLIF	Time Leaky Integrate-and-Fire
LLIF	Location Leaky Integrate-and-Fire
Hybrid_SRM_FC	The hybrid model that combines a fully-connected SNN with TSRM neurons and a fully-connected SNN with LSRM neurons
Hybrid_LIF_GNN	The hybrid model that fuses the spatial spiking graph neural network with TLIF neurons and temporal spiking graph neural network with LLIF neurons
ν	$\nu = t$ for existing spiking neurons and $\nu = l$ for location spiking neurons
$u_i(\nu)$	The membrane potential of neuron i at ν
$\eta_i(\cdot)$	The refractory kernel of neuron i
$\epsilon_{ij}(\cdot)$	The incoming spike response kernel between neurons i and j
Γ_i	The set of presynaptic neurons of neuron i
w_{ij}	The connection strength between neurons i and j
$x_j(\nu)$	The presynaptic input from pre-neuron j at ν
$I(\nu)$	The weighted summation of the presynaptic inputs at ν
τ	The time constant of TLIF neurons
α	The decay factor of TLIF neurons
τ'	The location constant of LLIF neurons
β	The location decay factor of LLIF neurons
u_{th}	The firing thresholds of neurons
N	The number of taxels of NeuTouch
T	The number of total time length of event sequences
K	The number of classes for the tasks
X_{in}	The event-based tactile input
X'_{in}	The transposed event-based tactile input
O_1	Output spikes from the SNN with existing spiking neurons
$o_i(t)$	The output spiking state of existing spiking neuron i at time t
O_2	Output spikes from the SNN with location spiking neurons
$o_i(l)$	The output spiking state of location spiking neuron i at location l
O	Output spikes from the Hybrid_SRM_FC
$G_s(t)$	The tactile spatial graph at time t
$G_t(n)$	The tactile temporal graph of taxel n
O'_1	The predicted label vector of the spatial spiking graph neural network
O'_2	The predicted label vector of the temporal spiking graph neural network
O'	The predicted label vector of the Hybrid_LIF_GNN

for the two models to show their applicability to the spike-based temporal data.

Lastly, we conduct experiments on three challenging event-driven tactile learning tasks. Specifically, the first task requires models to determine the type of objects being handled. The second task requires models to determine the type of containers being handled and the amount of liquid held within, which is more challenging than the first task. And the third task asks models to accurately detect the rotational slip ("stable" or "rotate") within 0.15 s. Extensive experimental results demonstrate the significant improvements of our models over the state-of-the-art methods on event-driven tactile learning. Moreover, the

experiments show that existing spiking neurons are better at capturing spatial dependencies, while location spiking neurons are better at modeling mid-and-long temporal dependencies. Furthermore, compared to the counterpart ANNs, our models are 10× to 100× energy-efficient, which shows the superior energy efficiency of our models and may bring new opportunities to neuromorphic engineering.

Portions of this work "Event-Driven Tactile Learning with Location Spiking Neurons (Kang et al., 2022)" were accepted by IJCNN 2022 and an oral presentation was given at the IEEE WCCI 2022. In the conference paper, we proposed location spiking neurons and demonstrated the dynamics of LSRM neurons. By exploiting the LSRM neurons, we developed the model Hybrid_SRM_FC for event-driven tactile learning and experimental results on benchmark datasets demonstrated the extraordinary performance and high energy efficiency of the Hybrid_SRM_FC and LSRM neurons. We highlight the additional contributions in this paper below.

- To make the location spiking neurons user-friendly in various spike-based learning frameworks, we expand the idea of location spiking neurons to the most commonly-used TLIF neurons and propose the LLIF neurons. Specifically, the LLIF neurons update their membrane potentials based on locations and enable the models to extract features with spatial recurrent neuronal dynamics. We can incorporate the LLIF neurons into popular spike-based learning frameworks like STBP (Wu et al., 2018) and tap their feature representation potential. We believe such neuron models can have a broad impact on the SNN community and spur the research on spike-based learning.
- To demonstrate the advantage of LLIF neurons and further boost the event-based tactile learning performance, we build the Hybrid_LIF_GNN, which fuses the spatial spiking graph neural network with TLIF neurons and the temporal spiking graph neural network with LLIF neurons. The model extracts features from tactile spatial graphs and tactile temporal graphs concurrently. To the best of our knowledge, this is the first work to construct tactile temporal graphs based on event sequences and build a temporal spiking graph neural network for event-driven tactile learning.
- We further include more data, experiments, and interpretation to demonstrate the effectiveness and energy efficiency of the proposed neurons and models. Extensive experiments on real-world datasets show that the Hybrid_LIF_GNN significantly outperforms the state-of-the-art methods for event-driven tactile learning, including the Hybrid_SRM_FC (Kang et al., 2022). Moreover, the computational cost evaluation demonstrates the high-efficiency benefits of the Hybrid_LIF_GNN and LLIF neurons, which may unlock their potential on neuromorphic hardware. The source code is available at: https://github.com/pkang2017/TactileLSN.
- We thoroughly discuss the advantages and limitations of existing spiking neurons and location spiking neurons. Moreover, we provide preliminary results on event-driven audio learning and discuss the broad applicability

and potential impact of this work on other spike-based learning applications.

The rest of the paper is organized as follows. In Section 2, we provide an overview of related work on SNNs and event-driven tactile sensing and learning. In Section 3, we start by introducing notations for existing spiking neurons and extend them to the specific location spiking neurons. We then propose various models with location spiking neurons for event-driven tactile learning. Last, we provide implementation details and algorithms related to the proposed models. In Section 4, we demonstrate the effectiveness and energy efficiency of our models on benchmark datasets. Finally, we discuss and conclude in Section 5.

2. Related work

In the following, we provide a brief overview of related work on SNNs and event-driven tactile sensing and learning.

2.1. Spiking Neural Networks (SNNs)

With the prevalence of Artificial Neural Networks (ANNs), computers today have demonstrated extraordinary abilities in many cognition tasks. However, ANNs only imitate brain structures in several ways, including vast connectivity and structural and functional organizational hierarchy (Roy et al., 2019). The brain has more information processing mechanisms like the neuronal and synaptic functionality (Felleman and Van Essen, 1991; Bullmore and Sporns, 2012). Moreover, ANNs are much more energy-consuming than human brains. To integrate more brain-like characteristics and make artificial intelligence models more energy-efficient, researchers propose Spiking Neural Networks (SNNs), which can be executed on power-efficient neuromorphic processors like TrueNorth (Merolla et al., 2014) and Loihi (Davies et al., 2021). Similar to ANNs, SNNs can adopt general network topologies like convolutional layers and fully-connected layers, but use different neuron models (Gerstner and Kistler, 2002), such as the Time Leaky Integrate-and-Fire (TLIF) model (Abbott, 1999) and the Time Spike Response Model [TSRM; Gerstner (1995)]. Due to the non-differentiability of these spiking neuron models, it still remains challenging to train SNNs. Nevertheless, several solutions have been proposed, such as converting the trained ANNs to SNNs (Cao et al., 2015; Sengupta et al., 2019) and approximating the derivative of the spike function (Wu et al., 2018; Cheng et al., 2020). In this work, we propose location spiking neurons to enhance the representative abilities of existing spiking neurons. These location spiking neurons maintain the spiking characteristic but employ the spatial recurrent neuronal dynamics, which enable us to build energy-efficient SNNs and extract features of event-based data in a novel way. Moreover, based on the optimization methods for SNNs with existing spiking neurons, we design new loss functions for SNNs with location spiking neurons and utilize the backpropagation methods with surrogate gradients to optimize the proposed models.

2.2. Event-driven tactile sensing and learning

With the prevalence of material science and robotics, several tactile sensors have been developed, including non-event-based tactile sensors like the iCub RoboSkin (Schmitz et al., 2010) and the SynTouch BioTac (Fishel and Loeb, 2012) and event-driven tactile sensors like the NeuTouch (Taunyazoz et al., 2020) and the NUSkin (Taunyazov et al., 2021). In this paper, we focus on event-driven tactile learning with SNNs. Since the development of event-driven tactile sensors is still in its infancy (Gu et al., 2020), little prior work exists on learning event-based tactile data with SNNs. The work (Taunyazov et al., 2020) employed a neural coding scheme to convert raw tactile data from non-event-based tactile sensors into event-based spike trains. It then utilized an SNN to process the spike trains and classify textures. A recent work (Taunyazoz et al., 2020) released the first publicly-available event-driven visual-tactile dataset collected by NeuTouch and proposed an SNN based on SLAYER (Shrestha and Orchard, 2018) to solve the event-driven tactile learning. Moreover, to naturally capture the spatial topological relations and structural knowledge in the event-based tactile data, a very recent work (Gu et al., 2020) utilized the spiking graph neural network (Xu et al., 2021) to process the event-based tactile data and conduct the tactile object recognition. In this paper, different from previous works building SNNs with spiking neurons that employ the temporal recurrent neuronal dynamics, we construct SNNs with location spiking neurons to capture the complex spatio-temporal dependencies in the event-based tactile data and improve event-driven tactile learning.

3. Methods

In this section, we first demonstrate the spatial recurrent neuronal dynamics of location spiking neurons by introducing notations for the existing spiking neurons and extending them to the location spiking neurons. We then introduce two models with location spiking neurons for event-driven tactile learning. Last, we provide implementation details and algorithms related to the proposed models.

3.1. Existing spiking neuron models vs. location spiking neuron models

Spiking neuron models are mathematical descriptions of specific cells in the nervous system. They are the basic building blocks of SNNs. In this section, we first introduce the mechanisms of existing spiking neuron models – the TSRM (Gerstner, 1995) and the TLIF (Abbott, 1999). To enrich their representative abilities, we transform them into location spiking neuron models – the LSRM and the LLIF.

In the TSRM, the temporal recurrent neuronal dynamics of neuron i are described by its membrane potential $u_i(t)$. When $u_i(t)$ exceeds a predefined threshold u_{th} at the firing time $t_i^{(f)}$, the neuron i will generate a spike. The set of all firing times of neuron i is

denoted by

$$\mathcal{F}_i = \{t_i^{(f)}; 1 \leq f \leq n\} = \{t|u_i(t) = u_{th}\}, \quad (1)$$

where $t_i^{(n)}$ is the most recent spike time $t_i^{(f)} < t$. The value of $u_i(t)$ is governed by two different spike response processes:

$$u_i(t) = \sum_{t_i^{(f)} \in \mathcal{F}_i} \eta_i(t - t_i^{(f)}) + \sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in \mathcal{F}_j} w_{ij} x_j(t_j^{(f)}) \epsilon_{ij}(t - t_j^{(f)}), \quad (2)$$

where Γ_i is the set of presynaptic neurons of neuron i and $x_j(t_j^{(f)}) = 1$ is the presynaptic spike at time $t_j^{(f)}$. $\eta_i(t)$ is the refractory kernel, which describes the response of neuron i to its own spikes at time t. $\epsilon_{ij}(t)$ is the incoming spike response kernel, which models the neuron i's response to the presynaptic spikes from neuron j at time t. w_{ij} accounts for the connection strength between neuron i and neuron j and scales the incoming spike response. Figure 1A of $\nu = t$ visualizes the refractory dynamics of the TSRM neuron i and Figure 1B of $\nu = t$ visualizes the incoming spike dynamics of the TSRM neuron i.

