NORTHWESTERN
UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2024-05

June, 2024

Event-Driven Processing and Learning with Spiking Neural Networks

Peng Kang

Abstract

The drive to engineer technology replicating human capabilities has recently led to the development of event-driven sensors, which emulate the energy-efficient neural mechanisms of biological systems such as the retina and cochlea. Like biological sensors that generate spikes to the changing environment, these bio-inspired event-driven sensors build circuits that dynamically produce the binary events to the changing environmental stimuli. In general, 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, processing and learning with these sensors remain in their infancy.

In this dissertation, we propose to utilize Spiking Neural Networks (SNNs) to tackle event-driven processing and learning. Unlike traditional Artificial Neural Networks (ANNs), SNNs draw inspiration from the brain, processing information in a binary spiking fashion that mirrors natural neural activity. Given the common spiking mechanism between event-driven data and SNNs, it naturally follows that SNNs are well-suited for processing and learning from event-driven data. In this thesis, our exploration focuses on three pivotal areas: 1) Developing SNNs for event-driven classification tasks, including tactile object recognition and slip detection, showcasing their superior performance, energy efficiency, and broad impact. 2) Advancing SNNs for complex event-driven regression challenges like surface normal estimation, demonstrating comparable state-of-the-art accuracy with less energy consumption. 3) Innovating spiking neural architectures by integrating insights from neuroscience, resulting in two models that excel in object recognition with additional robustness and interpretability, respectively.

By pushing the boundaries of SNN capabilities and exploring their applications in event-driven processing and learning, this work not only highlights the potential of bio-inspired technologies but also sets the stage for future research in the field.

Keywords

Spiking Neural Networks, Event-Driven Processing and Learning, Robotic Manipulation, Tactile Learning, Shape from Polarization, Bio-Inspired Models

NORTHWESTERN UNIVERSITY

Event-Driven Processing and Learning with Spiking Neural Networks

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Peng Kang

EVANSTON, ILLINOIS

June 2024

ABSTRACT

Event-Driven Processing and Learning with Spiking Neural Networks

Peng Kang

The drive to engineer technology replicating human capabilities has recently led to the development of event-driven sensors, which emulate the energy-efficient neural mechanisms of biological systems such as the retina and cochlea. Like biological sensors that generate spikes to the changing environment, these bio-inspired event-driven sensors build circuits that dynamically produce the binary events to the changing environmental stimuli. In general, 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, processing and learning with these sensors remain in their infancy.

In this dissertation, we propose to utilize Spiking Neural Networks (SNNs) to tackle event-driven processing and learning. Unlike traditional Artificial Neural Networks (ANNs), SNNs draw inspiration from the brain, processing information in a binary spiking fashion that mirrors natural neural activity. Given the common spiking mechanism between event-driven data and SNNs, it naturally follows that SNNs are well-suited for processing and learning from event-driven data. In this thesis, our exploration focuses on three pivotal

areas: 1) Developing SNNs for event-driven classification tasks, including tactile object recognition and slip detection, showcasing their superior performance, energy efficiency, and broad impact. 2) Advancing SNNs for complex event-driven regression challenges like surface normal estimation, demonstrating comparable state-of-the-art accuracy with less energy consumption. 3) Innovating spiking neural architectures by integrating insights from neuroscience, resulting in two models that excel in object recognition with additional robustness and interpretability, respectively.

By pushing the boundaries of SNN capabilities and exploring their applications in event-driven processing and learning, this work not only highlights the potential of bio-inspired technologies but also sets the stage for future research in the field.

Acknowledgements

Over the course of the six years dedicated to my PhD journey, I have been fortunate to encounter many outstanding individuals who have contributed significantly to my development as a researcher, educator, and individual. It is with profound gratitude that I acknowledge their contributions in this section of my thesis.

First and foremost, I extend my heartfelt thanks to my supervisor, Prof. Oliver Cossairt, for his unwavering guidance, support, and encouragement throughout this journey. His insight and wisdom have been instrumental in shaping my research trajectory. Similarly, I am deeply grateful to my co-supervisor, Prof. Aggelos Katsaggelos, who played a pivotal role in my development, challenging me to become a better researcher with his constructive critiques and enriching discussions. I would also like to thank my dissertation committee member, Prof. Jack Tumblin, for his insightful feedback and support during my whole PhD journey.

I would also like to express my sincere appreciation to Prof. Erin Leddon, Prof. Zach Wood-Doughty, and Prof. Mohammed Alam for their exemplary roles as educators. Their dedication to teaching and mentoring has profoundly impacted my approach to academia and teaching, instilling in me the qualities of a committed educator.

During the unprecedented times brought about by the COVID-19 pandemic, Tzung-Han Juang and Dr. Kimberly Seipel Carrow extended their support and assistance, for which I am eternally grateful. Their help was a beacon of hope and resilience in the face of adversity, and without it, my journey would have been significantly more challenging.

Special thanks are due to Prof. Chen Ma and Prof. Kai Huang, who provided substantial support during a particularly difficult challenge in 2020. Their generosity and willingness to help were crucial in navigating the obstacles that arose during that period.

I am also immensely thankful for the camaraderie, collaboration, and support of my colleagues, collaborators, and friends: Prof. Emma Alexander, Prof. Diego Klabjan, Dr. Srutarshi Banerjee, Dr. Henry Chopp, Dr. Sinan Seymen, Dr. Tim Tsz-Kit Lau, Dr. Zhipeng Hou, Dr. Jing Chen, Jianping Zhang, Yukun Ma, Weijian Li, Jingya Xun, Zhihan Zhou, Mingfu Liang, Jipeng Sun, Hao Hu, Aniket Dashpute, and Florian Schiffers. Each one of them has contributed to my PhD journey in unique and meaningful ways, fostering an environment of mutual learning and growth.

To my dearest parents, Xiuhong Yan and Jihong Kang, your love, sacrifice, and unwavering support have been the bedrock of my resilience and success. Your belief in me and my aspirations, coupled with the endless encouragement and care, have been my source of strength and inspiration throughout this journey. I am forever indebted to you both for your selflessness and for instilling in me the values of perseverance and hard work. Thank you for being my guiding light and for everything you have done for me.

In closing, I am profoundly grateful to everyone who has been a part of my PhD journey. Your contributions, support, and belief in my abilities have been the cornerstone of my achievements. Thank you all for being an integral part of my journey toward academic and personal development.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Throughout history, the promise of developing technology that emulates human or animal capabilities has fueled innovation. For instance, in the quest to replicate the visual image-capturing ability of the retina, innovators have developed various frame-based cameras. While these cameras do emulate certain features of the retina and have become ubiquitous in everyday use, their performance can suffer in demanding situations, such as those requiring high speed or high dynamic range. To some extent, this shortfall is attributed to the fundamental differences in the low-level circuit architectures between frame-based cameras and the biological structure of the retina. To replicate the intricate low-level neural architectures and functionalities of biological sensors such as the retina and cochlea, research on event-driven perception has started to gain momentum and several asynchronous event-based sensors have been proposed, including event cameras [**1**] and event-driven tactile sensors [**2**].

Like biological sensors that generate spikes to the changing environment, these bio-inspired event-driven sensors build circuits to dynamically produce the binary events to the changing environmental stimuli. For example, instead of capturing images at a fixed rate like the conventional frame-based cameras, event cameras asynchronously measure per-pixel brightness changes in the environment and output a stream of events that encode the location, time, and sign. In contrast to standard synchronous sensors, such event-based sensors can achieve higher energy efficiency, better scalability, and lower latency.

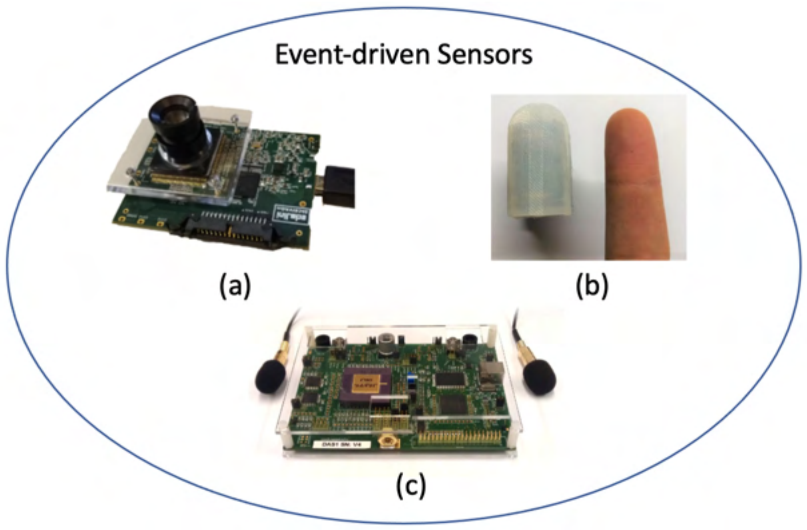


Figure 1.1. Event-driven sensors: (a) event camera [**1**]; (b) event tactile sensor compared to a human finger [**2**]; (c) dynamic audio sensor [**3**].

However, due to the high sparsity and complexity of event-driven data, learning with these sensors is still in their infancy [**6**].

1.1. Event Sensors and Event Data

Figure 1.1 illustrates three kinds of event-driven sensors, including event cameras, event tactile sensors, and dynamic audio sensors.

Inspired by the function of biological visual pathways [**7, 8**], researchers innovated event cameras. Unlike conventional cameras that capture entire images at a fixed frame rate determined by an external clock (e.g., 30 fps), event cameras, such as the Dynamic Vision Sensor (DVS) [**9, 10, 11, 12**], operate on a different principle. They detect changes in brightness asynchronously across each pixel, responding individually to changes in the scene's light intensity. As a result, event cameras produce a stream of digital "events" or "spikes" at a variable data rate. Each event signifies a predefined change in brightness

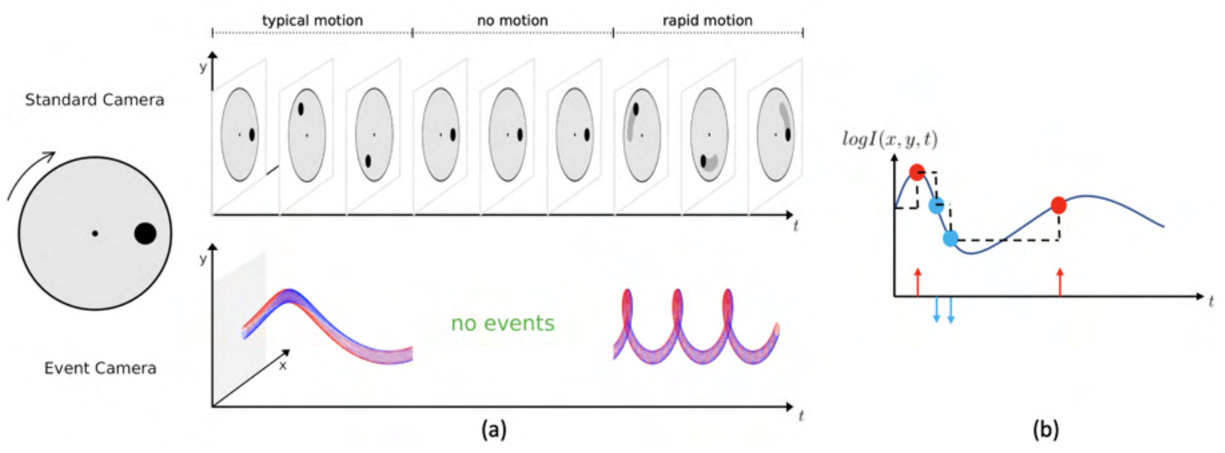


Figure 1.2. (a) Standard video frames from a conventional camera vs. a stream of events from an event camera. Red and blue dots represent positive and negative events, respectively [**4**]. (b) Changes in brightness (log intensity) over time at a specific pixel location. Red and blue arrows represent positive and negative events, respectively.

(log intensity) at a specific pixel and time, offering a distinct approach to visual data capture. Figure 1.2 (a) presents different working mechanisms between standard cameras and event cameras. Figure 1.2 (b) demonstrates the event generation at a specific location. Specifically, each pixel records its log intensity value every time it generates an event and persistently watches for a sufficient deviation from this remembered value. Upon detecting a change that surpasses a predefined threshold, the camera generates a new event. This event encodes the pixel's location (x, y), the time of the change t, and its polarity $p \in (-1, +1)$, indicating whether the brightness has increased (”positive events”) or decreased (”negative events”).

In the event-driven tactile sensing, event tactile sensors are proposed to emulate the functionalities of fingertips. Figure 1.3 presents an event tactile sensor called NeuTouch [**2**]. Specifically, each NeuTouch is composed of many taxels, which are elliptically-shaped to

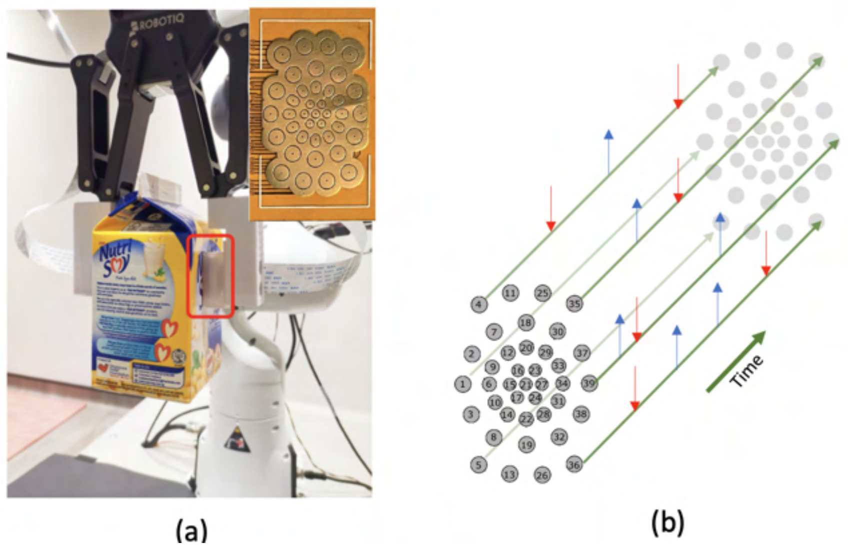


Figure 1.3. (a) Event tactile sensors – NeuTouch attached on Robotic grippers that hold the object; Spatial distribution of the 39 taxels on NeuTouch. [**2**]. (b) The event tactile sensor – NeuTouch generates streams of events that encode the location, time, and sign information. Red and blue arrows represent positive and negative events, respectively.

resemble the human fingertips fast-adapting (FA) mechanoreceptors [**13**]. These taxels can asynchronously measure pressure changes in the environment and output a stream of events that encode the location, time, and sign, as shown in Fig. 1.3 (b). Each event is generated when the change in pressure surpasses a certain threshold. The positive sign indicates the increase in pressure, while the negative sign represents the pressure decrease.

Similar to event cameras and event tactile sensors emulating visual pathways and fingertips respectively, the dynamic audio sensor is an asynchronous event-based silicon cochlea. Specifically, the device takes stereo audio inputs and asynchronously outputs a stream of address-events representing activity in different frequency ranges [**3**].

We can source event data not only directly from event sensors but also from frame-based data, which enables the extension of event-based processing benefits, such as energy

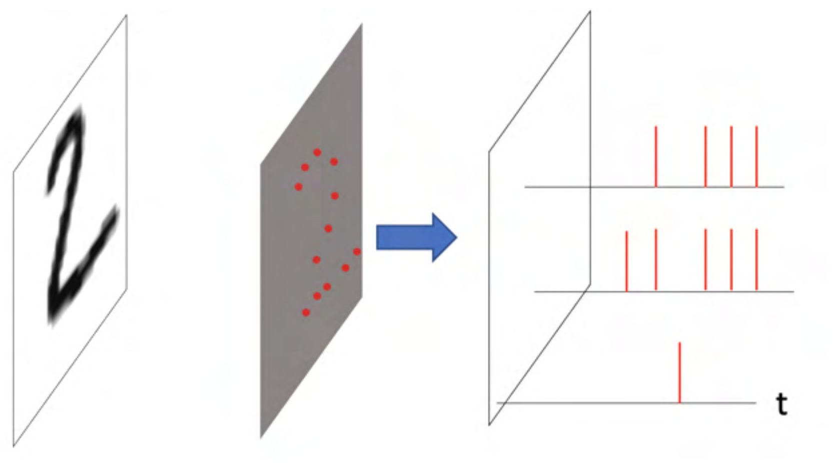


Figure 1.4. The rate coding method transforms an MNIST image into a spike train, where areas with greater intensity exhibit higher firing frequencies. Positions with strokes, marked by red dots, display higher firing rates. In contrast, areas without strokes, such as the image's corners, exhibit lower firing frequencies.

efficiency, into traditional application areas. For example, we can encode static frame-based data into events through rate coding [**14, 15**]. This approach represents the frame-based input by producing a spike train across T timesteps, with the aggregated spike count correlating with the input values' magnitude. The spike generation follows a Poisson distribution. Figure 1.4 illustrates this rate coding mechanism that converts an MNIST image to spiking trains. From the figure, we can see that this method ensures positions that have larger values have higher firing frequency.

1.2. Artificial Neural Networks

In the realm of Artificial Intelligence, Artificial Neural Networks (ANNs) invariably capture our attention. The advancement of optimization methods and training platforms has led to the development of increasingly larger and more effective models. The evolution from AlexNet [**16**] to VGG [**17**], and from ResNet [**18**] to DenseNet [**19**], reflects two

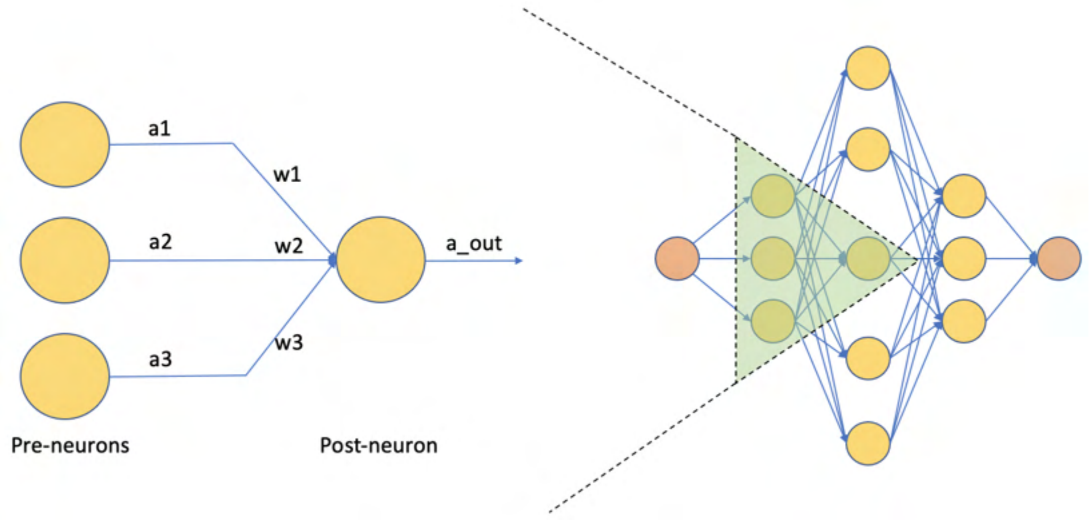


Figure 1.5. General structure of Artificial Neural Networks (ANNs).

predominant trends in artificial neural network design: denser connections and a more hierarchical organization of neurons and layers. These trends are evident at a high level on the right part of Fig. 1.5. Beyond these characteristics, a closer examination of artificial neural networks, as shown on the left part, reveals their reliance on real-valued computations for information transfer. This means that both pre-neurons and post-neurons engage in the exchange of real-valued data, with a significant amount of matrix-vector multiplications occurring during the propagation of information.

With the prevalence of ANNs, research on event-driven processing and learning with ANNs has begun to soar, including event-driven classification tasks, such as image classification [**20**], tactile object recognition [**21, 22, 23**], slip detection [**24**], and texture recognition [**25, 26**], and event-driven regression tasks, such as image reconstruction [**27, 28, 29, 30**], depth estimation [**31, 32, 33**], and optical flow estimation [**34, 35**]. Although ANNs demonstrate promising performance on these tasks, they have several limitations that hinder their potential in the event-driven processing and learning and neuromorphic engineering. First, ANNs operate in a synchronous and dense way, which is incompatible with the asynchronous and sparse nature of events. Most state-of-the-art ANN pipelines require expensive transformations from asynchronous discrete events to synchronous real-valued frames and utilize a stateless feedforward architecture to process the frames. Second, ANNs employ artificial neurons and conduct real-valued computations, which are usually power-hungry compared to human brains that require far less energy to perform the same tasks robustly [**36**].

1.3. Spiking Neural Networks

With the development of ANNs, computers today have demonstrated extraordinary abilities in many cognition tasks, such as computer vision, natural language processing, and recommender systems [**37, 38, 39**]. Although the performance is truly impressive, a key question remains: what is the computing cost involved in such activities? The human brain performs impressive tasks with a power budget of nearly 20W. However, a general computer needs around 250W to perform only recognition among 1000 different kinds of objects [**36**]. What makes such a big energy gap between the general computers and human brains?

To answer this question, we can find some clues from our brains. Although the brain remains vastly unexplored, its remarkable capability may be attributed to three foundational observations from neuroscience: the first one is vast connectivity, each neuron in the brain is connected on average to the other 10,000 neurons [**36**]. The second one is the structural and functional hierarchy. For example, the perceptive abilities are based

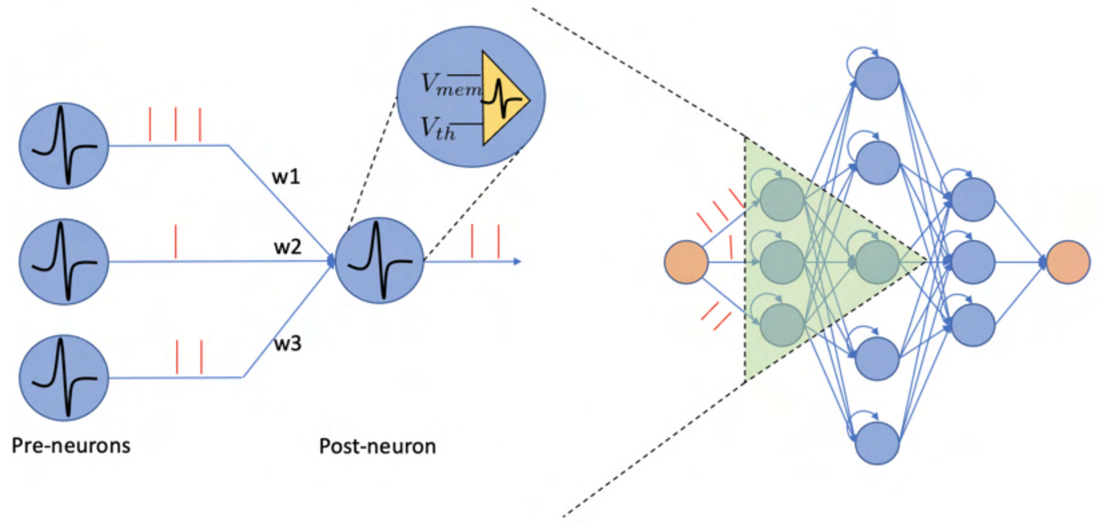


Figure 1.6. General structure of Spiking Neural Networks (SNNs).

on the structural hierarchy in the visual cortex. And the third one is time-dependent neuronal functionality. This means neurons are the computational primitive elements of the brain. And the neurons and synapses are utilized to transfer information through discrete time-dependent spikes. Such spike-based temporal processing allows sparse and efficient information transfer in the brain. Based on these three characteristics, we can find that ANNs majorly imitate brain structures in two ways including vast connectivity and structural hierarchy. But the brain has more information processing mechanisms like the time-dependent neuronal functionality. This explains why ANNs and our human brains have a such energy gap. To integrate such brain-like characteristics, researchers propose Spiking Neural Networks (SNNs).

Figure 1.6 presents the general structure of an SNN. From the right part of the figure, we can see that different from ANNs majorly imitating the vast connectivity and structural hierarchy of the brain structure, SNNs also involve spike-based temporal processing. Red vertical lines in the figure represent spikes, whose values are 1. If there is no spike, the value is 0. Each blue neuron is a spiking neuron. Each round arrow represents the recurrent temporal processing mechanism involved in the spiking neurons. We can take a closer examination of SNNs on the left part of Fig. 1.6. Different from ANNs using real-valued computations, SNNs use spikes to process information. These spikes are essentially binary events whose values can be 0 or 1. Each neuron in the SNNs keeps recurrently updating its membrane potential at each discrete time step. The neuron generates a spike if the membrane potential value exceeds some threshold at some specific time. Thus, a neuron in the SNN can only be active when it receives or generates spikes. Otherwise, the neuron can remain idle. The process is event-driven and contributes to energy efficiency over a given period. This is quite different from ANNs, where all neurons are always active since they always receive or generate real values. Moreover, since the inputs in the SNNs are either 1 or 0, this reduces the dot-product operations on the synapses to less computationally intensive addition operations. All of these explain why there is such a big energy gap between ANNs and human brains and make us embrace the SNNs.

Inspired by human brains, several recent works utilized Spiking Neural Networks (SNNs) to tackle event-driven tasks, including event-driven classification tasks, such as image classification [**40, 41, 14**], tactile object recognition [**2, 5**], slip detection [**2**], and texture recognition [**42**], and event-driven regression tasks, such as image reconstruction [**43**] and optical flow estimation [**44**]. Unlike ANNs, which require expensive transformations from asynchronous discrete events to synchronous real-valued frames, SNNs can directly process sparse event-based sensor data. Moreover, unlike ANNs that employ artificial neurons [**45, 46, 47**] and conduct real-valued computations, SNNs adopt spiking neurons [**48, 49, 50**] 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 capabilities of existing SNN models and high spatio-temporal complexity in the event-based data, it is still challenging to build effective and efficient SNN models to conduct event-driven processing and learning.

1.4. Organization of the Thesis

In this dissertation, we propose to build effective and efficient SNNs for event-driven processing and learning. Our exploration focuses on three pivotal areas: 1) How can we build effective and efficient SNNs to solve event-driven classification problems? Specifically, we focus on event-driven tactile learning with SNNs. We propose several fully SNN models and demonstrate the significant improvements of our models over other works on event-driven classification problems, such as event-driven tactile object recognition and event-driven tactile slip detection. In addition, we show the superior energy efficiency of our models, which may unlock their potential on neuromorphic hardware. Furthermore, we discuss the potential impact of our work on broad event-driven classification tasks. 2) How can we build effective and efficient SNNs to solve event-driven regression problems? Specifically, we tackle event-based shape from polarization with SNNs. We introduce the Single-Timestep and Multi-Timestep Spiking UNets for effective and efficient surface normal estimation. Extensive evaluations on synthetic and real-world datasets demonstrate that our models match the performance of state-of-the-art ANNs in estimating surface normals, with the added advantage of superior energy efficiency. Our work not only contributes to the advancement of SNNs in event-based shape from polarization but also sets the stage for future explorations in optimizing SNN architectures for event-driven regression tasks. 3) Can we bring the lessons from other fields into the novel spiking architecture design? Specifically, inspired by the evidence in neuroscience that the visual processing in human vision is performed hierarchically in the combination of analog and digital processing, we proposed the ANN-SNN models and demonstrated their robustness on object recognition. In addition, inspired by the GLOM model, a hypothetical model in the neuroscience that can imitate the human ability to parse visual scenes and represent the scene's part-whole hierarchies, we proposed energy-efficient and interpretable Spiking GLOM models by incorporating spiking neurons.

