ASRC-SNN: Adaptive Skip Recurrent Connection Spiking Neural Network

Anonymous Authors[1]

Abstract

In recent years, Recurrent Spiking Neural Networks (RSNNs) have shown promising potential in long-term temporal modeling. Many studies focus on improving neuron models and also integrate recurrent structures, leveraging their synergistic effects to improve the long-term temporal modeling capabilities of Spiking Neural Networks (SNNs). However, these studies often place an excessive emphasis on the role of neurons, overlooking the importance of analyzing neurons and recurrent structures as an integrated framework. In this work, we consider neurons and recurrent structures as an integrated system and conduct a systematic analysis of gradient propagation along the temporal dimension, revealing a challenging gradient vanishing problem. To address this issue, we propose the Skip Recurrent Connection (SRC) as a replacement for the vanilla recurrent structure, effectively mitigating the gradient vanishing problem and enhancing long-term temporal modeling performance. Additionally, we propose the Adaptive Skip Recurrent Connection (ASRC), a method that can learn the skip span of skip recurrent connection in each layer of the network. Experiments show that replacing the vanilla recurrent structure in RSNN with SRC significantly improves the model's performance on temporal benchmark datasets. Moreover, ASRC-SNN outperforms SRC-SNN in terms of temporal modeling capabilities and robustness.

1. Introduction

In artificial neural networks (ANNs), activation values are continuous, and computations involve numerous computationally expensive multiply-accumulation (MAC) operations. In contrast, in spiking neural networks (SNNs), the activation is represented by binary spike signals, where most MAC operations can be replaced by energy-efficient accumulation (AC) operations. Moreover, the sparse generation of spikes (Roy et al., 2019; Nunes et al., 2022) in SNN further reduces the number of AC operations. As a result, SNNs have a significant advantage in energy consumption compared to ANNs. This theoretical advantage has been validated in practice with SNNs deployed on neuromorphic hardware that demonstrate fast inference and low power consumption (Akopyan et al., 2015; Davies et al., 2018; Pei et al., 2019).

Leaky Integrate-and-Fire (LIF) neuron model (Gerstner & Kistler, 2002), due to their computational efficiency and similarity to biological neurons, have become the most widely used in SNN. Furthermore, the favorable temporal dynamics of LIF neurons makes LIF neuron-based SNNs well suited for handling temporal tasks. Nowadays, for static image classification tasks, replicating the image multiple times along the temporal dimension to introduce simple temporal features has enabled SNNs to achieve performance comparable to that of ANNs (Ding et al., 2021; Zhou et al., 2023; Yao et al., 2024; Zhou et al., 2024). For tasks that align with the event-driven paradigm and incorporate intrinsic temporal features, such as neuromorphic image datasets like CIFAR-10-DVS (Li et al., 2017) and DVS-128-Gesture (Amir et al., 2017), which are captured using Dynamic Vision Sensors (DVS) (Leñero-Bardallo et al., 2011), SNNs have demonstrated exceptional performance (Deng et al., 2023; Ma et al., 2023; Wang et al., 2023; Huang et al., 2024).

The tasks mentioned above are characterized by short time steps and simple temporal dependencies. For complex tasks that require the establishment of long-term temporal dependencies, such as speech recognition and sequence recognition, SNNs that rely solely on LIF neurons to capture temporal relationships are generally less competitive in terms of performance compared to ANNs (Yin et al., 2021). To enhance the competitiveness of SNNs, a common approach is to introduce the recurrent structure (Elman, 1990) and improve the neuron model based on simple SNNs (Yin et al., 2021; Bittar & Garner, 2022; Zhang et al., 2024; Baronig et al., 2024). In analyzing the temporal modeling capabilities of these improved models, these studies primarily focus on the function of neurons, while treating the recurrent connection as an additional mechanism aimed at enhancing