Without loss of generality, such temporal recurrent neuronal dynamics also apply to other spiking neuron models, such as the TLIF, which is a special case of the TSRM (Maass and Bishop, 2001). Since the TLIF model is computationally tractable and maintains biological fidelity to a certain degree, it becomes the most commonly-used spiking neuron model and there are many popular SNN frameworks powered with it (Wu et al., 2018). The dynamics of the TLIF neuron i are governed by

$$\tau \frac{du_i(t)}{dt} = -u_i(t) + I(t), \quad (3)$$

where $u_i(t)$ represents the internal membrane potential of the neuron i at time t, τ is a time constant, and $I(t)$ signifies the presynaptic input obtained by the combined action of synaptic weights and pre-neuronal activities. To better understand the membrane potential update of TLIF neurons, the Euler method is used to transform the first-order differential equation of Equation 3 into a recursive expression:

$$u_i(t) = (1 - \frac{dt}{\tau})u_i(t - 1) + \frac{dt}{\tau} \sum_j w_{ij} x_j(t), \quad (4)$$

where $\sum_j w_{ij} x_j(t)$ is the weighted summation of the inputs from pre-neurons at the current time step. Equation 4 can be further simplified as:

$$u_i(t) = \alpha u_i(t - 1) + \sum_j w'_{ij} x_j(t), \quad (5)$$

where $\alpha = 1 - \frac{dt}{\tau}$ can be considered a decay factor, and w'_{ij} is the weight incorporating the scaling effect of $\frac{dt}{\tau}$. When $u_i(t)$ exceeds a certain threshold u_{th}, the neuron emits a spike, resets its membrane potential to u_{reset}, and then accumulates $u_i(t)$ again in subsequent time steps. Figure 1C of $\nu = t$ visualizes the temporal dynamics of a TLIF neuron i.

From the above descriptions, we find that existing spiking neuron models have explicit temporal recurrence but do not

FIGURE 1
Recurrent neuronal dynamic mechanisms for the existing spiking neurons of $\nu = t$ and location spiking neurons of $\nu = l$. Unlike existing spiking neuron models that update their membrane potentials based on time steps $\nu = t$, location spiking neurons update their membrane potentials based on locations $\nu = l$. **(A)** The refractory dynamics of a TSRM neuron i or an LSRM neuron i. Immediately after firing an output spike at $\nu_i^{(f)}$, the value of $u_i(\nu)$ is lowered or reset by adding a negative contribution $\eta_i(\cdot)$. The kernel $\eta_i(\cdot)$ vanishes for $\nu < \nu_i^{(f)}$ and decays to zero for $\nu \to \infty$. **(B)** The incoming spike dynamics of a TSRM neuron i or an LSRM neuron i. A presynaptic spike at $\nu_j^{(f)}$ increases the value of $u_i(\nu)$ for $\nu \geq \nu_j^{(f)}$ by an amount of $w_{ij}x_j(\nu_j^{(f)})\epsilon_{ij}(\nu - \nu_j^{(f)})$. The kernel $\epsilon_{ij}(\cdot)$ vanishes for $\nu < \nu_j^{(f)}$. "<" and "≥" indicate the location order when $\nu = l$. **(C)** The recurrent neuronal dynamics of a TLIF neuron i or an LLIF neuron i. The neuron i takes as input binary spikes and outputs binary spikes. x_j represents the input signal to the neuron i from neuron j, u_i is the neuron's membrane potential, and o_i is the neuron's output. An output spike will be emitted from the neuron when its membrane potential surpasses the firing threshold u_{th}, after which the membrane potential will be reset to u_{reset}. This figure is adapted from Kang et al. (2022).

possess explicit spatial recurrence, which, to some extent, limits their representative abilities.

To enrich the representative abilities of existing spiking neuron models, we propose location spiking neurons, which adopt the spatial recurrent neuronal dynamics and update their membrane potentials based on locations.[1] These neurons exploit explicit spatial recurrence. Specifically, the spatial recurrent neuronal dynamics of the LSRM neuron i are described by its location membrane potential $u_i(l)$. When $u_i(l)$ exceeds a predefined threshold u_{th} at the firing location $l_i^{(f)}$, the neuron i will generate a spike. The set of all firing locations of neuron i is denoted by

$$\mathcal{G}_i = \{l_i^{(f)}; 1 \leq f \leq n\} = \{l | u_i(l) = u_{th}\}, \qquad (6)$$

where $l_i^{(n)}$ is the nearest firing location $l_i^{(f)} < l$. "<" indicates the location order, which is manually set and will be discussed in Section 3.3. The value of $u_i(l)$ is governed by two different spike response processes:

$$u_i(l) = \sum_{l_i^{(f)} \in \mathcal{G}_i} \eta_i(l - l_i^{(f)}) + \sum_{j \in \Gamma_i} \sum_{l_j^{(f)} \in \mathcal{G}_j} w_{ij}x_j(l_j^{(f)})\epsilon_{ij}(l - l_j^{(f)}), \quad (7)$$

where Γ_i is the set of presynaptic neurons of neuron i and $x_j(l_j^{(f)}) = 1$ is the presynaptic spike at location $l_j^{(f)}$. $\eta_i(l)$ is the refractory kernel, which describes the response of neuron i to its own spikes at location l. $\epsilon_{ij}(l)$ is the incoming spike response kernel, which models the neuron i's response to the presynaptic spikes from neuron j at location l. Figure 1A of $\nu = l$ visualizes the refractory dynamics of the LSRM neuron i and Figure 1B of $\nu = l$ visualizes the incoming spike dynamics of the LSRM neuron i. The threshold u_{th} of LSRM neurons can be different from that of TSRM neurons, while we set the same for simplicity. In Section 3.2.1, we will apply

the LSRM neurons to event-driven tactile learning and show how the proposed neurons enable feature extraction in a novel way.

To make the location spiking neurons user-friendly and compatible with various spike-based learning frameworks, we expand the idea of location spiking neurons to the most commonly-used TLIF neurons and propose the LLIF neurons. Different from the temporal dynamics shown in Equation 3, the LLIF neuron i employs the spatial dynamics:

$$\tau' \frac{du_i(l)}{dl} = -u_i(l) + I(l), \qquad (8)$$

where $u_i(l)$ represents the internal membrane potential of an LLIF neuron i at location l, τ' is a location constant, and $I(l)$ represents the presynaptic input. We use the Euler method again to transform the first-order differential equation of Equation 8 into a recursive expression:
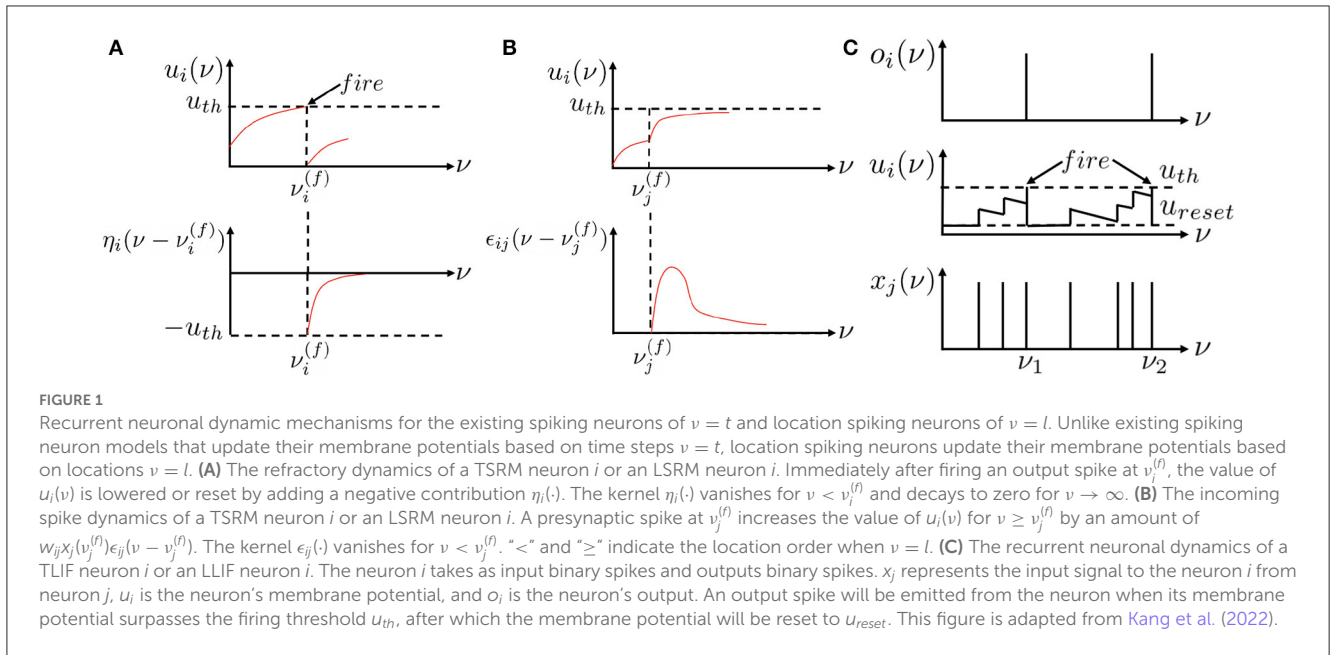
$$u_i(l) = (1 - \frac{dl}{\tau'})u_i(l_{prev}) + \frac{dl}{\tau'} \sum_j w_{ij}x_j(l), \qquad (9)$$

where $\sum_j w_{ij}x_j(l)$ is the weighted summation of the inputs from pre-neurons at the current location. Equation 9 can be further simplified as:

$$u_i(l) = \beta u_i(l_{prev}) + \sum_j w'_{ij}x_j(l), \qquad (10)$$

where $\beta = 1 - \frac{dl}{\tau'}$ can be considered a location decay factor, and w'_{ij} is the weight incorporating the scaling effect of $\frac{dl}{\tau'}$. When $u_i(l)$ exceeds a certain threshold u_{th}, the neuron emits a spike, resets its membrane potential to u_{reset}, and then accumulates $u_i(l)$ again at subsequent locations. u_{th} and u_{reset} of LLIF neurons can be different from those of TLIF neurons, while we set the same for simplicity. Figure 1C of $\nu = l$ visualizes the spatial recurrent neuronal dynamics of an LLIF neuron i. To enable the dynamics of LLIF neurons, we still need to specify the location order like the

---

1 Locations could refer to pixel or patch locations for images or taxel locations for tactile sensors.

LSRM neurons. In Section 3.2.2, we will demonstrate how the LLIF neurons can be incorporated into the popular spike-based learning framework and further boost the performance of event-driven tactile learning.

3.2. Event-driven tactile learning with location spiking neurons

To investigate the representation effectiveness of location spiking neurons and boost the event-driven tactile learning performance, we propose two models with location spiking neurons, which capture complex spatio-temporal dependencies in the event-based tactile data. In this paper, we focus on processing the data collected by NeuTouch (Taunyazoz et al., 2020), a biologically-inspired event-driven fingertip tactile sensor with 39 taxels arranged spatially in a radial fashion (see Figure 2).

3.2.1. Event-driven tactile learning with the LSRM neurons

In this section, we introduce event-driven tactile learning with the LSRM neurons. Specifically, we propose the Hybrid_SRM_FC to capture the complex spatio-temporal dependencies in the event-driven tactile data.

Figure 2 presents the network structure of the Hybrid_SRM_FC. From the figure, we can see that the model has two components, including the fully-connected SNN with TSRM neurons and the fully-connected SNN with LSRM neurons. Specifically, the fully-connected SNN with TSRM neurons employs the temporal recurrent neuronal dynamics to extract spiking feature representations from the event-based tactile data $X_{in} \in \mathbb{R}^{N \times T}$, where N is the total number of taxels and T is the

total time length of event sequences. The fully-connected SNN with LSRM neurons utilizes the spatial recurrent neuronal dynamics to extract spiking feature representations from the event-based tactile data $X'_{in} \in \mathbb{R}^{T \times N}$, where X'_{in} is transposed from X_{in}. The spiking representations from two networks are then concatenated to yield the final task-specific output.

