

Event-Driven Tactile Learning with Various Location Spiking Neurons

Peng Kang*, Srutarshi Banerjee†, Henry Chopp†, Aggelos Katsaggelos*†, Oliver Cossairt*†

*Department of Computer Science, Northwestern University, Evanston, IL 60208, USA

†Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208, USA

Email: {pengkang2022, SrutarshiBanerjee2022, henrychopp2017}@u.northwestern.edu,
{a-katsaggelos, oliver.cossairt}@northwestern.edu

Abstract—Tactile sensing is essential for a variety of daily tasks. New 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 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) model¹. By exploiting the novel location spiking neurons, we propose several models to capture the complex spatio-temporal dependencies in the event-driven tactile data. Extensive experiments demonstrate the significant improvements of our models over other works on event-driven tactile learning and show the superior energy efficiency of our models and location spiking neurons, which may unlock their potential on neuromorphic hardware.

Index Terms—Spiking Neural Networks, spiking neuron models, location spiking neurons, event-driven tactile learning, robotic manipulation

I. INTRODUCTION

With the prevalence of artificial intelligence, computers today have demonstrated extraordinary abilities in the 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 [1].

With the recent advances in material science and Artificial Neural Networks (ANNs), research on tactile perception has begun to soar, including tactile object recognition [2]–[4], slip detection [5], and texture recognition [6], [7]. 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 [8], [9].

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 [10] and event-based tactile sensors [11]. 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 [12]. Recently, several works [1], [11], [13] utilized Spiking Neural Networks (SNNs) [12], [14], [15] 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 [16]–[18] and conduct real-valued computations, SNNs adopt spiking neurons [19]–[21] 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. 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 current spiking neuron models and high spatio-temporal complexity in the event-based tactile data, 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 advance event-driven tactile learning.

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 [22], location spiking neurons update their membrane potentials based on locations. Specifically, based on the Time Spike Response Model (TSRM) [19], we develop the “Location Spike Response Model”, henceforth referred to as “LSRM”. Moreover, based on the most commonly-used Time Leaky Integrate-and-Fire (TLIF) model [20], we develop the “Location Leaky Integrate-and-Fire” model, henceforth

¹TSRM is the classical SRM in the literature and TLIF is the classical LIF in the literature. We add the character “T” to highlight their differences from LSRM and LLIF.

referred to as “LLIF”. TSRM and TLIF are the classical SRM and 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 research on learning from event sensors like NeuTouch [11], Dynamic Audio Sensors [23], or Dynamic Vision Sensors [10].

Next, we investigate the effectiveness of location spiking neurons and propose two models to capture the complex spatio-temporal dependencies in the event-driven tactile data. Specifically, we build the **Hybrid_SRM_FC** that combines a fully-connected SNN with TSRM neurons and a full-connected SNN with LSRM neurons. Moreover, to capture more spatio-temporal topology knowledge in the data, we propose the **Hybrid_LIF_GNN** that fuses spatial and temporal spiking graph neural networks for event-driven tactile learning. 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 temporal spiking graph neural network with LLIF neurons to extract features of these graphs. Finally, it fuses the spiking tactile features from two networks and provides the final tactile learning prediction. To enable the dynamics of location spiking neurons and optimize the proposed models, we propose three bio-inspired location orders and several loss functions involved with locations. We also propose the timestep-wise inference algorithms for the two models to show their applicability to the temporal data.

Lastly, we apply our proposed models to event-driven tactile learning, including event-driven tactile object recognition and event-driven slip detection, and test them on three challenging 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 0.15s. Extensive experimental results demonstrate the significant improvements of our models over other state-of-the-art methods on event-driven tactile learning. Moreover, the experiments show that conventional spiking neurons are better at capturing spatial dependencies, while location spiking neurons are better at modeling mid-and-long temporal dependencies. Furthermore, our models outperform ANNs by around **10 \times** to **100 \times** on energy consumption, which shows the superior energy efficiency of our models and may bring new opportunities and unlock their potential on neuromorphic hardware.

Portions of this work “Event-Driven Tactile Learning with Location Spiking Neurons [24]” were accepted by IJCNN 2022 and an oral presentation was given at the IEEE WCCI

2022. In this paper, we expand the idea of location spiking neurons to TLIF neurons and propose LLIF neurons. Moreover, we build the Hybrid_LIF_GNN for event-driven tactile learning, which adopts spatio-temporal topologies and employs a more complicated network structure compared to the Hybrid_SRM_FC. We further include more data, experiments, and interpretation to demonstrate the effectiveness and energy efficiency of the proposed neurons and models. Last but not least, we provide preliminary results of the location spiking neurons on event-driven audio learning to show the broad applicability and potential impact of this work.

The rest of the paper is organized as follows. In Section II, we provide an overview of related work on SNNs and event-driven tactile sensing and learning. In Section III, 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 IV, we demonstrate the effectiveness and energy efficiency of our proposed models on benchmark datasets. Finally, we discuss the broad impact of this work and conclude in Section V.

II. RELATED WORK

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

A. 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 [22]. The brain has more information processing mechanisms like the neuronal and synaptic functionality [25], [26]. 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 [27] and Loihi [28]. Similar to ANNs, SNNs can adopt general network topologies like convolutional layers and fully-connected layers, but use different neuron models [21]. Commonly-used neuron models for SNNs are the Time Leaky Integrate-and-Fire (TLIF) model [20] and the Time Spike Response Model (TSRM) [19]. 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 trained ANNs to SNNs [29], [30] and approximating the derivative of the spike function [31], [32]. 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,

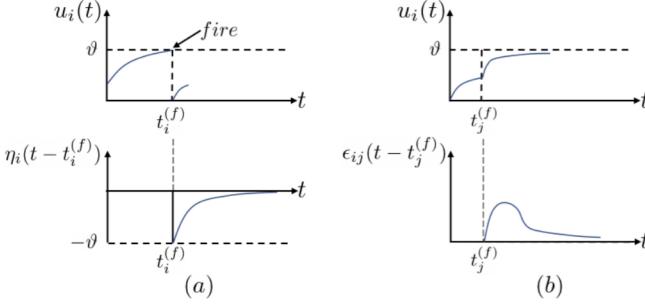


Fig. 1. The **temporal** recurrent neuronal dynamics of TSRM neuron i . (a) the refractory dynamics of TSRM neuron i . Immediately after firing an output spike at time $t_i^{(f)}$, the value of $u_i(t)$ is lowered or reset by adding a negative contribution $\eta_i(\cdot)$. The kernel $\eta_i(\cdot)$ vanishes for $t < t_i^{(f)}$ and decays to zero for $t \rightarrow \infty$. (b) the incoming spike dynamics of TSRM neuron i . A presynaptic spike at time $t_j^{(f)}$ increases the value of $u_i(t)$ for $t \geq t_j^{(f)}$ by an amount of $w_{ij}x_j(t_j^{(f)})\epsilon_{ij}(t - t_j^{(f)})$. The kernel $\epsilon_{ij}(\cdot)$ vanishes for $t < t_j^{(f)}$. A transmission delay may be included in the definition of $\epsilon_{ij}(\cdot)$. This figure is adopted from [24].

we can derive the approximate backpropagation methods for SNNs with location spiking neurons.

B. 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 [33] and the SynTouch BioTac [34] and event-driven tactile sensors like the NeuTouch [11] and the NUSkin [35]. 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 [13], little prior work exists on learning event-based tactile data with SNNs. The work [1] 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 [11] released the first publicly-available event-driven visual-tactile dataset collected by NeuTouch and proposed an SNN based on SLAYER [14] 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 [13] adopted the spiking graph neural network [15] 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 the event-driven tactile learning.

III. METHODS

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

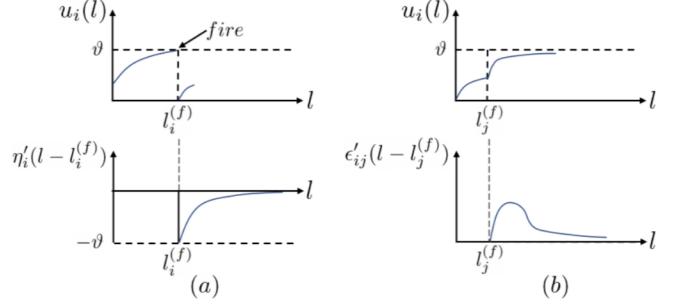


Fig. 2. The **spatial** recurrent neuronal dynamics of LSRM neuron i . (a) the refractory dynamics of LSRM neuron i . Immediately after firing an output spike at location $l_i^{(f)}$, the value of $u_i(l)$ is lowered or reset by adding a negative contribution $\eta'_i(\cdot)$. The kernel $\eta'_i(\cdot)$ vanishes for $l < l_i^{(f)}$ and decays to zero for $l \rightarrow \infty$. (b) the incoming spike dynamics of LSRM neuron i . A presynaptic spike at location $l_j^{(f)}$ increases the value of $u_i(l)$ for $l \geq l_j^{(f)}$ by an amount of $w'_{ij}x'_j(l_j^{(f)})\epsilon'_{ij}(l - l_j^{(f)})$. The kernel $\epsilon'_{ij}(\cdot)$ vanishes for $l < l_j^{(f)}$. A location delay may be included in the definition of $\epsilon'_{ij}(\cdot)$. ‘ $<$ ’ and ‘ \geq ’ indicate the location order. This figure is adopted from [24].