The rest of this dissertation is structured as follows. Chapter 2 details the event-driven tactile learning with spiking neural networks. Chapter 3 presents the event-based shape from polarization with spiking neural networks. Chapter 4 introduces the bio-inspired novel spiking architectures. And Chapter 5 provides a summary of my contributions, discusses directions for future work, and concludes this dissertation.

1.5. Bibliographic Notes

The content of this dissertation is mainly based on these five research papers: [**51, 52, 53, 54, 55**].

CHAPTER 2

Event-Driven Tactile Learning with Spiking Neural Networks

This chapter is based on papers [**52, 53**]. The paper [**52**] © 2022 IEEE. Reprinted, with permission, from 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. 1-9.

2.1. 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 [**42**].

With the recent advances in material science and Artificial Neural Networks (ANNs), research on tactile perception has begun to soar, including tactile object recognition [**21, 22, 23**], slip detection [**24**], and texture recognition [**25, 26**]. 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 [**56, 57**].

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 [**1**] and event-based tactile sensors [**2**]. 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 [**6**]. Recently, several works [**2, 5, 42**] utilized Spiking Neural Networks (SNNs) [**41, 6, 58**] 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 [**45, 46, 47**] and conduct real-valued computations, SNNs adopt spiking neurons [**48, 49, 50**] 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 [**36**]. 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 [**2**], these works still cannot sufficiently capture spatio-temporal dependencies and thus hinder the performance of event-driven tactile learning.

In this work, 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 A.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 [**36**], location spiking neurons update their membrane potentials based on locations. Specifically, based on the Time Spike Response Model (TSRM) [**48**], 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 [**49**]. 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 [**2**], Dynamic Audio Sensors [**3**], or Dynamic Vision Sensors [**1**].

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 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 classification 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 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.

We summarize the contributions in this work below.

- We proposed location spiking neurons and demonstrated the dynamics of LSRM neurons and LLIF neurons.
- By exploiting the location spiking neurons, we developed two fully spiking models Hybrid_SRM_FC and Hybrid_LIF_GNN for event-driven tactile learning.
- Experimental results on benchmark datasets demonstrated the extraordinary performance and high energy efficiency of the proposed models and neurons.
- 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 source code is available at `https://github.com/pkang2017/TactileLSN`.

The rest of the chapter is organized as follows. In Section 2.2, we provide an overview of related work on SNNs and event-driven tactile sensing and learning. In Section 2.3, we start by introducing notations for existing spiking neurons and extend them to the specific location spiking neurons. Then, in Section 2.4, we propose various models with location spiking neurons for event-driven tactile learning. Last, in Section 2.5, we provide implementation details and algorithms related to the proposed models. In Section 2.6, we demonstrate the effectiveness and energy efficiency of our models on benchmark datasets. Finally, we discuss and conclude in Section 2.7.

2.2. Related Work

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

2.2.1. 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 [**36**]. The brain has more information processing mechanisms like the neuronal and synaptic functionality [**59, 60**]. 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 [**61**] and Loihi [**62**]. Similar to ANNs, SNNs can adopt general network topologies like convolutional layers and fully-connected layers, but use different neuron models [**50**], such as the Time Leaky Integrate-and-Fire (TLIF) model [**49**] and the Time Spike Response Model (TSRM) [**48**]. 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 [**63, 64**] and approximating the derivative of the spike function [**40, 65**]. 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.

2.2.2. Event-Driven Tactile Sensing and Learning

Inspired by human brains, several recent works utilized Spiking Neural Networks (SNNs) to tackle event-driven classification tasks. However, most of these works are limited to problems of limited temporal complexity like image classification [**40, 41, 14, 63, 64, 65**] due to the lack of novel event-driven sensors and datasets.

With the prevalence of material science and robotics, several tactile sensors have been developed, including non-event-based tactile sensors like the iCub RoboSkin [**66**] and the SynTouch BioTac[**67**] and event-driven tactile sensors like the NeuTouch [**2**] and the NUSkin [**68**]. In this work, we focus on event-driven tactile learning with SNNs. Since the development of event-driven tactile sensors is still in its infancy [**5**], little prior work exists on learning event-based tactile data with SNNs. The work [**42**] 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 [**2**] released the first publicly-available event-driven visual-tactile dataset collected by NeuTouch and proposed an SNN based on SLAYER [**41**] 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 [**5**] utilized the spiking graph neural network [**58**] to process the event-based tactile data and conduct the tactile object recognition. In this work, 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.

2.3. 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 [**48**] and the TLIF [**49**]. 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

$$\mathcal{F}_i = \{t_i^{(f)}; 1 \leq f \leq n\} = \{t | u_i(t) = u_{th}\}, \tag{2.1}$$

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

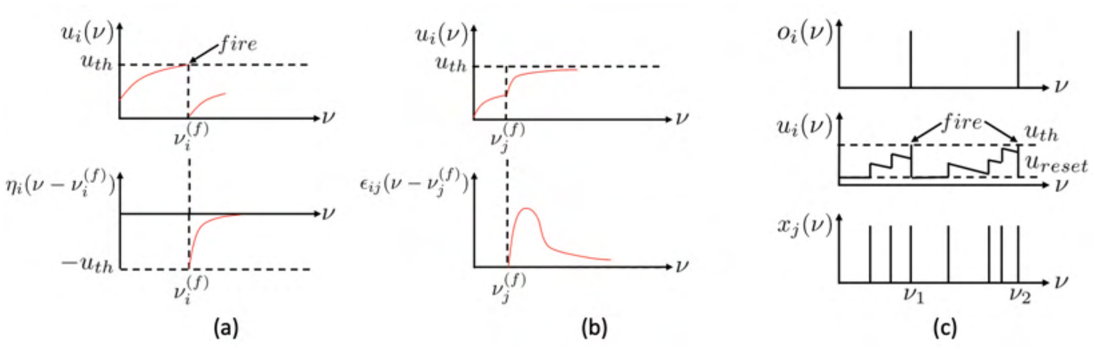


Figure 2.1. **Recurrent neuronal dynamic mechanisms for the existing spiking neurons of $\nu = t$ and location spiking neurons of $\nu = l$. Unlike existing spiking neuron models that update their membrane potentials based on time steps $\nu = t$, location spiking neurons update their membrane potentials 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 "$\geq$" 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}.

different spike response processes:

$$u_i(t) = \sum_{t_i^{(f)} \in \mathcal{F}_i} \eta_i(t - t_i^{(f)}) + \sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in \mathcal{F}_j} w_{ij} x_j(t_j^{(f)}) \epsilon_{ij}(t - t_j^{(f)}), \tag{2.2}$$

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 2.1(a) of $\nu = t$ visualizes the refractory dynamics of the TSRM neuron i and Figure 2.1(b) 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 [**69**]. 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 [**40**]. The dynamics of the TLIF neuron i are governed by

$$\tau \frac{du_i(t)}{dt} = -u_i(t) + I(t), \tag{2.3}$$

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 Eq. (2.3) into a recursive expression:

$$u_i(t) = (1 - \frac{dt}{\tau})u_i(t-1) + \frac{dt}{\tau}\sum_j w_{ij}x_j(t), \tag{2.4}$$

Written on the Wall at West Forest Temple

*By **Su Shi (1037 AD ~1101 AD)***
Tr. Qin Dachuan

From level view it's a ridge, from side view a peak,
The scene changes with angles -- far, near, high or ground.
Th' true features of Mount Lu I can't exactly see,
It is all because I'm within its scope and bounds.

Figure 2.2. The words of Su Shi offer a timeless lesson on perspective: our understanding is deeply influenced by where we stand. His poem encourages us to change our viewpoint, to see the broader context or a different facet of the same situation. This approach doesn't just enrich our appreciation of art or nature, it also applies to analyzing event data.

where $\sum_j w_{ij} x_j(t)$ is the weighted summation of the inputs from pre-neurons at the current time step.

Equation (2.4) can be further simplified as:

$$u_i(t) = \alpha u_i(t-1) + \sum_j w'_{ij} x_j(t), \tag{2.5}$$

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 2.1(c) of $\nu = t$ visualizes the temporal dynamics of a TLIF neuron i.

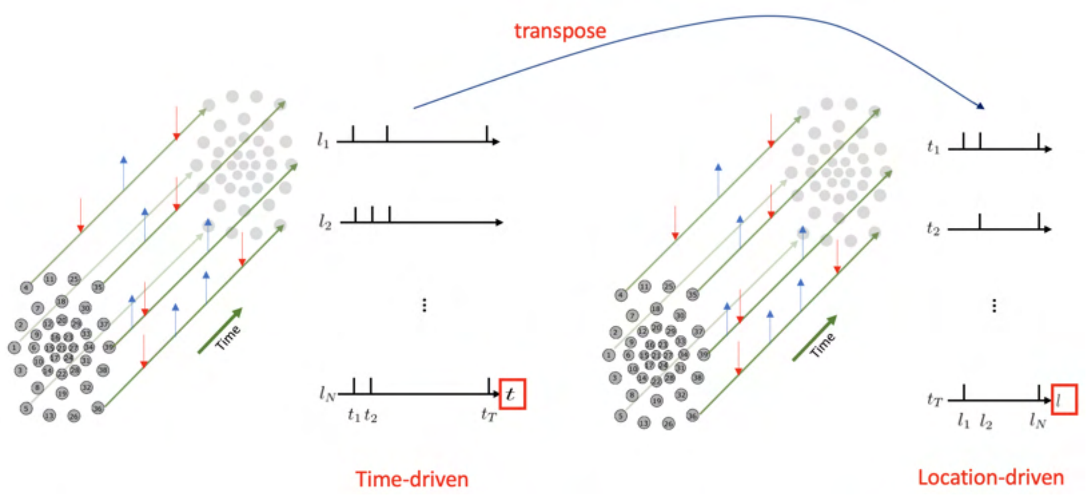


Figure 2.3. Explore event data from a location-driven viewpoint by transposing.

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.

Traditionally, our analysis of event data focuses on the temporal dimension, observing how events unfolded over time T with existing spiking neurons. However, drawing from the insight of Su Shi's poem in Fig. 2.2, we are prompted to consider a spatial perspective as well. As shown in Fig. 2.3, by transposing the time-oriented event data, we shift to examining the sequence of events based on their location N. This reorientation from time to space allows us to uncover patterns and relationships that may not be apparent when viewed solely through the lens of time. It's a transformative shift that enables us to analyze event data from a location-driven viewpoint.

To analyze event data from this location-driven viewpoint and enrich the representative abilities of existing spiking neuron models, we propose the 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

$$\mathcal{G}_i = \{l_i^{(f)}; 1 \leq f \leq n\} = \{l | u_i(l) = u_{th}\}, \tag{2.6}$$

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

$$u_i(l) = \sum_{l_i^{(f)} \in \mathcal{G}_i} \eta_i(l - l_i^{(f)}) + \sum_{j \in \Gamma_i} \sum_{l_j^{(f)} \in \mathcal{G}_j} w_{ij} x_j(l_j^{(f)}) \epsilon_{ij}(l - l_j^{(f)}), \tag{2.7}$$

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 2.1(a) of $\nu = l$ visualizes the refractory dynamics of the LSRM neuron i and Figure 2.1(b) 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

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

for simplicity. In Section 2.4.1, 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 Eq. (2.3), the LLIF neuron i employs the spatial dynamics:

$$\tau' \frac{du_i(l)}{dl} = -u_i(l) + I(l), \tag{2.8}$$

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 Eq. (2.8) into a recursive expression:

$$u_i(l) = (1 - \frac{dl}{\tau'})u_i(l_{prev}) + \frac{dl}{\tau'} \sum_j w_{ij} x_j(l), \tag{2.9}$$

where $\sum_j w_{ij} x_j(l)$ is the weighted summation of the inputs from pre-neurons at the current location. Equation (2.9) can be further simplified as:

$$u_i(l) = \beta u_i(l_{prev}) + \sum_j w'_{ij} x_j(l), \tag{2.10}$$

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 2.1(c) 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 LSRM neurons. In Section 2.4.2, 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.

2.4. 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 work, we focus on processing the data collected by NeuTouch [**2**], a biologically-inspired event-driven fingertip tactile sensor with 39 taxels arranged spatially in a radial fashion (see Fig. 2.4).

2.4.1. 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 2.4 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

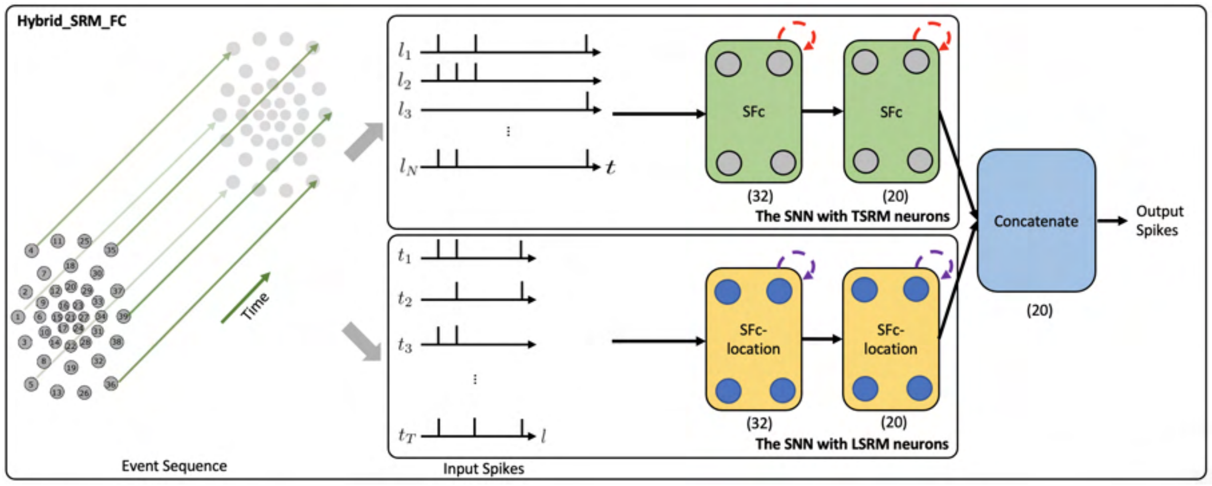


Figure 2.4. The network structure of the Hybrid_SRM_FC. **The Upper Panel:** The SNN with TSRM neurons processes the input spikes X_{in} and adopts the temporal recurrent neuronal dynamics (shown with red dashed arrows) of TSRM neurons to extract features from the data, where SFc is the spiking fully-connected 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.

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 Fig. 2.4 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:

$$
\begin{aligned}
u_i(t) &= \sum_{t_i^{(f)} \in \mathcal{F}_i} \eta(t - t_i^{(f)}) + \underbrace{\sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in \mathcal{F}_j} w_{ij} o_j(t_j^{(f)}) \epsilon(t - t_j^{(f)})}_{\textbf{capture spatial dependencies}}, \\
o_i(t) &= \begin{cases} 1 & \text{if } u_i(t) \geq u_{th}; \\ 0 & \text{otherwise,} \end{cases} \\
\mathcal{F}_i &= \begin{cases} \mathcal{F}_i \cup t & \text{if } o_i(t) = 1; \\ \mathcal{F}_i & \text{otherwise,} \end{cases}
\end{aligned}
\tag{2.11}
$$

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 Fig. 2.4 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 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:

$$
\begin{aligned}
u_i(l) &= \sum_{l_i^{(f)} \in \mathcal{G}_i} \eta(l - l_i^{(f)}) + \underbrace{\sum_{j \in \Gamma'_i} \sum_{l_j^{(f)} \in \mathcal{G}_j} w_{ij} o_j(l_j^{(f)}) \epsilon(l - l_j^{(f)})}_{\textbf{model temporal dependencies}}, \\
o_i(l) &= \begin{cases} 1 & \text{if } u_i(l) \geq u_{th}; \\ 0 & \text{otherwise,} \end{cases} \qquad (2.12) \\
\mathcal{G}_i &= \begin{cases} \mathcal{G}_i \cup l & \text{if } o_i(l) = 1; \\ \mathcal{G}_i & \text{otherwise,} \end{cases}
\end{aligned}
$$

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.** The 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 $T + N$.

2.4.2. 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

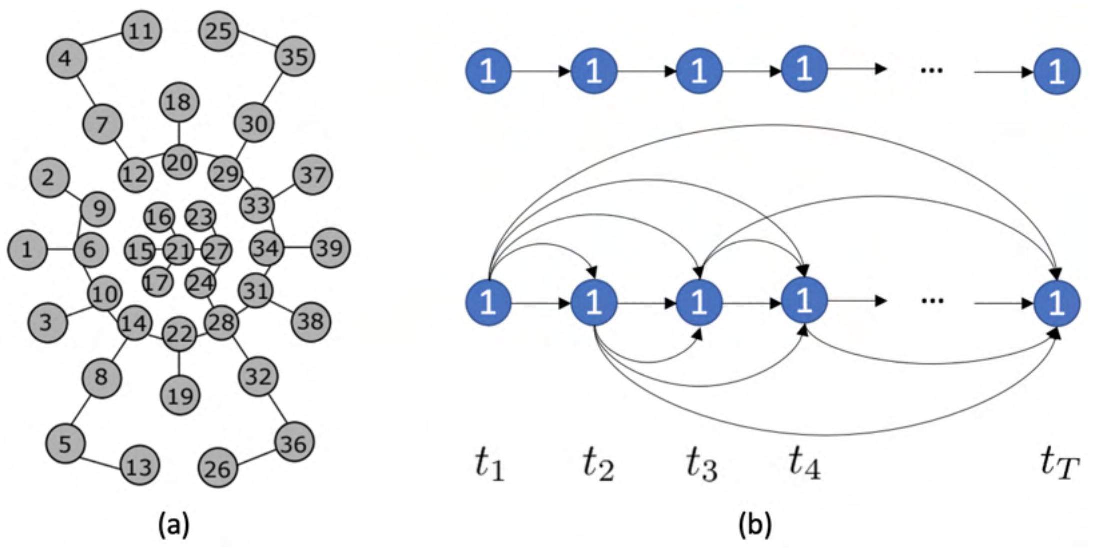


Figure 2.5. **(a)** The tactile spatial graph G_s at time step t generated by the Minimum Spanning Tree (MST) algorithm [**5**]. Each circle represents a taxel of NeuTouch. **(b)** Based on event sequences, we propose two different tactile temporal graphs G_t for a specific taxel $n = 1$: the above one is the sparse tactile temporal graph, while the below one is the dense tactile temporal graph.

Hybrid_LIF_GNN, which fuses spatial and temporal spiking graph neural networks and captures complex spatio-temporal dependencies in the event-based tactile data.

2.4.2.1. 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 Fig. 2.5.

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**. $V^t = \{v_n^t | n = 1, ..., N\}$ 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 Fig. 2.5(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.

2.4.2.2. 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 Fig. 2.6), which fuses spatial and temporal spiking graph neural networks. Specifically, we adopt the spatial spiking graph neural network with TLIF neurons [**5**], which is a spike-based tactile learning framework powered by STBP [**40**]. It uses temporal recurrent 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.

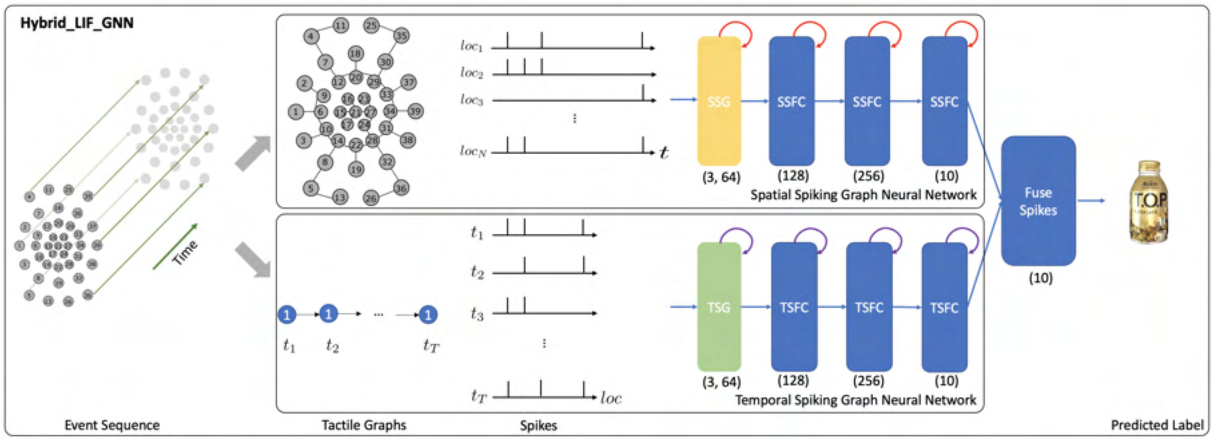


Figure 2.6. The structure of the Hybrid_LIF_GNN, where "SSG" is the spatial spiking graph layer, "SSFC" is the spatial spiking fully-connected layer, "TSG" is the temporal spiking graph layer, and "TSFC" is the temporal spiking fully-connected layer. The spatial spiking graph neural network processes 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.

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 Eq. (2.5), 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:

$$\begin{aligned} & u_i(t) = \alpha u_i(t-1)(1 - o_i(t-1)) + I(t), \\ & o_i(t) = \begin{cases} 1 & \text{if } u_i(t) \geq u_{th}; \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{2.13}$$

where $I(t) = \mathbf{GNN}(G_s(t))$ **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 Eq. (2.13), where $I(t) = \mathbf{FC}(Pre(t))$ and $Pre(t)$ is the previous layer's output at time step t.

Based on Eq. (2.10), 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:

$$\begin{aligned} & u_i(l) = \beta u_i(l_{prev})(1 - o_i(l_{prev})) + I(l), \\ & o_i(l) = \begin{cases} 1 & \text{if } u_i(l) \geq u_{th}; \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{2.14}$$

where $I(l) = \mathbf{GNN}(G_t(l))$ **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 Eq. (2.14), where $I(l) = \mathbf{FC}(Pre(l))$ 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 [**70**] as **GNN** in this work.

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,

$$O_1' = \frac{1}{T}\sum_{t}^{T} O_1(t), \tag{2.15}$$

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,

$$O_2' = \frac{1}{N}\sum_{l}^{N} O_2(l), \tag{2.16}$$

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.

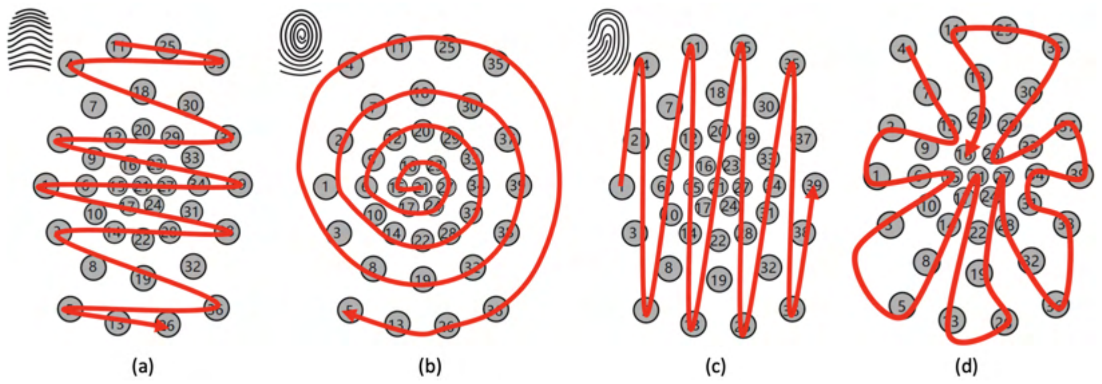


Figure 2.7. Location orders. **(a)** Arch-like location order. **(b)** Whorl-like location order. **(c)** Loop-like location order. **(d)** Random location order.

2.5. 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.

2.5.1. 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 Fig. 2.7, 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]

2.5.2. Hybrid SRM FC

Similar to the spike-count loss of prior works [**41, 2**], we propose a location spike-count loss to optimize the SNN with LSRM neurons:

$$\mathcal{L}_{LSRM} = \frac{1}{2}\sum_{k=0}^{K}\left(\sum_{l=0}^{N} o_k(l) - \sum_{l=0}^{N} \hat{o}_k(l)\right)^2, \tag{2.17}$$

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:

$$\mathcal{L}_1 = \frac{1}{2}\sum_{k=0}^{K}\left(\left(\sum_{t=0}^{T} o_k(t) + \lambda\sum_{l=0}^{N} o_k(l)\right) - \sum_{c=0}^{T+N} \hat{o}_k(c)\right)^2, \tag{2.18}$$

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 [**2**]. To overcome the non-differentiability of spikes and apply the backpropagation algorithm, we use the approximate gradient proposed in SLAYER [**41**]. 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.

Algorithm 1 Timestep-wise inference algorithm of the Hybrid_SRM_FC, adopted from [**52**]

```
Require: event-based tactile inputs $X_{in} \in \mathbb{R}^{N \times T}$, $N$ taxels, and the total time length $T$.
Ensure: 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 $\bar{X}' = concatenate(X', \mathbf{0}) \in \mathbb{R}^{T \times N}$, 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:     $O_1(t) =$ SNN_TSRM($X$)  ▷ SNN_TSRM for the fully-connected SNN with TSRM neurons
7:     $O_2(t) =$ SNN_LSRM($\bar{X}'$)  ▷ SNN_LSRM for the fully-connected SNN with LSRM neurons
8:     $O(t) = concatenate(O_1(t), O_2(t))$
9: end for
```

2.5.3. 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'.

$$\mathcal{L}_2 = \|y - O'\|^2. \tag{2.19}$$

We utilize the spatio-temporal backpropagation [**40**] to derive the weight update for the SNNs with LLIF neurons. Moreover, to overcome the non-differentiability of spikes, we use the rectangular function [**40**] to approximate the derivative of the spike function (Heaviside function) in Eqs. (2.13) and (2.14). Please check more implementation details in our Github repository. Algorithm 2 presents the timestep-wise inference algorithm of the Hybrid_LIF_GNN.

Algorithm 2 Timestep-wise inference algorithm of the Hybrid_LIF_GNN

```
Require: event-based tactile inputs $X_{in} \in \mathbb{R}^{N \times T}$, $N$ taxels, and the total time length $T$
Ensure: 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 $\bar{X}' = concatenate(X', \mathbf{0}) \in \mathbb{R}^{T \times N}$, 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:         $O_1'(t)$ += SSGNN($G_s(i)$)      ▷ SSGNN for the spatial spiking graph neural network
 8:     end for
 9:     $O_1'(t)$ /= $t$
10:     for $j \leftarrow 1$ 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
```

2.6. 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`.

2.6.1. 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.

2.6.1.1. Datasets. We use the datasets collected by NeuTouch [**2**], 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%, 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 provide more details about the datasets in the Supplementary Material. Following

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

Table 2.1. Accuracies on benchmark datasets for the Hybrid_SRM_FC

Method	Type	Objects-v1	Containers-v1	Slip Detection
Tactile-SNN [**2**]	SNN	0.75	0.57*	0.82*
TactileSGNet [**5**]	SNN	0.79	0.58	0.97
GRU-MLP [**2**]	ANN	0.72	0.46*	0.87*
CNN-3D [**2**]	ANN	0.90	0.67*	0.44*
Hybrid_SRM_FC	SNN	**0.91**	**0.86**	**1.0**

*These values come from [**2**]. The best performance is in bold.

the experimental setting of [**2**], 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.