To be more specific, the top part of Figure 2 shows the network structure of fully-connected SNN with TSRM neurons. It employs two spiking fully-connected layers with TSRM neurons to process X_{in} and generate the spiking representations $O_1 \in \mathbb{R}^{K \times T}$, where K is the output dimension determined by the task. The membrane potential $u_i(t)$, the output spiking state $o_i(t)$, and the set of all firing times $\mathcal{F}_i$ of TSRM neuron i in these layers are decided by:

$$u_i(t) = \sum_{t_i^{(f)} \in \mathcal{F}_i} \eta(t - t_i^{(f)}) + \underbrace{\sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in \mathcal{F}_j} w_{ij} o_j(t_j^{(f)}) \epsilon(t - t_j^{(f)})}_{\text{capture spatial dependencies}},$$

$$o_i(t) = \begin{cases} 1 & \text{if } u_i(t) \geq u_{th}; \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathcal{F}_i = \begin{cases} \mathcal{F}_i \cup t & \text{if } o_i(t) = 1; \\ \mathcal{F}_i & \text{otherwise,} \end{cases}$$

(11)

where w_{ij} are the trainable parameters, $\eta(t)$ and $\epsilon(t)$ model the temporal recurrent neuronal dynamics of TSRM neurons, Γ_i is the set of presynaptic TSRM neurons spanning over the spatial domain, which is utilized to capture the spatial dependencies in the event-based tactile data.

Moreover, the bottom part of Figure 2 shows the network structure of fully-connected SNN with LSRM neurons. It employs two spiking fully-connected layers with LSRM neurons to process X'_{in} and generate the spiking representations $O_2 \in \mathbb{R}^{K \times N}$, where K is the output dimension decided by the task. The membrane
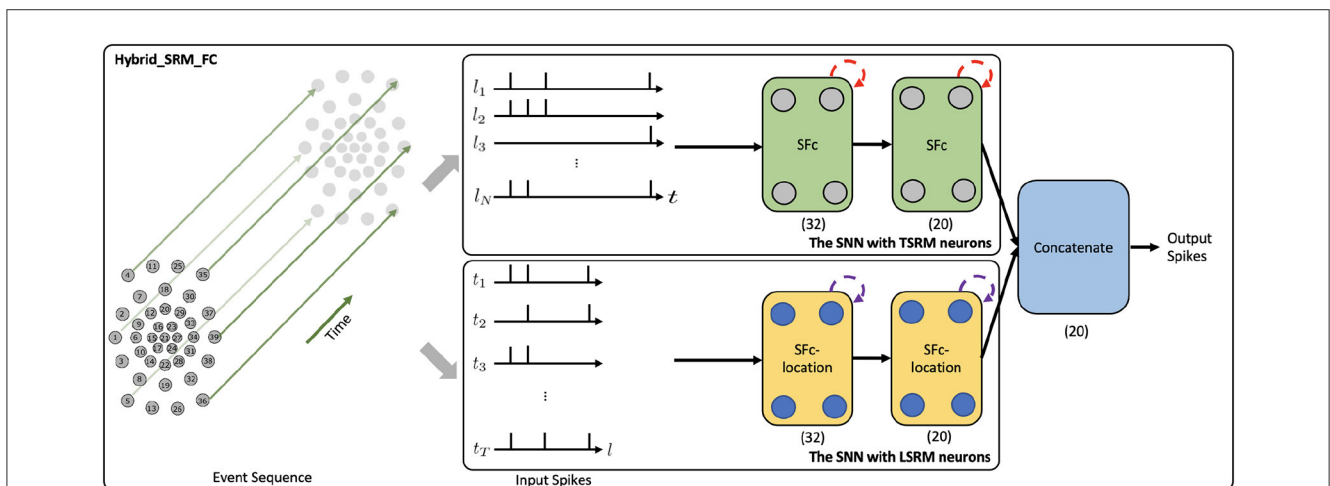

FIGURE 2
The network structure of the Hybrid_SRM_FC. **(The Upper Panel)** The SNN with TSRM neurons processes the input spikes X_{in} and adopts the temporal recurrent neuronal dynamics (shown with red dashed arrows) of TSRM neurons to extract features from the data, where SFc is the spiking fully-connected layer with TSRM neurons. **(The Lower Panel)** The SNN with LSRM neurons processes the transposed input spikes X'_{in} and employs the spatial recurrent neuronal dynamics (shown with purple dashed arrows) of LSRM neurons to extract features from the data, where SFc-location is the spiking fully-connected layer with LSRM neurons. Finally, the spiking representations from two networks are concatenated to yield the final predicted label. (32) and (20) represent the sizes of fully-connected layers, where we assume the number of classes (K) is 20. This figure is adapted from Kang et al. (2022).

potential $u_i(l)$, the output spiking state $o_i(l)$, and the set of all firing locations $\mathcal{G}_i$ of LSRM neuron i in these layers are decided by:

$$u_i(l) = \sum_{l_i^{(f)} \in \mathcal{G}_i} \eta(l - l_i^{(f)}) + \underbrace{\sum_{j \in \Gamma_i'} \sum_{l_j^{(f)} \in \mathcal{G}_j} w_{ij} o_j(l_j^{(f)}) \epsilon(l - l_j^{(f)})}_{\text{model temporal dependencies}},$$

$$o_i(l) = \begin{cases} 1 & \text{if } u_i(l) \geq u_{th}; \\ 0 & \text{otherwise}, \end{cases} \qquad (12)$$

$$\mathcal{G}_i = \begin{cases} \mathcal{G}_i \cup l & \text{if } o_i(l) = 1; \\ \mathcal{G}_i & \text{otherwise}, \end{cases}$$

where w_{ij} are the trainable connection weights, $\eta(l)$ and $\epsilon(l)$ determine the spatial recurrent neuronal dynamics of LSRM neurons, Γ_i' is the set of presynaptic LSRM neurons spanning over the temporal domain, which is utilized to model the temporal dependencies in the event-based tactile data. Such location spiking neurons tap the representative potential and enable us to capture features in this novel way.

Lastly, we concatenate the spiking representations of O_1 and O_2 along the last dimension and obtain the final output spike train $O \in \mathbb{R}^{K \times (T+N)}$. The predicted label is associated with the neuron $k \in K$ with the largest number of spikes in the domain of $T + N$.

3.2.2. Event-driven tactile learning with the LLIF Neurons

In this section, to demonstrate the usability of location spiking neurons and further boost the event-driven tactile learning performance, we utilize the LLIF neurons to propose the Hybrid_LIF_GNN, which fuses spatial and temporal spiking graph neural networks and captures complex spatio-temporal dependencies in the event-based tactile data.
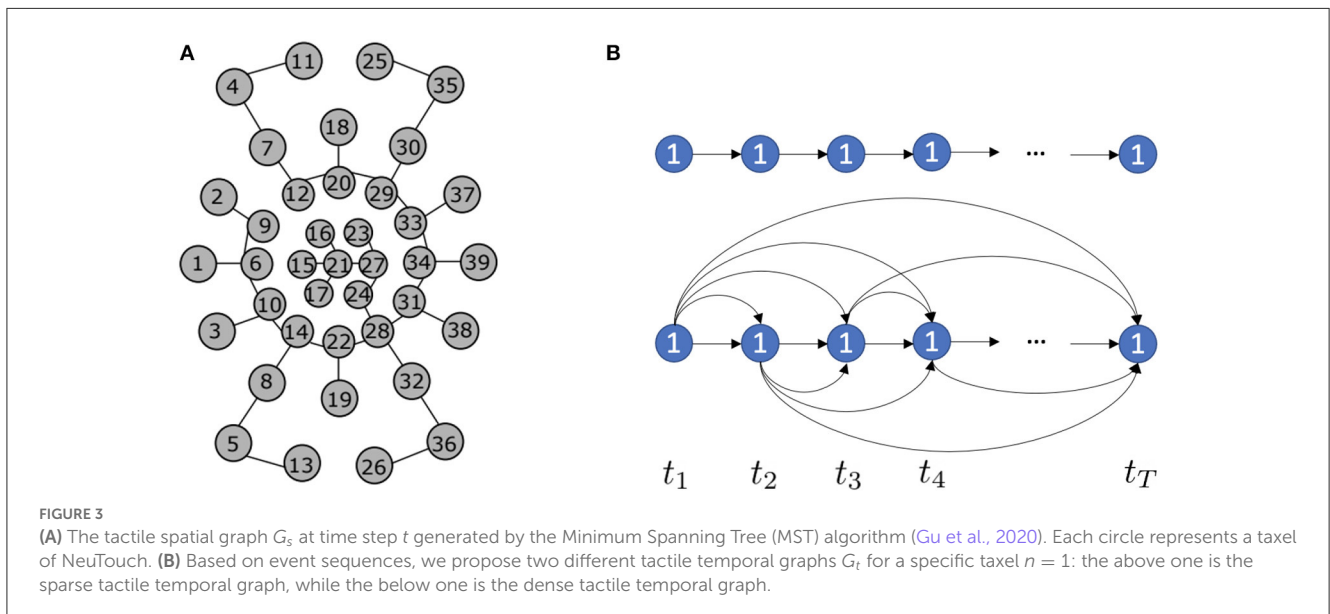
3.2.2.1. Tactile graph construction

Given event-based tactile inputs $X_{in} \in \mathbb{R}^{N \times T}$, we construct tactile spatial graphs and tactile temporal graphs as illustrated in Figure 3.
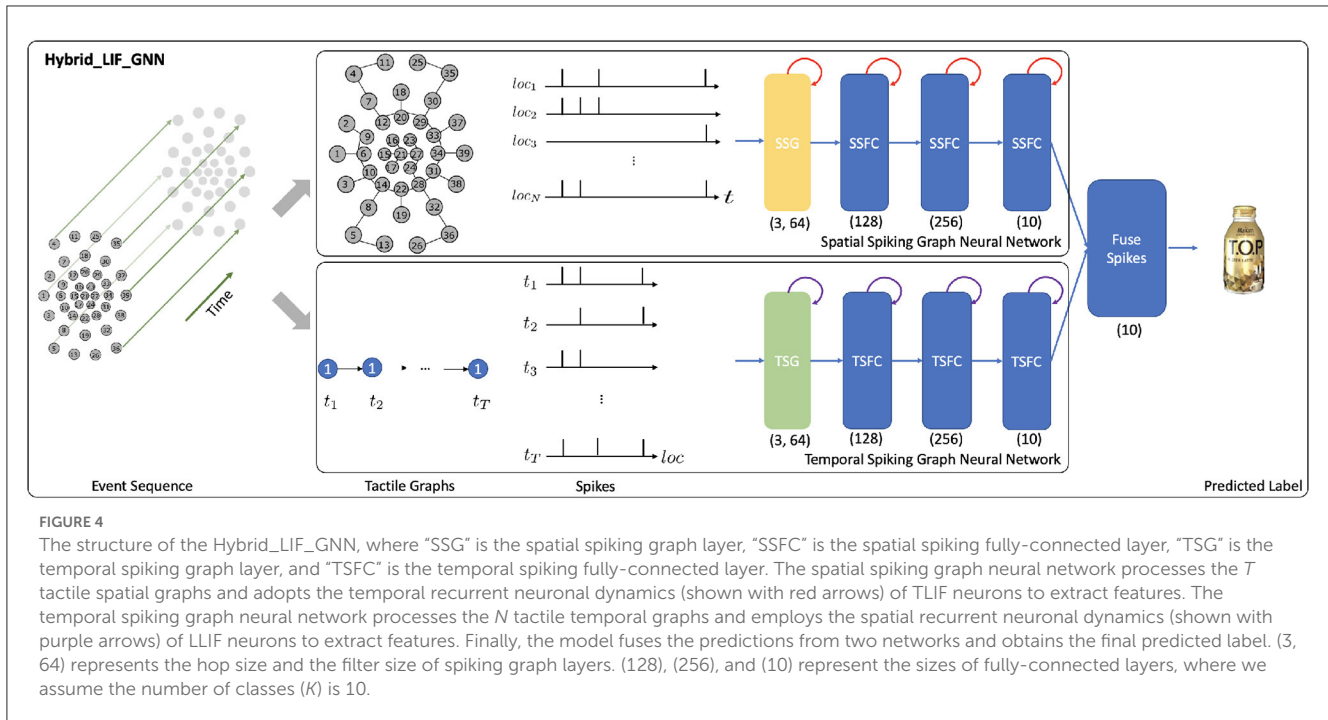
The tactile spatial graph $G_s(t) = (V^t, E^t)$ at time step t explicitly captures the spatial structural information in the data, while the tactile temporal graph $G_t(n) = (V_n, E_n)$ for a specific taxel n explicitly models the temporal dependency in the data. $V^t = \{v_n^t | n = 1, ..., N\}$ and $V_n = \{v_n^t | t = 1, ..., T\}$ represent nodes of $G_s(t)$ and $G_t(n)$, respectively, and the attribute of v_n^t is the event feature of the n-th taxel at time step t. $E^t = \{e_{i,j}^t | i, j = 1, ..., N\}$ represents the edges of $G_s(t)$, where $e_{i,j}^t \in \{0, 1\}$ indicates whether the nodes v_i^t, v_j^t are connected (denoted as 1) or disconnected (denoted as 0). E^t is formed by the Minimum Spanning Tree (MST) algorithm, where the Euclidean distance between taxels $d(v_i^t, v_j^t) = \|(x,y)_{v_i^t} - (x,y)_{v_j^t}\|_2$ is used to determine whether the edges are in the MST. Since the 2D coordinates (x,y) of taxels do not change with time, E^t remains the same throughout time. Moreover, the adjacency matrix of E^t is symmetric (i.e., the edges are indirect) as we assume the mutual spatial dependency in the data. $E_n = \{e_n^{p,q} | p, q = 1, ..., T\}$ represents the edges of $G_t(n)$, where $e_n^{p,q} \in \{0, 1\}$ and each edge is direct. Based on different temporal dependency assumptions, we propose two kinds of tactile temporal graphs shown in Figure 3B. One is sparse since we assume the current state only directly impacts the nearest future state. While the other is dense since we assume the current state has a broad impact on the future states. E_n remains the same for all N taxels.