A. TSRM vs. LSRM

Spiking neuron models are mathematical descriptions of specific cells in the nervous system. They are the basic building blocks of SNNs. Two commonly-used spiking neuron models are the TLIF and the TSRM [36]. In this section, we introduce the TSRM and transform it to a location spiking neuron model – the LSRM.

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 ϑ 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

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

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:

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

where $s_i = t - t_i^{(f)}$, $s_j = t - t_j^{(f)}$, Γ_i is the set of presynaptic neurons of neuron i , and $x_j(t_j^{(f)}) = 1$ is the presynaptic spike. $\eta_i(\cdot)$ is the refractory kernel, which describes the response of neuron i to its own spikes at time t . $\epsilon_{ij}(\cdot)$ 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 scale the incoming spike response. Figure 1(a) visualizes the refractory dynamics and Figure 1(b) visualizes the incoming spike dynamics. Without loss of generality, such temporal recurrent neuronal dynamics also apply to other spiking neuron models, such as the TLIF. From the above descriptions, we find that existing spiking neuron models have **explicit temporal recurrence** but do not 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 locations². These neurons exploit **explicit spatial recurrence** and enable us to extract features of event-based data in a novel way. Specifically, we adopt the TSRM and transform it to the LSRM. In the LSRM, the spatial recurrent neuronal dynamics of neuron i are described by its location membrane potential $u_i(l)$. When $u_i(l)$ exceeds a predefined threshold ϑ 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

$$\mathcal{G}_i = \{l_i^{(f)}; 1 \leq f \leq n\} = \{l | u_i(l) = \vartheta\}, \quad (3)$$

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 III-E. The value of $u_i(l)$ is governed by two different spike response processes:

$$u_i(l) = \sum_{l_i^{(f)} \in \mathcal{G}_i} \eta'_i(s_i') + \sum_{j \in \Gamma'_i} \sum_{l_j^{(f)} \in \mathcal{G}_j} w'_{ij} x'_j(l_j^{(f)}) \epsilon'_{ij}(s_j'), \quad (4)$$

where $s_i' = l - l_i^{(f)}$, $s_j' = l - l_j^{(f)}$, Γ'_i is the set of presynaptic neurons of neuron i , and $x'_j(l_j^{(f)}) = 1$ is the presynaptic spike. $\eta'_i(\cdot)$ is the refractory kernel, which describes the response of neuron i to its own spikes at location l . $\epsilon'_{ij}(\cdot)$ is the incoming spike response kernel, which models the neuron i 's response to the presynaptic spikes from neuron j at location l . w'_{ij} accounts for the connection strength between neuron i and neuron j and scale the incoming spike response. Figure 2(a) visualizes the refractory dynamics of LSRM neurons and Figure 2(b) visualizes the incoming spike dynamics of LSRM neurons. The threshold ϑ of LSRM neurons can be different from that of TSRM neurons, while we set the same for simplicity.

B. TLIF vs. LLIF

The TLIF model is the most commonly-used spiking neuron model since it is computationally tractable and maintains biological fidelity to a certain degree [31]. A similar transformation process can be applied to the TLIF to derive its corresponding location spiking neuron model – the LLIF.

The dynamics of the TLIF neuron i are governed by

$$\tau \frac{du_i(t)}{dt} = -u_i(t) + I(t), \quad (5)$$

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 pre-synaptic 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 Eq. (5) into a recursive expression:

$$u_i(t) = (1 - \frac{dt}{\tau})u_i(t-1) + \frac{dt}{\tau} \sum_j w_j x_j(t), \quad (6)$$

²locations could refer to pixel or patch locations for images or taxel locations for tactile sensors.

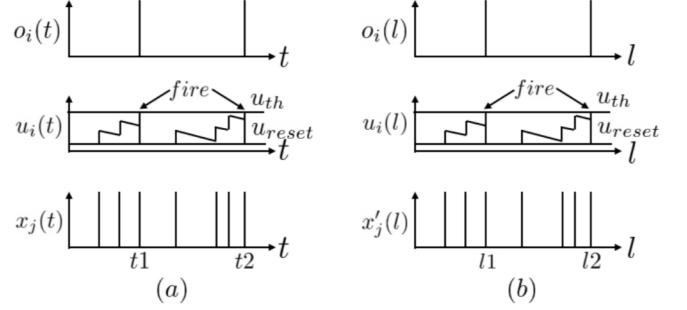


Fig. 3. (a) the **temporal** recurrent neuronal dynamics of a TLIF neuron i . (b) the **spatial** recurrent neuronal dynamics of a LLIF neuron i . Both of them take as input binary spikes and output binary spikes. x_j and x'_j represents the input signal to the neuron i from neuron j , u_i is the neuron's 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} .

where $\sum_j w_j x_j(t)$ is the weighted summation of the inputs from pre-neurons at the current time step. Equation (6) can be further simplified as:

$$u_i(t) = \alpha u_i(t-1) + \sum_j w'_j x_j(t), \quad (7)$$

where $\alpha = 1 - \frac{dt}{\tau}$ can be considered a decay factor, and w'_j 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 3(a) visualizes the dynamics of a TLIF neuron. From the above descriptions, like the TSRM, we find that the TLIF also has **explicit temporal recurrence** but does not possess **explicit spatial recurrence**, which, to some extent, limits its representation abilities.

To enrich the representation abilities of the TLIF, we propose the LLIF. Different from the temporal dynamics shown in Eq. (5), the LLIF neuron i employs the spatial dynamics:

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

where $u_i(l)$ represents the internal membrane potential of a LLIF neuron i at location l , τ' is a location constant, and $I(l)$ represents the pre-synaptic input. To better understand the membrane potential update of LLIF neurons, we use the Euler method to transform the first-order differential equation of Eq. (8) into a recursive expression:

$$u_i(l) = (1 - \frac{dl}{\tau'})u_i(l_{prev}) + \frac{dl}{\tau'} \sum_j w'_j x'_j(l), \quad (9)$$

where $\sum_j w'_j x'_j(l)$ is the weighted summation of the inputs from pre-neurons at the current location. Equation (9) can be further simplified as:

$$u_i(l) = \beta u_i(l_{prev}) + \sum_j w'_j x'_j(l), \quad (10)$$

where $\beta = 1 - \frac{dl}{\tau'}$ can be considered a location decay factor, and w'_j 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

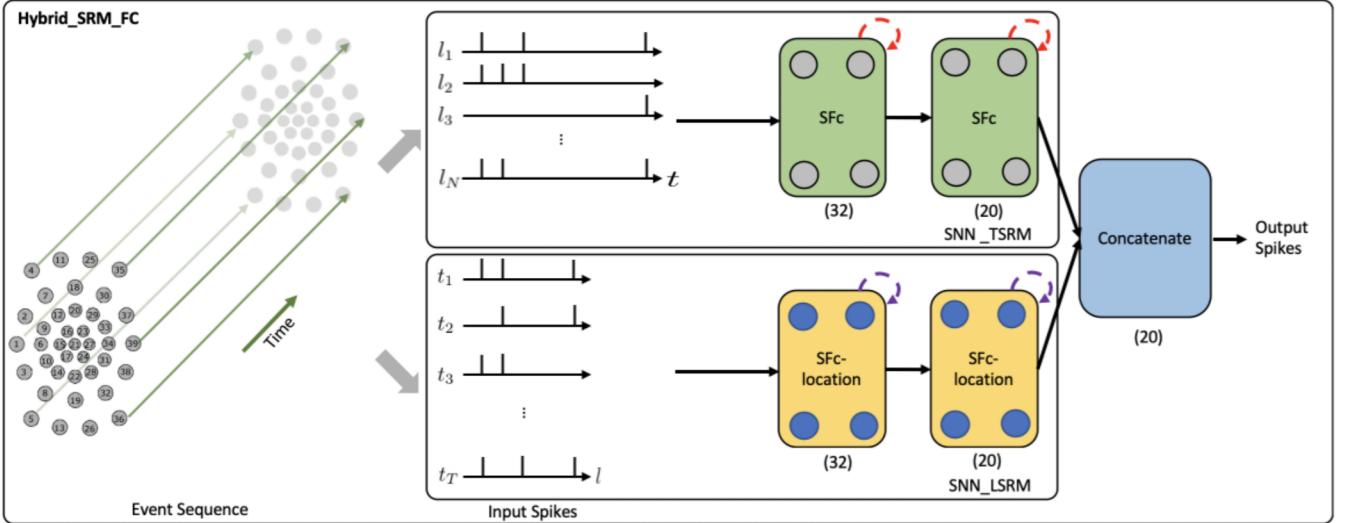


Fig. 4. The network structure of the Hybrid_SRM_FC. The SNN with TSRM neurons (SNN_TSRM) processes the input spikes X_{in} and adopts the temporal recurrent neuronal dynamics (shown with red dashed arrows) of TSRM neurons to extract features from the data. The SNN with LSRM neurons (SNN_LSRM) processes 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. 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) to be equal to 20. This figure is adapted from [24].

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. The location order is manually set and will be discussed in Section III-E. Figure 3(b) visualizes the spatial recurrent neuronal dynamics of a LLIF neuron.

C. Event-Driven Tactile Learning with the LSRM Neurons

To boost the event-based tactile learning performance, we propose 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 [11], a biologically-inspired event-driven fingertip tactile sensor with 39 taxels arranged spatially in a radial fashion (see Fig. 4). In this section, we introduce the event-driven tactile learning with the LSRM neurons.