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

the model performance. We believe that both the recurrent connection and the neuron play a synergistic role in capturing temporal dependencies. This work is the first to treat both components as a unified system for gradient analysis along the temporal dimension. This reveals that vanilla recurrent spiking neural networks (RSNNs) based on LIF neurons suffer from vanishing and exploding gradients when gradients propagate along the temporal dimension. At the same time, we provide the corresponding solutions to address these issues. Specifically, the exploding gradient problem, which arises from the recurrent structure, can be mitigated by orthogonal initialization (Henaff et al., 2016); the challenging vanishing gradient problem, which arises from both the LIF neurons and the recurrent structure, can be alleviated by replacing the vanilla recurrent connection with the skip recurrent connection (SRC). Experiments show that SRC-SNN significantly outperforms vanilla RSNN in long-term temporal tasks.

Furthermore, We identified two limitations of SRC: The uniform skip connection span across layers in SRC-SNN constrains the network's temporal modeling ability; the hyperparameter tuning process is complex. To address these limitations, we propose the adaptive skip recurrent connection (ASRC). For each layer in the ASRC-SNN, we introduce a temperature-scaled Softmax kernel. This Softmax kernel enables competition among multiple skip connections of varying temporal spans, while the temperature parameter gradually decreasing to intensify this competition. Specifically, initially, the kernel assigns equal weights to multiple skip connections; as training progresses, the temperature parameter decreases, and the Softmax kernel progressively concentrates the weights on the most relevant skip connection. Experiments show that ASRC-SNN demonstrates superior long-term temporal modeling capabilities and robustness compared to SRC-SNN.

2. Related Work

2.1. Long-term temporal modeling in RSNNs

ALIF (Yin et al., 2021) extends the LIF neuron model by incorporating a dynamic threshold mechanism, thereby enhancing their ability to model temporal sequences. (Bittar & Garner, 2022; Baronig et al., 2024) argues that the dynamics of LIF neurons is relatively simple and proposes the introduction of a second time-varying variable to model the oscillatory behavior. (Zhang et al., 2024) propose a novel biologically inspired two-compartment leaky integrative-and-fire (TC-LIF) spiking neuron model, which integrates specifically designed somatic and dendritic compartments aimed at improving the learning of long-term temporal dependencies. To further improve model performance, these enhanced neuron models incorporate the recurrent structure (Elman, 1990). Inspired by the success of gated recurrent units (GRUs) in ANNs, (Dampfhoffer et al., 2022) have proposed the spiking GRU model. (Wang & Yu, 2024) introduces a novel spatial-temporal circuit (STC) model that incorporates two learnable adaptive pathways, improving the temporal memory capabilities of spiking neurons.

2.2. Long-term temporal modeling in other SNNs

By reformulating the neuronal dynamics without reset into a general mathematical form, (Fang et al., 2024) introduces a series of parallel spiking neuron (PSN) models, which transform the membrane potential charging dynamics into a learnable decay matrix. The parallel multi-compartment spiking neuron (PMSN) (Chen et al., 2024) mimics biological neurons by incorporating multiple interacting substructures, enabling effective representation of temporal information across diverse timescales. Both of these studies propose corresponding parallelization methods to enhance computational efficiency. The balanced resonate-and-fire neuron (BRFN) (Higuchi et al., 2024) builds upon the resonate-and-fire neuron by incorporating a dynamic threshold mechanism to simulate the refractory period, thus effectively maintaining the stability of the oscillatory process. DCLS-delays (Hammouamri et al., 2024) demonstrates strong performance in speech datasets by learning synaptic delays. Additionally, some studies have drawn inspiration from successful architectures in ANNs and effectively applied them to SNNs. For example, (Yao et al., 2021) introduces the attention mechanism, (Sadovsky et al., 2023) utilizes convolutional neural networks (CNNs), (Liu et al., 2024) incorporates the legendre memory unit (LMU) and (Stan & Rhodes, 2024; Shen et al., 2024) explore state space models (SSMs).