3.2.2.2. Hybrid_LIF_GNN

To process the data from tactile graphs and capture the complex spatio-temporal dependencies in the event-based tactile data, we propose the Hybrid_LIF_GNN (see Figure 4), which fuses spatial and temporal spiking graph neural networks. Specifically, we adopt the spatial spiking graph neural network with TLIF neurons (Gu et al., 2020), which is a spike-based tactile learning framework powered by STBP (Wu et al., 2018). It uses temporal recurrent

FIGURE 3
(A) The tactile spatial graph G_s at time step t generated by the Minimum Spanning Tree (MST) algorithm (Gu et al., 2020). Each circle represents a taxel of NeuTouch. **(B)** Based on event sequences, we propose two different tactile temporal graphs G_t for a specific taxel $n = 1$: the above one is the sparse tactile temporal graph, while the below one is the dense tactile temporal graph.

FIGURE 4
The structure of the Hybrid_LIF_GNN, where "SSG" is the spatial spiking graph layer, "SSFC" is the spatial spiking fully-connected layer, "TSG" is the temporal spiking graph layer, and "TSFC" is the temporal spiking fully-connected layer. The spatial spiking graph neural network processes the T tactile spatial graphs and adopts the temporal recurrent neuronal dynamics (shown with red arrows) of TLIF neurons to extract features. The temporal spiking graph neural network processes the N tactile temporal graphs and employs the spatial recurrent neuronal dynamics (shown with purple arrows) of LLIF neurons to extract features. Finally, the model fuses the predictions from two networks and obtains the final predicted label. (3, 64) represents the hop size and the filter size of spiking graph layers. (128), (256), and (10) represent the sizes of fully-connected layers, where we assume the number of classes (K) is 10.

neuronal dynamics to capture the spatial structure information from the tactile spatial graphs. Inspired by this model, we develop the temporal spiking graph neural network with LLIF neurons, which is also powered by STBP. Our temporal spiking graph neural network utilizes spatial recurrent neuronal dynamics to extract the temporal dependencies in the tactile temporal graphs. Finally, we fuse the spiking features from two networks and obtain the final prediction.

To be more specific, the spatial spiking graph neural network takes as input tactile spatial graphs, and it has one spatial spiking graph layer and three spatial spiking fully-connected layers, where TLIF neurons that employ the temporal recurrent neuronal dynamics are the basic building blocks. On the other hand, the temporal spiking graph neural network takes as input tactile temporal graphs, and it has one temporal spiking graph layer and three temporal spiking fully-connected layers, where LLIF neurons that possess the spatial recurrent neuronal dynamics are the basic building blocks.

Based on Equation 5, the membrane potential $u_i(t)$ and output spiking state $o_i(t)$ of TLIF neuron i in the spatial spiking graph layer are decided by:

$$u_i(t) = \alpha u_i(t-1)(1 - o_i(t-1)) + I(t),$$
$$o_i(t) = \begin{cases} 1 & \text{if } u_i(t) \geq u_{th}; \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

where $I(t) = \text{GNN}(G_s(t))$ is to capture the spatial structural information. The membrane potential $u_i(t)$ and output spiking state $o_i(t)$ of TLIF neuron i in spatial spiking fully-connected layers are also decided by Equation 13, where $I(t) = \text{FC}(Pre(t))$ and $Pre(t)$ is the previous layer's output at time step t.

Based on Equation 10, the membrane potential $u_i(l)$ and output spiking state $o_i(l)$ of LLIF neuron i in the temporal spiking graph

layer are decided by:

$$u_i(l) = \beta u_i(l_{prev})(1 - o_i(l_{prev})) + I(l),$$
$$o_i(l) = \begin{cases} 1 & \text{if } u_i(l) \geq u_{th}; \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

where $I(l) = \text{GNN}(G_t(l))$ is to model the temporal dependencies in the data. The membrane potential $u_i(l)$ and output spiking state $o_i(l)$ of LLIF neuron i in temporal spiking fully-connected layers are also decided by Equation 14, where $I(l) = \text{FC}(Pre(l))$ and $Pre(l)$ is the previous layer's output at location l. l is the taxel $n \in N$ in event-driven tactile learning. To fairly compare with other baselines, we use TAGConv (Du et al., 2017) as GNN in this paper.

The spatial spiking graph neural network finally outputs the spiking feature $O_1 \in \mathbb{R}^{K \times T}$ and predicts the label vector $O_1' \in \mathbb{R}^K$ by averaging O_1 over the time window T,

$$O_1' = \frac{1}{T} \sum_t^T O_1(t), \tag{15}$$

where $O_1(t) \in \mathbb{R}^K$. The temporal spiking graph neural network finally outputs the spiking features $O_2 \in \mathbb{R}^{K \times N}$ and predicts the label vector $O_2' \in \mathbb{R}^K$ by averaging O_2 over the spatial domain N,

$$O_2' = \frac{1}{N} \sum_l^N O_2(l), \tag{16}$$

where $O_2(l) \in \mathbb{R}^K$. To fuse the predictions from these two networks, we take the *mean* or element-wise *max* of these two label vectors O_1' and O_2' and obtain the final predicted label vector $O' \in \mathbb{R}^K$. The predicted label is associated with the neuron with the largest value.

3.3. Implementations

In this section, we first introduce the location orders to enable the spatial recurrent neuronal dynamics of location spiking neurons. Then, we present the implementation details and timestep-wise inference algorithms for the proposed models.

3.3.1. Location orders

To enable the spatial recurrent neuronal dynamics of location spiking neurons, we need to manually set the location orders of location spiking neurons. Specifically, we propose four kinds of location orders for event-driven tactile learning and explore their robustness on the event-driven tactile tasks. As shown in Figure 5, three location orders are designed based on the major fingerprint patterns of humans – arch, whorl, and loop. And one location order randomly traverses all the taxels. Four concrete examples are shown below. Each number in the brackets represents the taxel index.

- An example for the arch-like location order: [11, 25, 35, 4, 18, 30, 7, 2, 20, 37, 29, 12, 9, 33, 23, 16, 1, 6, 15, 21, 27, 34, 39, 24, 17, 10, 31, 38, 28, 14, 3, 22, 32, 8, 19, 36, 5, 13, 26]
- An example for the whorl-like location order: [21, 15, 16, 23, 27, 24, 17, 6, 9, 12, 20, 29, 33, 34, 31, 28, 22, 14, 10, 1, 2, 7, 18, 30, 37, 39, 38, 32, 19, 8, 3, 4, 11, 25, 35, 36, 26, 13, 5]
- An example for the loop-like location order: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]
- An example for the random location order: [4, 7, 12, 9, 2, 1, 6, 15, 10, 3, 5, 8, 14, 17, 21, 22, 13, 26, 19, 24, 27, 28, 32, 36, 38, 31, 34, 39, 37, 33, 23, 29, 30, 35, 25, 11, 18, 20, 16].

3.3.2. Hybrid_SRM_FC

Similar to the spike-count loss of prior works (Shrestha and Orchard, 2018; Taunyazoz et al., 2020), we propose a location

spike-count loss to optimize the SNN with LSRM neurons:

$$\mathcal{L}_{LSRM} = \frac{1}{2} \sum_{k=0}^{K} \left(\sum_{l=0}^{N} o_k(l) - \sum_{l=0}^{N} \hat{o}_k(l) \right)^2 , \quad (17)$$

which captures the difference between the observed output spike count $\sum_{l=0}^{N} o_k(l)$ and the desired spike count $\sum_{l=0}^{N} \hat{o}_k(l)$ across the K neurons. Moreover, to optimize the Hybrid_SRM_FC, we develop a weighted spike-count loss:
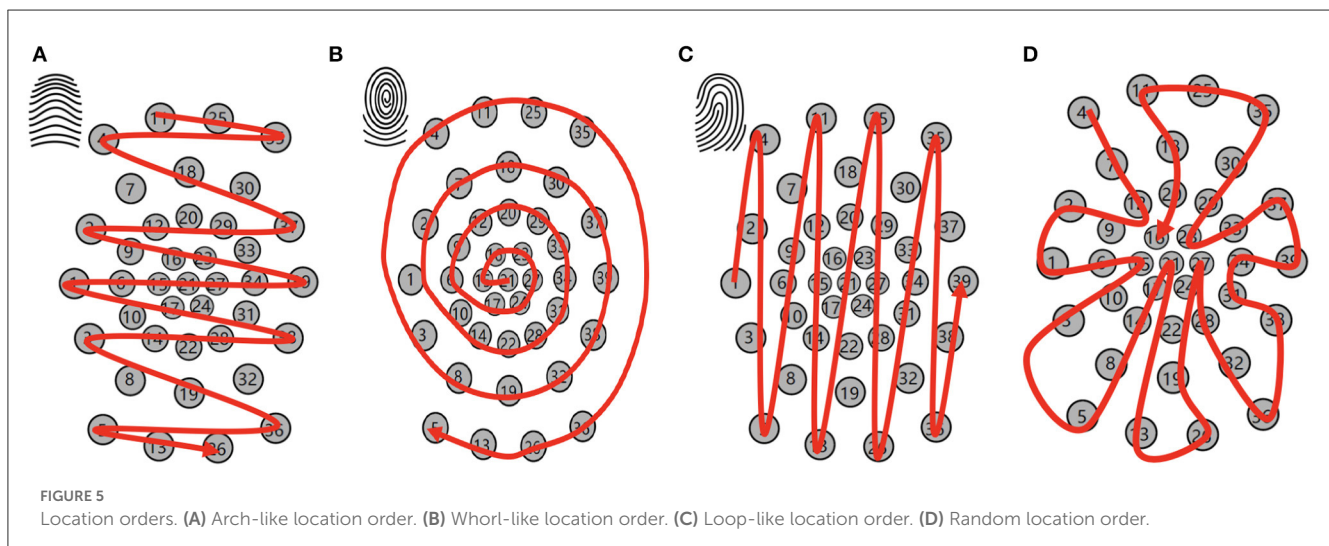
$$\mathcal{L}_1 = \frac{1}{2} \sum_{k=0}^{K} \left(\left(\sum_{t=0}^{T} o_k(t) + \lambda \sum_{l=0}^{N} o_k(l) \right) - \sum_{c=0}^{T+N} \hat{o}_k(c) \right)^2 , \quad (18)$$

which first balances the contributions from two SNNs and then captures the difference between the observed balanced output spike count $\sum_{t=0}^{T} o_k(t) + \lambda \sum_{l=0}^{N} o_k(l)$ and the desired spike count $\sum_{c=0}^{T+N} \hat{o}_k(c)$ across the K output neurons. For both $\mathcal{L}_{LSRM}$ and $\mathcal{L}_1$, the desired spike counts have to be specified for the correct and incorrect classes and are task-dependent hyperparameters. We set these hyperparameters as in Taunyazoz et al. (2020). To overcome the non-differentiability of spikes and apply the backpropagation algorithm, we use the approximate gradient proposed in SLAYER (Shrestha and Orchard, 2018). Moreover, based on the SLAYER's weight update in the temporal domain, we can derive the weight update for the SNNs with LSRM neurons in the spatial domain. Please check more details in our Github repository.