2.6.1.2. Comparing Models. We compare our model with the state-of-the-art SNN methods for event-driven tactile learning, including Tactile-SNN [**2**] and TactileSGNet [**5**]. 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 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 [**2**], we also compare our model against conventional deep learning, specifically Gated Recurrent Units (GRUs) [**71**] with Multi-layer Perceptrons (MLPs) and 3D convolutional neural networks (3D_CNN) [**72**]. 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.

2.6.1.3. Basic Performance. Table 2.1 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 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 2.1, 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.

2.6.1.4. 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 2.2, 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 two-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,

Table 2.2. Ablation studies on the Hybrid_SRM_FC

Method	Type	Objects-v1	Containers-v1	Slip Detection
Tactile-SNN [**2**]	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

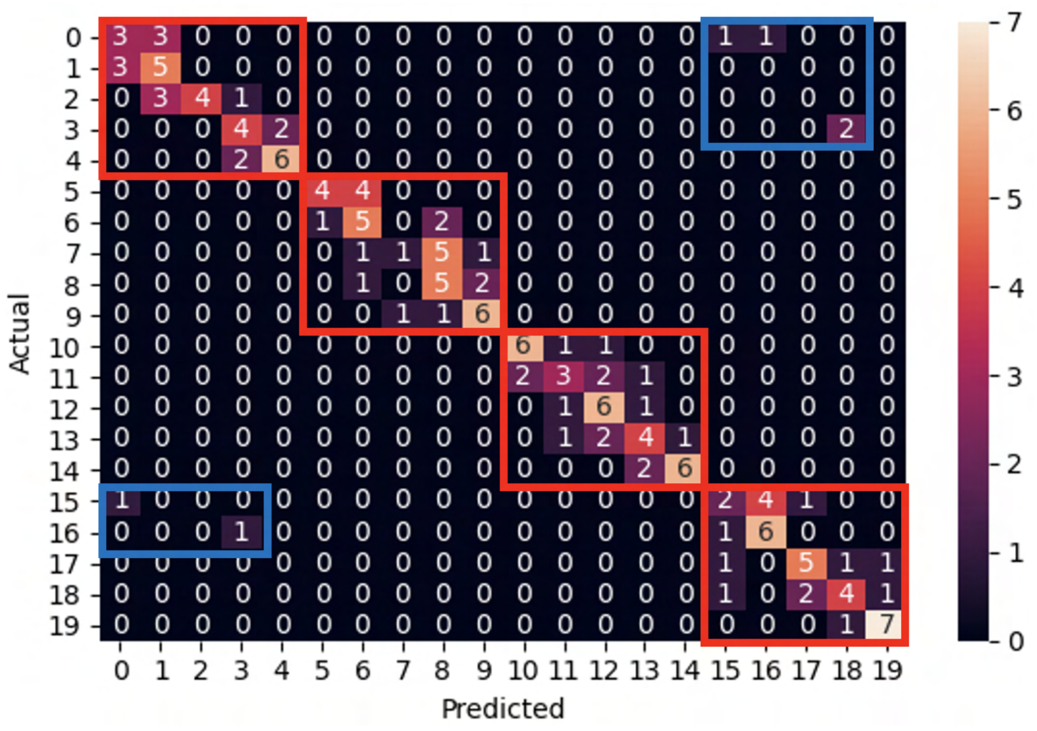


Figure 2.8. The confusion matrix of Tactile-SNN on “Containers”.

we examine the sensitivity of λ in Eq.(2.18) and the robustness of location orders. From Table 2.2, we notice the results of related models are close, proving that the λ tuning and location orders do not significantly impact task performance.

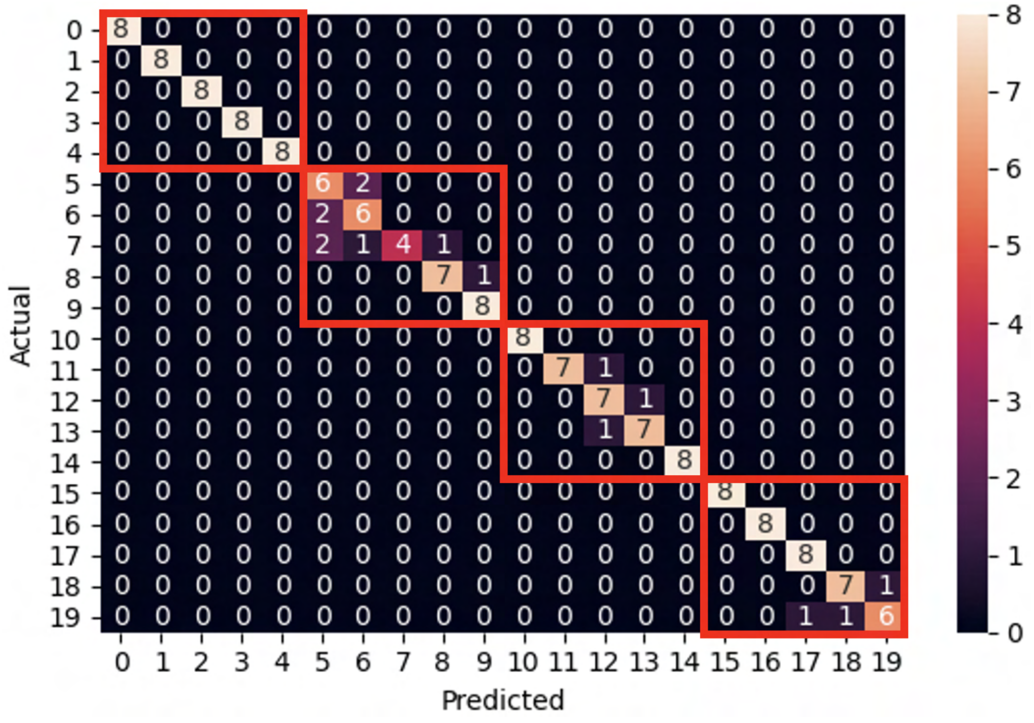


Figure 2.9. The confusion matrix of Hybrid_SRM_FC on "Containers".

2.6.1.5. Confusion Matrices. We calculate the confusion matrices of Tactile-SNN (Fig. 2.8) and Hybrid_SRM_FC (Fig. 2.9) on "Containers" since it is a more challenging event-driven tactile object recognition dataset. From the two figures, we can see that our hybrid model can perfectly distinguish the different containers. Each red box in the figures represents a type of container, and each blue box in the figures represents the container misclassification. Moreover, compared to Tactile-SNN, we observe that our model can recognize the container fullness with a higher accuracy since the misclassification number in each red box is fewer for our model.

2.6.1.6. 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

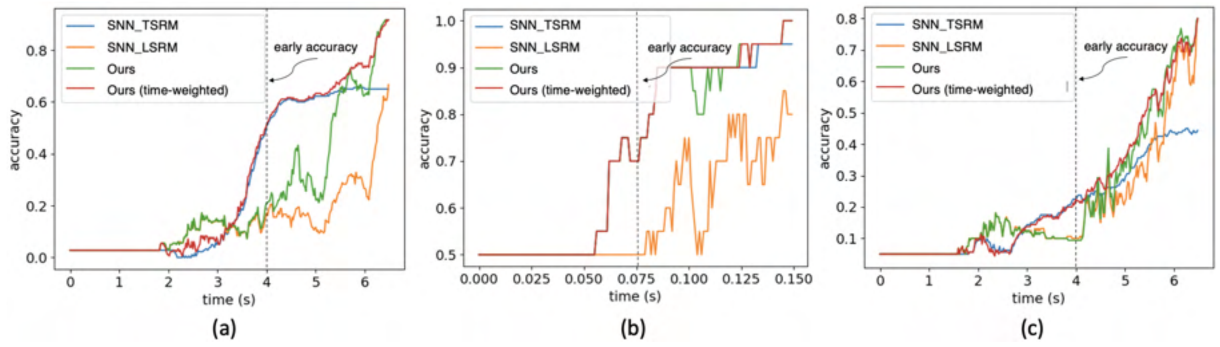


Figure 2.10. The timestep-wise inference (Alg. 1) for the SNN with TSRM neurons (SNN_TSRM), the SNN with LSRM neurons (SNN_LSRM), the Hybrid_SRM_FC, and the time-weighted Hybrid_SRM_FC on **(a)** "Objects-v1", **(b)** "Slip Detection", **(c)** "Containers-v1". Please note that we use the same event sequences as [**2**] and the first spike occurs at around 2.0s for "Objects-v1" and "Containers-v1".

components' contributions and achieve better overall performance. Figure 2.10 shows the timestep-wise inference accuracies of 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

$$\begin{aligned} O_{tw}(t) &= concatenate((1-\omega) * O_1(t), \omega * O_2(t)), \\ \omega &= \frac{1}{1+e^{-\psi*(\frac{t}{T}-1)}}, \end{aligned} \tag{2.20}$$

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 Eq. (2.11). 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 Eq. (2.12). 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.

2.6.1.7. 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 [**58, 73**]. In addition, we can estimate the total energy consumption of a model based on CMOS technology [**74**].

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:

$$F_1^{(l)} = \frac{1}{T} \sum_{t \in T} \frac{\#\text{spikes of layer } l \text{ at time } t}{\#\text{neurons of layer } l}, \tag{2.21}$$

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:

$$F_2^{(l)} = \frac{1}{N} \sum_{n \in N} \frac{\#\text{spikes of layer } l \text{ at location } n}{\#\text{neurons of layer } l}, \tag{2.22}$$

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:

$$OP_{Hybrid} = \sum_{l} M^{(l)} \times C^{(l)} \times F_1^{(l)} \times T + \sum_{l'} M^{(l')} \times C^{(l')} \times F_2^{(l')} \times N, \tag{2.23}$$

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 2.3, 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 example, an AC is reported to be **5.1**× more energy-efficient than a MAC in the case of 32-bit floating-point numbers (45nm CMOS process) [**74**]. Based on this principle, we obtain the computational energy benefits of SNNs over ANNs in Table 2.3. 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.

Table 2.3. The number of synaptic operations (#*op*, $\times 10^6$) and the compute-energy benefit (the compute-energy of ANNs / the compute-energy of SNNs, 45nm) 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×

2.6.2. 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 [**40**]. 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]

2.6.2.1. Datasets. To fairly compare with other published models with TLIF neurons [**5**], 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.

2.6.2.2. Comparing Models. We compare the Hybrid_LIF_GNN with the state-of-the-art methods with TLIF neurons and GNN structures [**5**] 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

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

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 Eq. (2.13). 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 Eqs. (2.13) and (2.14) while our model utilizes the spiking neurons and takes the Heaviside function as the activation function.

2.6.2.3. Basic Performance. We report the test accuracies on the two event-driven tactile object recognition datasets in Table 2.4. From this table, we can see that the Hybrid_LIF_GNN significantly outperforms the TactileSGNet series [**5**]. The reason why our model can achieve the better performance could be two-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 Eqs. (2.13) and (2.14) with activation functions, such as linear, elu, or

Table 2.4. Accuracies (%) on datasets for the Hybrid_LIF_GNN

Method	Type	Objects-v0	Containers-v0
TactileSGNet-MLP [**5**]	SNN	85.97*	58.83*
TactileSGNet-CNN [**5**]	SNN	88.40*	60.17*
TactileSGNet-GCN [**5**]	SNN	85.14*	58.83*
TactileSGNet-TAGConv [**5**]	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 [**5**]. 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.

LeakyRelu. These models provide fair comparison baselines for fully GNN architectures since they employ the same network architecture as ours. From Table 2.4, 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.

2.6.2.4. Ablation Studies. We further provide ablation studies for exploring the optimal design choices. From Table 2.4, 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.

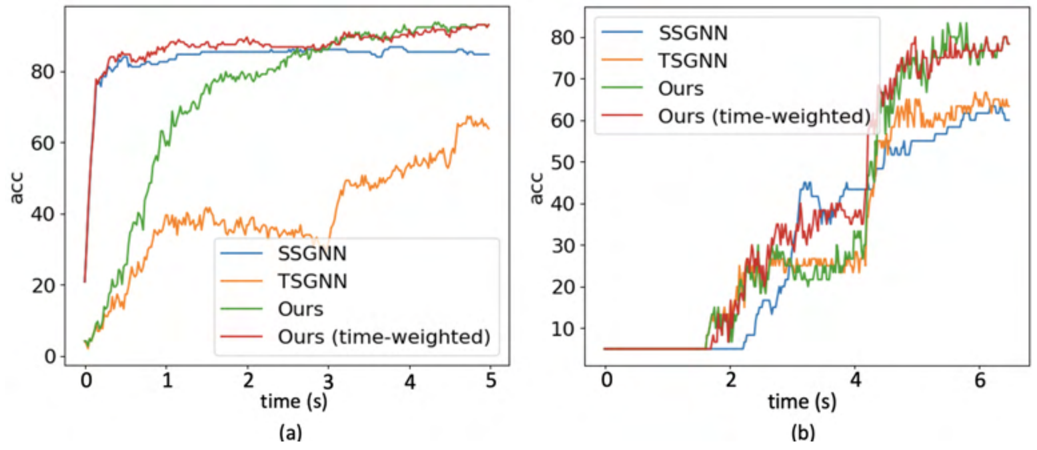


Figure 2.11. The timestep-wise inference (Alg. 2) accuracies (%) for the spatial spiking graph neural network (SSGNN), the temporal spiking graph neural network (TSGNN), the Hybrid_LIF_GNN, and the time-weighted Hybrid_LIF_GNN on **(a)** "Objects-v0" and **(b)** "Containers-v0".

2.6.2.5. Timestep-wise Inference. Figure 2.11 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

$$O'_{tw}(t) = O'_1(t)(1 - \frac{t}{\zeta T}) + O'_2(t)\frac{t}{\zeta T}, \tag{2.24}$$

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 Fig. 2.11.

2.6.2.6. Energy Efficiency. Following the estimation methods in Section 2.6.1.7, 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 2.5 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 45nm CMOS technology energy principle in Section 2.6.1.7, we obtain the computational energy benefits of SNNs over GNNs in Table 2.5. From the table, we can see that the SNN models are **10**$\times$ to **100**$\times$ energy-efficient than GNNs. Furthermore, by 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 2.6.1.7 and demonstrate the potential of our models and location spiking neurons (LLIF neurons) on neuromorphic hardware.

Table 2.5. The number of synaptic operations (#*op*, $\times 10^8$) and the compute-energy benefit (the compute-energy of GNNs / the compute-energy of SNNs, 45nm) on benchmark datasets for the Hybrid_LIF_GNN

Method	Type	Objects-v0	Containers-v0
#*op* Recurrent GNNs in Table 2.4	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×

2.6.2.7. Performance Comparison with the Hybrid SRM FC. To fairly compare with the Hybrid_SRM_FC (Fig.2.4), we further test the Hybrid_LIF_GNN (Fig.2.6) on "Objects-v1", "Containers-v1", and "Slip Detection". From Table 2.6, 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 2.1 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.

2.7. 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

Table 2.6. 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.

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 chapter.

2.7.1. Advantages and Limitations of Conventional and Location Spiking Neurons

This work 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 Eqs. (2.2), (2.5), (2.11), and (2.13). 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 Eqs. (2.7), (2.10), (2.12), and (2.14).

Moreover, based on experimental results, we can see that conventional spiking neurons are better at capturing **spatial 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.

2.7.2. Potential Impact on Broad Spike-Based Learning Applications

In this work, 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.

2.7.2.1. Event-Driven Audio Learning. To show the potential impact of our work, we apply the Hybrid_SRM_FC (see Fig. 2.4) 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.

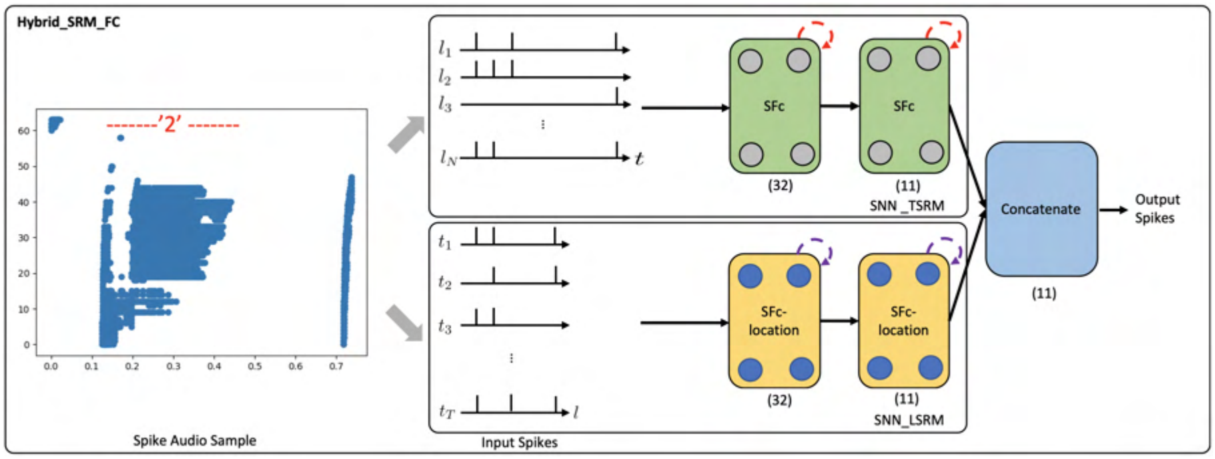


Figure 2.12. The Hybrid_SRM_FC processes a spike audio sequence and predict its label. The network structure of this model is the same as what we show in Fig. 2.4.

In the experiment, we use the N-TIDIGITS18 dataset [**3**], which is collected by playing the audio files from the TIDIGITS dataset [**75**] to the dynamic audio sensor – the CochleaAMS1b sensor [**76**]. 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'. A spike audio sequence of digit '2' is shown in Fig. 2.12, 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 Fig. 2.12. 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.

2.7.2.2. Visual Processing. Besides event-driven audio learning, a contemporary work [**77**] 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 [**78**] or Transformer [**79**] frameworks.

2.7.2.3. 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.

CHAPTER 3

Event-based Shape from Polarization with Spiking Neural Networks

This chapter is based on the paper [**55**].

3.1. Introduction

Precise surface normal estimation can provide valuable information about a scene's geometry and is useful for many computer vision tasks, including 3D Reconstruction [**80**], Augmented Reality (AR) and Virtual Reality (VR) [**81, 82**], Material Classification [**83**], and Robotics Navigation [**84**]. Depending upon the requirements of the application, surface normal estimation can be carried out using a variety of methods [**85, 86, 87, 88, 89, 90**]. In this work, we are interested in estimating surface normal from polarization images – shape from polarization [**91, 92, 93, 94, 95, 96**]. In particular, shape from polarization leverages the polarization state of light to infer the shape of objects. When light reflects off surfaces, it becomes partially polarized. This method uses this property to estimate the surface normals of objects, which are then used to reconstruct their 3D shape. Compared to other 3D sensing methods, shape from polarization has many advantages [**92, 97**], such as its suitability for capturing fine details on a variety of surface materials, including reflective and transparent ones, and its reliance on passive sensing, which eliminates the need for external light sources or emitters. Additionally, shape from polarization can provide high-precision data with relatively low-cost and low-energy equipment, making it an efficient and versatile option for 3D imaging in various applications.

Typically, a polarizing filter is used in conjunction with a camera to capture the polarization images and infer the polarization information. Generally, there are two ways to capture the polarization images and estimate the surface normals from them, one is *Division of Time (DoT)* [**92, 98, 99**] and the other one is *Division of Focal Plane (DoFP)* [**95, 96, 100**]. The DoT approaches add a rotatable linear polarizer in front of the lens of an ordinary camera. The filter is rotated to different orientations, and full-resolution polarization images are captured for each orientation at different times. By analyzing the changes in the polarization state of light across these images, the surface normals of objects can be estimated. The DoT methods use the full resolution of the sensor but trade-off against acquisition time. On the other hand, the DoFP methods place an array of micro-polarizers in front of the camera [**100**]. This allows the camera to capture polarization information at different orientations in a single shot. Despite the reduced latency, this system is limited by the low resolution of polarization images, as each pixel only captures polarization at a specific orientation. This can result in lower accuracy compared to the DoT methods.

To bridge the accuracy of DoT with the speed of DoFP, researchers propose event-based shape from polarization following the DoT design scheme [**101**]. Specifically, a polarizer is rotating in front of an event camera [**1**] and this creates sinusoidal changes in brightness intensity. Unlike traditional DoT methods utilize standard cameras to capture full-resolution polarization images at fixed rates, event-based shape from polarization employs event cameras to asynchronously measure changes in brightness intensity for each pixel within the full-resolution scene and trigger the events with microsecond resolution if the difference in brightness exceeds a threshold. The proposed event-based method uses the continuous event stream to reconstruct relative intensities at multiple polarizer angles. These reconstructed polarized images are then utilized to estimate surface normals using physics-based and learning-based methods [**101**]. Due to the DoT-driven characteristic and low latency event cameras provide, the event-based shape from polarization mitigates the accuracy-speed trade-off in the traditional shape from polarization field.

Although the event-based shape from polarization brings many advantages, we still need to carefully choose models that process the data from event cameras. With the prevalence of Artificial Neural Networks (ANNs), one recent method [**101**] employs ANNs to process event data and demonstrates the better surface normal estimation performance compared to physics-based methods. However, ANNs are not compatible with the working mechanism of event cameras and incur the high energy consumption. To be more compatible with event cameras and maintain the high energy efficiency, research on Spiking Neural Networks (SNNs) [**36**] has started to gain momentum. Similar to event cameras that mimic the human retina's way of responding to changes in light intensity, SNNs are also bio-inspired and designed to emulate the neural dynamics of human brains. Unlike ANNs employing artificial neurons [**45, 46, 47**] and conducting real-valued computation, SNNs adopt spiking neurons [**48, 49, 50**] and utilize binary 0-1 spikes to process information. This difference reduces the mathematical dot-product operations in ANNs to less computational summation operations in SNNs [**36**]. Due to such the advantage, SNNs are always energy-efficient and suitable for power-constrained devices. Although SNNs demonstrate the higher energy efficiency and much dedication has been devoted to SNN research, ANNs still present the better performance and dominate in a wide range of learning applications [**102**].

Recently, more research efforts have been invested to shrink the performance gap between ANNs and SNNs. And SNNs have achieved comparable performance in various tasks, including image classification [**103**], object detection [**104**], graph prediction [**105**], natural language processing [**106**], etc. Nevertheless, we have not yet witnessed the establishment of SNN in the accurate surface normal estimation with an advanced performance. To this end, this naturally raises an issue: *could bio-inspired Spiking Neural Networks estimate surface normals from event-based polarization data with an advanced quality at low energy consumption?*

In this work, we investigate the event-based shape from polarization with a spiking approach to answer the above question. Specifically, inspired by the feed-forward UNet [**107**] for event-based shape from polarization [**101**], we propose the Single-Timestep Spiking UNet, which treats the event-based shape from polarization as a non-temporal task. This model processes event-based inputs in a feed-forward manner, where each spiking neuron in the model updates its membrane potential only once. Although this approach may not maximize the temporal processing capabilities of SNNs, it significantly reduces the computational and energy requirements. To further exploit the rich temporal information from event-based data and enhance model performance in the task of event-based shape from polarization, we propose the Multi-Timestep Spiking UNet. This model processes inputs in a sequential, timestep-by-timestep fashion, allowing each spiking neuron to utilize its temporal recurrent neuronal dynamics to more effectively extract information from event data. We extensively evaluate the proposed models on the synthetic and real-world datasets for event-based shape from polarization. The results of these experiments, both quantitatively and qualitatively, indicate that our models are capable of estimating dense surface normals from polarization events with performance comparable to current state-of-the-art ANN models. Additionally, we perform ablation studies to assess the impact of various design components within our models, further validating their effectiveness. Furthermore, our models exhibit superior energy efficiency compared to their ANN counterpart, which highlights their potential for application on neuromorphic hardware and energy-constrained edge devices.

The remainder of this chapter is structured as follows: Section 3.2 provides a comprehensive review of existing literature on shape from polarization and SNNs. Section 3.3 describes the input event representation and Section 3.4 presents the spiking neuron models involved in this work. In Section 3.5, we detail our proposed SNN models for event-based shape from polarization, including their structures, training protocols, and implementation details. Section 3.6 showcases the effectiveness and energy efficiency of our proposed models on different benchmark datasets. The chapter concludes with Section 3.7, where we summarize our findings and outline potential avenues for future research.

3.2. Related Work

In the following, we will first give an overview of the related work on shape from polarization, including the traditional shape from polarization and event-based shape from polarization. Then, we will give a comprehensive review of SNNs and their applications in 3D scenes.

3.2.1. Shape from Polarization

Shurcliff proposed the method of shape recovery by polarization information in 1962 [**108**]. Essentially, when unpolarized light reflects off a surface point, it becomes partially polarized. And the observed scene radiance varies with changing the polarizer angle, which encodes some relationship with surface normals. Therefore, by analyzing such relationship at each surface point through Fresnel equations [**109**], shape from polarization methods can measure the azimuthal and zenithal angles at each pixel and recover the per-pixel surface normal with high resolution. Generally, two schemes are utilized to collect polarization images. One is *Division of Time (DoT)* [**92, 98, 99**] that provides full-resolution polarization images but increases the acquisition time significantly, while the other one is *Division of Focal Plane (DoFP)* [**95, 96, 100**] that trade-offs spatial resolution for low latency. After collecting the polarization images, various physical-based or learning-based methods [**110**] can be utilized to estimate the surface normals. However, since a linear polarizer cannot distinguish between polarized light that is rotated by π radians, this results in two confounding estimates for the azimuth angle at each pixel [**95, 111**]. To solve such ambiguity, we have to carefully design the estimation methods by exploring additional constraints from various aspects, such as geometric cues [**112, 113, 114**], spectral cues [**93, 115, 116**], photometric cues [**94, 117, 118**], or priors learned from deep learning techniques [**95, 96**].

Recently, with the prevalence of bio-inspired neuromorphic engineering, researchers have begun to shift their focus to high-speed energy-efficient event cameras and propose solutions that combine polarization information with event cameras. Specifically, inspired by the polarization vision in the mantis shrimp eye [**119**], [**120**] proposed the PDAVIS polarization event camera. The researchers employed the DoFP scheme to design such the camera, which involved fabricating an array of pixelated polarization filters and strategically positioning them atop the sensor of an event camera. While this camera is adept at capturing high dynamic range polarization scenes with high speeds, it still faces challenges with low spatial resolution, a common issue inherent in the DoFP methods. To bridge the high resolution of DoT with the low latency of DoFP, [**101**] adopted the DoT scheme and collected polarization events by placing a rotating polarizing filter in front of an event camera. Due to the high resolution of DoT and the low latency of event cameras, this method facilitates shape from polarization at both high speeds and with high spatial resolution. Typically, the captured polarization events are transformed into frame-like event representations [**35**], which are then processed using ANN models [**101**] to estimate surface normals. While these learning-based methods demonstrate superior performance over traditional physics-based methods, they significantly increase the energy consumption of the overall system, primarily due to the lower energy efficiency of ANNs. Through processing event polarization data collected by the promising DoT scheme, this work aims to address this challenge by conducting event-based shape from polarization using SNNs, presenting a more energy-efficient alternative in this domain.

3.2.2. Spiking Neural Networks

With the development of ANNs, artificial intelligence models today have demonstrated extraordinary abilities in many tasks, such as computer vision, natural language processing, and robotics. Nevertheless, ANNs only mimic the brain's architecture in a few aspects, including vast connectivity and structural and functional organizational hierarchy [**36**].

The brain has more information processing mechanisms like the neuronal and synaptic functionality [**59, 60**]. Moreover, ANNs are much more energy-consuming compared to human brains.