3. Methods

In this section, we first introduce the LIF neuron and then present the basic paradigm of RSNNs based on LIF neurons, with a focus on analyzing gradient propagation over the time dimension. Through this analysis, we identify the issues of vanishing and exploding gradients. For the vanishing gradient issue, we propose replacing the vanilla recurrent structure with the SRC structure. Finally, we discuss the limitations of the SRC-SNN model and introduce the ASRC-SNN model to overcome these limitations.

3.1. Preliminary

3.1.1. LIF Neuron Model

The LIF neuron is the most commonly used neuron model in the field of SNNs, known for its simplicity and computational efficiency while still capturing key aspects of neuronal dynamics. Its mathematical expressions are given by the

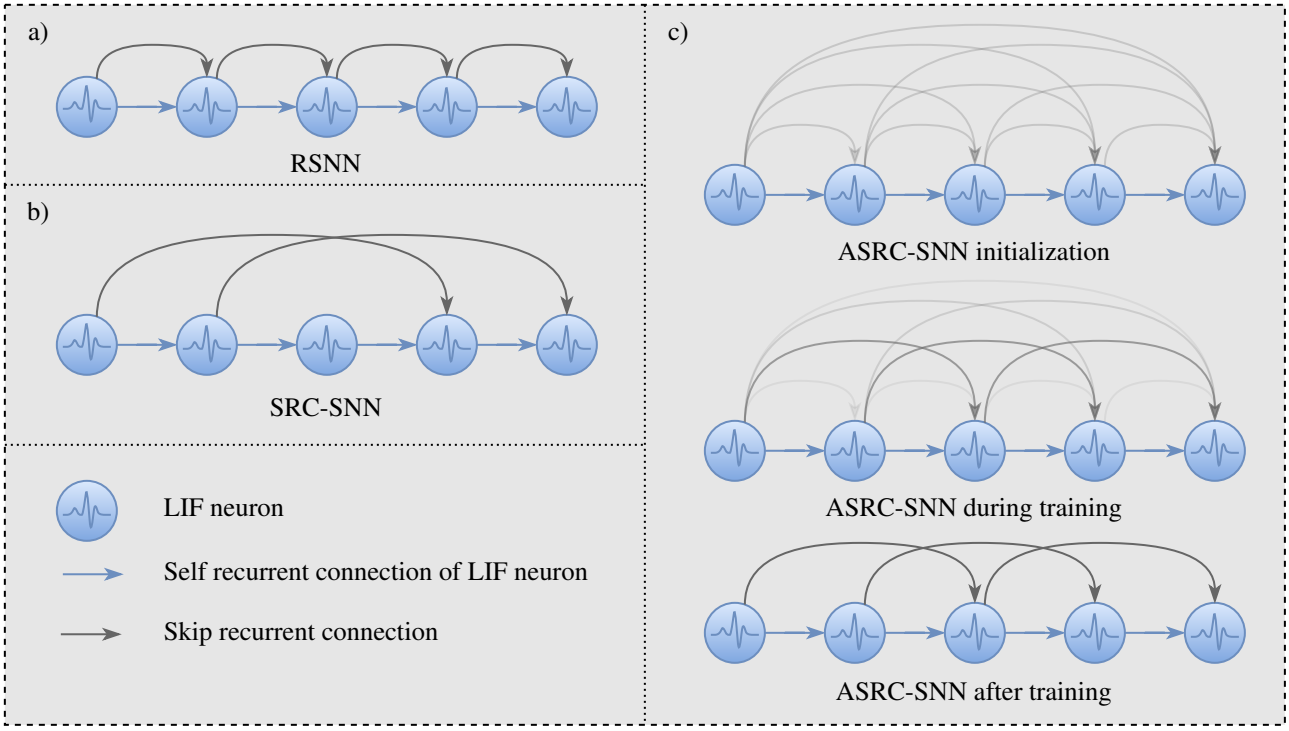


Figure 1: This figure demonstrates the flow of temporal information within the LIF neurons of the vanilla RSNN, SRC-SNN, and ASRC-SNN models. a) illustrates that in RSNN, recurrent connections are restricted to adjacent time steps. b) illustrates that in SRC-SNN, recurrent connections can span multiple time steps. c) illustrates the dynamic evolution of skip recurrent connections in ASRC-SNN from the start to the end of training. ASRC-SNN initialization: At the beginning of training, each LIF neuron in ASRC-SNN is connected to T_λ skip recurrent connections, with their weights initialized to $\frac{1}{T_\lambda}$. ASRC-SNN during training: During training, the weight distribution of the T_λ skip recurrent connections becomes more concentrated. ASRC-SNN after training: After training, the weights of the T_λ skip recurrent connections converge onto a single skip recurrent connection.

following equations:

$$U^l[t] = \alpha U^l[t-1] + I^l[t] \tag{1}$$

$$S^l[t] = H(U^l[t] - V_{th}) \tag{2}$$

$$U^l[t] = U^l[t] - V_{th} S^l[t] \tag{3}$$