To demonstrate the applicability of our model to the spike-based temporal data, we propose the timestep-wise inference algorithm of the Hybrid_SRM_FC, which is shown in Algorithm 1. The corresponding timestep-wise training algorithm can be derived by incorporating the weighted spike-count loss.

3.3.3. Hybrid_LIF_GNN

To train the Hybrid_LIF_GNN, we define the loss function that captures the mean squared error between the ground truth label

FIGURE 5
Location orders. **(A)** Arch-like location order. **(B)** Whorl-like location order. **(C)** Loop-like location order. **(D)** Random location order.

```
Input: event-based tactile inputs X_in  ∈   ℝ^(N×T), N
    taxels, and the total time length T.
Output: timestep-wise predictions of O_1, O_2, and O.
1: for t ← 1 to T do
2:     obtain X ∈ ℝ^(N×t)
3:     obtain X̄' = concatenate(X', 0) ∈ ℝ^(T×N), where X' ∈ ℝ^(t×N),
    and 0 ∈ ℝ^((T-t)×N)
4:     O_1(t) = 0 ∈ ℝ^(K×t),  O_2(t) = 0 ∈ ℝ^(K×N)
5:     O(t) = 0 ∈ ℝ^(K×(t+N))
6:     O_1(t) = SNN_TSRM(X)          ▷ SNN_TSRM for the
    fully-connected SNN with TSRM neurons
7:     O_2(t) = SNN_LSRM(X̄')         ▷ SNN_LSRM for the
    fully-connected SNN with LSRM neurons
8:     O(t) = concatenate(O_1(t), O_2(t))
9: end for
```

Algorithm 1. Timestep-wise inference algorithm of the Hybrid_SRM_FC, adopted from Kang et al. (2022).

```
Input: event-based tactile inputs X_in  ∈   ℝ^(N×T), N
    taxels, and the total time length T
Output: timestep-wise label vectors of O'_1, O'_2, and O'
1: for t ← 1 to T do
2:     form t tactile spatial graphs G_s with X ∈ ℝ^(N×t)
3:     obtain X̄' = concatenate(X', 0) ∈ ℝ^(T×N), where X' ∈ ℝ^(t×N),
    and 0 ∈ ℝ^((T-t)×N)
4:     form N tactile temporal graphs G_t with X̄'
5:     O'_1(t), O'_2(t), O'(t) = 0 ∈ ℝ^K
6:     for i ← 1 to t do
7:         O'_1(t) += SSGNN(G_s(i))    ▷ SSGNN for the spatial
    spiking graph neural network
8:     end for
9:     O'_1(t) /= t
10:    for j ← 1 to N do
11:        O'_2(t) += TSGNN(G_t(j))    ▷ TSGNN for the temporal
    spiking graph neural network
12:    end for
13:    O'_2(t) /= N
14:    O'(t) = mean(O'_1(t), O'_2(t))    ▷ max can be used
15: end for
```

Algorithm 2. Timestep-wise inference algorithm of the Hybrid_LIF_GNN.

vector y and the final predicted label vector O'.

$$\mathcal{L}_2 = \|y - O'\|^2. \tag{19}$$

We utilize the spatio-temporal backpropagation (Wu et al., 2018) to derive the weight update for the SNNs with LLIF neurons. Moreover, to overcome the non-differentiability of spikes, we use the rectangular function (Wu et al., 2018) to approximate the derivative of the spike function (Heaviside function) in Equations 13, 14. Please check more implementation details in our Github repository. Algorithm 2 presents the timestep-wise inference algorithm of the Hybrid_LIF_GNN.

4. Experiments

We extensively evaluate our proposed models and demonstrate their effectiveness and efficiency on event-driven tactile learning, including event-driven tactile object recognition and event-driven slip detection. Specifically, we first conduct experiments on the Hybrid_SRM_FC to show that location spiking neurons can improve event-driven tactile learning. Then, we utilize the experiments on the Hybrid_LIF_GNN to show that location spiking neurons are user-friendly and can be incorporated into more powerful spike-based learning frameworks to further boost event-driven tactile learning. The source code and experimental configuration details are available at: https://github.com/pkang2017/TactileLSN.

4.1. Hybrid_SRM_FC

In this section, we first introduce the datasets and models for the experiments. Next, to show the effectiveness of the Hybrid_SRM_FC, we extensively evaluate it on the benchmark datasets and compare it with state-of-the-art models. Finally, we demonstrate the superior energy efficiency of the Hybrid_SRM_FC over the counterpart ANNs and show the high-efficiency benefit of LSRM neurons. We implement our models using slayerPytorch[2] and employ RMSProp with the l_2 regularization to optimize them.

4.1.1. Datasets

We use the datasets collected by NeuTouch (Taunyazoz et al., 2020), including "Objects-v1" and "Containers-v1" for event-driven tactile object recognition and "Slip Detection" for event-driven slip detection. Unlike "Objects-v1" which only requires models to determine the type of objects being handled, "Containers-v1" asks models about the type of containers being handled and the amount of liquid (0, 25, 50, 75, and 100%) held within. Thus, "Containers-v1" is more challenging for event-driven tactile object recognition. Moreover, the task of event-driven slip detection is also challenging since it requires models to detect the rotational slip within a short time, like 0.15 s for "Slip detection." We provide more details about the datasets in the Supplementary material. Following the experimental setting of Taunyazoz et al. (2020), we split the data into a training set (80%) and a testing set (20%), repeat each experiment for five rounds, and report the average accuracy.

4.1.2. Comparing models

We compare our model with the state-of-the-art SNN methods for event-driven tactile learning, including Tactile-SNN (Taunyazoz et al., 2020) and TactileSGNet (Gu et al., 2020). Tactile-SNN employs TSRM neurons as the building blocks, and the network structure of Tactile-SNN is the same as the fully-connected SNN with TSRM neurons in the Hybrid_SRM_FC. TactileSGNet

---

2 https://github.com/bamsumit/slayerPytorch

TABLE 2 Accuracies on benchmark datasets for the Hybrid_SRM_FC.

Method	Type	Objects-v1	Containers-v1	Slip detection
Tactile-SNN (Taunyazoz et al., 2020)	SNN	0.75	0.57*	0.82*
TactileSGNet (Gu et al., 2020)	SNN	0.79	0.58	0.97
GRU-MLP (Taunyazoz et al., 2020)	ANN	0.72	0.46*	0.87*
CNN-3D (Taunyazoz et al., 2020)	ANN	0.90	0.67*	0.44*
Hybrid_SRM_FC	SNN	**0.91**	**0.86**	**1.0**

*These values come from Taunyazoz et al. (2020). The best performance is in bold.

TABLE 3 Ablation studies on the Hybrid_SRM_FC.

Method	Type	Objects-v1	Containers-v1	Slip detection
Tactile-SNN (Taunyazoz et al., 2020)	SNN	0.75	0.57	0.82
Location Tactile-SNN	SNN	0.89	0.88	0.82
Hybrid_SRM_FC $\lambda = 1$	SNN	0.91	0.86	1.0
Hybrid_SRM_FC $\lambda = 0.5$	SNN	0.92	0.89	0.98
Hybrid_SRM_FC-loop	SNN	0.91	0.86	1.0
Hybrid_SRM_FC-arch	SNN	0.91	0.86	0.99
Hybrid_SRM_FC-whorl	SNN	0.92	0.86	0.98
Hybrid_SRM_FC-random	SNN	0.91	0.86	0.99

utilizes TLIF neurons as the building blocks and the network structure of TactileSGNet is the same as the spatial spiking graph neural network in the Hybrid_LIF_GNN. As in Taunyazoz et al. (2020), we also compare our model against conventional deep learning, specifically Gated Recurrent Units [GRUs; Cho et al. (2014)] with Multi-layer Perceptrons (MLPs) and 3D convolutional neural networks [3D_CNN; Gandarias et al. (2019)]. The network structure of GRU-MLP is Input-GRU-MLP, where MLP is only utilized at the final time step. And the network structure of CNN-3D is Input-3D_CNN1-3D_CNN2-FC, where FC is for the fully-connected layer.

4.1.3. Basic performance

Table 2 presents the test accuracies on the three datasets. We observe that the Hybrid_SRM_FC significantly outperforms the state-of-the-art SNNs. The reason why our model is superior to other SNNs could be 2-fold: (1) different from state-of-the-art SNNs that only extract features with existing spiking neurons, our model employs an SNN with location spiking neurons that enhance the representative ability and enable the model to extract features in a novel way; (2) our model fuses the SNN with TSRM neurons and the SNN with LSRM neurons to better capture complex spatio-temporal dependencies in the data. We also compare our model with ANNs, which provide fair comparison baselines for fully ANN architectures since they employ similar lightsome network architectures as ours. From Table 2, we find out that our model outperforms the counterpart ANNs on the three tasks, which might be because our model is more compatible with event-based tactile data and better maintains the sparsity to prevent overfitting.

4.1.4. Ablation studies

To examine the effectiveness of each component in the proposed model and validate the representation ability of location spiking neurons on event-driven tactile learning, we separately train the SNN with TSRM neurons (which is exactly Tactile-SNN) and the SNN with LSRM neurons (which is referred to as Location Tactile-SNN). From Table 3, we surprisingly find out that Location Tactile-SNN significantly surpasses Tactile-SNN on the datasets for event-driven tactile object recognition and provides comparable performance on event-driven slip detection. The reason for this could be 2-fold: (1) the time durations of event-driven tactile object recognition datasets are longer than that of "Slip detection," and Location Tactile-SNN with LSRM neurons is good at capturing the mid-and-long term dependencies in these object recognition datasets; (2) like Tactile-SNN, Location Tactile-SNN with LSRM neurons can still capture the spatial dependencies in the event-driven tactile data ("Slip detection") due to the spatial recurrent neuronal dynamics of location spiking neurons. Moreover, we examine the sensitivity of λ in Equation 18 and the robustness of location orders. From Table 3, we notice the results of related models are close, proving that the λ tuning and location orders do not significantly impact task performance.