To integrate more brain-like characteristics and make artificial intelligence models more energy-efficient, researchers propose SNNs, which can be executed on power-efficient neuromorphic processors like TrueNorth [**61**] and Loihi [**62**]. Like ANNs, SNNs are capable of implementing common network architectures, such as convolutional and fully-connected layers, yet they distinguish themselves by utilizing spiking neuron models [**50**], such as the Leaky Integrate-and-Fire (LIF) model [**49**] and the Spike Response Model (SRM) [**48**]. Due to the non-differentiability of these spiking neuron models, training SNNs can be challenging. However, progress has been made through innovative approaches such as converting pre-trained ANNs to SNNs [**63, 64**] and developing methods to approximate the derivative of the spike function [**40, 65**]. Thanks to the developement of these optimization techniques, many learning-based fully SNNs have been proposed recently. Nevertheless, most of the learning-based fully SNN work has so far focused on event-driven classification problems, little prior work exists on event-driven regression problems. Specifically, [**121**] presented the first learning-based fully SNN on a simple regression problem. It utilized a convolutional spiking encoder with SRM neurons to predict angular velocities from event data. [**44**] presented the first fully-deep SNN model on large-scale event-driven optical flow estimation. And [**43**] built the first fully-deep SNN model on large-scale event-driven image reconstruction.

In this work, we focus on tackling the complex regression tasks in 3D scenes. Notably, StereoSpike [**122**] and MSS-DepthNet [**123**] have pioneered the development of deep SNNs for depth estimation, achieving performance on par with the state-of-the-art ANN models. Additionally, SpikingNeRF [**124**] has successfully adapted SNNs for radiance field reconstruction, yielding synthesis quality comparable to ANN baselines while maintaining high energy efficiency. In this work, our emphasis is on employing SNNs to tackle event-based shape from polarization, aiming to establish a method that is not only effective but also more efficient for event-based surface normal estimation.

3.3. Input Event Representation

In this work, we focus on building SNNs to estimate surface normals through the use of a polarizer paired with an event camera. In this setup, the polarizer is mounted in front of the event camera and rotates at a constant high speed driven by a motor. This rotation changes the illumination of the incoming light. Event cameras generate an asynchronous event $e_i = (x_i, y_i, t_i, p_i)$ when the illumination variation at a given pixel reaches a given contrast threshold C:

$$L(x_i, y_i, t_i) - L(x_i, y_i, t_i - \Delta t_i) = p_i C, \tag{3.1}$$

where $L \doteq log(I)$ is the log photocurrent ("brightness"), $p_i \in \{-1, +1\}$ is the sign of the brightness change, and Δt_i is the time since the last event at the pixel (x_i, y_i).

The surface normal vector can be represented by its azimuth angle α and zenith angle θ in a spherical coordinate system. And the proposed models predict the surface normal $\mathbf{N}$ as a 3-channel tensor $\mathbf{N} = (\sin\theta\cos\alpha, \sin\theta\sin\alpha, \cos\theta)$ through the event steam.

To ensure a fair comparison between our proposed methods and those utilizing ANNs for event-based shape from polarization, we transform the sparse event stream into frame-like event representations, which serve as the input for our methods. Specifically, similar to [**101**], we take the CVGR-I representation due to its superior performance. The CVGR-I representation combines the Cumulative Voxel Grid Representation (CVGR) with a single polarization image (I) taken at a polarizer angle of 0 degrees. The CVGR is a variation of the voxel grid [**35**]. Similar to previous works on learning with events [**28, 125**], the CVGR first encodes the events in a spatio-temporal voxel grid V. Specifically, the time domain of the event stream is equally discretized into B temporal bins indexed by integers in the range of $[0, B-1]$. Each event $e_i = (x_i, y_i, t_i, p_i)$ distributes its sign value p_i to the two closest spatio-temporal voxels as follows:

$$V(x,y,t) = \sum_{x_i=x, y_i=y} p_i \max(0, 1-|t-t_i^*|), \quad t_i^* = \frac{B-1}{\Delta T}(t_i - t_0), \tag{3.2}$$

where (x, y, t) is a specific location of the spatio-temporal voxel grid V, ΔT is the time domain of the event stream, and t_0 is the timestep of the initial event in the event stream. Then, the CVGR calculates the cumulative sum across the bins and multiplies this total by the contrast threshold:

$$E(x,y,b) = C \sum_{i=0}^{b} V(x,y,i), \quad b = \{0,1,2,3,...,B-1\}, \tag{3.3}$$

Finally, to enhance surface normal estimation in areas with insufficient event information, a single polarization image of 0 polarizer degree is incorporated, resulting in $E = I[0] + E$, thereby providing additional context. This resulting event representation E will serve as

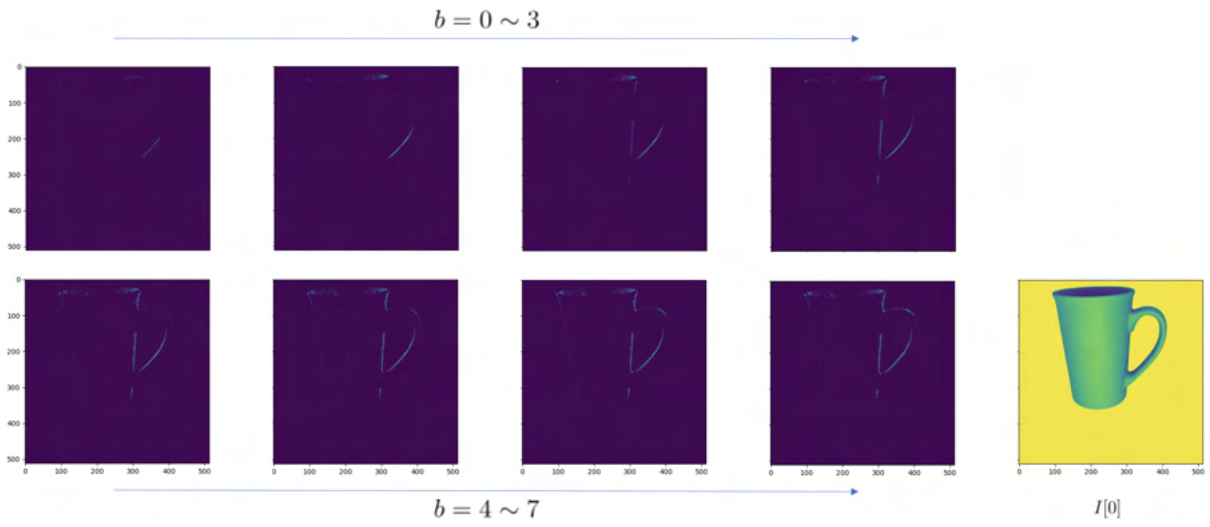


Figure 3.1. The CVGR-I input representation comprises CVGR frames spanning B temporal bins, along with a single polarization image captured at a polarizer angle of 0 degrees. In this example, we set $B = 8$.

the input of our models. Its dimensions are $B \times H \times W$, where H and W represent the height and width of the event camera, respectively. We present a concrete input example of "cup" in Fig. 3.1.

3.4. 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 work, we primarily concentrate on using the Integrate-and-Fire (IF) model [**49**] to develop our proposed SNNs. The IF model is one of the earliest and simplest spiking neuron models. The dynamics of IF neuron i is defined as:

$$u_i(t) = u_i(t-1) + \sum_j w_{ij} x_j(t), \tag{3.4}$$

where $u_i(t)$ represents the internal membrane potential of neuron i at time t, $u_i(t-1)$ is the membrane potential of neuron i at the previous timestep $t-1$, and $\sum_j w_{ij}x_j(t)$ is the weighted summation of the inputs from pre-neurons at the current time step t. 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.

In addition to the IF model, we also build our proposed models with the Leaky Integrate-and-Fire (LIF) model [**49**]. Compared to the IF model, LIF model contains a leaky term to mimic the diffusion of ions through the membrane. The dynamics of LIF neuron i can be expressed as:

$$u_i(t) = \alpha u_i(t-1) + \sum_j w_{ij}x_j(t), \tag{3.5}$$

where α is a leaky factor that decays the membrane potential over time. Drawing inspiration from previous work [**126**], we also construct models using the Parametric Leaky Integrate-and-Fire (PLIF) model, which enables automatic learning of the leaky factor. In our experiments, we demonstrate that the IF model can offer better performance as it retains more information by not incorporating the leaky factor, thus striking a balance between high performance and biological plausibility.

3.5. SNNs for Event-based Shape from Polarization

In this section, we propose two SNNs that take the CVGR-I event representation as the input and estimate the surface normals **N**. Both of them can process the information through the spiking neuron models mentioned above. Due to the potential of IF neurons

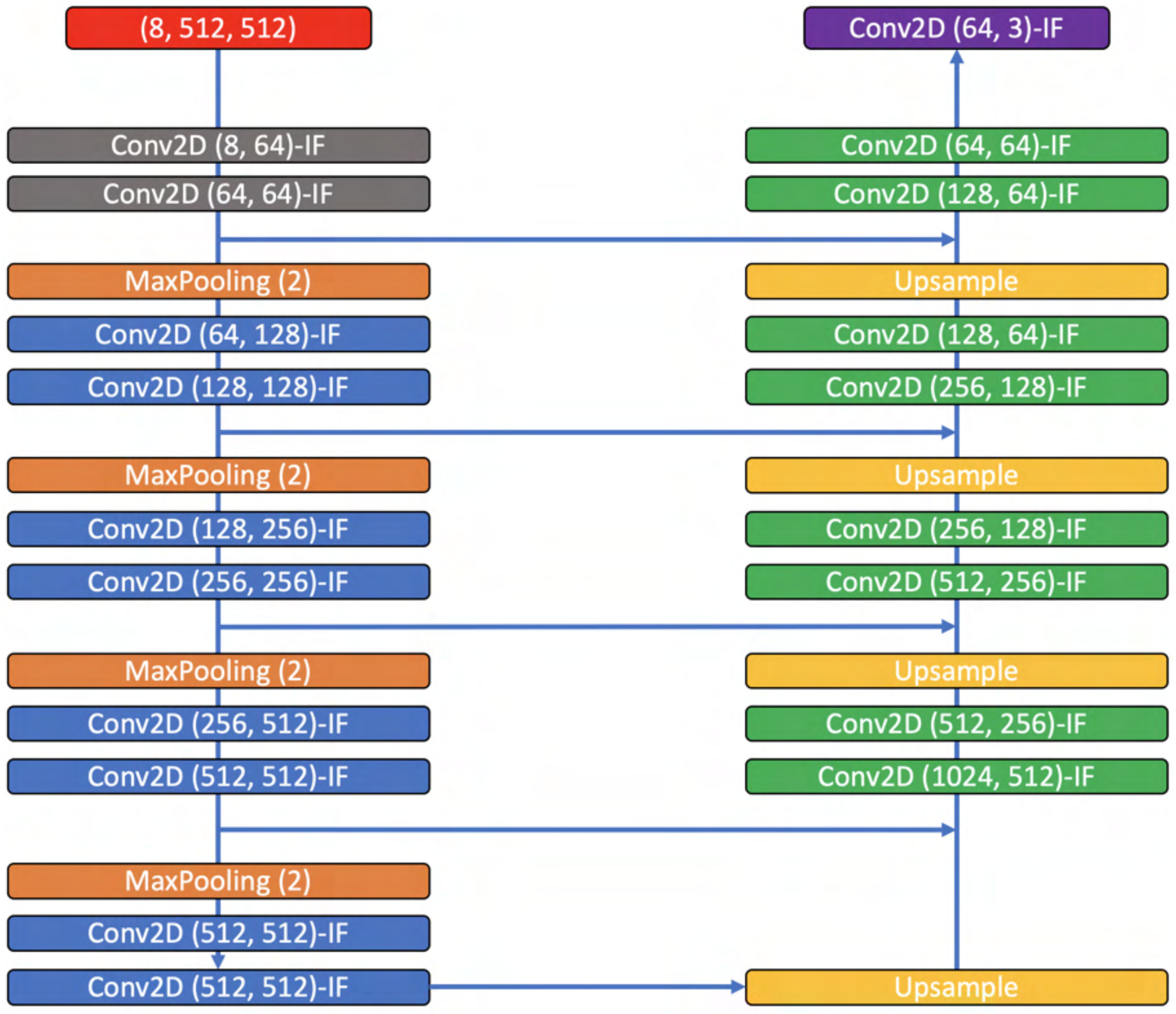


Figure 3.2. The network structure of Single-Timestep Spiking UNet: The network is designed in a fully convolutional manner according to the UNet architecture. Specifically, it consists of an event encoding module (gray), an encoder (orange and blue), a decoder (yellow and green), and a final prediction layer (purple). The size of the CVGR-I input representation is $(8 \times 512 \times 512)$. Conv2D($a$, b)-IF represents the spiking convolutional layer with a input channels and b output channels. Each max pooling layer downsamples the feature map by a factor of 2. And the spatial resolution is doubled after each upsampling layer.

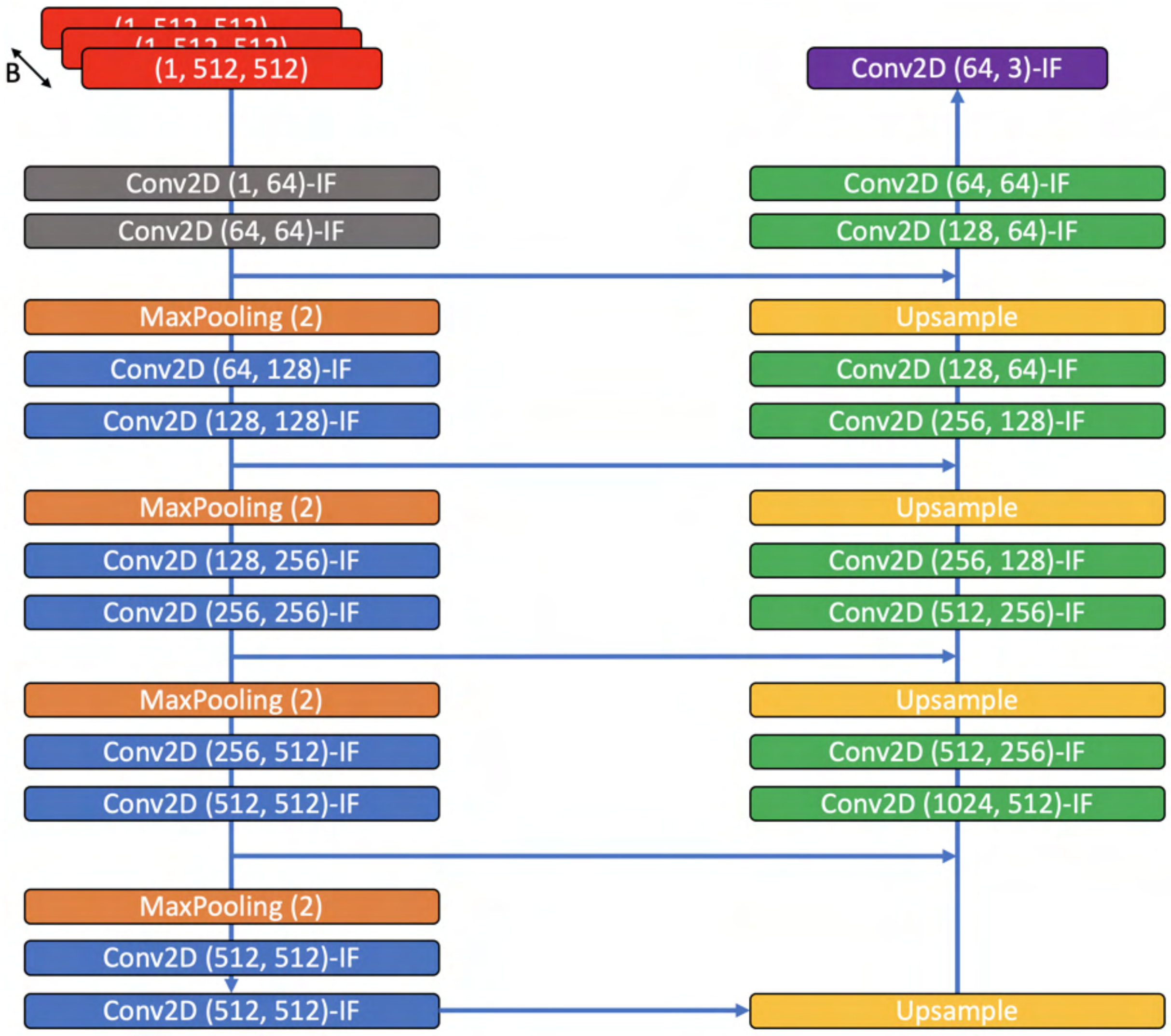


Figure 3.3. The network structure of Multi-Timestep Spiking UNet: The network is designed according to the UNet architecture in a fully convolutional manner. Specifically, it consists of an event encoding module (gray), an encoder (orange and blue), a decoder (yellow and green), and a final prediction layer (purple). Unlike the Single-Timestep Spiking UNet processing the CVGR-I representation as a whole and updating the membrane potential of its spiking neurons only once, the Multi-Timestep Spiking UNet processes the $B \times H \times W$ CVGR-I representation along its temporal dimension B. The settings for Conv2D(a, b)-IF layers, max pooling layers, and upsampling layers are the same as those for the Single-Timestep Spiking UNet.

in event-based shape from polarization, we will present the proposed models based on the dynamics of IF neurons.

3.5.1. Single-Timestep Spiking UNet

In this work, we have chosen a UNet [**107**], a commonly utilized architecture in semantic segmentation, as the backbone for surface normal estimation. Specifically, we propose the Single-Timestep Spiking UNet as shown in Fig. 3.2. This model is composed of several key components: an event encoding module, an encoder, a decoder, and a final layer dedicated to making surface normal predictions. As a Single-Timestep feed-forward SNN, this model processes the entire $B \times H \times W$ CVGR-I representation as its input and updates the membrane potential of its spiking neurons once per data sample. The event encoding module utilizes two spiking convolutional layers to transform the real-valued $B \times H \times W$ CVGR-I representation to the binary spiking representation with the size of $N_c \times H \times W$. Based on Eq. 3.4, the membrane potential u_i and output spiking state o_i of IF neuron i in the spiking convolutional layer are decided by:

$$\begin{aligned} u_i &= Conv(X), \\ o_i &= \begin{cases} 1 & \text{if } u_i \geq u_{th}; \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{3.6}$$

where $Conv(X)$ is the weighted convolutional summation of the inputs from previous layers and t in Eq. 3.4 is ignored since the model only updates once. After spiking feature extraction, there are N_e encoder blocks to encode the spiking representation. Each encoder employs a max pooling layer and multiple spiking convolutional layers to capture surface normal features. The neuronal dynamics of IF neurons in these layers are still controlled by Eq. 3.6. The encoded features are subsequently decoded using N_d decoder blocks, where $N_d = N_e$. Since transposed convolutions are often associated with the creation of checkerboard artifacts [**127**], each decoder consists of an upsampling layer followed by multiple spiking convolutional layers, where the IF neurons are governed by Eq. 3.6. For the upsampling operations, we have two options: nearest neighbor upsampling and bilinear upsampling. Through our experiments, we will show that nearest neighbor upsampling can achieve performance comparable to bilinear upsampling in event-based surface normal estimation while preserving the fully spiking nature of our proposed model. As suggested in the UNet architecture, to address the challenge of information loss during down-sampling and up-sampling, skip connections are utilized between corresponding encoder and decoder blocks at the same hierarchical levels. To preserve the spiking nature and avoid introducing non-binary values, the proposed model utilizes concatenations as skip connections. Lastly, the final prediction layer employs the potential-assisted IF neurons [**128, 129**] to estimate the surface normals. Unlike traditional IF neurons generate spikes based on Eq. 3.6, the potential-assisted IF neurons are non-spiking neurons which output membrane potential driven by:

$$\begin{aligned} u_i &= Conv(X), \\ o_i &= u_i, \end{aligned} \tag{3.7}$$

where o_i denotes the real-valued output of neuron i. These potential-assisted dynamics can be extended to both LIF and PLIF neurons, facilitating the construction of a Single-Timestep Spiking UNet using these types of neurons. By producing real-valued membrane potential outputs, potential-assisted neurons retain rich information that enhances surface normal estimation and boosts the expressivity of SNNs, especially for large-scale regression tasks.

3.5.2. Multi-Timestep Spiking UNet

To take advantage of temporal neuronal dynamics of spiking neurons and extract rich temporal information from event-based data, we propose the Multi-Timestep Spiking UNet for event-based shape from polarization. Figure 3.3 shows the network structure of the Multi-Timestep Spiking UNet. Similar to the Single-Timestep Spiking UNet, the Multi-Timestep Spiking UNet also consists of an event encoding module, an encoder, a decoder, and a final surface normal prediction layer. However, unlike the Single-Timestep Spiking UNet processing the CVGR-I representation as a whole and updating the membrane potential of its spiking neurons only once per data sample, the Multi-Timestep Spiking UNet processes the $B \times H \times W$ CVGR-I representation for each data sample along its temporal dimension B. At each time step, a $1 \times H \times W$ CVGR-I representation is fed in to the event encoding module and transformed as the size of $N_c \times H \times W$, followed by N_e encoder blocks, N_d decoder blocks, and a final prediction layer. Based on Eq. 3.4, the membrane potential $u_i(t)$ and output spiking state $o_i(t)$ of IF neuron i in the spiking convolutional layers of the Multi-Timestep Spiking UNet are decided by:

$$\begin{aligned} u_i(t) &= u_i(t-1)(1-o_i(t-1)) + Conv(X(t)), \\ o_i(t) &= \begin{cases} 1 & \text{if } u_i(t) \geq u_{th}; \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{3.8}$$

where $Conv(X(t))$ is the weighted convolutional summation of the inputs from previous layers at the time step t. The final prediction layer continues to use potential-assisted IF neurons, but with temporal dynamics as outlined below:

$$\begin{aligned} u_i(t) &= u_i(t-1) + Conv(X(t)), \\ o_i(t) &= u_i(t), \end{aligned} \tag{3.9}$$

where the potential-assisted IF neuron i accumulates its membrane potential to maintain the rich temporal information, $o_i(t)$ is the output of neuron i at time step t, and we use **the outputs at the last time step** as the final surface normal predictions.

3.5.3. Training and Implementation Details

We normalize outputs from spiking neurons into unit-length surface normal vectors $\hat{\mathbf{N}}$ and then apply the cosine similarity loss function:

$$\mathcal{L} = \frac{1}{H \times W} \sum_{i}^{H} \sum_{j}^{W} (1 - \left\langle \hat{\mathbf{N}}_{i,j}, \mathbf{N}_{i,j} \right\rangle), \tag{3.10}$$

where $\langle \cdot, \cdot \rangle$ indicates the dot product, $\hat{\mathbf{N}}_{i,j}$ refers to the estimated surface normal at the pixel location (i, j), while $\mathbf{N}_{i,j}$ denotes the ground truth surface normal at the same location. The objective is to minimize this loss, which is achieved when the orientations of $\hat{\mathbf{N}}_{i,j}$ and $\mathbf{N}_{i,j}$ align perfectly.

To optimize the Single-Timestep Spiking UNet, we utilize the backpropagation method [**130**] to calculate the weight updates:

$$\Delta w^l = \frac{\partial \mathcal{L}}{\partial w^l} = \frac{\partial \mathcal{L}}{\partial o^l} \frac{\partial o^l}{\partial u^l} \frac{\partial u^l}{\partial w^l}, \tag{3.11}$$

where w^l is the weight for layer l, o^l is the output of spiking neurons in layer l, and u^l is the membrane potential of spiking neurons in layer l. Similarly, to optimize the Multi-Timestep Spiking UNet, we utilize the BackPropagation Through Time (BPTT) [**131**] to calculate the weight updates. In BPTT, the model is unrolled for all discrete time steps, and the weight update is computed as the sum of gradients from each time step as follows:

$$\Delta w^l = \sum_{t=0}^{B-1} \frac{\partial \mathcal{L}}{\partial o_t^l} \frac{\partial o_t^l}{\partial u_t^l} \frac{\partial u_t^l}{\partial w^l}, \tag{3.12}$$

where w^l is the weight for layer l, o_t^l is the output of spiking neurons in layer l at the time step t, and u_t^l is the membrane potential of spiking neurons in layer l at the time step t. Based on the Heaviside step functions in Eq. 3.6 and Eq. 3.8, we can see that both $\frac{\partial o^l}{\partial u^l}$ and $\frac{\partial o_t^l}{\partial u_t^l}$ cannot be differentiable in spiking convolutional layers. To overcome the non-differentiability, we use the differentiable ArcTan function $g(x) = \frac{1}{\pi} arctan(\pi x) + \frac{1}{2}$ as the surrogate function of the Heaviside step function [**132**]. For the final prediction layer with potential-assisted spiking neurons, since they output membrane potential instead of spikes, we have $\frac{\partial o^l}{\partial u^l} = 1$ and $\frac{\partial o_t^l}{\partial u_t^l} = 1$ for these layers' weight updates.

3.6. Experiments and Results

In this section, we evaluate the effectiveness and efficiency of our proposed SNN models on event-based shape from polarization. We begin by introducing the experimental setup, datasets, baselines, and performance metrics for event-based shape from polarization. Then, extensive experiments on these datasets showcase the capabilities of our models, both in quantitative and qualitative terms, across synthetic and real-world scenarios.

Lastly, we analyze the computational costs of our models to highlight their enhanced energy efficiency compared to the counterpart ANN models.

3.6.1. Experimental Setup

Our models are implemented with SpikingJelly [**133**], an open-source deep learning framework for SNNs based on PyTorch [**134**]. To fairly compare with the counterpart ANN models, we ensure our models have similar settings to the ANN models in [**101**]. Specifically, we set $B = 8$ for the input event representation. In addition, our models have $N_e = 4$ encoder blocks and $N_d = 4$ decoder blocks. And the event encoding module outputs the binary spiking representation with the channel size of $N_c = 64$. For the spiking-related settings, all the spiking neurons in the spiking convolutional layers are set with a reset value (u_{reset}) of 0 and a threshold value (u_{th}) of 1. Following [**126, 135**], normalization techniques are applied after each convolution (Conv) operation for faster convergence. We train our models for 1000 epochs with a batch size of 2 on Quadro RTX 8000. We use the Adam [**136**] with a learning rate of $1e - 4$ to optimize our models.

3.6.2. Datasets

We evaluate our proposed models on two latest large-scale datasets for event-based shape from polarization, including the ESfP-Synthetic Dataset and ESfP-Real Dataset.

The ESfP-Synthetic Dataset was generated using the Mitsuba renderer [**137**], which created scenes with textured meshes illuminated by a point light source. For each scene, a polarizer lens, positioned in front of the camera, was rotated through angles ranging from 0 to 180 degrees with 15-degree intervals, producing a total of 12 polarization images. With these images, events were simulated using ESIM [**138**] with a 5% contrast threshold. Therefore, each scene in the dataset is accompanied by rendered polarization images, simulated events, and groundtruth surface normals provided by the renderer.

The ESfP-Real Dataset is the first large-scale real-world dataset for event-based shape from polarization. It contains various scenes with different objects, textures, shapes, illuminations, and scene depths. The dataset was collected using a Prophesee Gen 4 event camera [**139**], a Breakthrough Photography X4 CPL linear polarizer [**140**], a Lucid Polarisens camera [**100**], and a laser point projector. Specifically, the polarizer rotated in front of the event camera that captured the events for each scene in the dataset. The Lucid Polarisens camera was used to collect polarization images of the same scene at 4 polarization angles {0, 45, 90, 135}. And the groundtruth surface normals were generated using Event-based Structured Light [**141**], a technique that involves integrating the laser point projector with the event camera.