1) *Hybrid_SRM_FC*: Figure 4 presents the network structure of the Hybrid_SRM_FC. From the figure, we can see that the model has two components, including the SNN with TSRM neurons (SNN_TSRM) and the SNN with LSRM neurons (SNN_LSRM). Specifically, SNN_TSRM employs the temporal recurrent neuronal dynamics to extract spiking feature representations from the event-based tactile data $\mathbf{X}_{in} \in \mathbb{R}^{N \times T}$, where N is the total number of taxels and T is the total time length of event sequences. SNN_LSRM utilizes the spatial recurrent neuronal dynamics to extract spiking feature representations from the event-based tactile data $\mathbf{X}'_{in} \in \mathbb{R}^{T \times N}$, where X'_{in} is transposed from X_{in} . The spiking representations from two networks are then concatenated to yield the final task-specific output.

2) *SNN_TSRM vs. SNN_LSRM*: The network structure of SNN_TSRM is shown in the top part of Fig. 4. It employs two spiking fully-connected layers with TSRM neurons (SFC) to process X_{in} and generate the spiking representations $O_1 \in \mathbb{R}^{K \times T}$,

$\mathbb{R}^{K \times T}$, 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 SFC are decided by:

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

$$o_i(t) = \begin{cases} 1 & \text{if } u_i(t) \geq \vartheta; \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

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

where w_{ij} are the trainable parameters, $\eta(\cdot)$ and $\epsilon(\cdot)$ are predefined by hyperparameters, Γ_i is the set of presynaptic neurons spanning over the spatial domain, which is utilized to capture the spatial dependencies in the event-based data.

The network structure of SNN_LSRM is shown in the bottom part of Fig. 4. It employs two spiking fully-connected layers with LSRM neurons (SFC-location) 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 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 SFC-location are:

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

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

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

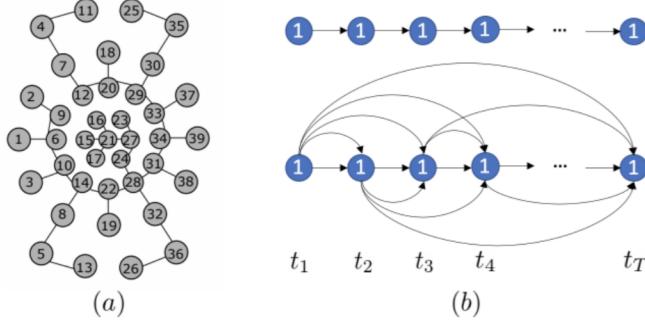


Fig. 5. (a) the tactile spatial graph G_s at time step t generated by the Minimum Spanning Tree (MST) algorithm [13]. Each circle represents a taxel of NeuTouch. (b) two different tactile temporal graphs G_t for a specific taxel $n = 1$: the above one is the sparse tactile temporal graph, while the below one is the dense tactile temporal graph.

where w'_{ij} are the trainable parameters, $\eta'(\cdot)$ and $\epsilon'(\cdot)$ are predefined by hyperparameters, Γ'_i is the set of presynaptic neurons spanning over the temporal domain, which is utilized to model the temporal dependencies in the event-based data. Such location spiking neurons tap the representative potential and enable us to capture features in this way.

3) *Concatenate*: 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 $T + N$.

D. Event-Driven Tactile Learning with the LLIF Neurons

In this section, 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.

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 Fig. 5.

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**. $V^t = \{v_n^t | n = 1, \dots, N\}$ and $V_n = \{v_n^t | t = 1, \dots, T\}$ 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 . $E^t = \{e_{i,j}^t | i, j = 1, \dots, N\}$ 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. $E_n = \{e_n^{p,q} | p, q = 1, \dots, T\}$ 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 Fig. 5(b). 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.

2) *Hybrid_LIF_GNN*: To process the data from our tactile graphs and capture the complex spatio-temporal dependencies in the event-based tactile data, we propose the Hybrid_LIF_GNN (Fig. 6), which fuses spatial and temporal spiking graph neural networks. Specifically, we adopt the spatial spiking graph neural network (SSGNN) with TLIF neurons [13] to process the tactile spatial graphs. And we use temporal recurrent neuronal dynamics to extract features from the tactile spatial graphs. Moreover, we develop the temporal spiking graph neural network (TSGNN) with LLIF neurons to process the tactile temporal graphs. And we use spatial recurrent neuronal dynamics to extract features from the tactile temporal graphs. Finally, we fuse the spiking features from two networks and obtain the final prediction.

a) *SSGNN vs. TSGNN*: SSGNN takes as input tactile spatial graphs, while TSGNN takes as input tactile temporal graphs. SSGNN has one spatial spiking graph (SSG) layer and three spatial spiking fully-connected (SSFc) layers, where TLIF neurons that employ the temporal recurrent neuronal dynamics are the basic building blocks. On the other hand, TSGNN has one temporal spiking graph (TSG) layer and three temporal spiking fully-connected (TSFc) layers, where LLIF neurons that possess the spatial recurrent neuronal dynamics are the basic building blocks.

b) *SSG vs. TSG*: Based on Eq. (7), the membrane potential $u_i(t)$ and output spiking state $o_i(t)$ of TLIF neuron i in SSG are decided by:

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

where $I(t) = \text{GNN}(G_s(t))$ is to capture the spatial structural information. Based on Eq. (10), the membrane potential $u_i(l)$ and output spiking state $o_i(l)$ of LLIF neuron i in TSG are decided by:

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

where $I(l) = \text{GNN}(G_t(l))$ is to model the temporal dependencies in the data. l is represented by the taxel $n \in N$ in this paper. To fairly compare with other baselines, we use TAGConv [37] as GNN in this paper.

c) *SSFc vs. TSFc*: The membrane potential $u_i(t)$ and output spiking state $o_i(t)$ of TLIF neuron i in SSFc are decided by Eq. (13), where $I(t) = \text{Fc}(\text{Pre}(t))$ and $\text{Pre}(t)$ is the previous layer's output at time step t . While the membrane potential $u_i(l)$ and output spiking state $o_i(l)$ of LLIF neuron i in TSFc are decided by Eq. (14), where $I(l) = \text{Fc}(\text{Pre}(l))$ and $\text{Pre}(l)$ is the previous layer's output at location l (taxel n).

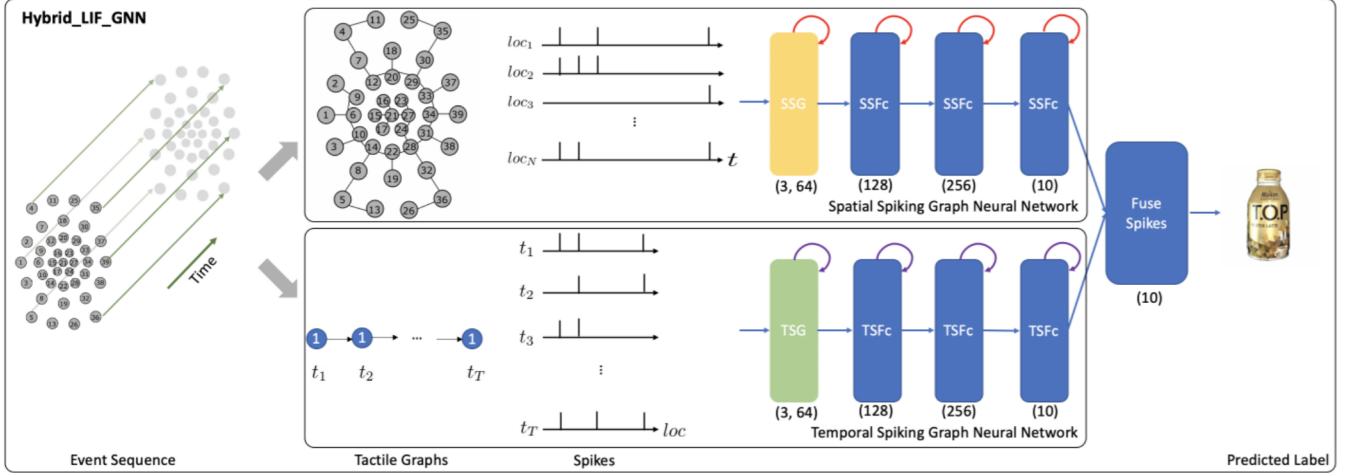


Fig. 6. The structure of the Hybrid_LIF_GNN. The spatial spiking graph neural network processes the T tactile spatial graphs and adopts the temporal recurrent neuronal dynamics (shown with red arrows) of TLIF neurons to extract features. The temporal spiking graph neural network processes the N tactile 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 TAGConv's hop size and filter size. (128), (256), and (10) represent the sizes of fully-connected layers, where we assume the number of classes (K) to be equal to 10.