Here, $U^l[t]$ and $I^l[t]$ represent the membrane potential and the input current of the neuron in the l-th layer at time step t. α is a decay factor that controls membrane potential leakage and ranges from 0 to 1. Eq. (1) describes how the membrane potential evolves over time by integrating the previous membrane potential with the input current, while the decay factor reflects the natural leakage of the potential. $H(\cdot)$ is Heaviside function. $S^l[t]$ represent the spike state of the l-th layer neuron at time step t, where $S^l[t] = 1$ if the neuron fires a spike and $S^l[t] = 0$ otherwise. Eq. (2) represents that if the membrane potential $U^l[t]$ exceeds the threshold V_{th}, the neuron fires; otherwise, no spike occurs. Eq. (3) describes the reset process of the membrane potential after a spike is fired. This soft reset mechanism is designed to preserve more information, as suggested by (Rueckauer et al., 2017; Han et al., 2020; Huang et al., 2024).

3.1.2. Paradigm of LIF-based RSNN

The SNN without vanilla recurrent connections has $I^l[t] = W_1^l S^{l-1}[t]$, while the SNN with vanilla recurrent connections has $I^l[t] = W_1^l S^{l-1}[t] + W_2^l S^l[t-1]$. Here W_1^l and W_2^l represent the parameters of the feedforward connections and recurrent connections, respectively, in the l-th layer. This paper focuses on the mechanisms of RSNNs. By substituting $I^l[t] = W_1^l S^{l-1}[t] + W_2^l S^l[t-1]$ into Eq. (1) and integrating Equations Eq. (1) and Eq. (3), the membrane potential update equation for the neurons in the l-th layer of the RSNN can be derived:

$$U^l[t] = \alpha(U^l[t-1] - V_{th} S^l[t-1]) + W_1^l S^{l-1}[t] + W_2^l S^l[t-1] \tag{4}$$

In a vanilla LIF-based RSNN, the l-th layer can be described by Eq. (4) and Eq. (2), with S^0 considered as the network's input.

3.2. Temporal Gradient Analysis of LIF-based RSNN

(Yin et al., 2021; Bittar & Garner, 2022; Baronig et al., 2024; Zhang et al., 2024) treat the recurrent connection as an additional mechanism aimed at improving the model's performance. In contrast, we consider the recurrent structure and neurons as working synergistically, analyzing them within a unified framework. Considering the propagation of gradients across adjacent time steps, we have:

$$\frac{\partial U^l[t+1]}{\partial U^l[t]} = \alpha + (W_2^l - \alpha V_{th})\frac{\partial S^l[t]}{\partial U^l[t]} \tag{5}$$