3.6.3. Baselines and Performance Metrics

We evaluate our models against the state-of-the-art physics-based and learning-based methods in the field of shape from polarization. Smith *et al.* [**118**] combined the physics-based shape from polarization with the photometric image formation model. The method directly estimates lighting information and calculates the surface height using a single polarization image under unknown illumination. Mahmoud *et al.* [**142**] presented a physics-based method to conduct shape recovery using both polarization and shading information. Recently, Muglikar *et al.* [**101**] have been pioneers in addressing event-based shape from polarization, employing both physics-based and learning-based approaches. Their models are notable for directly using event data as inputs. In this work, our focus is on comparing our proposed models with the learning-based model developed by Muglikar *et al.* We aim to demonstrate that our SNN-based models can match their performance while offering greater energy efficiency.

To evaluate the accuracy of the predicted surface normals, we employ four metrics: Mean Angular Error (MAE), % Angular Error under 11.25 degrees (AE<11.25), % Angular Error under 22.5 degrees (AE<22.5), and % Angular Error under 30 degrees (AE<30). MAE is a commonly used metric that quantifies the angular error of the predicted surface normal, where a lower value indicates better performance [**95, 96**]. The latter three metrics, collectively referred to as angular accuracy, assess the proportion of pixels with angular errors less than 11.25, 22.5, and 30 degrees, respectively, with higher percentages indicating better accuracy [**101**].

3.6.4. Performance on ESfP-Synthetic

We thoroughly evaluate our proposed models on the ESfP-Synthetic Dataset, using both quantitative metrics and qualitative analysis. Specifically, Table 3.1 presents the performance of both baselines and our methods in surface normal estimation on the ESfP-Synthetic Dataset. In addition, Figure 3.4 showcases the qualitative results of our models and the ANN counterpart on the ESfP-Synthetic Dataset. Column (a) shows the scene photographs for context. Column (b) is for the counterpart ANN models. Columns (c-d) are for the Single-Timestep Spiking UNets with bilinear upsampling and nearest neighbor upsampling, respectively. Columns (e-f) are for the Multi-Timestep Spiking UNets with bilinear upsampling and nearest neighbor upsampling, respectively. Column (g) presents

Table 3.1. Shape from polarization performance on the ESfP-Synthetic Dataset in terms of Mean Angular Error (MAE) and the percentage of pixels under specific angular errors (AE< ·). The "Input" column specifies whether the method utilizes events (E) or polarization images (I). E+I[0] means the CVGR-I representation. "Single" is for the Single-Timestep Spiking UNet. "Multi" is for the Multi-Timestep Spiking UNet. "Bilinear" and "Nearest" represent the bilinear upsampling and nearest neighbor upsampling, respectively. We highlight the top performance in bold, and underline the second-best results.

Method	Input	Task	MAE↓	AE<11.25↑	AE<22.5↑	AE<30↑
Mahmoud *et al.* [**142**]	I	Physics	80.923	0.034	0.065	0.085
Smith *et al.* [**118**]	I	Physics	67.684	0.010	0.047	0.106
Muglikar *et al.* [**101**]	E	Physics	58.196	0.007	0.046	0.095
Muglikar *et al.* [**101**]	E+I[0]	Learning	**27.953**	**0.263**	**0.527**	**0.655**
Single_Bilinear	E+I[0]	Learning	36.432	0.181	0.403	0.525
Single_Nearest	E+I[0]	Learning	36.824	0.141	0.370	0.491
Multi_Bilinear	E+I[0]	Learning	<u>31.296</u>	<u>0.200</u>	<u>0.438</u>	<u>0.578</u>
Multi_Nearest	E+I[0]	Learning	31.724	0.193	0.425	0.562

Table 3.2. Ablation study on various spiking neurons.

Method	Input	Task	MAE↓	AE<11.25↑	AE<22.5↑	AE<30↑
Multi_Nearest_IF	E+I[0]	Learning	31.724	0.193	0.425	0.562
Multi_Nearest_LIF	E+I[0]	Learning	35.250	0.154	0.384	0.523
Multi_Nearest_PLIF	E+I[0]	Learning	35.086	0.154	0.393	0.530

the ground truth normals. The MAE for the reconstructions is shown on the top left of each cell in Columns (b-f). For each scene, we highlight the best result using the green colorbox.

From Table 3.1, we can see that our proposed models significantly outperform the physics-based methods. The reason why our model can achieve the better performance is that our models benefit from the large-scale dataset and utilize the spiking neurons to extract useful information for event-based shape from polarization. Despite this success, our models do not quite match the **overall performance** of their ANN counterpart on this dataset, likely due to the limited representation capacity of spiking neurons. However, as Fig. 3.4 illustrates, our Multi-Timestep Spiking UNets still manage to **achieve comparable, and in some cases superior, results in shape recovery across various objects in the test set**, compared to the ANN models.

Table 3.1 clearly demonstrates that the temporal dynamics inherent in spiking neurons enable the Multi-Timestep Spiking UNets to surpass the Single-Timestep versions in surface normal estimation. Additionally, nearest neighbor sampling, as compared to bilinear upsampling, shows comparable performance while preserving the binary nature and compatibility with SNNs.

Recognizing the effectiveness of Multi-Timestep Spiking UNets, we undertook an ablation study aimed at identifying the ideal spiking neurons to fully leverage their temporal dynamic capabilities. The results, detailed in Table 3.2, indicate that IF neurons offer superior performance. This is largely due to their ability to retain more extensive temporal information, as they operate without the influence of a leaky factor.

3.6.5. Performance on ESfP-Real

We also compare these methods on the ESfP-Real Dataset. Specifically, we show the quantitative performance in Table 3.3 and illustrate the qualitative results in Fig. 3.5.

Similar to the results on the ESfP-Synthetic Dataset, our models demonstrate superior performance compared to physics-based methods on the real-world dataset. Moreover, as indicated by Table 3.3 and Fig. 3.5, our models not only match the overall performance of the ANN counterpart but also excel in qualitative results across diverse scenes in the test dataset. This enhanced performance on the ESfP-Real Dataset can be attributed to

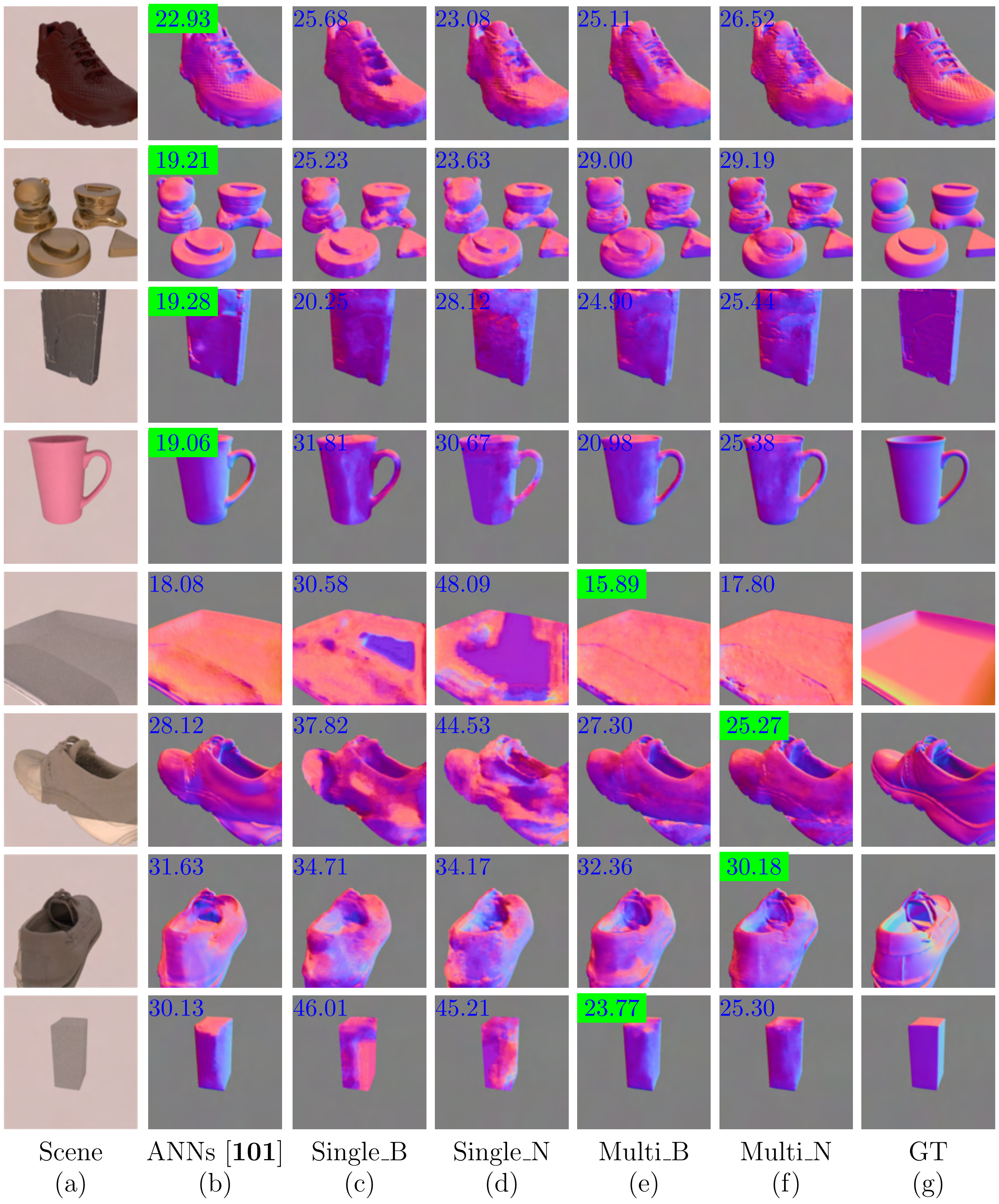


Figure 3.4. Qualitative results on the ESfP-Synthetic Dataset.

Table 3.3. Shape from polarization performance on the ESfP-Real Dataset in terms of Mean Angular Error (MAE) and the percentage of pixels under specific angular errors (AE< ·). The "Input" column specifies whether the method utilizes events (E) or polarization images (I). E+I[0] means the CVGR-I representation. "Single" is for the Single-Timestep Spiking UNet. "Multi" is for the Multi-Timestep Spiking UNet. "Bilinear" and "Nearest" represent the bilinear upsampling and nearest neighbor upsampling, respectively. We highlight the top performance in bold, and underline the second-best results.

Method	Input	Task	MAE↓	AE<11.25↑	AE<22.5↑	AE<30↑
Mahmoud *et al.* [**142**]	I	Physics	56.278	0.032	0.091	0.163
Smith *et al.* [**118**]	I	Physics	72.525	0.009	0.034	0.058
Muglikar *et al.* [**101**]	E	Physics	38.786	0.087	0.220	0.452
Muglikar *et al.* [**101**]	E+I[0]	Learning	<u>26.851</u>	0.099	0.449	**0.691**
Single_Bilinear	E+I[0]	Learning	27.134	**0.109**	**0.458**	0.685
Single_Nearest	E+I[0]	Learning	27.391	<u>0.106</u>	<u>0.450</u>	0.684
Multi_Bilinear	E+I[0]	Learning	26.886	0.093	0.439	<u>0.689</u>
Multi_Nearest	E+I[0]	Learning	**26.781**	0.089	<u>0.450</u>	0.688

the sparser nature of this real-world dataset [**101**]. In addition, compared to the ANN counterpart, our model is more compatible with the sparse events and better maintains the sparsity to prevent overfitting on this dataset.

Mirroring the outcomes observed on the ESfP-Synthetic Dataset, results from Table 3.3 and Fig. 3.5 also show that the Multi-Timestep Spiking UNet slightly outperforms the Single-Timestep Spiking UNet. Additionally, nearest neighbor upsampling is on par with bilinear upsampling in terms of surface normal estimation performance.

3.6.6. Energy Analysis

In earlier sections, we demonstrated that our models, employing nearest neighbor upsampling, can achieve performance comparable to those using bilinear upsampling in event-based shape from polarization. To delve deeper into the advantages of these fully

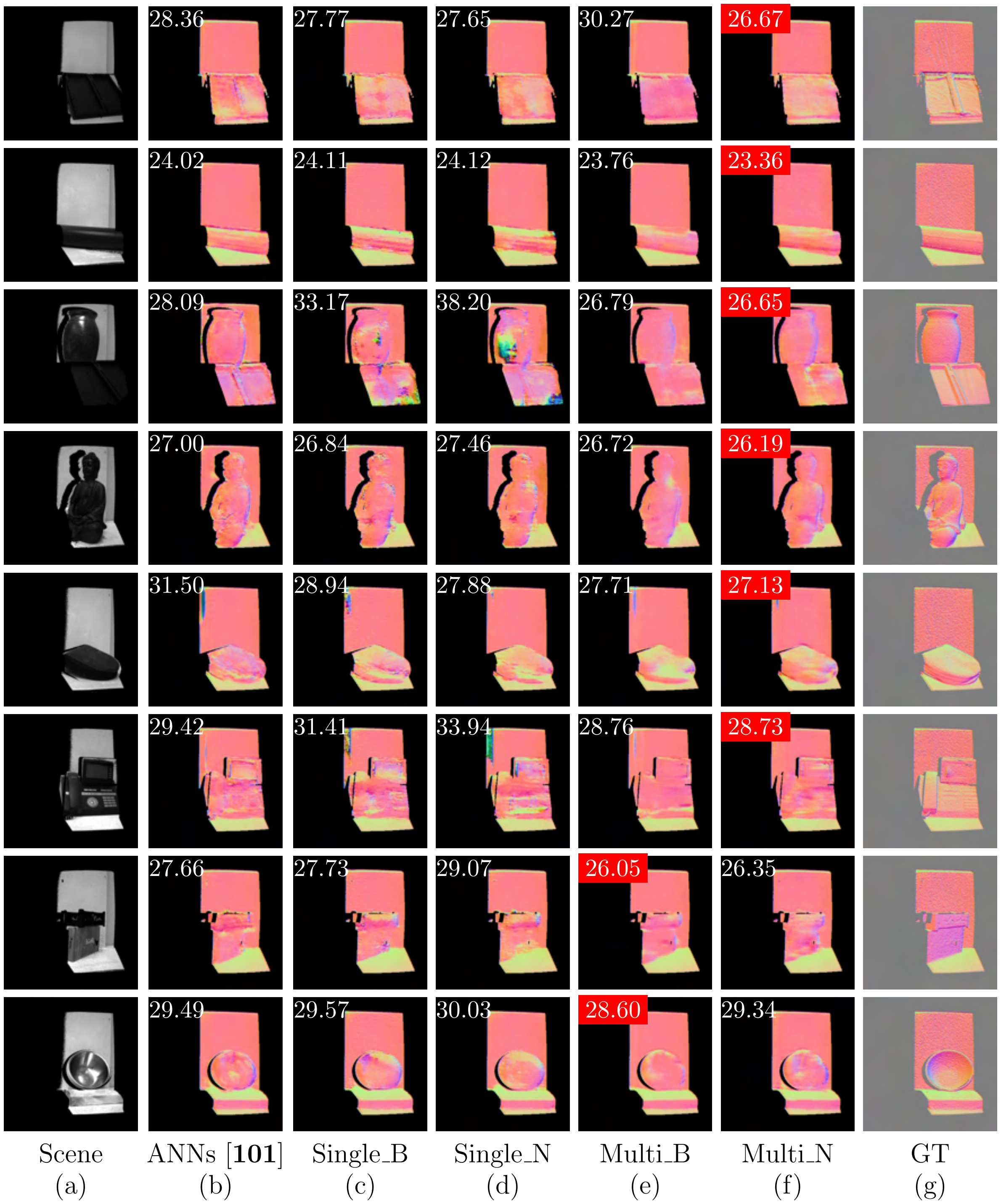


Figure 3.5. Qualitative results on the ESfP-Real Dataset. The meanings of columns are the same as those for Fig. 3.4. The MAE for the reconstructions is shown on the top left of each cell in Columns (b-f). For each scene, we highlight the best result using the red colorbox.

spiking models, we will now estimate the computational cost savings they offer compared to their fully ANN counterpart [**101**] on the ESfP-Real Dataset. Commonly, the number of synaptic operations serves as a benchmark for assessing the computational energy of SNN models, as referenced in studies like [**58**] and [**73**]. Moreover, we can approximate the total energy consumption of a model using principles based on CMOS technology, as outlined in [**74**].

Unlike ANNs, which consistently perform real-valued matrix-vector multiplication operations regardless of input sparsity, SNNs execute computations based on events, triggered only upon receiving input spikes. Therefore, we initially assess the mean spiking rate of layer l in our proposed model. In particular, the mean spiking rate for layer l in an SNN is calculated as follows:

$$F^{(l)} = \frac{1}{T} \sum_{t \in T} \frac{S_t^{(l)}}{K^{(l)}} \tag{3.13}$$

where T is the total time length, $S_t^{(l)}$ is the number of spikes of layer l at time t, and $K^{(l)}$ is the number of neurons of layer l. Table 3.4 shows the mean spiking rates for all layers in our fully spiking models, including the Single-Timestep Spiking UNet and Multi-Timestep Spiking UNet. Notice that we do not consider the components without trainable weights, such as max pooling and nearest neighbor upsampling layers. From the table, we can see that the Multi-Timestep Spiking UNet exhibits a higher average spiking rate across all its layers compared to the Single-Timestep Spiking UNet. This increased spiking rate aids in preserving more information, thereby enhancing the accuracy of surface normal estimation.

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 in layer l is calculated as $M^{(l)} \times C^{(l)} \times F^{(l)}$. Thus, the total number of synaptic operations in an SNN is calculated by:

$$\#OP = \sum_{l} M^{(l)} \times C^{(l)} \times F^{(l)} \times T. \tag{3.14}$$

In contrast, the total number of synaptic operations in the ANNs is $\sum_l M^{(l)} \times C^{(l)}$. 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. Based on these, we estimate the number of synaptic operations in the our proposed models and their ANN counterpart. Table 3.5 illustrates that, in comparison to ANNs, our models primarily perform AC operations with significantly fewer MAC operations that transform real-valued event inputs into binary spiking representations. Furthermore, the Multi-Timestep Spiking UNet executes more AC operations than the Single-Timestep Spiking UNet due to its higher average spiking rate and the utilization of temporal dynamics across multiple timesteps.

In general, AC operation is considered to be significantly more energy-efficient than MAC. For example, an AC is reported to be **5.1**$\times$ more energy-efficient than a MAC in the case of 32-bit floating-point numbers (0.9pJ vs. 4.6pJ, 45nm CMOS process) [**74**]. Based on this principle, we obtain the computational energy benefits of SNNs over ANNs in Table 3.5. From the table, we can see that the SNN models are **3.14**$\times$ to **28.80**$\times$ more energy-efficient than ANNs on the ESfP-Real Dataset.

Table 3.4. Mean spiking rates for all layers in the Single-Timestep Spiking UNet and Multi-Timestep Spiking UNet, both utilizing nearest neighbor upsampling and being fully spiking. Layers 1 to 19 correspond to the spiking convolutional layers depicted in Fig. 3.2 and Fig. 3.3. Given that the CVGR-I inputs are real-valued, the first layer in both models does not involve spike calculation.

	Single-Timestep Spiking UNet_Nearest		Multi-Timestep Spiking UNet_Nearest	
	Spiking rates	Spikes	Spiking rates	Spikes
Layer 1	0.3070	No	0.3070	No
Layer 2	0.0901	Yes	0.2484	Yes
Layer 3	0.1342	Yes	0.2304	Yes
Layer 4	0.1057	Yes	0.1626	Yes
Layer 5	0.1467	Yes	0.2482	Yes
Layer 6	0.1174	Yes	0.1719	Yes
Layer 7	0.1485	Yes	0.2733	Yes
Layer 8	0.1153	Yes	0.1870	Yes
Layer 9	0.1717	Yes	0.3607	Yes
Layer 10	0.1691	Yes	0.2149	Yes
Layer 11	0.1278	Yes	0.1991	Yes
Layer 12	0.1513	Yes	0.2075	Yes
Layer 13	0.1175	Yes	0.1840	Yes
Layer 14	0.1540	Yes	0.1923	Yes
Layer 15	0.1391	Yes	0.1810	Yes
Layer 16	0.1937	Yes	0.1867	Yes
Layer 17	0.1624	Yes	0.1881	Yes
Layer 18	0.2323	Yes	0.2058	Yes
Layer 19	0.2080	Yes	0.2099	Yes
Average	0.1575	-	0.2189	-

Table 3.5. Energy comparison of our models and their ANN counterpart on the ESfP-Real Dataset. The energy benefit is equal to $\frac{Energy_{ANNs}}{Energy_{SNNs}}$.

	ANNs [**101**]	Single_Nearest	Multi_Nearest
Average Spiking Rate	-	0.1575	0.2189
#OP_MAC ($\times 10^9$)	161.11	1.21	1.21
#OP_AC ($\times 10^9$)	0	22.36	255.35
Energy (10^{-3}J, 45nm CMOS process)	741.11	25.69	235.38
Energy Benefit ($\times$)	1.0	28.80	3.14

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 models on neuromorphic hardware and energy-constrained devices.

3.7. Conclusion and Future Work

In this work, we build SNNs that solve event-driven regression tasks. Specifically, we explore the domain of event-based shape from polarization with SNNs.

Drawing inspiration from the feed-forward UNet, we introduce the Single-Timestep Spiking UNet, which processes event-based shape from polarization as a non-temporal task, updating the membrane potential of each spiking neuron only once. This method, while not fully leveraging the temporal capabilities of SNNs, significantly cuts down on computational and energy demands. To better harness the rich temporal data in event-based information, we also propose the Multi-Timestep Spiking UNet. This model operates sequentially across multiple timesteps, enabling spiking neurons to employ their temporal recurrent neuronal dynamics for more effective data extraction. Through extensive evaluation on both synthetic and real-world datasets, our models demonstrate their ability to estimate dense surface normals from polarization events, achieving results comparable to those of state-of-the-art ANN models. Moreover, our models present enhanced energy efficiency over their ANN counterpart, underscoring their suitability for neuromorphic hardware and energy-sensitive edge devices. This research not only advances the field of spiking neural networks but also opens up new possibilities for efficient and effective event-based shape recovery in various applications.

Building on this foundation, future work could focus on several promising directions. One key area is the further optimization of SNN architectures to enhance their ability to process complex, dynamic scenes, potentially by integrating more sophisticated temporal dynamics or learning algorithms. Additionally, exploring the integration of our models with other sensory data types, like depth information, could lead to more robust and versatile systems. Moreover, adapting these models for real-time applications in various fields, from autonomous vehicles to augmented reality, presents an exciting challenge. Finally, there is significant potential in further reducing the energy consumption of these networks, making them even more suitable for deployment in low-power, edge computing scenarios. Through these explorations, we can continue to push the boundaries of what's possible with SNNs in event-based sensing and beyond.

CHAPTER 4

Not Just Spiking Neural Networks: Bio-Inspired Novel Spiking Architectures

This chapter is based on papers [**51, 54**]. The paper [**51**] © 2021 IEEE. Reprinted, with permission, from Peng Kang, Hao Hu, Srutarshi Banerjee, Henry Chopp, Aggelos Katsaggelos, and Oliver Cossairt, "Human vision-like robust object recognition," in 2021 IEEE International Conference on Image Processing (ICIP). IEEE, 2021, pp. 709-713. The paper [**54**] © 2023 IEEE. Reprinted, with permission, from Peng Kang, Srutarshi Banerjee, Henry Chopp, Aggelos Katsaggelos, and Oliver Cossairt, "Spiking glom: Bio-inspired architecture for next-generation object recognition," in 2023 IEEE International Conference on Image Processing (ICIP). IEEE, 2023, pp. 950-954.

In this chapter, we introduce innovative spiking architectures that draw inspiration from neuroscience findings. Drawing on the understanding that human visual processing operates hierarchically, combining both analog and digital processing, we proposed the ANN-SNN models and demonstrated their robustness in object recognition. In addition, inspired by the GLOM model, a hypothetical model in neuroscience that can imitate the human ability to parse visual scenes and represent the scene's part-whole hierarchies, we proposed energy-efficient and interpretable Spiking GLOM models by incorporating spiking neurons.

4.1. Human Vision-Like Robust Object Recognition

In this section, inspired by the evidence in neuroscience that the visual processing in human vision is performed hierarchically in the combination of analog and digital processing, we introduce the ANN-SNN models and demonstrate their robustness in object recognition.

4.1.1. Introduction

With the prevalence of Artificial Neural Networks (ANNs), computers today have demonstrated extraordinary abilities in many cognition tasks, such as image classification, speech recognition, and natural language processing [**102**]. However, ANNs only imitate brain structures in several ways including vast connectivity, and structural and functional organizational hierarchy [**36**]. The brain has more information processing mechanisms like the time-dependent neuronal and synaptic functionality [**59, 60**]. To integrate more brain-like characteristics, researchers propose Spiking Neural Networks (SNNs). Different from ANNs using real-valued computation, SNNs use binary 0-1 spikes to process information, which reduces the mathematical dot-product operations to less computationally summation operations. Besides, in most instances, the neurons in SNNs do not excite until they receive or generate spikes. Thus SNNs are potentially energy-efficient and may have a wide range of application scenarios with emerging neuromorphic hardwares [**6**].

There are two kinds of approaches in SNN learning, namely learning-based methods and conversion-based methods. In learning-based approaches, researchers have adopted two main directions including unsupervised learning with spike-timing-dependent plasticity (STDP) [**143**] and supervised learning with gradient descent and error back-propagation

[**40, 144, 145, 146**]. In conversion-based approaches, trained ANNs are converted to SNNs with the same network topology using weight rescaling and normalization methods to match the neuronal characteristics [**63, 64**].

Existing SNNs for object recognition are always pure SNNs [**36**], which only utilize the digital signals to transfer the information. However, evidence in neuroscience suggests that the visual processing in human vision is performed hierarchically in the combination of analog and digital processing [**147**]. Most neurons at low-level visual processing, like rod, cone, and bipolar cells, exhibit graded membrane potentials [**148, 149, 150**]. Unlike the spikes (digital signals) following the all-or-none principle, the graded membrane potentials (analog signals) refer to neuron outputs that are proportional to the inputs. At low-level visual processing, the visual information is extensively pre-processed and a great number of features are extracted [**151, 152**], e.g., whether the light intensity at a certain place increases or decreases, in which direction a light source moves, or whether there is an edge in the image [**153**]. Then, the output signals from the low-level visual processing are conveyed to retinal ganglion cells which make spikes dominant during further propagation. Finally, the brain processes spikes and reacts for specific vision tasks. To fully take advantage of the human vision system, we propose a general hierarchical ANN-SNN model for robust object recognition.