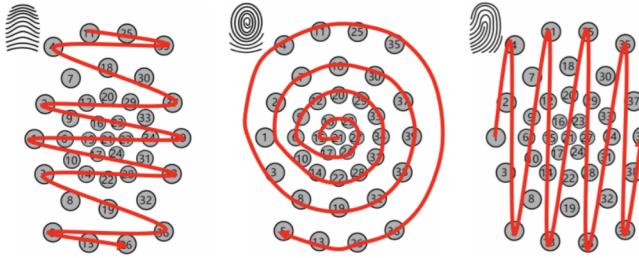


Fig. 7. Three bio-inspired location orders. Left to right: arch-like location order, whorl-like location order, loop-like location order. This figure is adopted from [24].

d) Fuse Spikes: SSGNN extracts spiking features $O_1 \in \mathbb{R}^{K \times T}$, where K is the output dimension determined by the task. Then, SSGNN predicts its label vector $O'_1 \in \mathbb{R}^K$ by averaging O_1 over the time window T ,

$$O'_1 = \frac{1}{T} \sum_t^T O_1(t), \quad (15)$$

where $O_1(t) \in \mathbb{R}^K$. TSGNN extracts spiking features $O_2 \in \mathbb{R}^{K \times N}$. It predicts its label vector $O'_2 \in \mathbb{R}^K$ by averaging O_2 over the spatial window N ,

$$O'_2 = \frac{1}{N} \sum_l^N O_2(l), \quad (16)$$

where $O_2(l) \in \mathbb{R}^K$. Then, to fuse 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.

E. Implementations and Algorithms

1) Location Orders: To enable location spiking neurons' spatial recurrent neuronal dynamics, we propose three location

orders for event-based tactile learning (Fig. 7) based on three major fingerprint patterns of humans – arch, whorl, and loop. Three examples are shown here. Each number in the brackets represents the taxel index, see Fig. 5(a).

- 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]

2) Hybrid_SRM_FC: Similar to the spike-count loss of prior works [11], [14], 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 [11]. To overcome the non-differentiability of spikes and apply the backpropagation algorithm, we use the approximate gradient proposed in SLAYER [14]. The timestep-wise inference algorithm of the Hybrid_SRM_FC is shown in Algorithm 1. And the corresponding timestep-wise training algorithm can be derived by incorporating the weighted spike-count loss.

Algorithm 1 Timestep-wise inference algorithm of the Hybrid_SRM_FC, adopted from [24]

Input: event-based tactile inputs $X_{in} \in \mathbb{R}^{N \times T}$, N taxels, and the total time length T .
Output: timestep-wise predictions of O_1 , O_2 , and O .

- 1: **for** $t \leftarrow 1$ to T **do**
- 2: obtain $X \in \mathbb{R}^{N \times t}$
- 3: obtain $\bar{X}' = \text{concatenate}(X', \mathbf{0}) \in \mathbb{R}^{T \times N}$, where $X' \in \mathbb{R}^{t \times N}$, and $\mathbf{0} \in \mathbb{R}^{(T-t) \times N}$
- 4: $O_1(t) = \mathbf{0} \in \mathbb{R}^{K \times t}$, $O_2(t) = \mathbf{0} \in \mathbb{R}^{K \times N}$
- 5: $O(t) = \mathbf{0} \in \mathbb{R}^{K \times (t+N)}$
- 6: $O_1(t) = \text{SNN_TSRM}(X)$
- 7: $O_2(t) = \text{SNN_LSRM}(\bar{X}')$
- 8: $O(t) = \text{concatenate}(O_1(t), O_2(t))$
- 9: **end for**

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 vector y and the final predicted label vector O' .

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

To overcome the non-differentiability of spikes and apply the backpropagation algorithm, we use the rectangular function [31] to approximate the derivative of the spike function (Heaviside function) in Eqs. (13) and (14). The timestep-wise inference algorithm of the model is shown in Algorithm 2.

Algorithm 2 Timestep-wise inference algorithm of the Hybrid_LIF_GNN

Input: event-based tactile inputs $X_{in} \in \mathbb{R}^{N \times T}$, N taxels, and the total time length T
Output: timestep-wise label vectors of O'_1 , O'_2 , and O'

- 1: **for** $t \leftarrow 1$ to T **do**
- 2: form t tactile spatial graphs G_s with $X \in \mathbb{R}^{N \times t}$
- 3: obtain $\bar{X}' = \text{concatenate}(X', \mathbf{0}) \in \mathbb{R}^{T \times N}$, where $X' \in \mathbb{R}^{t \times N}$, and $\mathbf{0} \in \mathbb{R}^{(T-t) \times N}$
- 4: form N tactile temporal graphs G_t with \bar{X}'
- 5: $O'_1(t), O'_2(t), O'(t) = \mathbf{0} \in \mathbb{R}^K$
- 6: **for** $i \leftarrow 1$ to t **do**
- 7: $O'_1(t) += \text{SSGNN}(G_s(i))$
- 8: **end for**
- 9: $O'_1(t) /= t$ ▷ the label vector from SSGNN
- 10: **for** $j \leftarrow 1$ to N **do**
- 11: $O'_2(t) += \text{TSGNN}(G_t(j))$
- 12: **end for**
- 13: $O'_2(t) /= N$ ▷ the label vector from TSGNN
- 14: $O'(t) = \text{mean}(O'_1(t), O'_2(t))$ ▷ max can be used
- 15: **end for**

IV. EXPERIMENTS

In this section, we extensively evaluate our proposed models. Specifically, we first conduct experiments on the Hybrid_SRM_FC built with TSRM and LSRM neurons. Then, we demonstrate the superior performance of the Hybrid_LIF_GNN built with TLIF and LLIF neurons. Finally, we compare the two models on the benchmark datasets. The source code and experimental configuration details are available at <https://github.com/pkang2017/TactileLSN>.

A. 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 its performance on the benchmark datasets and compare it with state-of-the-art models. Finally, we demonstrate the superior energy efficiency of Hybrid_SRM_FC over ANNs and show the high-efficiency benefit of LSRM neurons. We utilize the slayerPytorch framework³ to implement the proposed model and employ RMSProp with the l_2 regularization to optimize them.

TABLE I
BENCHMARK DATASETS FOR THE HYBRID_SRM_FC

Datasets	TD (s)	SSR (s)	T	N	K	#Samples
Objects-v1	6.5	0.02	325	78	36	900
Containers-v1	6.5	0.02	325	78	20	800
Slip Detection	0.15	0.001	150	78	2	100

1) *Datasets*: We use the datasets collected by NeuTouch [11], including “Objects-v1” and “Containers-v1” for event-driven tactile object recognition and “Slip Detection” for event-driven slip detection. Unlike “Objects-v1” only requiring 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%, 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 0.15s for “Slip Detection”. We summarize the dataset statistics in Table I, where TD is for time durations, SSR is for spike sampling rates, $T=TD / SSR$ is the total time length, N is the total number of taxels (each tactile sensor has 39 taxels and two tactile sensors are used), and K is the number of classes. We split the data into a training set (80%) and a test set (20%) with an equal class distribution in the experiments. We repeat each experiment for five rounds and report the average accuracy.

2) *Comparing Models*: We compare our model with the state-of-the-art SNN methods for event-driven tactile learning, including Tactile-SNN [11] and TactileSGNet [13]. Tactile-SNN employs **TSRM neurons** as the building blocks, and the network structure of Tactile-SNN is Input-SFc0-SFc1. TactileSGNet utilizes **TLIF neurons** as the building blocks and proposes the spiking graph neural network (SGNet).

³<https://github.com/bamsumit/slayerPytorch>

TABLE II
ACCURACIES ON BENCHMARK DATASETS FOR THE HYBRID_SRM_FC

Method	Type	Objects-v1	Containers-v1	Slip Detection
Tactile-SNN [11]	SNN	0.75	0.57*	0.82*
TactileSGNet [13]	SNN	0.79	0.58	0.97
GRU-MLP [11]	ANN	0.72	0.46*	0.87*
CNN-3D [11]	ANN	0.90	0.67*	0.44*
Hybrid_SRM_FC	SNN	0.91	0.86	1.0

*These values come from [11]. The best performance is in bold.

The network structure of TactileSGNet is Input-SGNet-SFc1-SFc2-SFc3. As in [11], we also compare our model against conventional deep learning, specifically Gated Recurrent Units (GRUs) [38] with Multi-layer Perceptrons (MLPs) and 3D convolutional neural networks [39]. 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-Fc1.

3) *Basic Performance*: Table II 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 two-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 to extract features in a novel way; (2) our model fuses SNN_TSRM and SNN_LSRM 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 II, we find out that our model outperforms ANNs on the three tasks, which might be because our model is more compatible with various kinds of event-based data and better maintains the sparsity to prevent overfitting.

4) *Ablation Studies*: To examine the effectiveness of each component in the proposed model, we **separately train** SNN_TSRM (which is exactly Tactile-SNN) and SNN_LSRM (which is referred to as Location Tactile-SNN). From Table III, 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 the event-driven slip detection. The reason for this could be two-fold: (1) the time durations of event-driven tactile object recognition datasets are longer than that of “Slip Detection”, and Location Tactile-SNN is good at capturing the mid-and-long term dependencies in these object recognition datasets; (2) like Tactile-SNN, Location Tactile-SNN 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. Furthermore, we examine the sensitivities of λ in Eq.(18) and location orders. From Table III, we notice the results of related models are close, proving that the λ tuning and location orders do not significantly impact the task performance.