The Heaviside function is non-differentiable, and a common approach is to use a surrogate gradient function to approximate its derivative (Neftci et al., 2019). Similar to (Deng et al., 2022; Zhang et al., 2024), we use the triangle function as the surrogate gradient function:

$$\frac{\partial S^l[t]}{\partial U^l[t]} \approx \mathbb{H}(U^l[t]) = \frac{1}{\gamma^2}\max(0, \gamma - |U^l[t] - V_{th}|) \tag{6}$$

where γ represents the constraint factor that governs the range of samples required to activate the gradient. In this work, we set $\gamma = V_{th}$. Considering the propagation of the gradient over a longer time span, we have:

$$\begin{aligned}\frac{\partial U^l[t+k]}{\partial U^l[t]} &= \frac{\partial U^l[t+k]}{\partial U^l[t+k-1]}\frac{\partial U^l[t+k-1]}{\partial U^l[t+k-2]}\cdots\frac{\partial U^l[t+1]}{\partial U^l[t]} \\ &= \prod_{t'=0}^{k-1}(\alpha + (W_2^l - \alpha V_{th})\mathbb{H}(U^l[t+t']))\end{aligned} \tag{7}$$

Considering the extreme case where $\mathbb{H}(U^l[t+t'])$ is always equal to $\frac{1}{V_{th}}$, we have $\frac{\partial U^l[t+k]}{\partial U^l[t]} = (\frac{W_2^l}{V_{th}})^k$. When the value of k is sufficiently large, and $|W_2^l| > V_{th}$, the gradient explosion problem occurs in the temporal dimension. The gradient explosion problem caused by the recurrent structure. This issue can be addressed using recurrent neural networks (RNNs) techniques (Pascanu, 2013; Henaff et al., 2016). To mitigate this, we employ orthogonal initialization (Henaff et al., 2016), a simple and effective approach.

When $|W_2^l| \leq V_{th}$, combining Equations Eq. (6) and Eq. (7), we have:

$$|\frac{\partial U^l[t+k]}{\partial U^l[t]}| \leq \max(\alpha^k, |\frac{W_2^l}{V_{th}}|^k) \leq 1 \tag{8}$$

When k is large enough, the problem of gradient vanishing in the temporal dimension can be avoided only if both inequality signs in Eq. (8) become equalities. To meet this condition, it is essential that $|W_2^l| = V_{th}$ and $\mathbb{H}(U^l[t+t']) = \frac{1}{V_{th}}$ for $0 \leq t' < k$, which is a very stringent requirement. As a result, RSNNs are prone to vanishing gradients in the temporal dimension, limiting their ability to capture long-term dependencies. To mitigate the gradient vanishing problem, improvements can be made from either the LIF neuron or the recurrent structure perspective while maintaining their coordination. This work primarily focuses on optimizing the recurrent structure.

3.3. Skip Recurrent Connection

Skip recurrent connections (SRC) can alleviate the vanishing gradient problem by introducing direct pathways between temporal steps. We propose SRC-SNN by replacing the vanilla recurrent structure in RSNN with SRC. In SRC-SNN, The membrane potential update equation for the l-th layer neuron is given as follows:

$$\begin{aligned}U^l[t] = &\underbrace{\alpha(U^l[t-1] - V_{th}S^l[t-1])}_{\text{self-connections of LIF neurons}} + \underbrace{W_1^l S^{l-1}[t]}_{\text{feedforward connections}} \\ &+ \underbrace{W_2^l S^l[t-\lambda]}_{\text{skip recurrent connections}}\end{aligned} \tag{9}$$

Here, λ represents the skip coefficient, which is typically greater than 1 (Figure.1b). When $\lambda = 1$, SRC-SNN degenerates into vanilla RSNN (Figure.1a). Furthermore, we found that (Zhang et al., 2016) systematically analyzed and validated the effectiveness of SRC in modeling long-term dependencies. It is worth noting that, unlike (Zhang et al., 2016), where adjacent time steps are connected through the vanilla recurrent structure, in SRC-SNN, the connections between adjacent time steps are inherently supported by the intrinsic self-connections of LIF neurons.