To the best of our knowledge, there is no pre-existing work on imitating the human vision systems in this way. We believe that this study should be helpful for further exploration of the human vision system. We evaluate our model and its variants on two popular datasets to show its effectiveness, robustness, efficiency, and generality. Extensive

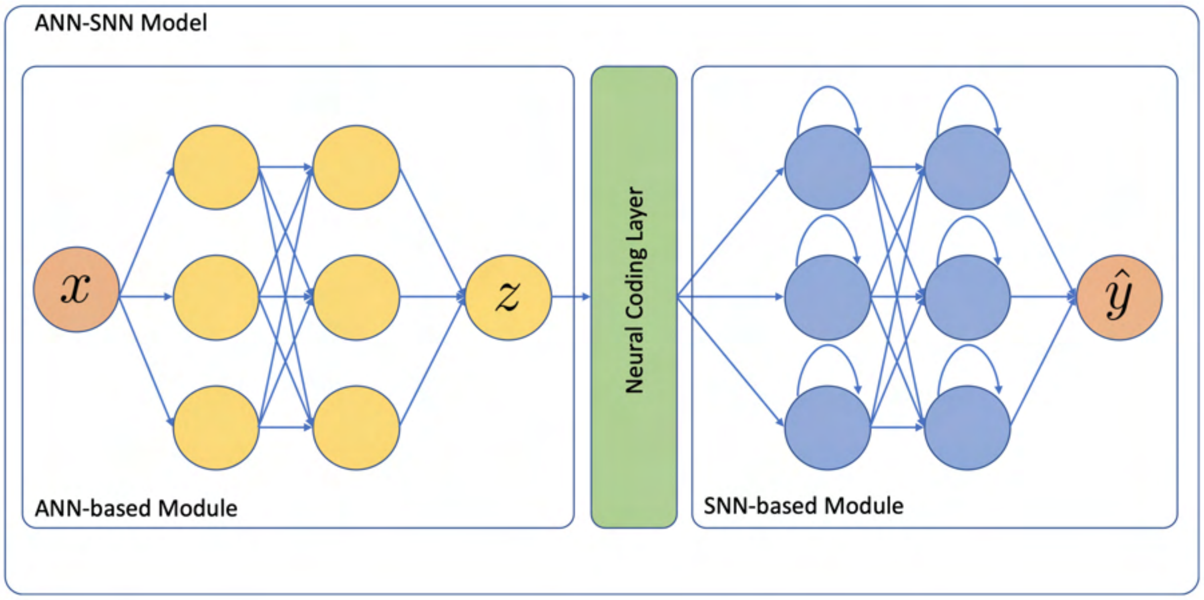


Figure 4.1. The network structure of the general ANN-SNN, where orange circles are the inputs and outputs, yellow circles are ANN neurons, and blue circles are LIF neurons.

experiments clearly demonstrate the superiority of our proposed models over previous SNNs for robust object recognition.

4.1.2. Models

In this section, we first introduce the network structures of the general ANN-SNN model. Then, we demonstrate how to convert ANN signals to SNN signals via neural coding. Finally, to show the generality of the proposed ANN-SNN model, we illustrate its variants and provide implementation details.

4.1.2.1. Network Structures. Inspired by the human vision system, we propose the general ANN-SNN model shown in Fig. 4.1. There are two components in the model. The first component is the ANN-based module, where each neuron is the ANN neuron

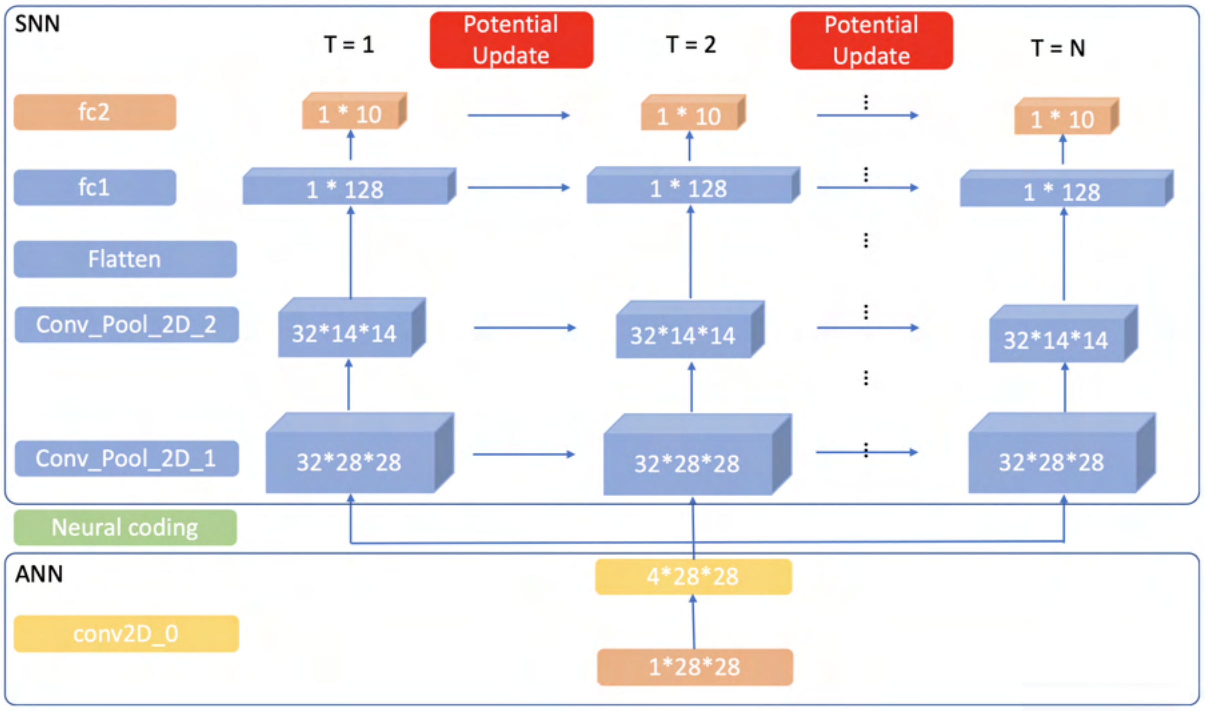


Figure 4.2. The unrolled network structure of a specific ANN-SNN, where orange cubes are the inputs and outputs, yellow cubes are ANN neurons, and blue cubes are LIF neurons. Round arrows in Fig. 4.1 are unrolled in this figure.

accepting and generating real-valued signals. The second component is the SNN-based module, which contains the Leaky Integrate-and-Fire (LIF) neurons accepting and generating spikes. Specifically, we utilize the ANN-based module to imitate the low-level visual processing, which involves complex analog computations like noise smoothing and edge detection. After that, to simulate the process of ganglion reformatting the analog signals to digital signals and sending the digital signals to the brain, we employ the neural coding layer to conduct the analog-to-digital conversion. Finally, to mimic the behavior of dynamic neuronal activities in the brain, we use the LIF model to construct the neurons in the SNN-based modules. LIF model is commonly used to describe the behavior of neuronal activities, including the update of membrane potential and spike firing. Specifically, the SNN-based module follows the basic structure of many other SNNs, which can be trained with backpropagation. We use discretized time steps and divide the forward propagation inside spiking neurons into two phases: update of membrane potential (represented by horizontal arrows in Fig. 4.2) and firing (represented by vertical arrows in Fig. 4.2). Membrane potential updates following

$$u(t) = k * u(t-1) * (1 - \delta(u(t-1) - V_{th})) + I(t), \tag{4.1}$$

where $u(t)$ is the membrane potential at time step t, k is a decay factor, V_{th} is a given fire threshold, $\delta(\cdot)$ is the Heaviside function whose gradient is approximated [**145**], $I(t)$ is the pre-neuron input at time step t.

We show the unrolled network structure of a specific ANN-SNN in Fig. 4.2, which can be applied to the object recognition task. Specifically, its ANN-based module is a light convolutional module that imitates the low-level human vision process extracting low-level noise-robust features. Its SNN-based module is a LIF neuron-based convolutional module that simulates the dynamic neuronal characteristic in the brain.

4.1.2.2. Neural Coding Layer. To encode the ANN's outputs into spiking trains, which is suitable for the inherent dynamic of the SNN-based module, we use the rate coding method to convert analog signals to digital signals. Specifically, at each simulation time step, we normalize the outputs of ANNs and generate the random feature map $Q \sim U[0, 1]$ which has the same size as the outputs of ANNs. Then we compare every generated random number in Q with its corresponding ANN's outputs. If the generated random number is greater than the ANN's output data, the corresponding position is set

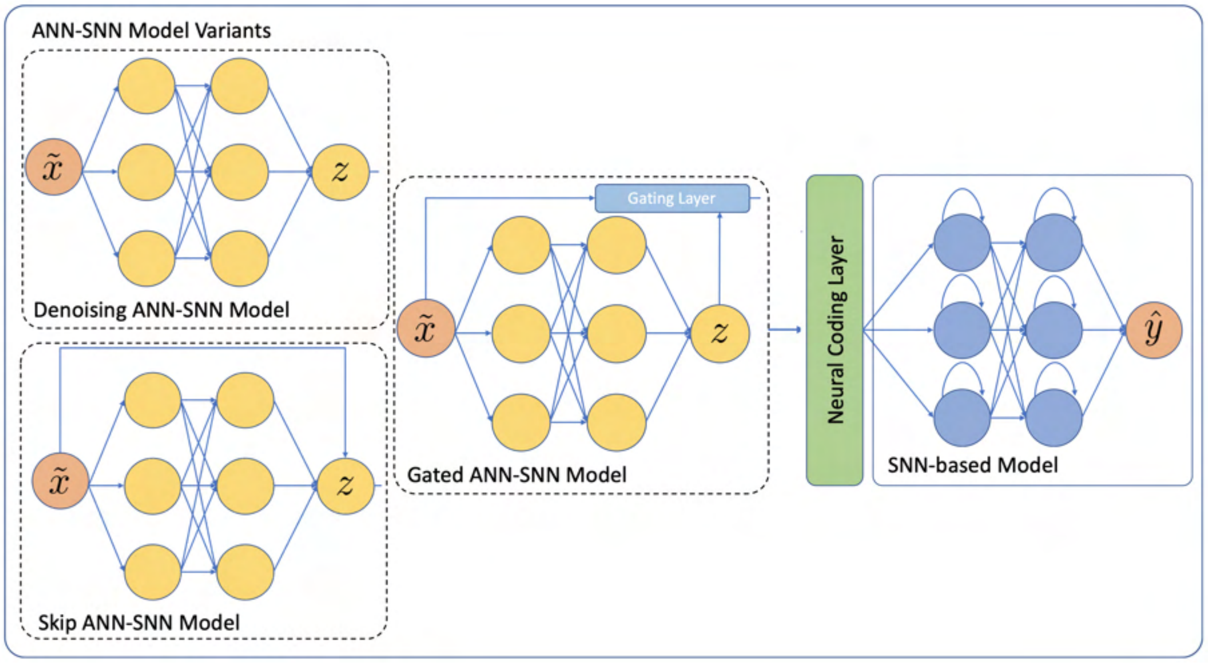


Figure 4.3. Three variants of the ANN-SNN model: Denoising ANN-SNN model, Skip ANN-SNN model, and Gated ANN-SNN model.

to 0. Otherwise, it is set to 1. By this means, we transform floating-point ANN's outputs into spiking trains, which ensures positions that have larger ANN's output data have higher firing frequency. Moreover, since we use mask operations to implement the neural coding layer in Pytorch and mask operations do not change the gradient characteristic in Pytorch, this means models with the neural coding layer can still be differentiable and optimized by autograd.

4.1.2.3. Model Variants and Implementation Details. To show the generality of the ANN-SNN model, we propose several variants (see Fig. 4.3) to make preceding ANNs gain the denoising ability explicitly. The first variant is Denoising ANN-SNN model, which reconstructs the clean input x from its (partially) corrupted version $\tilde{x}$ and forces the ANN-based module to discover more noise-robust features. We utilize the additive Gaussian noise whose variance range is $[0.0, 0.5]$ to corrupt the clean inputs and optimize the whole model on the task of object recognition via the loss in Eq. 4.2, where $z = ANN(\tilde{x})$, $\tilde{x} = x + Gaussian(0.0, var)$ and $var \sim U(0.0, 0.5)$, MSE is the mean squared error, and λ controls the balance between the object recognition loss and denoising loss. We set $\lambda = 1.0$ in all the experiments.

$$loss = MSE(y, \hat{y}) + \lambda * MSE(x, z) \tag{4.2}$$

However, if $\tilde{x}$ itself has already been a clean image, we tend to get a noisy image from $ANN(\tilde{x})$ at this time since it is hard for ANN to get the identity mapping, especially when ANN is deep [**154**]. At this time, it is necessary for us to keep the input information. To achieve this, we add a skip connection to the first variant and get Skip ANN-SNN model, where the outputs are the concatenation of $(\tilde{x}, z)$. However, Skip ANN-SNN model is too redundant to keep all the $\tilde{x}$. We thus use a gating module to distinguish the situation of keeping the input from the situation of discarding the input. Specifically, following Eq. 4.3, the gating module generates the weight G based on the input. Then, the ANN-based module outputs the weighted combination of x and z, i.e., $z = G \odot z + (1 - G) \odot \tilde{x}$, where $\odot$ is the element-wise multiplication.

$$G = Sigmoid(Conv(\tilde{x})) \tag{4.3}$$

4.1.3. Experiments

We conduct the experiments on NVIDIA RTX 2070S. The model is implemented under the Pytorch framework. We use Adam with the decay learning rate to optimize the model.

Table 4.1. The performance on MNIST. The best performance is in bold.

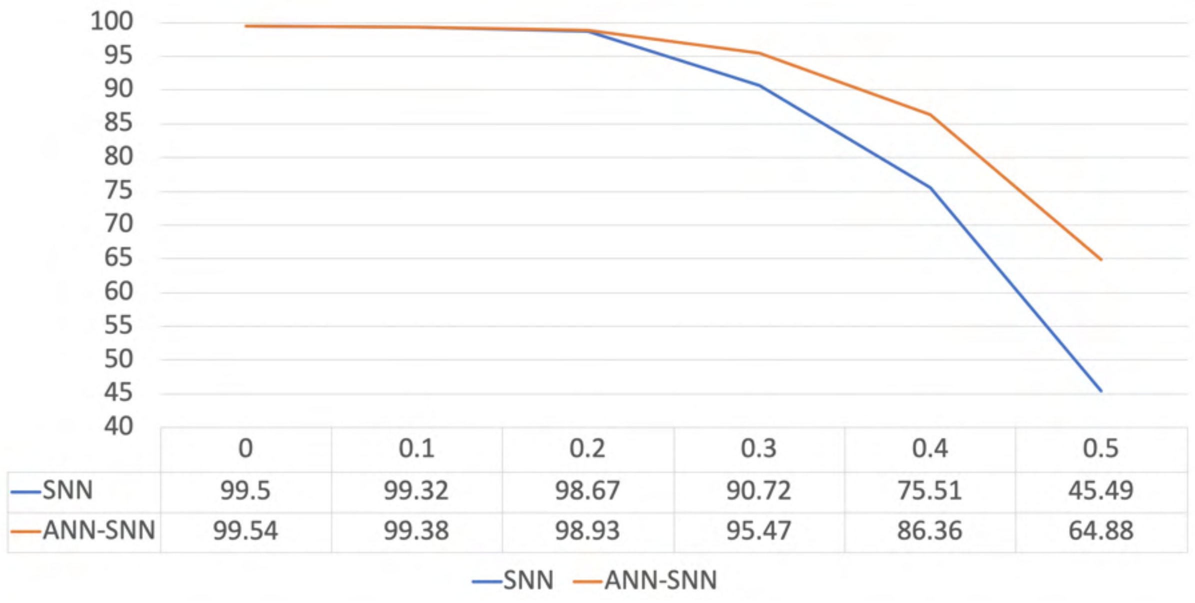
Models	Type	ACC(%)
Spiking CNN [**63**]	Conversion-based	99.31
SLAYER [**144**]	Learning-based	99.41
STBP [**40**]	Learning-based	99.42
HM2-BP [**146**]	Learning-based	99.49
LISNN [**145**]	Learning-based	99.50
SNN (without ANN)	Learning-based	99.50
ANN-SNN	Learning-based	**99.54**

To show the effectiveness, robustness, and efficiency of ANN-SNN model, we evaluate its performance on two classification tasks including the frame-based MNIST and event-based N-MNIST, and compare it with the current state-of-the-art performances of SNNs.

4.1.3.1. Frame-Based Dataset. MNIST is a frame-based dataset. It contains a training set of 60,000 images and a testing set of 10,000 images. Each image in the dataset is a 28×28 normalized grey-scale picture with a label from 0 to 9. To show the effectiveness of the ANN-SNN model, we compare the performance of our model with several other SNN models (to fairly compare, most of them are learning-based SNNs). From Table 4.1, it is clear that our ANN-SNN model outperforms other SNN models. The reason why our model can achieve the best performance could be two-fold: (1) the structure and functionality of our model are quite similar to those of the human vision system; (2) the preceding ANN can conduct the analog computations and extract effective information to boost the performance of latter networks – SNNs.

To show the robustness and generality of our proposed models, we evaluate the models in two experimental settings: out-domain noise and in-domain noise.

For the setting of out-domain noise, we train the models on the original MNIST and test them on the noisy MNIST testing set. We set the variance of the additive Gaussian

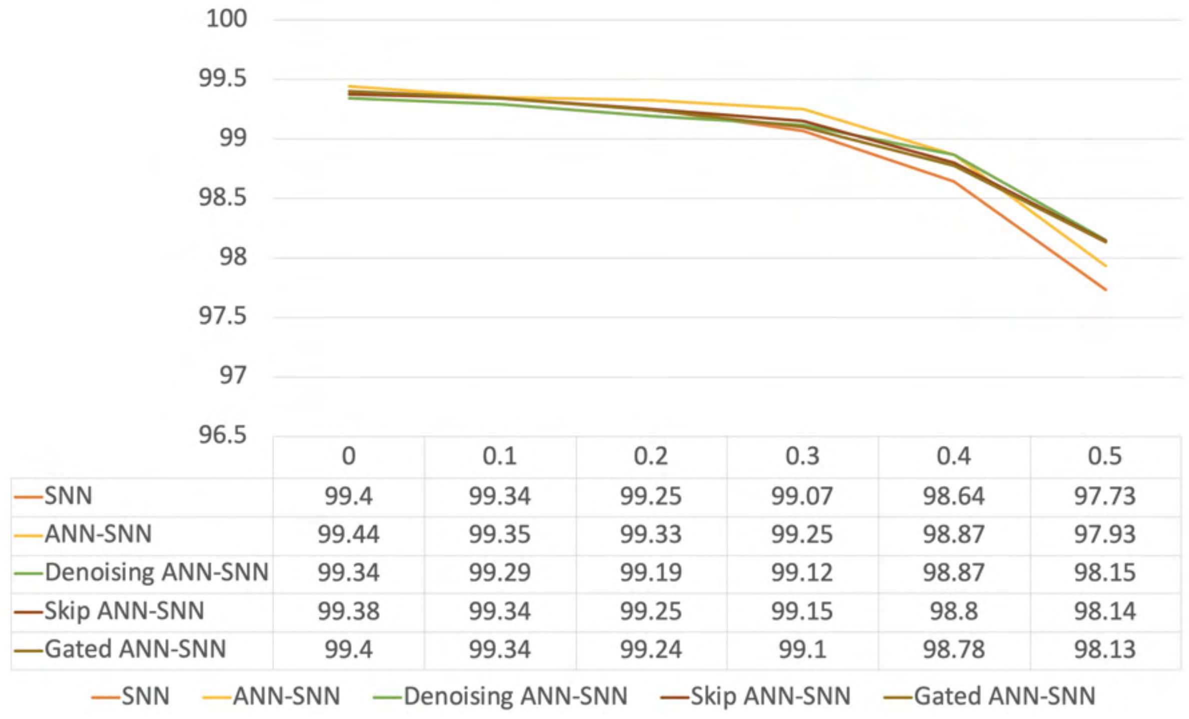


	0	0.1	0.2	0.3	0.4	0.5
—SNN	99.5	99.32	98.67	90.72	75.51	45.49
—ANN-SNN	99.54	99.38	98.93	95.47	86.36	64.88

Figure 4.4. Experimental results for the out-domain noise setting.

noise to be 0, 0.1, 0.2, 0.3, 0.4, 0.5. The results of out-domain noise experiments are shown in Fig. 4.4. From the figure, we can find that the ANN-SNN model consistently performs better than the SNNs without preceding ANNs. Moreover, the ANN-SNN model has more benefits as the noise level increases. The reason why the ANN-SNN model is out-domain noise-robust could be that the preceding ANNs imitate the low-level retina process to conduct the analog computations and extract low-level noise-robust features.

For the setting of in-domain noise, we train and test the models on the noisy MNIST. We set the variance of training noise to follow the uniform distribution $U[0.0, 0.5]$ and the variance of testing noise to be 0, 0.1, 0.2, 0.3, 0.4, 0.5. Fig. 4.5 shows the results of in-domain noise experiments. From the figure, we can see that the ANN-SNN model still consistently performs better than the pure SNNs. However, as the noise level increases, the benfits that the ANN-SNN model provides decrease. For example, when the noise



	0	0.1	0.2	0.3	0.4	0.5
SNN	99.4	99.34	99.25	99.07	98.64	97.73
ANN-SNN	99.44	99.35	99.33	99.25	98.87	97.93
Denoising ANN-SNN	99.34	99.29	99.19	99.12	98.87	98.15
Skip ANN-SNN	99.38	99.34	99.25	99.15	98.8	98.14
Gated ANN-SNN	99.4	99.34	99.24	99.1	98.78	98.13

Figure 4.5. Experimental results for the in-domain noise setting.

level is 0.5, the gain is 0.20. Such gain is slightly lower than the gain of 0.23 the ANN-SNN model obtains for the noise level of 0.4, To make the ANN-SNN model more in-domain noise-robust and gain more and more benefits as the noise level increases, we utilize the variants shown in Fig. 4.3 and optimize them with the loss function in Eq. (4.2). From Fig. 4.5, compared to the pure SNNs, we can see that the three variants can gain more and more benefit as the noise level increases. When the noise level is 0.5, all the variants can achieve the best performance. Compared to the original ANN-SNN model, because we use lighter ANNs in the variants, the performance of all the variants is lower at low noise levels.

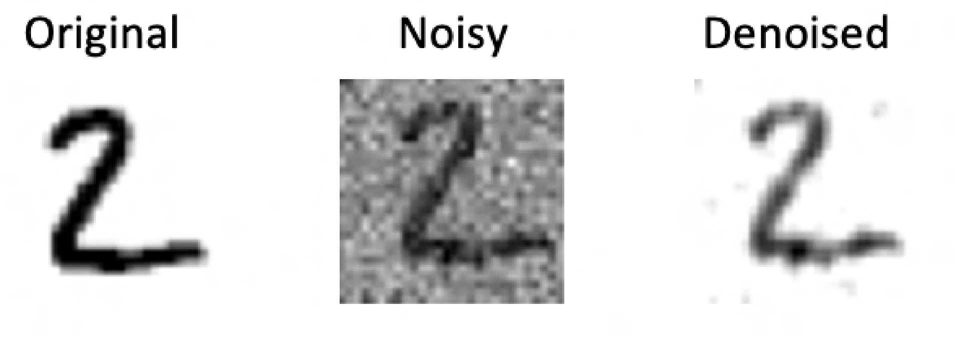


Figure 4.6. Qualitative denoising output.

Table 4.2. The performance on N-MNIST. The best performance is in bold.

Models	Type	ACC(%)
Spiking CNN [**63**]	Conversion-based	95.72
LSTM [**155**]	Learning-based	97.05
MLP [**156**]	Learning-based	97.80
CNN [**157**]	Learning-based	98.30
Spiking MLP [**156**]	Learning-based	98.74
STBP [**40**]	Learning-based	98.78
LISNN [**145**]	Learning-based	99.45
SNN (without ANN)	Learning-based	99.51
ANN-SNN	Learning-based	**99.53**

To qualitatively show the effectiveness of denoising in the variants, we present the original MNIST image, MNIST image with noise level of 0.5, ANN's output of Denoising ANN-SNN model in Fig. 4.6. From the figure, we can see that the preceding ANNs indeed perform denoising.

To show the efficiency of the ANN-SNN model, we record the training time for the ANN-SNN model and pure SNNs. Both of them need around **6.4s** to process 10,000 training images. This shows that the ANN-SNN model does not add too much complexity to the SNNs.

4.1.3.2. Event-Based Dataset. To show that our model can process different data formats, we also evaluate the ANN-SNN model on the event-based dataset N-MNIST, which is a spiking version of the frame-based MNIST dataset. It contains 60,000 sets of training events and 10,000 sets of testing events. The N-MNIST dataset is captured by displaying MNIST images on a monitor and moving the sensor on a motorized unit to view these images. N-MNIST images have two channels that record brighter pixels and darker pixels separately. In this case, we partition the triggered pixels into 10 simulation time steps according to its timestamp and transfer the event-based data to the frame-based one.

We compare the performance of our model with several other algorithms working on N-MNIST in Table 4.2, which demonstrates the effectiveness of ANN-SNN model on the event-based dataset. The reason why our model is effective and competitive could be two-fold: (1) the whole structure and functionality of our model are quite similar to those of the human vision system; (2) Dynamic Vision Sensors (DVS) create inevitable noise when capturing the object and this increases the difficulty for networks to extract the significative features from input data, while the preceding ANN in our model can conduct the analog computations and extract noise-robust features, which guarantees the superior performance. To verify the efficiency, we also record the training time for the ANN-SNN model and pure SNNs. Both of them need around **12.6s** to process 10,000 training images.

4.2. Spiking GLOM: Bio-inspired Architecture for Next-generation Object Recognition

In this section, inspired by the GLOM model, a hypothetical model in the neuroscience that can imitate the human ability to parse visual scenes and represent the scene's part-whole hierarchies, we proposed energy-efficient and interpretable Spiking GLOM models by incorporating spiking neurons.

4.2.1. Introduction

With the prevalence of Artificial Neural Networks (ANNs), computers today have achieved impressive performance in many cognition tasks, such as computer vision and natural language processing [**102**]. However, such extraordinary performance has come at the expense of model complexity, making ANN-based technologies energy-consuming and difficult to interpret. High energy efficiency and interpretability of artificial intelligence products are essential to many fields, such as autonomous driving [**158**] and healthcare [**159**], where energy is constrained and safety is at stake.

To reduce the energy consumption in neural networks, researchers proposed Spiking Neural Networks (SNNs) inspired by energy-efficient biological systems [**36, 51**]. Unlike ANNs that perform real-valued computations, SNNs use binary spikes to process information, which reduces the mathematical dot-product operations inherent in ANNs to just a few summations. Moreover, in most instances, the neurons in SNNs do not excite until they receive or generate spikes. Therefore, SNNs are potentially energy-efficient and have a wide range of application scenarios with emerging neuromorphic hardware [**160, 161**].

However, with the recent advance in deep SNNs, it becomes hard to explain the results of these models, which could pose challenges when we deploy them to the critical fields.

Recently, Geoffrey Hinton introduced a hypothetical system called GLOM to improve the interpretability of neural networks [**162**]. The GLOM model imitates the human ability to parse visual scenes and aims to represent the scene's part-whole hierarchies in neural networks. Specifically, the model utilizes inter-connected columns with sharing weights, which mimic the hyper-column structures of the human visual cortex [**163**], to process the input image patches. Each column comprises auto-encoders stacked in levels and processes a patch of the image. Ideally, the GLOM would learn different patch abstractions in the columns at different locations and levels and create part-whole hierarchies with rich representations. Inspired by this work, Garau et al. [**164**] turned it into a fully working system with the application to image classification. Although the work did not reach the state-of-the-art performance, it verifies the feasibility and interpretability of the GLOM model. However, since both the GLOM model [**162**] and its implementation [**164**] are ANN-based frameworks, they could be more energy-consuming compared to the bio-inspired SNNs that employ the binary spikes.