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

TABLE III
ABLATION STUDIES ON THE HYBRID_SRM_FC

Method	Type	Objects-v1	Containers-v1	Slip Detection
Tactile-SNN [11]	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

the two components’ contributions and achieve the better overall performance. Figure 8(a), 8(b), and 8(c) show the timestep-wise inference accuracies of SNN_TSRM, SNN_LSRM, 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

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

where the hyperparameter ψ controls the balance between SNN_TSRM’s contribution and SNN_LSRM’s contribution and T is the total time length. From the figures, we can see that SNN_TSRM has good “early” accuracies on the three tasks since it well captures the spatial dependencies with the help of Eq. (11). However, its accuracies do not improve too much at the later stage since it does not sufficiently capture the temporal dependencies. In contrast, SNN_LSRM has fair “early” accuracies, while its accuracies jump a lot at the later stage since it models the temporal dependencies in Eq. (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 SNN_TSRM at the early stage, the time-weighted Hybrid_SRM_FC can have a good “early” accuracy as well as an excellent “final” accuracy.

TABLE IV
OPERATION COST ($\times 10^6$) AND COMPRESSION RATIO ON BENCHMARK DATASETS FOR THE HYBRID_SRM_FC

Method	Objects-v1	Containers-v1	Slip Detection
GRU-MLP	11.87	11.87	5.48
CNN-3D	8.34	8.14	3.5
Hybrid_SRM_FC	0.60	0.83	0.045
Ratio	$9.80 \sim 14.30 \times$	$13.90 \sim 19.78 \times$	$77.78 \sim 121.78 \times$

6) *Energy Efficiency*: Following the estimation method in [15], [40]⁴, we estimate the computational costs of the Hybrid_SRM_FC and ANNs on the three datasets. As shown in Fig. 8(d), our proposed model has no multiplication operations and achieves far fewer addition operations than ANN models on the three datasets. Moreover, based on Table IV, the compression ratio of total operations (ANNS Opts. / Ours Opts.) is between $9.80 \times$ and $121.78 \times$. These results are consistent with the fact that the sparse spike communication and event-driven computation underlie the efficiency advantage of SNNs

⁴We consider the computational costs in feature matrix transformation.

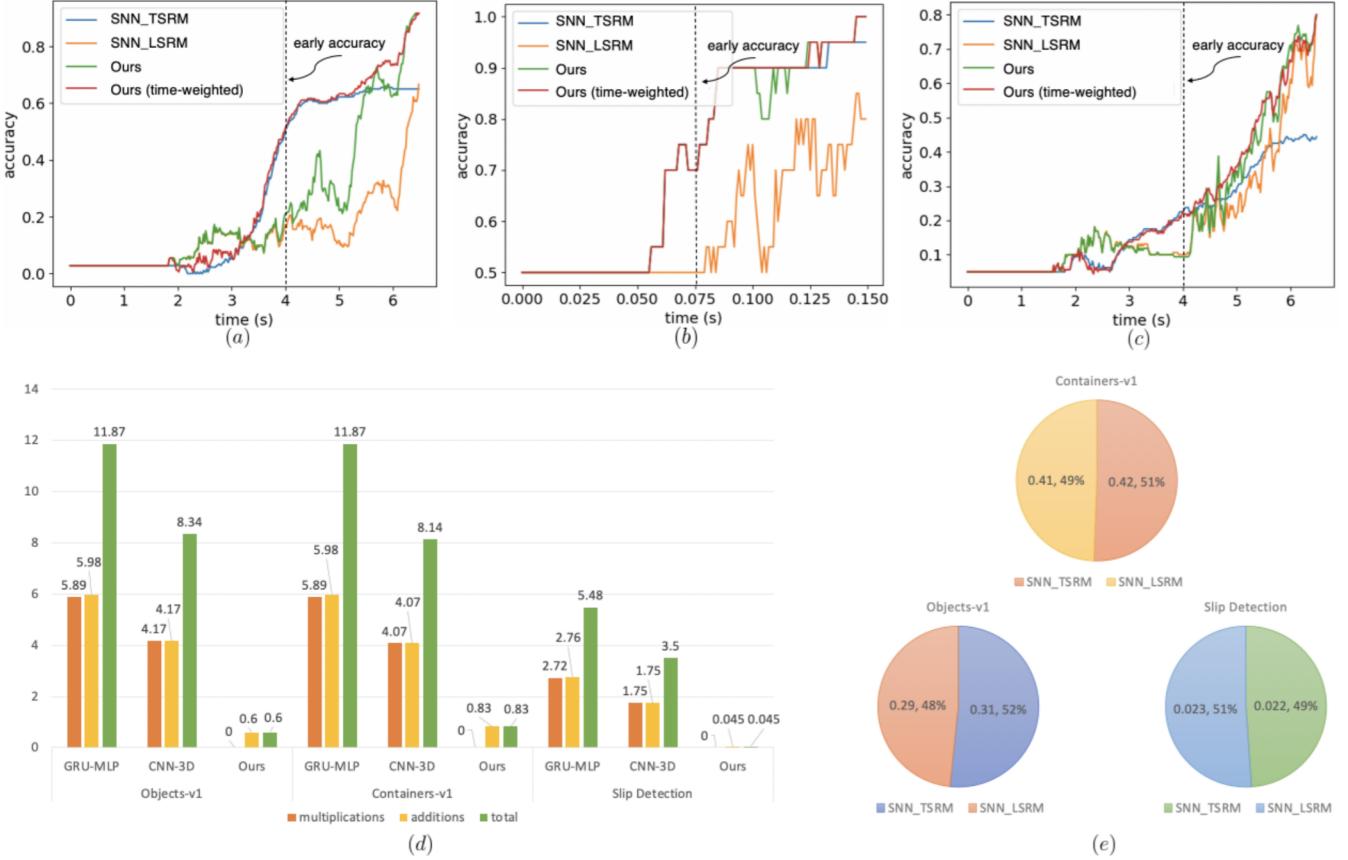


Fig. 8. The timestep-wise inference (Alg. 1) for SNN_TSRM – O_1 , SNN_LSRM – O_2 , the Hybrid_SRM_FC – O , and the time-weighted Hybrid_SRM_FC – O_{tw} on (a) “Objects-v1”, (b) “Slip Detection”, (c) “Containers-v1”. Note that we use the same event sequences as [11] and the first spike occurs at around 2.0s for “Objects-v1” and “Containers-v1”. (d) Operation cost ($\times 10^6$) comparison between the Hybrid_SRM_FC and ANNs on the benchmark datasets. (e) Operation cost ($\times 10^6$) and percentage comparison between SNN_TSRM and SNN_LSRM on the benchmark datasets.

and demonstrate the potentials of our model on neuromorphic hardware. We further compare the costs of SNN_TSRM and SNN_LSRM on the benchmark datasets. From Fig. 8(e), we can see that the cost of SNN_LSRM is almost equal to that of SNN_TSRM on each dataset, which shows that the location spiking neurons (LSRM neurons) have the similar energy efficiency compared to existing spiking neurons (TSRM neurons). Such high-efficiency benefits make location spiking neurons a perfect fit for neuromorphic hardware.

B. Hybrid_LIF_GNN

In this section, to fairly compare with other published models with TLIF neurons [13], we evaluate the Hybrid_LIF_GNN on two real-world datasets – “Objects-v0” and “Containers-v0”.⁵ Then, we conduct several ablation studies to examine the effectiveness of some designs in the Hybrid_LIF_GNN. Finally, we demonstrate the superior energy efficiency of our model over Graph Neural Networks (GNNs) and show the high-efficiency benefits of location spiking neurons. We utilize the Pytorch framework to implement the models and employ Adam with the decay learning rate to optimize them.

⁵In this section, to be consistent with [13], we use accuracies (%).

1) Datasets: To fairly compare with other published models with TLIF neurons [13], we evaluate the Hybrid_LIF_GNN built with TLIF and LLIF neurons on “Objects-v0” and “Containers-v0”. These two datasets are the initial versions of “Objects-v1” and “Containers-v1”. And we summarize the dataset statistics in Table V. We split the data into a training set (80%) and a test set (20%) with an equal class distribution in the experiments. We repeat each experiment for five rounds and report the average accuracy. Interested readers can find more details about the datasets in [13] and the corresponding website⁶.

TABLE V
BENCHMARK DATASETS FOR THE HYBRID_LIF_GNN

Datasets	TD (s)	SSR (s)	T	N	K	#Samples
Objects-v0	5.0	0.02	250	39	36	720
Containers-v0	6.5	0.02	325	39	20	300

2) Comparing Models: We compare the Hybrid_LIF_GNN with the state-of-the-art methods on event-based tactile object recognition, including the TactileSGNet series [13] and Tactile-SNN [11]. All the models in the TactileSGNet series adopt the neural networks with TLIF neurons. More precisely,

⁶<https://clear-nus.github.io/visuotactile/download.html>

TABLE VI
ACCURACIES (%) ON DATASETS FOR THE HYBRID_LIF_GNN

Method	Type	Objects-v0	Containers-v0
TactileSGNet-CNN [13]	SNN	88.40*	60.17*
TactileSGNet-MLP [13]	SNN	85.97*	58.83*
TactileSGNet-GCN [13]	SNN	85.14*	58.83*
TactileSGNet-TAGConv [13]	SNN	89.44*	64.17*
Tactile-SNN [11]	SNN	-	71.00
Recurrent GNN-linear	ANN	92.36	70.67
Recurrent GNN-elu	ANN	91.11	74.67
Recurrent GNN-LeakyRelu	ANN	89.31	73.00
Hybrid_LIF_GNN [‡]	SNN	93.33	79.33