The limitations of SRC Individually tuning the hyperparameters of the skip coefficients for each layer in SRC-SNN results in an exponential growth in the number of hyperparameter optimization experiments, making this approach impractical. Consequently, the skip coefficients across the layers in SRC-SNN are set to be identical, which, however, limits the temporal modeling capability of the SRC. Moreover, the optimal setting of the skip coefficients in the SRC-SNN models often differs on different datasets, and in some cases the model performance is highly sensitive to changes in the skip coefficient (see 4.3.1). Therefore, the hyperparameter tuning process in ASRC-SNN is complex.

3.4. Adaptive Skip Recurrent Connection

3.4.1. The Design of ASRC

To address the limitations of SRC, we propose an improved approach termed adaptive skip recurrent connection (ASRC), which can learn the span of skip connection. This approach is inspired by the asymptotic behavior of the Softmax function in the low-temperature regime (Guo

et al., 2017). Specifically, when the Softmax function is parameterized with a temperature $\tau > 0$, its form is given by:

$$\text{Softmax}_\tau(x_i) = \frac{\exp(x_i/\tau)}{\sum_j \exp(x_j/\tau)} \tag{10}$$

where $x = [x_1, x_2, \ldots, x_n]$ denotes the input vector and x_i represents the i-th element. As $\tau \to 0$ (the low-temperature limit), the Softmax function exhibits the following asymptotic behavior:

$$\lim_{\tau \to 0} \text{Softmax}_\tau(x_i) = \begin{cases} 1, & \text{if } i = \arg\max_j x_j \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

In this limit, the Softmax function converges to a Hardmax operation, where the output corresponds to a one-hot encoding of the position of the maximum value in the input vector.

In the ASRC-SNN, the membrane potential update equation for the l-th layer neurons is as follows:

$$\begin{aligned} U^l[t] &= \alpha(U^l[t-1] - V_{th}S^l[t-1]) \\ &+ W_1^l S^{l-1}[t] + W_2^l \sum_{t'=1}^{T_\lambda} p^l[t'] S^l[t-t'] \end{aligned} \tag{12}$$

Here, p^l represents the weights of multiple skip connections with varying temporal spans, which are computed using a Softmax kernel function with a temperature parameter τ. T_λ represents the length of the Softmax kernel, defining the maximum time span considered in the ASRC model, i.e., the longest time span that the skip recurrent connections can extend across. Specifically, p^l regulates the influence of states from the past T_λ time steps, distributing weights for skip connections and determining their contribution to the current neuron state update. The mathematical expression is as follows:

$$\forall t \in \{1, \ldots, T_\lambda\},\ p^l[t] = \frac{\exp(w^l[t]/\tau)}{\sum_{t'=1}^{T_\lambda} \exp(w^l[t']/\tau)} \tag{13}$$

Here, w^l is a vector at the l-th layer containing T_λ trainable parameters, initialized to zero. The temperature parameter τ is a non-trainable constant initialized to 1 and decreases after each epoch according to an exponential decay strategy. In our experiments, the exponential decay factor is set to 0.96. Indeed, during the model testing phase, we use Eq. (11) to compute p^l.

3.4.2. Dynamic Analysis

In the early stages of training, the distribution of the output of the Softmax kernel function exhibits high smoothness, enabling the simultaneous activation of multiple skip recurrent connections with varying temporal spans. During this phase, the model leverages this smooth selection mechanism to thoroughly explore the dependencies between the current state and multiple historical time steps, facilitating comprehensive temporal modeling. As training progresses, the temperature parameter gradually decreases, leading to a sharper distribution of weights in the Softmax kernel. Eventually, as the temperature approaches zero, the Softmax kernel function gradually converges to the Hardmax operation. At this stage, the model independently selects the most relevant time step for the skip connection at each layer, based on the current state of that layer. This hardening process (Figure.1c) allows the model to focus more precisely on critical temporal dependencies.