4.1.5. Timestep-wise inference

We evaluate the timestep-wise inference performance of the Hybrid_SRM_FC and validate the contributions of the two components in it. Moreover, we propose a time-weighted Hybrid_SRM_FC to better balance the two components' contributions and achieve better overall performance. Figures 6A–C show the timestep-wise inference accuracies of
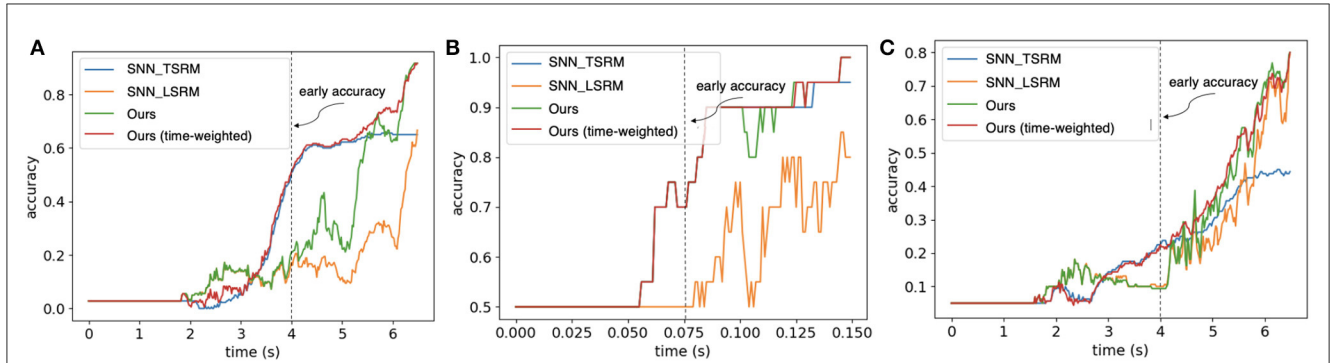
FIGURE 6
The timestep-wise inference (Algorithm 1) for the SNN with TSRM neurons (SNN_TSRM), the SNN with LSRM neurons (SNN_LSRM), the Hybrid_SRM_FC, and the time-weighted Hybrid_SRM_FC on **(A)** "Objects-v1," **(B)** "Slip Detection," and **(C)** "Containers-v1." Please note that we use the same event sequences as Taunyazoz et al. (2020) and the first spike occurs at around 2.0 s for "Objects-v1" and "Containers-v1." From the figure, we can see that the models with location spiking neurons have not reached the saturated levels while the blue line (the models with only traditional spiking neurons) has already reached the saturated levels. This demonstrates the potential of location spiking neurons and the models with location spiking neurons could provide the better performance by increasing the time on these tasks.

the SNN with TSRM neurons, the SNN with LSRM neurons, the Hybrid_SRM_FC, and the time-weighted Hybrid_SRM_FC on the three datasets. Specifically, the output of the time-weighted Hybrid_SRM_FC at time t is

$$O_{tw}(t) = concatenate((1 - \omega) * O_1(t), \omega * O_2(t)),$$
$$\omega = \frac{1}{1 + e^{-\psi * (\frac{t}{T} - 1)}}, \quad (20)$$

where the hyperparameter ψ balances the contributions of the two components in the hybrid model and T is the total time length. From the figures, we can see that the SNN with TSRM neurons has good "early" accuracies on the three tasks since it well captures the spatial dependencies with the help of Equation 11. However, its accuracies do not improve too much at the later stage since it does not sufficiently capture the temporal dependencies. In contrast, the SNN with LSRM neurons has fair "early" accuracies, while its accuracies jump a lot at the later stage since it models the temporal dependencies in Equation 12. The Hybrid_SRM_FC adopts the advantages of these two components and extracts spatio-temporal features from various views, which enables it to have a better overall performance. Furthermore, after employing the time-weighted output and shifting more weights to the SNN with TSRM neurons at the early stage, the time-weighted Hybrid_SRM_FC can have a good "early" accuracy as well as an excellent "final" accuracy.

4.1.6. Energy efficiency

To further analyze the benefits of the proposed model and location spiking neurons, we estimate the gain in computational costs compared to fully ANN architectures. Typically, the number of synaptic operations is used as a metric for benchmarking the computational energy of SNN models (Lee et al., 2020; Xu et al., 2021). In addition, we can estimate the total energy consumption of a model based on CMOS technology (Horowitz, 2014).

Different from ANNs that always conduct real-valued matrix-vector multiplication operations without considering the sparsity of inputs, SNNs carry out event-based computations only at the

arrival of input spikes. Hence, we first measure the mean spiking rate of layer l in our proposed model. Specifically, the mean spiking rate of the layer l in the SNN with existing spiking neurons is given by:

$$F_1^{(l)} = \frac{1}{T} \sum_{t \in T} \frac{\#\text{spikes of layer } l \text{ at time } t}{\#\text{neurons of layer } l}, \quad (21)$$

where T is the total time length. And the mean spiking rate of the layer l in the SNN with location spiking neurons is given by:

$$F_2^{(l)} = \frac{1}{N} \sum_{n \in N} \frac{\#\text{spikes of layer } l \text{ at location } n}{\#\text{neurons of layer } l}, \quad (22)$$

where N is the total number of locations. We show the mean spiking rates of Hybrid_SRM_FC layers in the Supplementary material. With the mean spiking rates, we can estimate the number of synaptic operations in the SNNs. Given M is the number of neurons, C is the number of synaptic connections per neuron, and F indicates the mean spiking rate, the number of synaptic operations at each time or location in layer l is calculated as $M^{(l)} \times C^{(l)} \times F^{(l)}$, where $F^{(l)}$ is $F_1^{(l)}$ or $F_2^{(l)}$. Thus, the total number of synaptic operations in our hybrid model is calculated by:

$$OP_{Hybrid} = \sum_l M^{(l)} \times C^{(l)} \times F_1^{(l)} \times T + \sum_{l'} M^{(l')} \times C^{(l')} \times F_2^{(l')} \times N, \quad (23)$$

where l is the spiking layer with existing spiking neurons and l' is the spiking layer with location spiking neurons. Generally, the total number of synaptic operations in the ANNs is $\sum_l M^{(l)} \times C^{(l)}$. Based on these, we estimate the number of synaptic operations in the Hybrid_SRM_FC and ANNs like the GRU-MLP and CNN-3D. As shown in Table 4, all the SNNs achieve far fewer operations than ANNs on the three datasets.

Moreover, due to the binary nature of spikes, SNNs perform only accumulation (AC) per synaptic operation, while ANNs perform the multiply-accumulate (MAC) computations since the operations are real-valued. In general, AC computation is considered to be significantly more energy-efficient than MAC. For

TABLE 4 The number of synaptic operations ($\#op$, $\times 10^6$) and the compute-energy benefit (the compute-energy of ANNs / the compute-energy of SNNs, 45 nm) on benchmark datasets for the Hybrid_SRM_FC.

Method	Type	Objects-v1	Containers-v1	Slip detection
$\#op$ GRU-MLP	ANN	5.89	5.89	2.72
$\#op$ CNN-3D	ANN	4.17	4.07	1.75
$\#op$ SNN with TSRM neurons	SNN	0.31	0.42	0.022
Compute-energy benefit		68.60–96.90×	49.42–71.52×	405.68–630.55×
$\#op$ SNN with LSRM neurons	SNN	0.29	0.41	0.023
Compute-energy benefit		73.33–103.58×	50.63–73.27×	388.04–603.13×
$\#op$ Hybrid_SRM_FC	SNN	0.60	0.83	0.045
Compute-energy benefit		35.45–50.07×	25.01–36.19×	198.33–308.27×

example, an AC is reported to be 5.1× more energy-efficient than a MAC in the case of 32-bit floating-point numbers [45 nm CMOS process; Horowitz (2014)]. Based on this principle, we obtain the computational energy benefits of SNNs over ANNs in Table 4. From the table, we can see that the SNN models are 10× to 100× more energy-efficient than ANNs and the location spiking neurons (LSRM neurons) have the similar energy efficiency compared to existing spiking neurons (TSRM neurons).

These results are consistent with the fact that the sparse spike communication and event-driven computation underlie the efficiency advantage of SNNs and demonstrate the potential of our model and location spiking neurons on neuromorphic hardware.

4.2. Hybrid_LIF_GNN

In this section, to show the usability of location spiking neurons and further boost event-driven tactile learning, we conduct a series of experiments with the Hybrid_LIF_GNN, which is powered by the popular spike-based learning framework – STBP (Wu et al., 2018). Specifically, we first compare our model with the state-of-the-art models with TLIF neurons and GNN structures. Then, we conduct several ablation studies to examine the effectiveness of some designs in the Hybrid_LIF_GNN. Next, we demonstrate the superior energy efficiency of our model over the counterpart Graph Neural Networks (GNNs) and show the high-efficiency benefits of location spiking neurons. Finally, we compare with the Hybrid_SRM_FC on the same benchmark datasets to validate the superiority of the Hybrid_LIF_GNN.[3]

4.2.1. Datasets

To fairly compare with other published models with TLIF neurons (Gu et al., 2020), we evaluate the Hybrid_LIF_GNN on "Objects-v0" and "Containers-v0." These two datasets are the initial versions of "Objects-v1" and "Containers-v1." We demonstrate their differences in the Supplementary material. To show the superiority of the Hybrid_LIF_GNN on event-driven tactile learning, we compare it with the Hybrid_SRM_FC on "Objects-v1," "Containers-v1," and "Slip detection." During the

experiments, we split the data into a training set (80%) and a testing set (20%) with an equal class distribution. We repeat each experiment for five rounds and report the average accuracy.

4.2.2. Comparing models

We compare the Hybrid_LIF_GNN with the state-of-the-art methods with TLIF neurons and GNN structures (Gu et al., 2020) on event-based tactile object recognition. Specifically, we compare the TactileSGNet series. The general network structure is the same as the spatial spiking graph neural network, which is Input-Spiking TAGConv-Spiking FC1-Spiking FC2-Spiking FC3. The other models in the series are obtained by substituting the Spiking TAGConv layer:

- TactileSGNet-MLP, which uses the Spiking FC layer with TLIF neurons to process the input. The network structure is Input-Spiking FC0-Spiking FC1-Spiking FC2-Spiking FC3.
- TactileSGNet-CNN, which takes the network structure of Input-Spiking CNN-Spiking FC1-Spiking FC2-Spiking FC3. The tactile input is organized in a grid structure according to the spatial distribution of taxels, and the Spiking CNN with TLIF neurons is utilized to extract features from this grid.
- TactileSGNet-GCN, where the graph convolutional network (GCN) is used as the GNN in Equation 13. The network structure is Input-Spiking GCN-Spiking FC1-Spiking FC2-Spiking FC3.

Moreover, we also compare the Hybrid_LIF_GNN against fully GNNs. Specifically, the GNNs have the same network structures as the Hybrid_LIF_GNN, including one recurrent TAGConv-FC1-FC2-FC3 for T tactile spatial graphs, one recurrent TAGConv-FC1-FC2-FC3 for N tactile temporal graphs, and one fusion module to fuse the predictions from two branches. The major difference between our model and GNNs is that GNNs employ artificial neurons and adopt different activation functions in Equations 13, 14 while our model utilizes the spiking neurons and takes the Heaviside function as the activation function.

4.2.3. Basic performance

We report the test accuracies on the two event-driven tactile object recognition datasets in Table 5. From this table, we can

---

3   In this section, to be consistent with Gu et al. (2020), we use accuracies (%).

TABLE 5 Accuracies (%) on datasets for the Hybrid_LIF_GNN.

Method	Type	Objects-v0	Containers-v0
TactileSGNet-MLP (Gu et al., 2020)	SNN	85.97*	58.83*
TactileSGNet-CNN (Gu et al., 2020)	SNN	88.40*	60.17*
TactileSGNet-GCN (Gu et al., 2020)	SNN	85.14*	58.83*
TactileSGNet-TAGConv (Gu et al., 2020)	SNN	89.44*	64.17*
Recurrent GNN-linear	GNN	92.36	70.67
Recurrent GNN-elu	GNN	91.11	74.67
Recurrent GNN-LeakyRelu	GNN	89.31	73.00
Hybrid_LIF_GNN-sparse-mean	SNN	**93.33**	**79.33**
Hybrid_LIF_GNN-dense-mean	SNN	92.50	78.67
Hybrid_LIF_GNN-sparse-max	SNN	85.56	77.00
Hybrid_LIF_GNN-dense-max	SNN	85.14	76.00

*These values come from Gu et al. (2020). All the Hybrid_LIF_GNN models use the loop-like location order. "Sparse" is for "sparse tactile temporal graph," "dense" is for "dense tactile temporal graph," "mean" is for "mean fusion," and "max" is for "max fusion." The best performance is in bold.

see that the Hybrid_LIF_GNN significantly outperforms the TactileSGNet series (Gu et al., 2020). The reason why our model can achieve the better performance could be 2-fold: (1) different from the TactileSGNet models that only utilize TLIF neurons to extract features from the tactile spatial graphs, our model also employs the temporal spiking graph neural network with LLIF neurons to extract features from the tactile temporal graphs; (2) our model fuses the spatial and temporal spiking graph neural networks to capture complex spatio-temporal dependencies in the data. We also compare our model with fully GNNs by replacing the spike functions in Equations 13, 14 with activation functions, such as linear, elu, or LeakyRelu. These models provide fair comparison baselines for fully GNN architectures since they employ the same network architecture as ours. From Table 5, we observe that the Hybrid_LIF_GNN outperforms the counterpart GNNs on the two datasets, which might be because our model is more compatible with event-based tactile data and better maintains the sparsity to prevent overfitting.