*These values come from [13]. And the value for Tactile-SNN comes from [11]. [‡] represents Hybrid_LIF_GNN-sparse-mean-loop, where we use “sparse tactile temporal graph”, “mean fusion”, and “loop-like location order”. The best performance is in bold.

the network structure for TactileSGNet-CNN is Input-Spiking CNN-SSFc1-SSFc2-SSFc3, where the tactile data are organized in a grid structure according to the spatial distribution of taxels, and the CNN with TLIF neurons is utilized to extract features from this grid. The network structure for TactileSGNet-MLP is Input-SSFc0-SSFc1-SSFc2-SSFc3, where SSF0 (the FC layer with TLIF neurons) is utilized to extract features from input data. The network structure for TactileSGNet-GCN is Input-Spiking GCN-SSFc1-SSFc2-SSFc3, where the tactile spatial graphs are built, and Spiking GCN [41] is utilized to extract features from the graphs. The network structure for TactileSGNet-TAGConv is Input-Spiking TAGConv-SSFc1-SSFc2-SSFc3 (exactly the same as SSGNN of our model), where the tactile spatial graphs are built, and TAGConv [37] with TLIF neurons is utilized to extract features from the graphs. Tactile-SNN employs TSRM neurons as the building blocks, which is the SNN_TSRM in Fig. 4. Please note that Tactile-SNN [11] utilized the data of both hands ($N = 78$) in their paper, while other models, including ours, use the data of one hand ($N = 39$). We also compare the Hybrid_LIF_GNN against GNNs. Specifically, the GNNs have the same network structures as the Hybrid_LIF_GNN, including one **recurrent TAGConv-Fc-Fc-Fc** for T **tactile spatial graphs**, one **recurrent TAGConv-Fc-Fc-Fc** 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 Eqs. (13) and (14). Our model utilizes Heaviside function as its activation function while GNNs utilize ANNs’ activation functions.

3) *Basic Performance*: We report the test accuracies on the two event-driven tactile object recognition datasets in Table VI. From this table, we can see that the Hybrid_LIF_GNN significantly outperforms state-of-the-art methods including TactileSGNet series [13] and Tactile-SNN [11]. The reason why our model can achieve the better performance could be two-fold: (1) different from state-of-the-art methods that only utilize TLIF neurons to extract features from the tactile spatial graphs, our model also employs TSGNN with LLIF neurons to extract features from the tactile temporal graphs; (2) our model fuses SSGNN and TSGNN to capture complex spatio-temporal dependencies in the data. We also compare our model with

TABLE VII
ACCURACIES (%) ON ABLATION STUDIES FOR THE HYBRID_LIF_GNN

Method	Objects-v0	Containers-v0
Ours-sparse-mean*	93.33	79.33
Ours-dense-mean*	92.50	78.67
Ours-sparse-max*	85.56	77.00
Ours-dense-max*	85.14	76.00
Ours-arch [†]	93.89	77.67
Ours-whorl [†]	93.47	81.00
Ours-loop [†]	93.33	79.33

* represents Hybrid_LIF_GNN-loop. [†] represents Hybrid_LIF_GNN-sparse-mean. The best performance is in bold.

GNNs by replacing the spike functions in Eqs. (13) and (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 VI, we observe that the Hybrid_LIF_GNN outperforms GNNs on the two datasets, which might be because our model is more compatible with event-based data and better maintains the sparsity to prevent overfitting.

4) *Ablation Studies*: We further provide ablation studies for exploring the optimal design choices. From Table VII, 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 two-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. From Table VII, we also notice that the results of three bio-inspired location orders are close, proving that the location order does not significantly impact the task performance.

5) *Timestep-wise Inference*: Figure 9 shows the timestep-wise inference accuracies (%) for SSGNN, TSGNN, 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

$$O'_{tw}(t) = O'_1(t)(1 - \frac{t}{\zeta T}) + O'_2(t)\frac{t}{\zeta T}. \quad (21)$$

where ζ controls the balance between SSGNN’s contribution and TSGNN’s contribution and T is the total time length. From the figure, we can see that SSGNN 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, TSGNN 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 set $\zeta = 2$ to shift more weights to SSGNN 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 Fig. 9.

6) *Energy Efficiency*: Following the estimation method in [15], we estimate the computational costs of Hybrid_LIF_GNN-sparse-mean-loop and GNNs from Table VI on the two datasets. As shown in Fig. 10(a), our model has

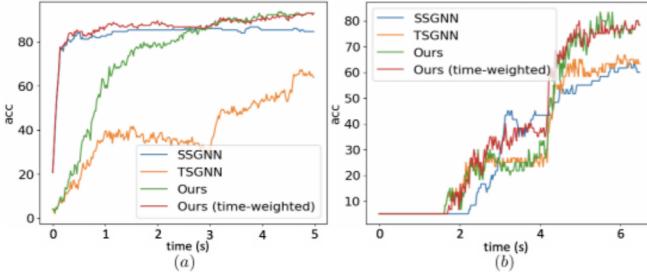


Fig. 9. The timestep-wise inference (Alg. 2) accuracies (%) for SSGNN – O'_1 , TSGNN – O'_2 , the Hybrid_LIF_GNN – O' , and the time-weighted Hybrid_LIF_GNN – O'_{tw} on (a) “Objects-v0”, (b) “Containers-v0”.

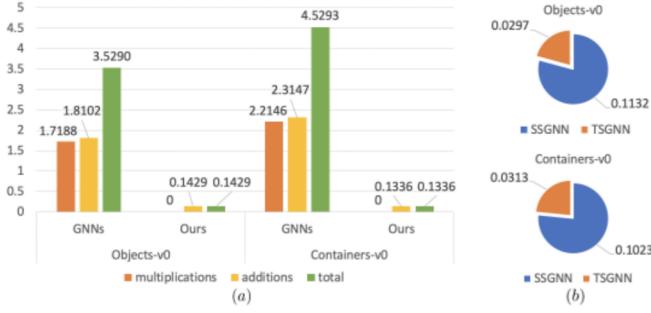


Fig. 10. Operation cost ($\times 10^8$) comparison (a) between Hybrid_LIF_GNN-sparse-mean-loop and GNNs. (b) between SSGNN and TSGNN of Hybrid_LIF_GNN-sparse-mean-loop.

no multiplication operations and achieves far fewer addition operations than GNNs on the two datasets. Moreover, the compression ratio of total operations (GNNs Opts. / Ours Opts.) is between **24.70** \times and **33.90** \times . These results are consistent with what we show in Section IV-A6 and demonstrate the potentials of our model on neuromorphic hardware.

We further compare the costs of SSGNN and TSGNN. From Fig. 10(b), we can see that the cost of TSGNN only accounts for 21% ~ 23% of total costs. The reason why TSGNN can achieve higher energy efficiency could be two-fold: (1) the **sparse tactile temporal graphs** require fewer operations in the TSG layer; (2) the number of taxels N (39 locations) is much less than the number of time steps T (250 for “Objects-v0” and 325 for “Containers-v0”), which leads to less recurrent neuronal dynamics in TSGNN. Such high-efficiency benefits make TSGNN and location spiking neurons (LLIF neurons) a perfect fit for neuromorphic hardware.

C. Comparison

We further compare the Hybrid_SRM_FC (Fig.4) and the Hybrid_LIF_GNN (Fig.6) on “Objects-v1”, “Containers-v1”,

and “Slip Detection”. From Table VIII, 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.

V. DISCUSSION AND CONCLUSION

This paper proposes a novel neuron model – “location spiking neuron”. 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 Eqs. (2), (7), (11) and (13). While location spiking neurons use **spatial** recurrent neural dynamics to update their potentials and model **temporal** dependencies by aggregating the information from presynaptic neurons, see Eqs. (4), (10), (12) and (14). Moreover, based on experimental results, we can see that conventional spiking neurons (in SNN_TSRM and SSGNN) are better at capturing **spatial dependencies** which benefit the “early” accuracy, while location spiking neurons (in SNN_LSRM and TSGNN) 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 tactile data. Thus, we always **concatenate or fuse** the networks to sufficiently capture spatio-temporal dependencies in the data and boost the task performance.

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 the task performance.

Besides event-driven tactile learning, we can apply the models with location spiking neurons to other event-driven learning fields. To show the broad impact of location spiking neurons, we apply the Hybrid_SRM_FC (Fig. 4) to event-driven audio learning and provide preliminary results⁷. Specifically, we use the N-TIDIGITS18 dataset [23], which is collected by playing the audio files from the TIDIGITS dataset [42] to the dynamic audio sensor – the CochleaAMS1b sensor [43]. 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’ to ‘9’. The single digit dataset has 4,950 samples, of which 2,486

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

⁷The objective of this experiment is not to necessarily obtain state-of-the-art results, but to demonstrate the effectiveness of location spiking neurons on other event-driven neuromorphic applications.