In this work, to alleviate the two limitations mentioned above, we extend the conceptual idea of the GLOM model and propose the Spiking GLOM by adopting the spiking neurons with neuronal dynamics as the building blocks. Moreover, we propose the potential-assisted Spiking GLOM and hybrid Spiking GLOM to further improve the applicability and performance of our model. Furthermore, we introduce the spike-based supervised contrastive loss to optimize our models. To the best of our knowledge, this is the first time using spikes to conduct the supervised contrastive learning. Finally, to demonstrate the feasibility of our models, we provide a preliminary implementation of these models and conduct experiments on object recognition. Extensive experiments on CIFAR-10 clearly demonstrate the effectiveness of our proposed models and show the superiority of our models in energy efficiency and interpretability.

4.2.2. Models

In this section, we first introduce the architecture of the Spiking GLOM. Then, we propose the variants of our model, including the potential-assisted Spiking GLOM and hybrid Spiking GLOM. Finally, we present the training procedure and introduce the spike-based supervised contrastive loss.

4.2.2.1. Spiking GLOM. To alleviate the low energy efficiency and poor interpretability in ANNs, we propose a novel bio-inspired model for next-generation object recognition – Spiking GLOM by adopting the spiking neurons with neuronal dynamics in the GLOM model. In this work, we use the integrate-and-fire (IF) neurons whose neuronal dynamics are controlled by

$$u(t) = u(t-1) * (1 - \delta(u(t-1) - V_{th})) + I(t), \tag{4.4}$$

where $u(t)$ is the membrane potential at time step t, V_{th} is a given fire threshold, $\delta(\cdot)$ is the Heaviside function whose gradient is approximated as in [**126**] to enable the back-propagation, $I(t)$ is the pre-neuron input at time step t. Figure 4.7 presents the network structure of the Spiking GLOM. Specifically, the proposed model has four major components: a spike-based patch embedding, spike-based columns, a spike-based contrast head, and a spike-based classification head.

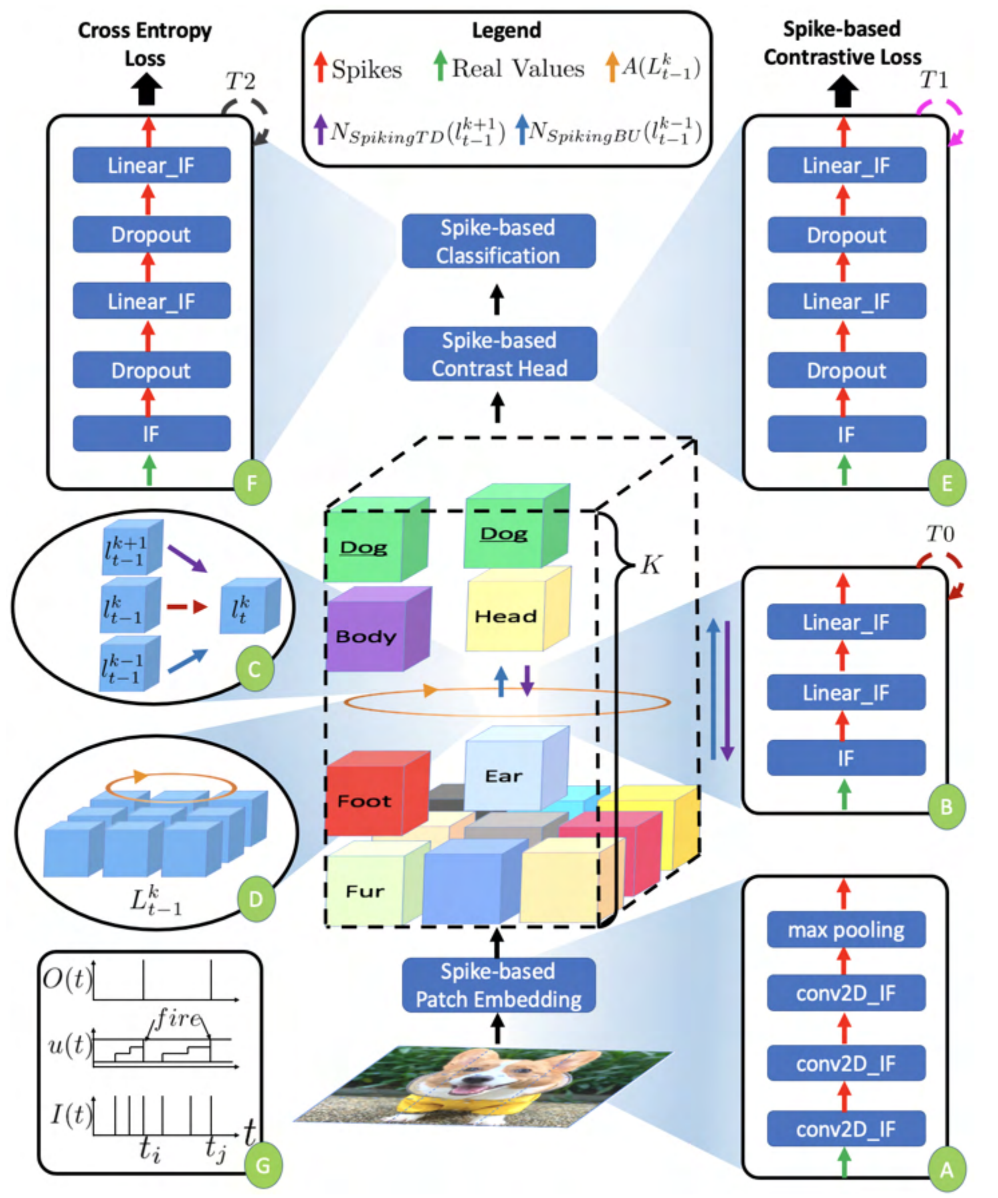


Figure 4.7. The architecture of the Spiking GLOM (center) with detailed structures of building elements (left and right) and neuronal dynamics of IF neurons (bottom-left). Each cube is a level representation l_t^k denoting various part-whole hierarchies. A: spike-based patch embedding; B: spike-based MLPs in the spike-based columns; C, D: the information routing in the spike-based columns, including the contributions from the same level, the bottom level, the top level, and the attentional neighbors; E: spike-based contrast head; F: spike-based classification head; G: neuronal dynamics of IF neurons. The neuron receives spikes, accumulates its potential, and generates spikes when the potential reaches the threshold.

The spike-based patch embedding utilizes a convolutional tokenizer with IF neurons to extract the rich representation of the input image. Compared to the ANN-based tokenizer, our module can be more energy-efficient since binary spikes are utilized in information processing. The neuronal dynamics of IF neurons are controlled by Eq. 4.4 with $I(t) = Conv2D(x(t))$, where $x(t)$ is real-valued input image for the first layer and binary spike representation for the deeper layers. As in [**164**], our spike-based patch embedding is feed-forward, which indicates that the IF neurons in this module only update their potentials once. And at each time step t, the module produces the same spike-based patch embedding L_t^0, which has the size of $H \times W \times d$, where $N = H \times W$ is the number of patches and d is the embedding dimension. We then feed each d-dimensional patch embedding to a spatially located spike-based column $C(h; w)$, which is the core component of the Spiking GLOM.

As shown in Fig. 4.7, each column $C(h; w)$ consists of K level representations $\{l_t^k | k = 1, ..., K; t = 1, ..., T0\}$. Ideally, at the end of time, the variances of representations across all the patches tend to reduce with the increase of levels, and the highest level representations tend to have similar values and represent the same whole ("Dog" in Fig.4.7). To obtain such part-whole hierarchies, we update the l_t^k of $C(h; w)$ in the form of the average-weighted spike representations:

$$\begin{aligned} l_t^k = {} & avg(w_l l_{t-1}^k + w_{BU} N_{SpikingBU}(l_{t-1}^{k-1}) \\ & + w_{TD} N_{SpikingTD}(l_{t-1}^{k+1}) + w_A A(L_{t-1}^k)), \end{aligned} \tag{4.5}$$

where l_t^k, l_{t-1}^k, l_{t-1}^{k-1}, and l_{t-1}^{k+1} are spatio-temporal consecutive level representations in $C(h; w)$. For $k = 1$, the spike-based patch embedding $l_t^0 \in L_t^0$ will get involved. $avg()$

indicates the arithmetical average, w_l, w_{BU}, w_{TD}, and w_A are four trainable scalars. As in [**164**], $A(L_{t-1}^k)$ is a standard attention weighting mechanism without trainable weights, which takes the attentional average of the level representations across all the patches at level k and time $t-1$. The spike-based Multi Layer Perceptrons (MLPs) $N_{SpikingBU}$ and $N_{SpikingTD}$ compute the bottom-up contribution of the level l_{t-1}^{k-1} and the top-down contribution of the level l_{t-1}^{k+1}, respectively. Since level representations in Eq. 4.5 are real-valued, these spike-based modules first utilize a layer of IF neurons to transform the representations into binary spikes. Such the transformation ensures energy-efficient summations in the deeper layers. The neuronal dynamics of IF neurons in the deeper layers are controlled by Eq. 4.4 with $I(t) = Linear(x(t))$, where $x(t)$ is the binary spike representation from the previous layer. As suggested in [**162**], all the $N_{SpikingBU}(\cdot)$ connecting $l_{t-1}^{k-1} \in L_{t-1}^{k-1}$ to $l_t^k \in L_t^k$ share the same weights. The same is true for all the $N_{SpikingTD}(\cdot)$ connecting $l_{t-1}^{k+1} \in L_{t-1}^{k+1}$ to $l_t^k \in L_t^k$. After $T0$ time steps, we obtain the final average-weighted spike-based column representations L_{T0}^K for N image patches. Then, we rearrange these N representations $l_{T0}^K \in L_{T0}^K$ to a vector with the dimension of $N \times d'$, where d' is the size of l_{T0}^K, and input this vector to the spike-based contrast head.

The spike-based contrast head further provides contrastive representations for the input image. Specifically, at each time step t, the spike-based contrast head first utilizes a layer of IF neurons to transform the real-valued L_{T0}^K to binary spikes. Then, it uses several $Linear\_IF$ layers to extract spiking contrastive features F_t. The dropout layer is applied before each $Linear\_IF$ layer to stabilize the model training. Finally, we obtain the time-weighted spiking contrastive features $F_a = \sum_{t=1}^{T1} F_t / T1$, which are fed to the spike-based classification head. Compared to the spike-based contrast head, the spike-based classification head has a similar structure, except that the output dimension is the number of classes and the total number of time steps is $T2$. The final time-weighted prediction is $O = \sum_{t=1}^{T2} O_t/T2$, where O_t is the spiking prediction from the spike-based classification head at each time step. The final predicted label is associated with the neuron with the largest value in O.

4.2.2.2. Model Variants. Inspired by the superiority of potential-assisted SNNs in regression tasks [**122, 43**], we propose the potential-assisted Spiking GLOM to obtain richer hierarchical level representations. Specifically, we change the spike-based MLPs in the column structure and employ the IF neurons with infinite thresholds **in the last layer of these MLPs**. Therefore, instead of generating binary spikes, these potential-assisted MLPs generate real-valued potentials at the last layer. Nevertheless, since all the other layers still employ the IF neurons with finite thresholds, such modules still conduct the binary summations and maintain the energy efficiency. Furthermore, inspired by the hybrid SNN-ANN model [**73**], we propose the hybrid Spiking GLOM that builds upon the potential-assisted Spiking GLOM. Specifically, we keep the spike-based patch embedding and spike-based columns but change the contrast head and classification head with ANNs in [**164**].

4.2.2.3. Training and Loss. The training procedure is divided into two steps: (i) a pre-training phase using the spike-based supervised contrastive loss and (ii) a training phase for object recognition using the Cross-Entropy loss. Specifically, given a batch of B samples, we duplicate each image in the batch for a total of $2B$ data points. Moreover, we apply the random data augmentation to all the samples and obtain their time-weighted spiking contrastive features F_a as described in Sec. 4.2.2.1. The spike-based supervised contrastive loss is shown as follows:

$$\mathcal{L} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log \frac{exp(F_a^i \cdot F_a^p / \tau)}{\sum_{q \in Q(i)} exp(F_a^i \cdot F_a^q / \tau)}, \tag{4.6}$$

where $i \in I \equiv \{1, ..., 2B\}$ is the index of an arbitrary augmented sample, $P(i)$ is the set of indices of all positives (with the same labels) in the augmented batch distinct from i, $Q(i) \equiv I \backslash i$, the $\cdot$ is the inner product, and τ is a scalar temperature parameter. Once the network is pre-trained using this loss, the weights are frozen, except for the spike-based classification head, which is optimized by the Cross-Entropy loss and provides the final predictions.

4.2.3. Experiments

We evaluate our proposed model on CIFAR-10 to show its effectiveness on object recognition. Moreover, we demonstrate the superior energy efficiency of our model compared to its ANN counterpart. Finally, we present the interpretability of our model. To make a fair comparison between the ANN-GLOM [**164**] and our model on effectiveness and energy efficiency, we maintain the same training epoch (300) and batch size (256). Additionally, we employ the same trainable weights (connections) as the ANN-GLOM but with IF neurons. Following [**126, 135**], normalization techniques are applied after each *Conv2D* or *Linear* operation for faster convergence. The experimental settings and codes are available at `https://github.com/pkang2017/slom`.

4.2.3.1. Basic Performance. Table 4.3 reports the validation accuracies of our proposed models and the ANN-GLOM [**164**] on CIFAR-10. As can be seen, our models

Table 4.3. The validation accuracies on CIFAR-10 dataset, where "GLOM" is ANN-based, "Ours" is for the Spiking GLOM, "Ours_PA" is for the potential-assisted Spiking GLOM, and "Ours_Hybrid" is for the hybrid Spiking GLOM.

Models	Acc(%)
GLOM [**164**]	87.61
Ours	84.35
Ours_PA	85.10
Ours_Hybrid	85.71

achieve the comparable performance to the ANN-GLOM, which shows the effectiveness of our models on object recognition. Moreover, we can see that the potential-assisted method and hybrid structure further boost the performance of Spiking GLOM on object recognition.

4.2.3.2. Energy Efficiency. Following [**73**], we estimate the gain in inference energy efficiency compared to the ANN-GLOM. Generally, the number of synaptic operations in an ANN layer is $M \times C \times T$, where M is the number of neurons, C is the number of synaptic connections per neuron, and T is the total time length for this ANN layer. And the number of synaptic operations in an SNN layer is $M \times C \times F \times T'$, where F is the layer's firing rate and T' is the total time length for this SNN layer. $T' = T = 1$ for the patch embedding module, $T' = T = T0 > 1$ for the columns, and $T' = T1, T2 > T = 1$ for the other modules. Moreover, due to the binary nature of spikes, SNNs perform only accumulation (AC) per synaptic operation, while ANNs perform the multiply-accumulate (MAC) computations. The AC is reported to be **5.1**$\times$ more energy-efficient than the MAC operation in the 45nm CMOS process [**74**]. Based on these principles, we obtain the computational energy benefits of our models over ANN-GLOM in Table 4.4. From the table, we can see that most SNN layers are **10**$\times$ to **40**$\times$ more energy-efficient than

Table 4.4. Firing rates (FR) and energy benefits (the compute-energy of ANNs / the compute-energy of SNNs, 45nm CMOS process) of layers ($Conv2D\_IF$ or $Linear\_IF$ in Fig. 4.7) in the the potential-assisted Spiking GLOM (_PA) and the hybrid Spiking GLOM (_Hybrid).

Layers	FR_PA	Energy Benefit_PA	FR_Hybrid	Energy Benefit_Hybrid
Patch Embedding_1	1.0	1.0×	1.0	1.0×
Patch Embedding_2	0.1843	27.6723×	0.1838	27.7476×
Patch Embedding_3	0.2768	18.4249×	0.3023	16.8707×
Column_BU_1	0.1712	29.7897×	0.1814	28.1147×
Column_BU_2	0.2427	21.0136×	0.2346	21.7391×
Column_TD_1	0.1045	48.8038×	0.2916	17.4897×
Column_TD_2	0.1668	30.5755×	0.1477	34.5295×
Contrast_1	0.1455	1.7526×	1.0	1.0×
Contrast_2	0.0089	28.6517×	1.0	1.0×
Classification_1	0.0160	15.9375×	1.0	1.0×
Classification_2	0.0807	3.1599×	1.0	1.0×

the counterpart ANN layers, except for the first layer of the spike-based patch embedding and the ANN-based contrast head and classification head in the hybrid Spiking GLOM. There are no energy benefits for these layers since they receive real-valued inputs.

4.2.3.3. Interpretability. As described in Sec. 4.2.2, the spike-based column is the core component of the Spiking GLOM. Moreover, the potential-assisted Spiking GLOM's columns employ the potentials of spiking neurons to maintain richer part-whole hierarchical representations. Therefore, to demonstrate the interpretability of our model, we train a potential-assisted Spiking GLOM with $K = 5$ levels following [**162, 164**] and provide a few CIFAR-10 examples in Fig. 4.8. Specifically, the bottom row presents the input images with their labels. And the rest of the rows show the column representations L_{T0}^{k} with their variances. Ideally, at the end of time $T0$, we expect the variances of representations across all the patches L_{T0}^{k} tend to reduce with the increase of level k, and the highest level representation L_{T0}^{K} tends to have similar values and represent the same whole. From the

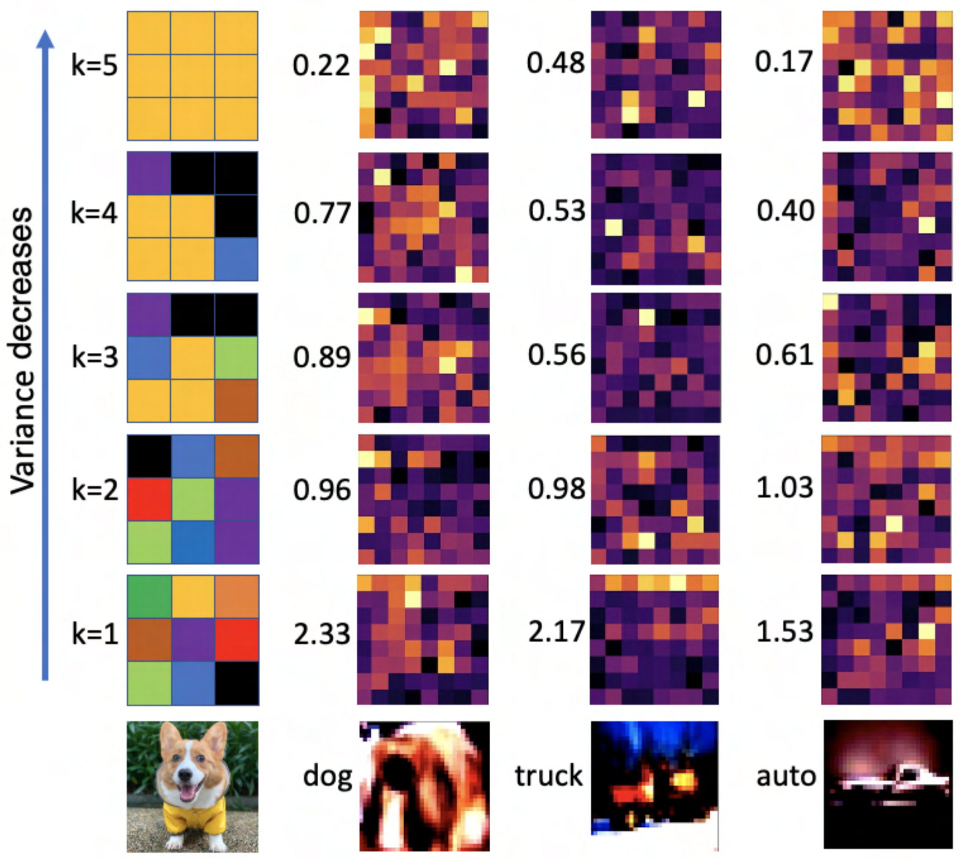


Figure 4.8. Input images and their 5 level representations. The representation variance decreases with the increase of level k.

figure, we can see that all the lowest level representations have the largest variances. In contrast, the level representations tend to agree on a common representation of the whole scene and achieve smaller variances with the increase of level k.

4.3. Discussion and Conclusion

In this chapter, we propose a general hierarchical ANN-SNN model, which is a human vision-like robust object recognition system. We illustrate the network structures of our model and its variants to show its generality and differentiability. Finally, we use extensive experiments to demonstrate the superiority and robustness of our proposed models over previous SNNs for object recognition. Moreover, inspired by the GLOM model, we propose various bio-inspired Spiking GLOM models, which make a step towards the effective, energy-efficient, and interpretable object recognition.

With this work, we intend to provide preliminary implementations and experimental object recognition results on these next-generation spiking models. Furthermore, we hope our work can inspire other researchers to explore more robust, energy-efficient, and interpretable AI techniques for diverse tasks with the lessons from other fields.

CHAPTER 5

Conclusion and Future Work

This chapter concludes the dissertation by summarizing the contributions and discussing directions for future work.

5.1. Conclusion

In this dissertation, we embarked on a journey to architect effective and efficient SNNs tailored for event-driven processing and learning, exploring three critical domains of inquiry. Our endeavors in developing SNNs for event-driven classification challenges, particularly in tactile learning, have yielded several fully SNN models that not only outperform existing approaches in tasks like tactile object recognition and slip detection but also highlight the models' exceptional energy efficiency. This efficiency, coupled with their performance, hints at the transformative potential of our models for deployment on energy-constrained hardware, broadening the scope of their applicability across various event-driven classification tasks.

Further, our exploration extended into resolving event-driven regression problems through SNNs, with a spotlight on event-based shape from polarization. The introduction of Single-Timestep and Multi-Timestep Spiking UNets for surface normal estimation stands as a testament to our commitment to advancing SNN capabilities in regression tasks. Our comprehensive evaluations on both synthetic and real-world datasets have confirmed that our models are on par with the performance of leading-edge ANNs, all the while bringing to the table superior energy efficiency. This achievement not only propels the field of SNNs in 3D scene tasks but also lays a solid groundwork for future efforts aimed at refining SNN architectures for regression-oriented tasks.

Lastly, our work has been enriched by cross-disciplinary inspirations, notably from neuroscience. By drawing parallels with the hierarchical visual processing observed in human vision, we have proposed ANN-SNN hybrid models that excel in robustness for object recognition tasks. Moreover, inspired by the hypothetical GLOM model from neuroscience, known for its capability to parse visual scenes and understand complex hierarchies, we have introduced Spiking GLOM models. These models, through the integration of spiking neurons, not only push the boundaries of energy efficiency but also enhance interpretability, marking a significant leap forward in our understanding and application of next-generation object recognition.

In conclusion, this dissertation not only advances the state-of-the-art in SNNs for event-driven processing and learning across classification and regression problems but also brings the lessons from neuroscience into practical spiking architecture design. Our work demonstrates the vast potential of SNNs in tackling complex computational tasks with superior energy efficiency, paving the way for their broader adoption and further exploration in the realm of neuromorphic computing and engineering.

5.2. Other Contributions from Collaborative Work

In collaboration with other researchers, we proposed a novel adaptive multimodal intensity-event algorithm to optimize an overall objective of object tracking under bit rate constraints for a host-chip architecture [**165**]. Please note that this work was funded

by Defense Advanced Research Projects Agency (DARPA) (Grant Number: HR0011-17-2-0044).

5.3. Future Directions

Moving forward, my research endeavors will pivot around pioneering enriched learning competencies in the forthcoming generation of intelligent computing systems and improving social welfare through neuromorphic technologies. A detailed expanse of this ambitious journey is demarcated below:

Large-Scale Neural Networks for Various On-Device Applications: The upcoming work will involve innovating algorithms for large-scale SNNs and ANNs, optimizing their compute complexity during both training and inference phases. The optimization will be grounded on metrics such as robustness, energy efficiency, and memory usage, particularly zeroing in on on-device applications, including but not limited to recommender systems, computer vision, natural language processing, and robotic manipulation.

Energy-Aware Software-Hardware Co-Design Frameworks for Edge AI: Progressing on a parallel path, the research will focus on designing energy-aware software-hardware co-design frameworks for ANNs and SNNs. The intention here is to perform a myriad of computational operations, such as pruning, quantization, and neural architecture search for learning or inference, tailoring them for various edge AI applications.

Innovative Computing Scenarios Exploration with a Multidisciplinary Canvas: The research endeavors will chart into the territories of novel computing scenarios, for instance, event-driven data processing and learning, multi-modal learning, continual learning, and federated learning. The multidisciplinary insights derived from understanding natural intelligence, including learning mechanisms and intricate brain architectures, will guide these explorations. The aim is to mold the next generation of AI systems, embarking on a journey where they not only perceive and reason but also make informed and real-time decisions.

Neuromorphic AI for Social Good: A promising direction in my future research will be the intersection of Neuromorphic AI with applications that amplify social good. While technological advancements have rapidly propelled AI systems' capabilities, there remains an imperative to channel these innovations toward social welfare. Outlined below is my envisioned trajectory in Neuromorphic AI tailored for societal benefits:

- Health and Well-being: Using Neuromorphic AI, I aim to develop assistive technologies for early detection, diagnosis, and therapeutic interventions for various health conditions. The energy-efficiency and real-time responsiveness of SNNs can be instrumental in wearable health devices, enabling continuous patient monitoring without cumbersome battery constraints.
- Environmental Sustainability: The efficient computational power of Neuromorphic AI can be harnessed to monitor environmental factors in real-time, aiding in the prediction of natural disasters, climate change impact analysis, and real-time pollution monitoring. These systems, being energy-efficient, align with the sustainability objective, reducing the carbon footprint of computational operations.
- Disaster Response and Management: Leveraging the rapid decision-making capability of Neuromorphic AI, I envisage developing systems that can efficiently coordinate disaster response. These systems can process vast amounts of data

from various sources in real-time, enabling quicker and more effective responses to emergencies.

- Assistive Technologies for Differently-abled: Drawing inspiration from the human brain's adaptability, I will explore Neuromorphic AI's potential in creating assistive technologies. These could range from mobility aids, communication devices, or environment interaction tools tailored for individuals with physical or cognitive challenges.

My envisioned future research is not merely a continuation of what has been accomplished, but a strategic and thought-out expansion into arenas that hold the promise of revolutionizing intelligent computing systems and fostering a society that is more inclusive, sustainable, and forward-thinking. I believe that these efforts will elevate the current paradigm of AI, especially in designing systems that are robust, energy-efficient, and capable of learning and inferring at the edge, thereby ushering us into a new era of technological advancement and discovery. Moreover, by intertwining the realms of advanced AI with societal well-being, I hope to carve a future where technology is an enabler of positive societal change.

References

[1] Guillermo Gallego, Tobi Delbruck, 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.

[2] Tasbolat Taunyazoz, 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.

[3] Jithendar Anumula, Daniel Neil, Tobi Delbruck, and Shih-Chii Liu, "Feature representations for neuromorphic audio spike streams," *Frontiers in neuroscience*, vol. 12, pp. 308889, 2018.

[4] Hanme Kim, Stefan Leutenegger, and Andrew J Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer, 2016, pp. 349–364.

[5] Fuqiang Gu, Weicong Sng, Tasbolat Taunyazov, and Harold Soh, "Tactilesgnet: 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.

[6] Michael Pfeiffer and Thomas Pfeil, "Deep learning with spiking neurons: opportunities and challenges," *Frontiers in neuroscience*, vol. 12, pp. 774, 2018.

[7] Christoph Posch, Teresa Serrano-Gotarredona, Bernabe Linares-Barranco, and Tobi Delbruck, "Retinomorphic event-based vision sensors: bioinspired cameras with spiking output," *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1470–1484, 2014.