4.2.4. Ablation studies

We further provide ablation studies for exploring the optimal design choices. From Table 5, we find out that the combination of "sparse tactile temporal graph" and "mean fusion" performs better than other combinations. The reason for this could be 2-fold: (1) the dense tactile temporal graph involves too many insignificant temporal dependencies and does not differentiate the importance of each dependency; (2) the max fusion results in information loss.

4.2.5. Timestep-wise inference

Figure 7 shows the timestep-wise inference accuracies (%) for the spatial spiking graph neural network, the temporal spiking graph neural network, the Hybrid_LIF_GNN, and the time-weighted Hybrid_LIF_GNN on the two datasets. Specifically, the

output of time-weighted Hybrid_LIF_GNN at time t is

$$O'_{tw}(t) = O'_1(t)(1 - \frac{t}{\zeta T}) + O'_2(t)\frac{t}{\zeta T}, \qquad (24)$$

where ζ balances the contributions of the two components in the hybrid model and T is the total time length. From the figure, we can see that the spatial spiking graph neural network has a good "early" accuracy with the help of tactile spatial graphs, while its accuracy does not improve too much at the later stage since it cannot well capture the temporal dependencies. In contrast, the temporal spiking graph neural network has a fair "early" accuracy, while its accuracy jumps a lot at the later stage since it models the temporal dependencies explicitly. The Hybrid_LIF_GNN adopts the advantages of these two models and extracts spatio-temporal features from multiple views, which enables it to have a better overall performance. Furthermore, after employing the time-weighted output and setting $\zeta = 2$ to shift more weights to the spatial spiking graph neural network at the early stage, the time-weighted model can have a good "early" accuracy as well as an excellent "final" accuracy, see red lines in Figure 7.

4.2.6. Energy efficiency

Following the estimation methods in Section 4.1.6, we estimate the computational costs of the Hybrid_LIF_GNN and its counterpart GNNs on the benchmark datasets.

We show the mean spiking rates of Hybrid_LIF_GNN layers in the Supplementary material. Table 6 provides the number of synaptic operations conducted in the Hybrid_LIF_GNN and the counterpart GNNs with the same network structure. From the table, we can see that the SNNs achieve far fewer operations than GNNs on the benchmark datasets. Moreover, following the 45 nm CMOS technology energy principle in Section 4.1.6, we obtain the computational energy benefits of SNNs over GNNs in Table 6. From the table, we can see that the SNN models are 10× to 100× energy-efficient than GNNs. Furthermore, by
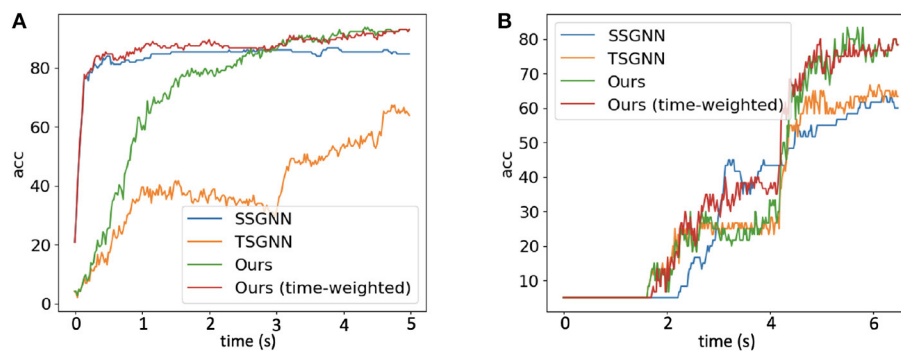
FIGURE 7
The timestep-wise inference (Algorithm 2) accuracies (%) for the spatial spiking graph neural network (SSGNN), the temporal spiking graph neural network (TSGNN), the Hybrid_LIF_GNN, and the time-weighted Hybrid_LIF_GNN on **(A)** "Objects-v0" and **(B)** "Containers-v0".

TABLE 6 The number of synaptic operations ($\#op$, $\times 10^8$) and the compute-energy benefit (the compute-energy of GNNs/the compute-energy of SNNs, 45 nm) on benchmark datasets for the Hybrid_LIF_GNN.

Method	Type	Objects-v0	Containers-v0
$\#op$ Recurrent GNNs in Table 5	GNN	1.7188	2.2146
$\#op$ Spatial spiking graph neural network	SNN	0.1132	0.1023
Compute-energy benefit		77.44×	110.41×
$\#op$ Temporal spiking graph neural network	SNN	0.0297	0.0313
Compute-energy benefit		295.15×	360.85×
$\#op$ Hybrid_LIF_GNN	SNN	0.1429	0.1336
Compute-energy Benefit		61.34×	84.54×

comparing the number of synaptic operations in the spatial spiking graph neural network with that in the temporal spiking graph neural network, we find that the temporal spiking graph neural network has the higher energy efficiency. The reason for this could be that we employ the sparse tactile temporal graphs in the temporal spiking graph neural network and such graphs require fewer operations.

These results are consistent with what we show in Section 4.1.6 and demonstrate the potential of our models and location spiking neurons (LLIF neurons) on neuromorphic hardware.

4.2.7. Performance comparison with the hybrid_SRM_FC

To fairly compare with the Hybrid_SRM_FC (Figure 2), we further test the Hybrid_LIF_GNN (Figure 4) on "Objects-v1," "Containers-v1," and "Slip detection." From Table 7, we can see that the Hybrid_LIF_GNN outperforms the Hybrid_SRM_FC on "Objects-v1" and "Containers-v1" and they both achieve the perfect slip detection. The reason for

this is that the Hybrid_LIF_GNN adopts graph topologies and has a more complicated structure than the Hybrid_SRM_FC. Such comparison results are consistent with the comparison between the Tactile-SNN and TactileSGNet in Table 2 and demonstrate the benefit of spiking graph neural networks and complex structures on event-driven tactile learning. Through this experiment, we show that the location spiking neurons can be incorporated into complex spike-based learning frameworks and further boost the performance of event-driven tactile learning.

5. Discussion and conclusion

In this section, we discuss the advantages and limitations of conventional spiking neurons and location spiking neurons. Moreover, we provide preliminary results of the location spiking neurons on event-driven audio learning and discuss the potential impact of this work on broad spike-based learning applications. Finally, we conclude the paper.

5.1. Advantages and limitations of conventional and location spiking neurons

This paper proposes location spiking neurons. Based on the neuronal dynamic equations of conventional spiking neurons and location spiking neurons, we can see that both of them can extract spatio-temporal dependencies from the data. Specifically, the conventional spiking neurons employ the temporal recurrent neural dynamics to update their membrane potentials and capture spatial dependencies by aggregating the information from presynaptic neurons, see Equations 2, 5, 11, and 13. However, location spiking neurons use spatial recurrent neural dynamics to update their potentials and model temporal dependencies by aggregating the information from presynaptic neurons, see Equations 7, 10, 12, and 14.
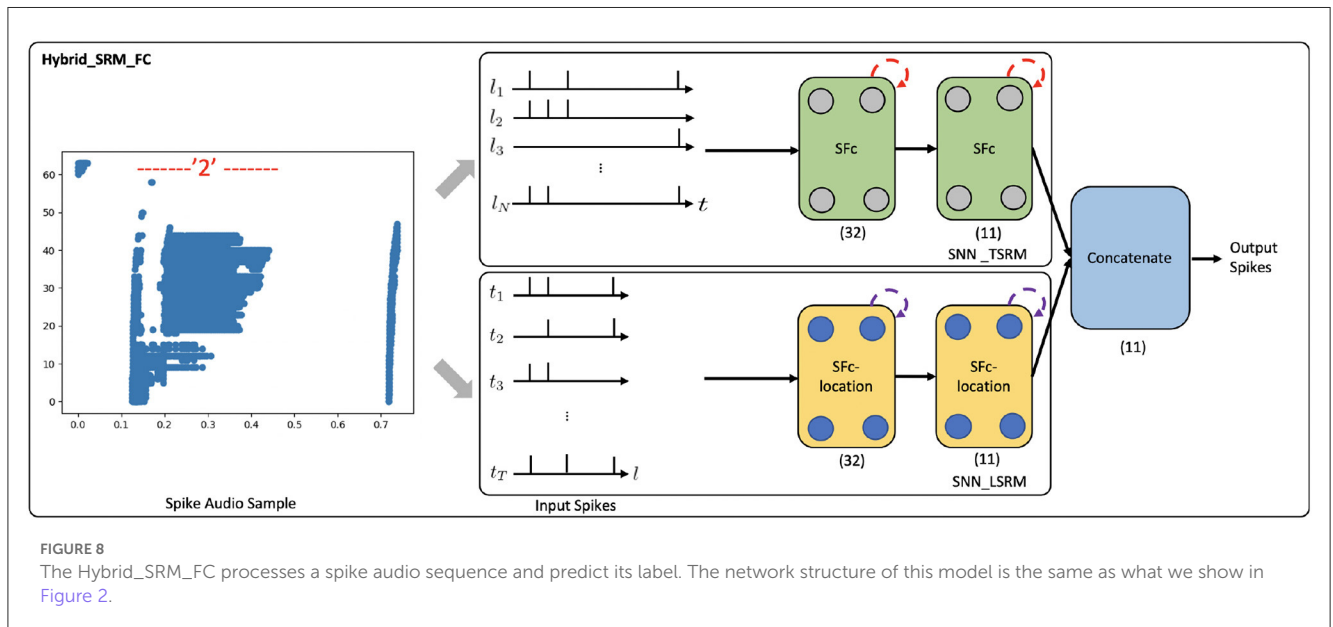
Moreover, based on experimental results, we can see that conventional spiking neurons are better at capturing spatial

TABLE 7 Performance comparison between the Hybrid_SRM_FC with LSRM neurons and the Hybrid_LIF_GNN with LLIF neurons.

Method	Type	Objects-v1	Containers-v1	Slip detection
Hybrid_SRM_FC	SNN	0.91	0.86	**1.0**
Hybrid_LIF_GNN[‡]	SNN	**0.96**	**0.90**	1.0

[‡]Represents Hybrid_LIF_GNN-sparse-mean-loop. The best performance is in bold.

FIGURE 8
The Hybrid_SRM_FC processes a spike audio sequence and predict its label. The network structure of this model is the same as what we show in Figure 2.

dependencies which benefit the "early" accuracy, while location spiking neurons are better at modeling mid-and-long temporal dependencies which benefit the "late" accuracy. Networks built only with conventional spiking neurons or networks built only with location spiking neurons cannot sufficiently capture spatio-temporal dependencies in the event-based data. Thus, we always concatenate or fuse the networks to sufficiently capture spatio-temporal dependencies in the data.

By introducing LSRM neurons and LLIF neurons, we verify that the idea of location spiking neurons can be applied to various existing spiking neuron models like TSRM neurons and TLIF neurons and strengthen their feature representation abilities. Moreover, we extensively evaluate the models built with these novel neurons and demonstrate their superior performance and energy efficiency. Furthermore, by comparing the Hybrid_LIF_GNN with the Hybrid_SRM_FC, we show that the location spiking neurons can be utilized to build more complicated models to further improve task performance.

5.2. Potential impact on broad spike-based learning applications

In this paper, we focus on boosting event-driven tactile learning with location spiking neurons. And extensive experimental results validate the effectiveness and efficiency of our models on the tasks. Besides event-driven tactile learning, we can also apply the models with location spiking neurons to other spike-based learning applications.

5.2.1. Event-driven audio learning

To show the potential impact of our work, we apply the Hybrid_SRM_FC (see Figure 2) to event-driven audio learning and provide preliminary results. Please note that the objective of this experiment is not necessarily to obtain state-of-the-art results on event-driven audio learning, but to demonstrate that location spiking neurons can bring benefits to the model built with conventional spiking neurons on other spike-based learning applications.