TABLE VIII

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

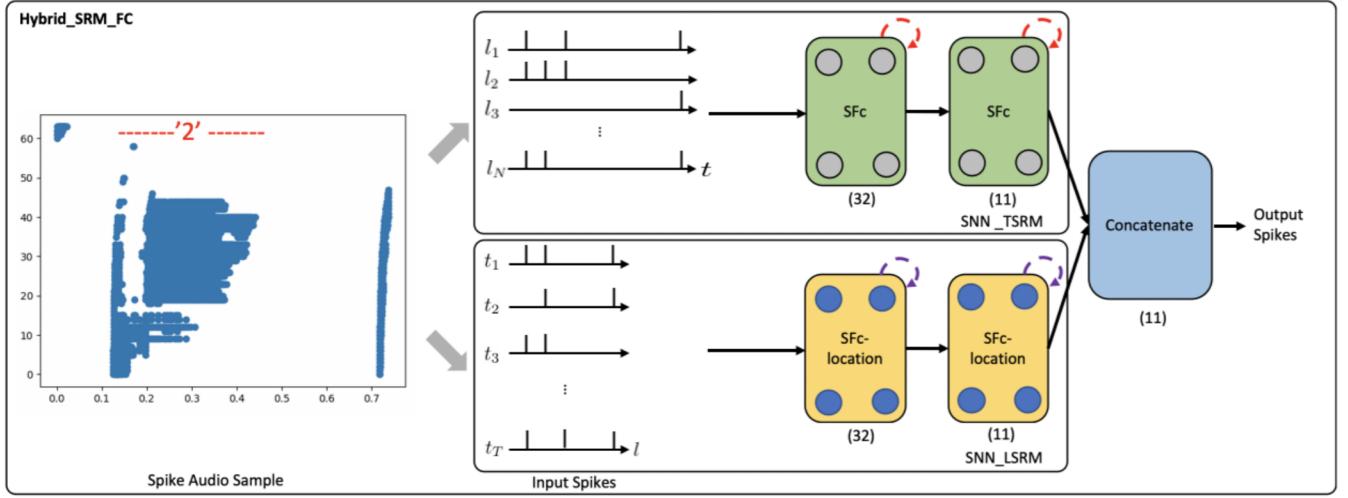


Fig. 11. The Hybrid_SRM_FC processes a spike audio sample and predict the sample's label. The network structure of this model is the same as what we show in Fig. 4.

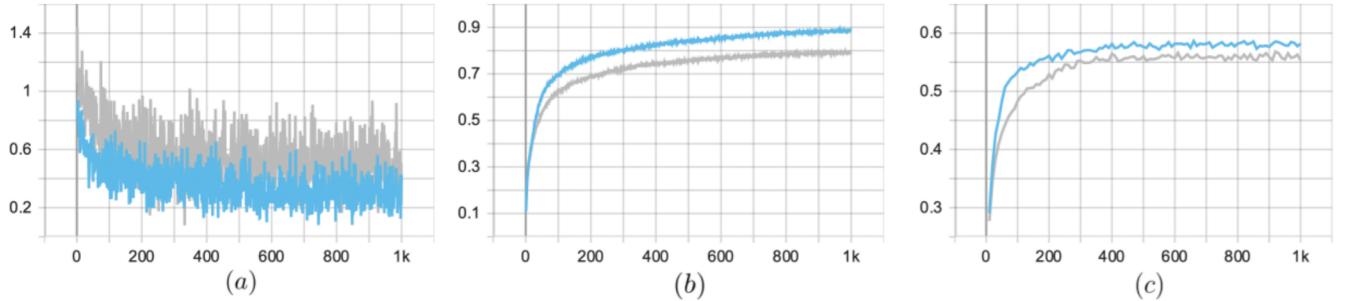


Fig. 12. Training and testing profiles for SNN_TSRM (gray) and Hybrid_SRM_FC (blue): (a) the training loss, (b) the training accuracy, (c) the testing accuracy.

TABLE IX
TEST ACCURACIES (%) ON THE SINGLE DIGIT N-TIDIGITS18

Method	Type	N-TIDIGITS18
SNN_TSRM (only TSRM)	SNN	56.34
Hybrid_SRM_FC (TSRM + LSRM)	SNN	58.58

samples are utilized for testing. A spike audio sample of digit '2' is shown in Fig. 11, where x-axis indicates the event time, and y-axis indicates the 64 frequency channels of the CochleaAMS1b sensor. Each blue dot in the sample 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 Fig. 11. Table IX presents the test accuracies on the single digit dataset. We observe that our model significantly outperforms SNN_TSRM⁸. Moreover, figure 12 shows the training and testing profiles for SNN_TSRM and our model on the dataset. From this figure, we can see that our model converges faster and attains a lower loss and a higher accuracy compared to SNN_TSRM.

These preliminary results show that location spiking neurons

⁸Please note that there are 2,486 testing samples. This means 1% improvement leads to 24.86 more samples are correctly classified.

can be applied to other event-driven learning fields. Moreover, location spiking neurons can extract useful features from the data and benefit the task.

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 several models for event-driven tactile learning to sufficiently capture the complex spatio-temporal dependencies in the 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 further unlocks their potential on neuromorphic hardware. Overall, this work sheds new light on SNN representation learning and event-driven learning, which may facilitate the understanding of advanced cognitive intelligence.

ACKNOWLEDGMENTS

Portions of this work "Event-Driven Tactile Learning with Location Spiking Neurons [24]" were accepted by IJCNN 2022 and orally presented at the IEEE WCCI in 2022.

APPENDIX
LIST OF MAJOR NOTATIONS IN THE PAPER

Notation	Description
$u_i(t)$	the potential of existing spiking neuron i at time t
$\eta_i(\cdot)$	the refractory kernel of TSRM neuron i
Γ_i	the set of presynaptic neurons of TSRM neuron i
$\epsilon_{ij}(\cdot)$	the incoming spike response kernel between TSRM neurons
w_{ij}	the connection strength between TSRM neurons
$x_j(t)$	the input from pre TSRM / TLIF neuron j at time t
$u_l(l)$	the potential of location spiking neuron i at location l
$\eta'_i(\cdot)$	the refractory kernel of LSRM neuron i
Γ'_i	the set of presynaptic neurons of LSRM neuron i
$\epsilon'_{ij}(\cdot)$	the incoming spike response kernel between LSRM neurons
w'_{ij}	the connection strength between LSRM neurons
$x'_j(l)$	the input from pre LSRM / LLIF neuron j at location l
ϑ	the firing threshold of TSRM and LSRM neurons
τ	the time constant of TLIF neurons
α	the decay factor of TLIF neurons
$I(t)$	the weighted summation of the inputs from pre TLIF neurons at time t
w_j	the connection strength between TLIF / LLIF neurons
w'_j	the scaling weight between TLIF / LLIF neurons
u_{th}	the firing threshold of TLIF / LLIF neurons
τ'	the location constant of LLIF neurons
β	the location decay factor of LLIF neurons
$I(l)$	the weighted summation of the inputs from pre LLIF neurons at location l
N	the number of taxels of NeuTouch
T	the number of total time length of event inputs
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 SNN_TSRM / SSGNN
$o_i(t)$	the output spiking state of TSRM / TLIF neuron i at time t
O_2	output spikes from SNN_LSRM / TSGNN
$o_i(l)$	the output spiking state of LSRM / LLIF 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 at taxel n
O'_1	the predicted label vector of SSGNN
O'_2	the predicted label vector of TSGNN
O^2	the predicted label vector of the Hybrid_LIF_GNN