[8] Lea Steffen, Daniel Reichard, Jakob Weinland, Jacques Kaiser, Arne Roennau, and Rüdiger Dillmann, "Neuromorphic stereo vision: A survey of bio-inspired sensors and algorithms," *Frontiers in neurorobotics*, vol. 13, pp. 28, 2019.

[9] P Lichtsteiner, "64x64 event-driven logarithmic temporal derivative silicon retina," in *Program 2003 IEEE Workshop on CCD and AIS*, 2003.

[10] Patrick Lichtsteiner and Tobi Delbruck, "A 64x64 aer logarithmic temporal derivative silicon retina," in *Research in Microelectronics and Electronics, 2005 PhD*. IEEE, 2005, vol. 2, pp. 202–205.

[11] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck, "A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change," in *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*. IEEE, 2006, pp. 2060–2069.

[12] Patrick Lichtensteiner, Christoph Posch, and T Delbruck, "A 128x128 120db 15μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, , no. 2, pp. 566–576, 2008.

[13] Roland S Johansson, "Tactile sensibility in the human hand: receptive field characteristics of mechanoreceptive units in the glabrous skin area.," *The Journal of physiology*, vol. 281, no. 1, pp. 101–125, 1978.

[14] Chankyu Lee, Syed Shakib Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers in neuroscience*, vol. 14, pp. 497482, 2020.

[15] Youngeun Kim, Hyoungseob Park, Abhishek Moitra, Abhiroop Bhattacharjee, Yeshwanth Venkatesha, and Priyadarshini Panda, "Rate coding or direct coding: Which one is better for accurate, robust, and energy-efficient spiking neural networks?," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 71–75.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[17] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[20] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5633–5643.

[21] 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.

[22] Zhanat Kappassov, Juan-Antonio Corrales, and Véronique Perdereau, "Tactile sensing in dexterous robot hands," *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3857–3866.

[28] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 6, pp. 1964–1980, 2019.

[29] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza, "Fast image reconstruction with an event camera," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 156–163.

[30] Federico Paredes-Vallés and Guido CHE de Croon, "Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3446–3455.

[31] Stepan Tulyakov, Francois Fleuret, Martin Kiefel, Peter Gehler, and Michael Hirsch, "Learning an event sequence embedding for dense event-based deep stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1527–1537.

[32] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza, "Learning monocular dense depth from events," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 534–542.

[33] Kaixuan Zhang, Kaiwei Che, Jianguo Zhang, Jie Cheng, Ziyang Zhang, Qinghai Guo, and Luziwei Leng, "Discrete time convolution for fast event-based stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8676–8686.

[34] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," *arXiv preprint arXiv:1802.06898*, 2018.

[35] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 989–997.

[36] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[37] Peng Kang, Jianping Zhang, Chen Ma, and Guiling Sun, "Atm: attentional text matting," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3902–3911.

[38] Chen Ma, Peng Kang, and Xue Liu, "Hierarchical gating networks for sequential recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 825–833.

[39] Chen Ma, Peng Kang, Bin Wu, Qinglong Wang, and Xue Liu, "Gated attentive-autoencoder for content-aware recommendation," in *Proceedings of the twelfth ACM international conference on web search and data mining*, 2019, pp. 519–527.

[40] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in neuroscience*, vol. 12, pp. 331, 2018.

[41] 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.

[42] 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.

[43] Lin Zhu, Xiao Wang, Yi Chang, Jianing Li, Tiejun Huang, and Yonghong Tian, "Event-based video reconstruction via potential-assisted spiking neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3594–3604.

[44] Jesse Hagenaars, Federico Paredes-Vallés, and Guido De Croon, "Self-supervised learning of event-based optical flow with spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7167–7179, 2021.

[45] 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.

[46] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[47] 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.

[48] Wulfram Gerstner, "Time structure of the activity in neural network models," *Physical review E*, vol. 51, no. 1, pp. 738, 1995.

[49] 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.

[50] Wulfram Gerstner and Werner M Kistler, *Spiking neuron models: Single neurons, populations, plasticity*, Cambridge university press, 2002.

[51] Peng Kang, Hao Hu, Srutarshi Banerjee, Henry Chopp, Aggelos Katsaggelos, and Oliver Cossairt, "Human vision-like robust object recognition," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 709–713.

[52] 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. 1–9.

[53] Peng Kang, Srutarshi Banerjee, and Henry Chopp, "Boost event-driven tactile learning with location spiking neurons," *Frontiers in Neuroscience*, vol. 17, pp. 1127537, 2023.

[54] Peng Kang, Srutarshi Banerjee, Henry Chopp, Aggelos Katsaggelos, and Oliver Cossairt, "Spiking glom: Bio-inspired architecture for next-generation object recognition," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 950–954.

[55] Peng Kang, Srutarshi Banerjee, Henry Chopp, Aggelos Katsaggelos, and Oliver Cossairt, "Event-based shape from polarization with spiking neural networks," *arXiv preprint arXiv:2312.16071*, 2023.

[56] 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.

[57] Emma Strubell, Ananya Ganesh, and Andrew McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[58] 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.

[59] Ed Bullmore and Olaf Sporns, “The economy of brain network organization,” *Nature Reviews Neuroscience*, vol. 13, no. 5, pp. 336–349, 2012.

[60] Daniel J Felleman and David C Van Essen, “Distributed hierarchical processing in the primate cerebral cortex,” in *Cereb cortex*. Citeseer, 1991.

[61] 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.

[62] 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.

[63] 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.

[64] Abhronil 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] Wolfgang Maass and Christopher M Bishop, *Pulsed neural networks*, MIT press, 2001.

[70] Jian Du, Shanghang Zhang, Guanhang Wu, José MF Moura, and Soummya Kar, "Topology adaptive graph convolutional networks," *arXiv preprint arXiv:1710.10370*, 2017.

[71] 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.

[72] 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.

[73] 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.

[74] Mark Horowitz, "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)*. IEEE, 2014, pp. 10–14.

[75] R Gary Leonard and George Doddington, "Tidigits speech corpus," *Texas Instruments, Inc*, 1993.

[76] 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.

[77] Wenshuo Li, Hanting Chen, Jianyuan Guo, Ziyang Zhang, and Yunhe Wang, "Brain-inspired multilayer perceptron with spiking neurons," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 783–793.

[78] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al., "Mlp-mixer: An all-mlp architecture for vision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24261–24272, 2021.

[79] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[80] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006, vol. 7, p. 0.

[81] Jundan Luo, Zhaoyang Huang, Yijin Li, Xiaowei Zhou, Guofeng Zhang, and Hujun Bao, "Niid-net: adapting surface normal knowledge for intrinsic image decomposition in indoor scenes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 12, pp. 3434–3445, 2020.

[82] Yakun Ju, Junyu Dong, and Sheng Chen, "Recovering surface normal and arbitrary images: A dual regression network for photometric stereo," *IEEE Transactions on Image Processing*, vol. 30, pp. 3676–3690, 2021.

[83] Matti Strese, Clemens Schuwerk, Albert Iepure, and Eckehard Steinbach, "Multimodal feature-based surface material classification," *IEEE transactions on haptics*, vol. 10, no. 2, pp. 226–239, 2016.

[84] Hernan Badino, Daniel Huber, Yongwoon Park, and Takeo Kanade, "Fast and accurate computation of surface normals from range images," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3084–3091.

[85] Jason Geng, "Structured-light 3d surface imaging: a tutorial," *Advances in optics and photonics*, vol. 3, no. 2, pp. 128–160, 2011.

[86] Berthold KP Horn, "Shape from shading: A method for obtaining the shape of a smooth opaque object from one view," 1970.

[87] Steven A Shafer and Takeo Kanade, “Using shadows in finding surface orientations,” *Computer Vision, Graphics, and Image Processing*, vol. 22, no. 1, pp. 145–176, 1983.

[88] Andrew P Witkin, “Recovering surface shape and orientation from texture,” *Artificial intelligence*, vol. 17, no. 1-3, pp. 17–45, 1981.

[89] Carlos Hernández, “Stereo and silhouette fusion for 3d object modeling from uncalibrated images under circular motion,” *These de doctorat, École Nationale Supérieure des Télécommunications*, vol. 2, 2004.

[90] Robert J Woodham, “Photometric method for determining surface orientation from multiple images,” *Optical engineering*, vol. 19, no. 1, pp. 139–144, 1980.

[91] Stefan Rahmann and Nikos Canterakis, “Reconstruction of specular surfaces using polarization imaging,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.* IEEE, 2001, vol. 1, pp. I–I.

[92] Achuta Kadambi, Vage Taamazyan, Boxin Shi, and Ramesh Raskar, “Polarized 3d: High-quality depth sensing with polarization cues,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3370–3378.

[93] Lawrence B Wolff and Terrance E Boult, “Constraining object features using a polarization reflectance model,” *Phys. Based Vis. Princ. Pract. Radiom*, vol. 1, pp. 167, 1993.

[94] Silvia Tozza, William AP Smith, Dizhong Zhu, Ravi Ramamoorthi, and Edwin R Hancock, “Linear differential constraints for photo-polarimetric height estimation,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2279–2287.

[95] Yunhao Ba, Alex Gilbert, Franklin Wang, Jinfa Yang, Rui Chen, Yiqin Wang, Lei Yan, Boxin Shi, and Achuta Kadambi, “Deep shape from polarization,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16.* Springer, 2020, pp. 554–571.

[96] Chenyang Lei, Chenyang Qi, Jiaxin Xie, Na Fan, Vladlen Koltun, and Qifeng Chen, “Shape from polarization for complex scenes in the wild,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12632–12641.

[97] Achuta Kadambi, Vage Taamazyan, Boxin Shi, and Ramesh Raskar, "Depth sensing using geometrically constrained polarization normals," *International Journal of Computer Vision*, vol. 125, pp. 34–51, 2017.

[98] Gary A Atkinson and Jürgen D Ernst, "High-sensitivity analysis of polarization by surface reflection," *Machine Vision and Applications*, vol. 29, pp. 1171–1189, 2018.

[99] Lawrence B Wolff, "Polarization vision: a new sensory approach to image understanding," *Image and Vision computing*, vol. 15, no. 2, pp. 81–93, 1997.

[100] "Lucid vision phoenix polarization camera," `https://thinklucid.com/product/phoenix-5-0-mp-polarized-model/`, 2018.

[101] Manasi Muglikar, Leonard Bauersfeld, Diederik Paul Moeys, and Davide Scaramuzza, "Event-based shape from polarization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1547–1556.

[102] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[103] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan, "Spikformer: When spiking neural network meets transformer," *arXiv preprint arXiv:2209.15425*, 2022.

[104] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang, "Spiking transformers for event-based single object tracking," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2022, pp. 8801–8810.

[105] Zulun Zhu, Jiaying Peng, Jintang Li, Liang Chen, Qi Yu, and Siqiang Luo, "Spiking graph convolutional networks," *arXiv preprint arXiv:2205.02767*, 2022.

[106] Rui-Jie Zhu, Qihang Zhao, and Jason K Eshraghian, "Spikegpt: Generative pre-trained language model with spiking neural networks," *arXiv preprint arXiv:2302.13939*, 2023.

[107] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[108] William A Shurcliff, *Polarized light: production and use*, Harvard University Press, 1962.

[109] Edward Collett, "Field guide to polarization," Spie Bellingham, WA, 2005.

[110] Boxin Shi, Jinfa Yang, Jinwei Chen, Ruihua Zhang, and Rui Chen, "Recent progress in shape from polarization," *Advances in Photometric 3D-Reconstruction*, pp. 177–203, 2020.

[111] Miyazaki, Kagesawa, and Ikeuchi, "Polarization-based transparent surface modeling from two views," in *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 1381–1386.

[112] Daisuke Miyazaki, Takuya Shigetomi, Masashi Baba, Ryo Furukawa, Shinsaku Hiura, and Naoki Asada, "Surface normal estimation of black specular objects from multiview polarization images," *Optical Engineering*, vol. 56, no. 4, pp. 041303–041303, 2017.

[113] Luwei Yang, Feitong Tan, Ao Li, Zhaopeng Cui, Yasutaka Furukawa, and Ping Tan, "Polarimetric dense monocular slam," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3857–3866.

[114] Dizhong Zhu and William AP Smith, "Depth from a polarisation+ rgb stereo pair," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7586–7595.

[115] Christophe Stolz, Mathias Ferraton, and Fabrice Meriaudeau, "Shape from polarization: a method for solving zenithal angle ambiguity," *Optics letters*, vol. 37, no. 20, pp. 4218–4220, 2012.

[116] Lawrence B Wolff, Shree K Nayar, and Michael Oren, "Improved diffuse reflection models for computer vision," *International Journal of Computer Vision*, vol. 30, pp. 55–71, 1998.

[117] Ye Yu, Dizhong Zhu, and William AP Smith, "Shape-from-polarisation: a nonlinear least squares approach," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2969–2976.

[118] William AP Smith, Ravi Ramamoorthi, and Silvia Tozza, "Height-from-polarisation with unknown lighting or albedo," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 12, pp. 2875–2888, 2018.

[119] N Justin Marshall, "A unique colour and polarization vision system in mantis shrimps," *Nature*, vol. 333, no. 6173, pp. 557–560, 1988.

[120] Germain Haessig, Damien Joubert, Justin Haque, Moritz B Milde, Tobi Delbruck, and Viktor Gruev, "Pdavis: Bio-inspired polarization event camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3962–3971.

[121] Mathias Gehrig, Sumit Bam Shrestha, Daniel Mouritzen, and Davide Scaramuzza, "Event-based angular velocity regression with spiking networks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4195–4202.

[122] Ulysse Rançon, Javier Cuadrado-Anibarro, Benoit R Cottereau, and Timothée Masquelier, "Stereospike: Depth learning with a spiking neural network," *IEEE Access*, vol. 10, pp. 127428–127439, 2022.

[123] Xiaoshan Wu, Weihua He, Man Yao, Ziyang Zhang, Yaoyuan Wang, and Guoqi Li, "Mss-depthnet: Depth prediction with multi-step spiking neural network," *arXiv preprint arXiv:2211.12156*, 2022.

[124] Xingting Yao, Qinghao Hu, Tielong Liu, Zitao Mo, Zeyu Zhu, Zhengyang Zhuge, and Jian Cheng, "Spiking nerf: Making bio-inspired neural networks see through the real world," *arXiv preprint arXiv:2309.10987*, 2023.

[125] Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza, "Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17755–17764.

[126] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian, "Incorporating learnable membrane time constant to enhance learning of spiking neural networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2661–2671.

[127] Augustus Odena, Vincent Dumoulin, and Chris Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, pp. e3, 2016.

[128] Beck Strohmer, Rasmus Karnøe Stagsted, Poramate Manoonpong, and Leon Bonde Larsen, "Integrating non-spiking interneurons in spiking neural networks," *Frontiers in neuroscience*, vol. 15, pp. 633945, 2021.

[129] Zhenzhi Wu, Hehui Zhang, Yihan Lin, Guoqi Li, Meng Wang, and Ye Tang, "Liaf-net: Leaky integrate and analog fire network for lightweight and efficient spatiotemporal information processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6249–6262, 2021.

[130] Robert Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, pp. 65–93. Elsevier, 1992.

[131] Paul J Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[132] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, 2019.

[133] Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, and Yonghong Tian, "Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence," *Science Advances*, vol. 9, no. 40, pp. eadi1480, 2023.

[134] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[135] Eimantas Ledinauskas, Julius Ruseckas, Alfonsas Juršėnas, and Giedrius Buračas, "Training deep spiking neural networks," *arXiv preprint arXiv:2006.04436*, 2020.

[136] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[137] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang, "Mitsuba 3 renderer," 2022, https://mitsuba-renderer.org.

[138] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza, "Esim: an open event camera simulator," in *Conference on robot learning*. PMLR, 2018, pp. 969–982.

[139] T Finateu, A Niwa, D Matolin, K Tsuchimoto, A Mascheroni, E Reynaud, P Mostafalu, F Brady, L Chotard, F LeGoff, et al., "A 1280x720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 um pixels, 1.066

geps readout, programmable event-rate controller and compressive data-formatting pipeline," in *IEEE International Solid-State Circuits Conference*, 2020.

[140] "Breakthrough photography x4 polarizer," `https://breakthrough.photography/products/x4-circular-polarizer`.

[141] Manasi Muglikar, Guillermo Gallego, and Davide Scaramuzza, "Esl: Event-based structured light," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 1165–1174.

[142] Ali H Mahmoud, Moumen T El-Melegy, and Aly A Farag, "Direct method for shape recovery from polarization and shading," in *2012 19th IEEE International Conference on Image Processing*. IEEE, 2012, pp. 1769–1772.

[143] Peter U Diehl and Matthew Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in computational neuroscience*, vol. 9, pp. 99, 2015.

[144] Sumit Bam Shrestha and Garrick Orchard, "Slayer: Spike layer error reassignment in time," *arXiv preprint arXiv:1810.08646*, 2018.

[145] Xiang Cheng, Yunzhe Hao, Jiaming Xu, and Bo Xu, "Lisnn: Improving spiking neural networks with lateral interactions for robust object recognition," in *IJCAI*, pp. 1519–1525.

[146] Yingyezhe Jin, Wenrui Zhang, and Peng Li, "Hybrid macro/micro level backpropagation for training deep spiking neural networks," *arXiv preprint arXiv:1805.07866*, 2018.

[147] Maximilian Riesenhuber and Tomaso Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.

[148] Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, and Sarah Mack, *Principles of neural science*, vol. 4, McGraw-hill New York, 2000.

[149] Peter Dayan and Laurence F Abbott, *Theoretical neuroscience: computational and mathematical modeling of neural systems*, Computational Neuroscience Series, 2001.

[150] Dale Purves, G Augustine, D Fitzpatrick, L Katz, A LaMantia, J McNamara, and S Williams, "Neuroscience 2nd edition. sunderland (ma) sinauer associates," *Types of Eye Movements and Their Functions*, 2001.

[151] DC Hood, "Lower-level visual processing and models of light adaptation," *Annual review of psychology*, vol. 49, no. 1, pp. 503–535, 1998.

[152] Tim Gollisch and Markus Meister, "Eye smarter than scientists believed: neural computations in circuits of the retina," *Neuron*, vol. 65, no. 2, pp. 150–164, 2010.

[153] Shannon Saszik and Steven H DeVries, "A mammalian retinal bipolar cell uses both graded changes in membrane voltage and all-or-nothing na+ spikes to encode light," *Journal of Neuroscience*, vol. 32, no. 1, pp. 297–307, 2012.

[154] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[155] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu, "Phased lstm: Accelerating recurrent network training for long or event-based sequences," *arXiv preprint arXiv:1610.09513*, 2016.

[156] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in neuroscience*, vol. 10, pp. 508, 2016.

[157] Daniel Neil and Shih-Chii Liu, "Effective sensor fusion with event-based sensors and deep network architectures," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 2282–2285.

[158] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[159] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.

[160] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *Ieee Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[161] Garrick Orchard, E Paxon Frady, Daniel Ben Dayan Rubin, Sophia Sanborn, Sumit Bam Shrestha, Friedrich T Sommer, and Mike Davies, "Efficient neuromorphic signal processing with loihi 2," in *2021 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, 2021, pp. 254–259.

[162] Geoffrey Hinton, “How to represent part-whole hierarchies in a neural network,” *Neural Computation*, pp. 1–40, 2022.

[163] Jeff Hawkins, *A thousand brains: A new theory of intelligence*, Basic Books, 2021.

[164] Nicola Garau, Niccoló Bisagno, Zeno Sambugaro, and Nicola Conci, “Interpretable part-whole hierarchies and conceptual-semantic relationships in neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13689–13698.

[165] Srutarshi Banerjee, Henry H Chopp, Jianping Zhang, Zihao W Wang, Peng Kang, Oliver Cossairt, and Aggelos Katsaggelos, “A joint intensity-neuromorphic event imaging system with bandwidth-limited communication channel,” *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[166] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar, “Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set,” *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.

APPENDIX A

Appendix for Chapter 2

A.1. Acronyms and Notations

A.2. Datasets

This section introduces the details of the datasets for the Hybrid_SRM_FC and the datasets for the Hybrid_LIF_GNN.

A.2.1. Datasets for the Hybrid_SRM_FC

Objects-v1: There are 36 object classes in this dataset, including 26 objects directly from the YCB dataset [**166**] and 10 deformable YCB objects. To collect each event-driven tactile data sequence, the gripper grasped an object, lifted it off the table by 20 cm, and placed it back onto the table. Each data sequence is 6.5 seconds, including the time from lifting the object to releasing it. And we sample the spikes every 0.02 seconds like [**2**], which results in the total time length of this dataset being 325. For each object class, 25 event sequences are collected, yielding a total of 900 event sequences. Each gripper has two event-driven tactile sensors. And we use the event data from these two tactile sensors (78 taxels) in the experiments.

Containers-v1: There are 20 object classes in this dataset, including four containers with five different volumes. Four containers are a coffee can, a plastic soda bottle, a soymilk carton, and a metal tuna can. Five different volumes are 0%, 25%, 50%, 75%, 100% of the respective maximum amount of water (or rice for the tuna can). The gripper grasped a container and lifted it off the table by 5 cm to collect each tactile data sequence. Each data sequence is 6.5 seconds, including the time from grasping the container to lifting and holding it for a while. And we sample the spikes every 0.02 seconds like [**2**], which results in the total time length of this dataset being 325. For each class, 40 event sequences are collected, yielding a total of 800 event sequences. Each gripper has two event-driven tactile sensors. And we use the event data from these two tactile sensors (78 taxels) in the experiments.

Slip Detection: Two objects are used in the experiments. One is stable and the other one is unstable. They are visually identical and have the same overall weight. The model is required to determine whether the held object is stable or rotational in a short time. To collect each event-driven tactile data sequence, the gripper is instructed to close upon the object, lift by 10cm off the table in 0.75 seconds and hold it for an additional 4.25 seconds. Since we are interested in rapid slip detection, we extract a 0.15s window around the start of the lift like [**2**]. And we sample the spikes every 0.001 seconds, which results in the total time length of this dataset being 150. For each object, 50 event sequences are collected, yielding a total of 100 event sequences. Each gripper has two event-driven tactile sensors. And we use the event data from these two tactile sensors (78 taxels) in the experiments.

Interested readers can find more details about the datasets from [**2**] and the corresponding website `https://clear-nus.github.io/visuotactile/download.html`.

A.2.2. Datasets for the Hybrid_LIF_GNN

Objects-v0: There are 36 object classes in this dataset, including 26 objects directly from the YCB dataset [**166**] and 10 deformable YCB objects. The gripper grasped an object, lifted it off the table by 20 cm, and placed it back onto the table to collect each event-driven tactile data sequence. Each data sequence is 5 seconds, including the time from lifting the object to releasing it. And we sample the spikes every 0.02 seconds like [**5**], which results in the total time length of this dataset being 250. For each object class, 20 event sequences are collected, yielding a total of 720 event sequences. Each gripper has two event-driven tactile sensors. And we use the event data from only one tactile sensor (39 taxels) in the experiments to fairly compare with other published results.

Containers-v0: There are 20 object classes in this dataset, including four containers with five different volumes. Four containers are a coffee can, a plastic soda bottle, a soymilk carton, and a metal tuna can. Five different volumes are 0%, 25%, 50%, 75%, 100% of the respective maximum amount of water (or rice for the tuna can). The gripper grasped a container and lifted it off the table by 5 cm to collect each tactile data sequence. Each data sequence is 6.5 seconds, including the time from grasping the container to lifting and holding it for a while. And we sample the spikes every 0.02 seconds like [**5**], which results in the total time length of this dataset being 325. For each class, 15 event sequences are collected, yielding a total of 300 event sequences. Each gripper has two event-driven tactile sensors. And we use the event data from only one tactile sensor (39 taxels) in the experiments to fairly compare with other published results.

Interested readers can find more details about the datasets from [**5**] and the corresponding website `https://clear-nus.github.io/visuotactile/download.html`.

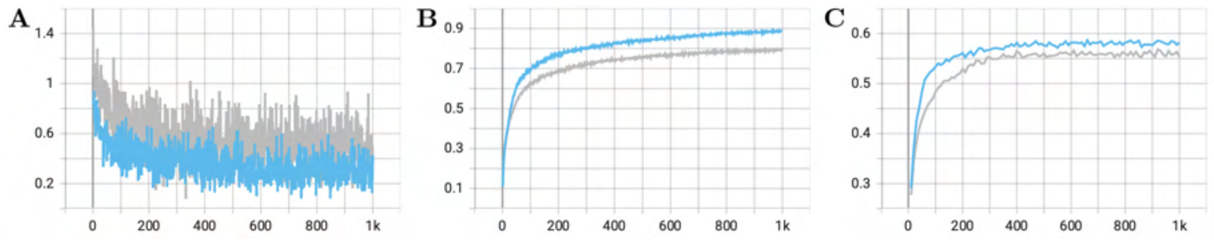


Figure A.1. Training and testing profiles for the SNN with TSRM neurons (gray) and the Hybrid_SRM_FC (blue): **(A)** the training loss, **(B)** the training accuracy, **(C)** the testing accuracy.

A.3. Mean Spiking Rates

We present the mean spiking rates of Hybrid_SRM_FC layers in Table A.2. And Table A.3 provides the mean spiking rates of Hybrid_LIF_GNN layers.

A.4. Training and Testing Profiles on Event-Driven Audio Learning

We show the training and testing profiles of the SNN with TSRM neurons and the Hybrid_SRM_FC in Fig. A.1. From these figures, we can see that our hybrid model converges faster and attains a lower loss and a higher accuracy compared to the SNN with TSRM neurons on event-driven audio learning. 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 the benefits to the models built with conventional spiking neurons and improve their performance on the tasks.

Table A.1. List of Acronyms and Notations in Chapter 2

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 (see Fig. 2.4)
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 (see Fig. 2.6)
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

Table A.2. Mean spiking rates of Hybrid_SRM_FC layers.

	Objects-v1	Containers-v1	Slip Detection
Input layer	0.1539	0.2316	0.0333
Spiking FC layer1 with TSRM neurons	0.2148	0.2461	0.0536
Spiking FC layer2 with TSRM neurons	0.0146	0.0176	0.1962
Spiking FC layer1 with LSRM neurons	0.2134	0.3576	0.0232
Spiking FC layer2 with LSRM neurons	0.0284	0.0376	0.0824

Table A.3. Mean spiking rates of Hybrid_LIF_GNN layers

	Objects-v0	Containers-v0
Input layer	0.1272	0.0994
Spatial Spiking Graph layer	0.0271	0.0106
Spatial Spiking Fully-connected layer1	0.1503	0.0857
Spatial Spiking Fully-connected layer2	0.1767	0.1077
Spatial Spiking Fully-connected layer3	0.0403	0.0580
Temporal Spiking Graph layer	0.0134	0.0067
Temporal Spiking Fully-connected layer1	0.1100	0.2196
Temporal Spiking Fully-connected layer2	0.1275	0.1489
Temporal Spiking Fully-connected layer3	0.0213	0.0419

Vita

Peng Kang (Member, IEEE) is currently a final 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 and journals, like KDD, WSDM, WACV, ICIP, IJCNN, and Frontiers in Neuroscience. He served as reviewers for several premier conferences and journals, including WACV 2024, 2023, 2022, 2021, IJCNN 2024, 2023, 2022, AISTATS 2020, IEEE CAI 2024, Neurocomputing, Neural Networks, PLoS One, IEEE TCI, IEEE TNNLS, IEEE CIM.