In the experiment, we use the N-TIDIGITS18 dataset (Anumula et al., 2018), which is collected by playing the audio files from the TIDIGITS dataset (Leonard and Doddington, 1993) to the dynamic audio sensor–the CochleaAMS1b sensor (Chan et al., 2007). The dataset includes both single digits and connected digit sequences. We use the single-digit part of the dataset, which consists of 11 categories, including "oh," "zero," and digits "1–9." A spike audio sequence of digit "2" is shown in Figure 8, where the x-axis indicates the event time, and the y-axis indicates the 64 frequency channels of the CochleaAMS1b sensor. Each blue dot in the sequence represents an event that occurs at time t_e and frequency f_e. In this application, we regard "frequency channels" as "locations" and apply the Hybrid_SRM_FC to process the spike audio inputs, see Figure 8. Through the experiments, the fully-connected SNN with TSRM neurons achieves the test accuracy of 0.563. However, with the help of LSRM neurons, the Hybrid_SRM_FC obtains the test accuracy of 0.586 and correctly classifies the additional 57 spike audio sequences. Moreover, we show the training and testing profiles of the fully-connected SNN with TSRM neurons and

the Hybrid_SRM_FC in the Supplementary material. From those figures, we can see that our hybrid model converges faster and attains a lower loss and a higher accuracy compared to the fully-connected SNN with TSRM neurons.

From this experiment, we can see that location spiking neurons can be applied to other spike-based learning applications. Moreover, the location spiking neurons can bring benefits to the models built with conventional spiking neurons and improve their task performance. We believe there will be further improvements on event-driven audio learning if we can incorporate the location spiking neurons into state-of-the-art event-driven audio learning frameworks.

5.2.2. Visual processing

Besides event-driven audio learning, a contemporary work (Li et al., 2022) also validates the effectiveness of spatial recurrent neuronal dynamics on conventional image classification. This work incorporates the spatial recurrent neuronal dynamics into the full-precision Multilayer Perceptron (MLP) and achieves the state-of-the-art top-1 accuracy on the ImageNet dataset. Since the model is full-precision and real-valued, it may lose the energy efficiency benefits of binary spikes. Our location spiking neurons employ the spatial recurrent neuronal dynamics but also keep the binary nature of spikes. Based on these, we think our proposed neurons could bring more potential to computer vision (e.g., event-based vision) when they are incorporated into MLP (Tolstikhin et al., 2021) or Transformer (Dosovitskiy et al., 2020) frameworks.

5.3. Conclusion

In this work, we propose a novel neuron model– "location spiking neuron." Specifically, we introduce two concrete location spiking neurons—the LSRM neurons and LLIF neurons. We demonstrate the spatial recurrent neuronal dynamics of these neurons and compare them with the conventional spiking neurons—the TSRM neurons and TLIF neurons. By exploiting these location spiking neurons, we develop two hybrid models for event-driven tactile learning to sufficiently capture the complex spatio-temporal dependencies in the event-based tactile data. The extensive experimental results on the event-driven tactile datasets demonstrate the extraordinary performance and high energy efficiency of our models and location spiking neurons. This could further unlock their potential on neuromorphic hardware. Overall, this work sheds new light on SNN representation learning and event-driven learning.

Data availability statement

Publicly available dataset were analyzed in the study. This data can be found here: https://clear-nus.github.io/visuotactile/download.html.

Author contributions

PK, AK, and OC brought up the core concept and architecture of this manuscript and wrote the paper. PK, SB, HC, AK, and OC designed the experiments and discussed the results. All authors contributed to the article and approved the submitted version.

Acknowledgments

Portions of this work Event-Driven Tactile Learning with Location Spiking Neurons (Kang et al., 2022) were accepted by IJCNN 2022 and orally presented at the IEEE WCCI in 2022.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2023.1127537/full#supplementary-material

References

Abbott, L. F. (1999). Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Res. Bullet.* 50, 303–304. doi: 10.1016/s0361-9230(99)00161-6

Anumula, J., Neil, D., Delbruck, T., and Liu, S.-C. (2018). Feature representations for neuromorphic audio spike streams. *Front. Neurosci.* 12, 23. doi: 10.3389/fnins.2018.00023

Baishya, S. S., and Bäuml, B. (2016). "Robust material classification with a tactile skin using deep learning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon: IEEE, 8–15.

Bullmore, E., and Sporns, O. (2012). The economy of brain network organization. *Nat. Rev. Neurosci.* 13, 336–349. doi: 10.1038/nrn3214

Calandra, R., Owens, A., Jayaraman, D., Lin, J., Yuan, W., Malik, J., et al. (2018). More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robot. Automat. Lett.* 3, 3300–3307. doi: 10.48550/arXiv.1805.11085

Cao, Y., Chen, Y., and Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *Int. J. Comput. Vis.* 113, 54–66. doi: 10.1007/s11263-014-0788-3

Chan, V., Liu, S.-C., and van Schaik, A. (2007). Aer ear: A matched silicon cochlea pair with address event representation interface. *IEEE Trans. Circuit. Syst. I* 54, 48–59. doi: 10.1109/ISCAS.2005.1465560

Cheng, X., Hao, Y., Xu, J., and Xu, B. (2020). "Lisnn: Improving spiking neural networks with lateral interactions for robust object recognition," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20.*Yokohama, 1519–1525.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint.* arXiv:1406.1078. doi: 10.48550/arXiv.1406.1078

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint.* arXiv:1511.07289. doi: 10.48550/arXiv.1511.07289

Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G. A. F., Joshi, P., et al. (2021). Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proc. IEEE* 109, 911–934. doi: 10.1109/JPROC.2021.3067593

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv Preprint.* arXiv:2010.11929. doi: 10.48550/arXiv.2010.11929

Du, J., Zhang, S., Wu, G., Moura, J. M., and Kar, S. (2017). Topology adaptive graph convolutional networks. *arXiv Preprint.* arXiv:1710.10370. doi: 10.48550/arXiv.1710.10370

Felleman, D. J., and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. cortex.* 1, 1–47. doi: 10.1093/cercor/1.1.1-a

Fishel, J. A., and Loeb, G. E. (2012). "Sensing tactile microvibrations with the biotac– comparison with human sensitivity," in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob).* (Rome: IEEE), 1122–1127.

Gallego, G., Delbruck, T., Orchard, G. M., Bartolozzi, C., Taba, B., Censi, A., et al. (2020). Event-based vision: A survey. *IEEE Trans. Pat. Anal. Machine Intell.* 2020, 8405. doi: 10.48550/arXiv.1904.08405

Gandarias, J. M., Pastor, F., García-Cerezo, A. J., and Gómez-de Gabriel, J. M. (2019). "Active tactile recognition of deformable objects with 3d convolutional neural networks," in *2019 IEEE World Haptics Conference (WHC).* Tokyo, 551–555. IEEE.

Gerstner, W. (1995). Time structure of the activity in neural network models. *Phys. Rev. E* 51, 738. doi: 10.1103/PhysRevE.51.738

Gerstner, W., and Kistler, W. M. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity.* Cambridge: Cambridge University Press.

Gu, F., Sng, W., Taunyazov, T., and Soh, H. (2020). "TactileSGNet: A spiking graph neural network for event-based tactile object recognition," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Las Vegas, NV: IEEE), 9876–9882.

Horowitz, M. (2014). "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (San Francisco, CA: IEEE), 10–14.

Kang, P., Banerjee, S., Chopp, H., Katsaggelos, A., and Cossairt, O. (2022). "Event-driven tactile learning with location spiking neurons," in *2022 International Joint Conference on Neural Networks (IJCNN)* (Padua: IEEE), 1–9.

Kappassov, Z., Corrales, J.-A., and Perdereau, V. (2015). Tactile sensing in dexterous robot hands. *Robot. Auton. Syst.* 74, 195–220. doi: 10.1016/j.robot.2015.07.015

Lee, C., Kosta, A. K., Zhu, A. Z., Chaney, K., Daniilidis, K., and Roy, K. (2020). "Spike-flownet: Event-based optical flow estimation with energy-efficient hybrid neural networks," in *European Conference on Computer Vision.* Springer, 366–382.

Leonard, R. G., and Doddington, G. (1993). *Tidigits Speech Corpus.* Dallas, TX: Texas Instruments, Inc.

Li, D., Chen, X., Becchi, M., and Zong, Z. (2016). "Evaluating the energy efficiency of deep convolutional neural networks on CPUs and GPUs," in *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom).* (Atlanta, GA: IEEE), 477–484.

Li, W., Chen, H., Guo, J., Zhang, Z., and Wang, Y. (2022). "Brain-inspired multilayer perceptron with spiking neurons," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (New Orleans, LA), 783–793.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the 30 th International Conference on Ma- chine Learning.* Atlanta, GA.

Maass, W., and Bishop, C. M. (2001). *Pulsed Neural Networks.* Cambridge, MA: MIT Press.

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642

Pfeiffer, M., and Pfeil, T. (2018). Deep learning with spiking neurons: Opportunities and challenges. *Front. Neurosci.* 12, 774. doi: 10.3389/fnins.2018.00774

Roy, K., Jaiswal, A., and Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature* 575, 607–617. doi: 10.1038/s41586-019-1677-2

Sanchez, J., Mateo, C. M., Corrales, J. A., Bouzgarrou, B.-C., and Mezouar, Y. (2018). "Online shape estimation based on tactile sensing and deformation modeling for robot manipulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Madrid: IEEE, 504–511.

Schmitz, A., Maggiali, M., Natale, L., Bonino, B., and Metta, G. (2010). "A tactile sensor for the fingertips of the humanoid robot iCub," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems.* (Taipei: IEEE), 2212–2217.

Sengupta, A., Ye, Y., Wang, R., Liu, C., and Roy, K. (2019). Going deeper in spiking neural networks: VGG and residual architectures. *Front. Neurosci.* 13, 95. doi: 10.3389/fnins.2019.00095

Shrestha, S. B., and Orchard, G. (2018). "SLAYER: Spike layer error reassignment in time," in *Advances in Neural Information Processing Systems 31*, eds S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (New York, Ny: Curran Associates, Inc.), 1419–1428.

Soh, H., and Demiris, Y. (2014). Incrementally learning objects by touch: Online discriminative and generative models for tactile-based recognition. *IEEE Trans. Hapt.* 7, 512–525. doi: 10.1109/TOH.2014.2326159

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. *arXiv Preprint.* arXiv:1906.02243. doi: 10.48550/arXiv.1906.02243

Taunyazov, T., Chua, Y., Gao, R., Soh, H., and Wu, Y. (2020). "Fast texture classification using tactile neural coding and spiking neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* (Las Vegas, NV: IEEE), 9890–9895.

Taunyazov, T., Koh, H. F., Wu, Y., Cai, C., and Soh, H. (2019). "Towards effective tactile identification of textures using a hybrid touch approach," in *2019 International Conference on Robotics and Automation (ICRA).* (Montreal, QC: IEEE), 4269–4275.

Taunyazov, T., Song, L. S., Lim, E., See, H. H., Lee, D., Tee, B. C., et al. (2021). Extended tactile perception: Vibration sensing through tools and grasped objects. *arXiv Preprint.* arXiv:2106.00489. doi: 10.48550/arXiv.2106.00489

Taunyazoz, T., Sng, W., See, H. H., Lim, B., Kuan, J., Ansari, A. F., et al. (2020). "Event-driven visual-tactile sensing and learning for robots," in *Proceedings of Robotics: Science and Systems.*

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., et al. (2021). MLP-mixer: An all-MLP architecture for vision. *Adv. Neural Inform. Process. Syst.* 34, 24261–24272. doi: 10.48550/arXiv.2105.01601

Wu, Y., Deng, L., Li, G., Zhu, J., and Shi, L. (2018). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* 12, 331. doi: 10.3389/fnins.2018.00331

Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv Preprint.* arXiv:1505.00853. doi: 10.48550/arXiv.1505.00853

Xu, M., Wu, Y., Deng, L., Liu, F., Li, G., and Pei, J. (2021). "Exploiting spiking dynamics with spatial-temporal feature normalization in graph learning," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, ed Z. H. Zhou (Montreal, QC), 3207–3213.