REFERENCES

- [1] Tasbolat Taunyazov, Yansong Chua, Ruihan Gao, Harold Soh, and Yan Wu, “Fast texture classification using tactile neural coding and spiking neural network,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9890–9895.
- [2] Harold Soh and Yiannis Demiris, “Incrementally learning objects by touch: Online discriminative and generative models for tactile-based recognition,” *IEEE transactions on haptics*, vol. 7, no. 4, pp. 512–525, 2014.
- [3] Zhanat Kappassov, Juan-Antonio Corrales, and Véronique Perdereau, “Tactile sensing in dexterous robot hands,” *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015.
- [4] Jose Sanchez, Carlos M Mateo, Juan Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar, “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)*. IEEE, 2018, pp. 504–511.
- [5] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine, “More than a feeling: Learning to grasp and regrasp using vision and touch,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.
- [6] Shiv S Baishya and Berthold Bäuml, “Robust material classification with a tactile skin using deep learning,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 8–15.
- [7] Tasbolat Taunyazov, Hui Fang Koh, Yan Wu, Caixia Cai, and Harold Soh, “Towards effective tactile identification of textures using a hybrid touch approach,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4269–4275.
- [8] Da Li, Xinbo Chen, Michela Becchi, and Ziliang Zong, “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)*. IEEE, 2016, pp. 477–484.
- [9] Emma Strubell, Ananya Ganesh, and Andrew McCallum, “Energy and policy considerations for deep learning in nlp,” *arXiv preprint arXiv:1906.02243*, 2019.
- [10] Guillermo Gallego, Tobi Delbrück, Garrick Michael Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jorg Conradt, Kostas Daniilidis, et al., “Event-based vision: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, 2020.
- [11] Tasbolat Taunyazov, Weicong Sng, Hian Hian See, Brian Lim, Jethro Kuan, Abdul Fatir Ansari, Benjamin Tee, and Harold Soh, “Event-driven visual-tactile sensing and learning for robots,” in *Proceedings of Robotics: Science and Systems*, July 2020.
- [12] Michael Pfeiffer and Thomas Pfeil, “Deep learning with spiking neurons: opportunities and challenges,” *Frontiers in neuroscience*, vol. 12, pp. 774, 2018.
- [13] Fuqiang Gu, Weicong Sng, Tasbolat Taunyazov, and Harold Soh, “Tactilesnet: A spiking graph neural network for event-based tactile object recognition,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9876–9882.
- [14] Sumit Bam Shrestha and Garrick Orchard, “SLAYER: Spike layer error reassignment in time,” in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., pp. 1419–1428. Curran Associates, Inc., 2018.
- [15] Mingkun Xu, Yujie Wu, Lei Deng, Faqiang Liu, Guoqi Li, and Jing Pei, “Exploiting spiking dynamics with spatial-temporal feature normalization in graph learning,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou, Ed. 8 2021, pp. 3207–3213, International Joint Conferences on Artificial Intelligence Organization, Main Track.
- [16] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al., “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*. Citeseer, 2013, vol. 30, p. 3.
- [17] Bing Xu, Naixian Wang, Tianqi Chen, and Mu Li, “Empirical evaluation of rectified activations in convolutional network,” *arXiv preprint arXiv:1505.00853*, 2015.
- [18] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *arXiv preprint arXiv:1511.07289*, 2015.
- [19] Wulfram Gerstner, “Time structure of the activity in neural network models,” *Physical review E*, vol. 51, no. 1, pp. 738, 1995.
- [20] Larry F Abbott, “Lapicque’s introduction of the integrate-and-fire model neuron (1907),” *Brain research bulletin*, vol. 50, no. 5–6, pp. 303–304, 1999.
- [21] Wulfram Gerstner and Werner M Kistler, *Spiking neuron models: Single neurons, populations, plasticity*, Cambridge university press, 2002.
- [22] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda, “Towards spike-based machine intelligence with neuromorphic computing,” *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.
- [23] Jithendar Anumula, Daniel Neil, Tobi Delbrück, and Shih-Chii Liu, “Feature representations for neuromorphic audio spike streams,” *Frontiers in neuroscience*, vol. 12, pp. 23, 2018.
- [24] Peng Kang, Srutarshi Banerjee, Henry Chopp, Aggelos Katsaggelos, and Oliver Cossairt, “Event-driven tactile learning with location spiking neurons,” in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.
- [25] Ed Bullmore and Olaf Sporns, “The economy of brain network organization,” *Nature Reviews Neuroscience*, vol. 13, no. 5, pp. 336–349, 2012.
- [26] Daniel J Felleman and David C Van Essen, “Distributed hierarchical processing in the primate cerebral cortex,” in *Cereb cortex*. Citeseer, 1991.
- [27] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al., “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science*, vol. 345, no. 6197, pp. 668–673, 2014.
- [28] Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R Risbud, “Advancing neuromorphic computing with loihi: A survey of

- results and outlook,” *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.
- [29] Yongqiang Cao, Yang Chen, and Deepak Khosla, “Spiking deep convolutional neural networks for energy-efficient object recognition,” *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, 2015.
- [30] Abhroni Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy, “Going deeper in spiking neural networks: Vgg and residual architectures,” *Frontiers in neuroscience*, vol. 13, pp. 95, 2019.
- [31] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi, “Spatiotemporal backpropagation for training high-performance spiking neural networks,” *Frontiers in neuroscience*, vol. 12, pp. 331, 2018.
- [32] Xiang Cheng, Yunzhe Hao, Jiaming Xu, and Bo Xu, “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*, Christian Bessiere, Ed. 7 2020, pp. 1519–1525, International Joint Conferences on Artificial Intelligence Organization, Main track.
- [33] Alexander Schmitz, Marco Maggiali, Lorenzo Natale, Bruno Bonino, and Giorgio Metta, “A tactile sensor for the fingertips of the humanoid robot icub,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2212–2217.
- [34] Jeremy A Fishel and Gerald E Loeb, “Sensing tactile microvibrations with the biotac—comparison with human sensitivity,” in *2012 4th IEEE RAS & EMBS international conference on biomedical robotics and biomechatronics (BioRob)*. IEEE, 2012, pp. 1122–1127.
- [35] Tasbolat Taunyazov, Luar Shui Song, Eugene Lim, Hian Hian See, David Lee, Benjamin CK Tee, and Harold Soh, “Extended tactile perception: Vibration sensing through tools and grasped objects,” *arXiv preprint arXiv:2106.00489*, 2021.
- [36] Wolfgang Maass and Christopher M Bishop, *Pulsed neural networks*, MIT press, 2001.
- [37] Jian Du, Shanghang Zhang, Guanhong Wu, José MF Moura, and Soummya Kar, “Topology adaptive graph convolutional networks,” *arXiv preprint arXiv:1710.10370*, 2017.
- [38] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [39] Juan M Gandarias, Francisco Pastor, Alfonso J García-Cerezo, and Jesús M Gómez-de Gabriel, “Active tactile recognition of deformable objects with 3d convolutional neural networks,” in *2019 IEEE World Haptics Conference (WHC)*. IEEE, 2019, pp. 551–555.
- [40] Chankyu Lee, Adarsh Kumar Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy, “Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks,” in *European Conference on Computer Vision*. Springer, 2020, pp. 366–382.
- [41] Thomas N Kipf and Max Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.
- [42] R Gary Leonard and George Doddington, “Tidigits speech corpus,” *Texas Instruments, Inc*, 1993.
- [43] Vincent Chan, Shih-Chii Liu, and Andr van Schaik, “Aer ear: A matched silicon cochlea pair with address event representation interface,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 1, pp. 48–59, 2007.



Peng Kang (Member, IEEE) is currently a fourth year Ph.D. candidate in the department of Computer Science at Northwestern University. His supervisor is Prof. Oliver Cossairt. And he works closely with Prof. Aggelos Katsaggelos. Prior to his Ph.D. study at Northwestern, he was a research assistant and obtained a research-based master at School of Computer Science of McGill University, under the supervision of Prof. Xue (Steve) Liu. Before that, he completed his bachelor degree in software engineering at Sun Yat-sen University. He has a broad interest in artificial intelligence and its applications. Previously, his research focused on 2nd generation neural networks – ANNs, including numerical analysis and their applications in recommender systems and computer vision. His current research focuses on 3rd generation neural networks – SNNs and their event-driven neuromorphic applications in vision, audio, and robotic learning. He published several papers on premier conferences, like KDD, WSDM, WACV, ICIP, and IJCNN.



Srutarshi Banerjee received his Bachelor degree in electrical engineering (with Honors) from Jadavpur University, Kolkata, India with a focus in control systems. After that, he obtained the Master of Technology in instrumentation from Indian Institute of Technology, Kharagpur, India and Ph.D. in electrical engineering from the Northwestern University. His research interests include image and signal processing, computer vision, machine learning as well system level implementation of these algorithms. His research focuses on learning based intelligent

imaging. He is using machine learning and deep learning techniques to build smart imaging systems which adapt itself to the changing scene and environment. Moreover, he is building learning based Room Temperature Semiconductor Detector (RTSD) models for Computed Tomography systems and algorithms for event reconstruction.



Henry Chopp is a Ph.D. candidate in electrical engineering at the Image and Video Processing Lab (IVPL), Northwestern University. He received his bachelor’s degree in electrical engineering at Northwestern University. His research interests are in compressed sensing, data fusion, and machine learning. He is currently working with The Center for Scientific Studies of the Arts on subsampling and reconstruction of paintings using data fusion techniques, as well rate-distortion optimization for object tracking.



Aggelos K. Katsaggelos (Life Fellow, IEEE) received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Greece, in 1979, and the M.S. and Ph.D. degrees in electrical engineering from the Georgia Institute of Technology, in 1981 and 1985, respectively. He is Professor and Joseph Cummings Chair in the ECE Department at Northwestern University. And he has courtesy appointments in the Departments of Computer Science and Radiology at Northwestern University. He has published extensively in the areas of multimedia signal processing and communications (over 300 journal papers, 600 conference papers and 45 book chapters) and he is the holder of 27 international patents. Prof. Katsaggelos is a Fellow of the IEEE (1998), SPIE (2009), EURASIP (2017) and Optical Society of America (2018) and the recipient of the IEEE Third Millennium Medal (2000), the IEEE Signal Processing Society Meritorious Service Award (2001), the IEEE Signal Processing Society Technical Achievement Award (2010), and co-author of several award-winning papers. He was elevated to a Life Fellow member by the IEEE (2022).



Oliver S. Cossairt received the B.S. degree in physics from the Evergreen State College in 2001, and the M.S. from the Massachusetts Institute of Technology in 2003, and Ph.D. in computer science from the Columbia University in 2011. He is currently an Associate Professor with the Computer Science and Electrical and Computer Engineering (ECE) Departments, Northwestern University, Evanston, IL, USA. He is the Director of the Computational Photography Laboratory (CPL), Northwestern University, whose research consists of a diverse portfolio, ranging in topics from optics/photonics, computer graphics, computer vision, machine learning and image processing. The general goal of CPL is to develop imaging hardware and algorithms that can be applied across a broad range of physical scales, from nanometer to astronomical. His research interests include active projects on event cameras, 3D nanotomography (10-9 m), computational microscopy (10-6 m), cultural heritage imaging analysis of paintings (10-3 m), structured light and ToF 3D-scanning of macroscopic scenes (1 m), de-scattering through fog for remote sensing (103 m), and coded aperture imaging for astronomy (106 m). Prof. Cossairt has garnered funding from numerous corporate sponsorships including Google, Rambus, Samsung, Omron, Oculus/Facebook, and Zoloz/Alibaba, and federal funding agencies including ONR, NIH, DOE, DARPA, IARPA, and NSF CAREER Award.