Enhanced Temporal Modeling Capacity By dynamically adjusting the weights of the skip connections with varying temporal spans, ASRC can more accurately select the relevant time step for the skip connection. This flexible adjustment allows ASRC-SNN to better accommodate the varying temporal dependencies required by different layers, compared to the fixed setting of identical coefficients across layers in SRC-SNN. As a result, ASRC exhibits enhanced temporal modeling capabilities.

Enhanced Robustness ASRC-SNN is capable of adaptively adjusting the skip recurrent connections based on the characteristics of different datasets, optimizing the structure to align with the data. When the length of the Softmax kernel T_λ exceeds a certain value, ASRC-SNN shows reduced sensitivity to variations in T_λ, while its performance approaches the optimal level (see 4.3.2). These characteristics enhance the robustness of ASRC-SNN, enabling it to preserve stability across diverse datasets and under fluctuations in hyperparameters.

4. Experiments

4.1. Experimental Setup

We chose to evaluate our method on various temporal classification benchmarks, including sequential MNIST (S-MNIST), permuted sequential MNIST (PS-MNIST), Google Speech Commands v0.01 (GSC) and Spiking Google Speech Commands (SSC). We use a simple model architecture consisting of three hidden layers. More experimental details can be found in the Appendix.A.

The MNIST dataset consists of 70,000 handwritten grayscale digit images with a resolution of 28×28 pixels, intended for classification tasks. Of these, 60,000 images are used for training, while 10,000 images are used for testing. In the S-MNIST dataset, the MNIST images are transformed into 784×1 vectors, where 784 represents the length of the temporal dimension. Building upon S-MNIST, the PS-MNIST dataset introduces random shuffling of the image sequences before inputting them into the network

Table 1: Classification accuracy on S-MNIST, PS-MNIST, SSC and GSC datasets. The symbol * indicates that the results are derived from (Zhang et al., 2024).

Dataset	Method	Recurrent	Parameters	Accuracy(%)
S-MNIST	PLIF (Fang et al., 2021)	Y	0.15M*	91.79
	GLIF (Yao et al., 2022)	Y	0.15M*	96.64
	ALIF (Yin et al., 2021)	Y	0.15M	98.70
	BRFN (Higuchi et al., 2024)	N	0.068M	99.1
	TC-LIF (Zhang et al., 2024)	Y	0.063M/0.15M	98.79/99.20
	PMSN (Chen et al., 2024)	N	0.066M/0.15M	99.40/99.53
	SRC-SNN (ours)	Y	**0.063M/0.15M**	**99.32/99.38**
	ASRC-SNN (ours)	Y	**0.063M**	**99.57**
PS-MNIST	GLIF (Yao et al., 2022)	Y	0.15M*	90.47
	ALIF (Yin et al., 2021)	Y	0.15M	94.30
	BRFN (Higuchi et al., 2024)	N	0.068M	95.2
	TC-LIF (Zhang et al., 2024)	Y	0.063M/0.15M	92.69/95.36
	PMSN (Chen et al., 2024)	N	0.066M/0.15M	97.16/97.78
	SRC-SNN (ours)	Y	**0.063M/0.15M**	**94.78/96.36**
	ASRC-SNN (ours)	Y	**0.063M/0.15M**	**95.40/96.62**
SSC	TC-LIF (Zhang et al., 2024)	Y	0.11M	61.90
	SNN-CNN (Sadovsky et al., 2023)	N	N/A	72.03
	ALIF (Yin et al., 2021)	Y	N/A	74.20
	SpikGRU (Dampfhoffer et al., 2022)	Y	0.28M	77.00
	RadLIF (Bittar & Garner, 2022)	Y	3.9M	77.40
	DCLS-Delays (2L-1KC) (Hammouamri et al., 2024)	N	0.70M	79.77
	DCLS-Delays (3L-2KC) (Hammouamri et al., 2024)	N	2.5M	80.69
	SRC-SNN (ours)	Y	**0.37M**	**81.83**
	ASRC-SNN (ours)	Y	**0.37M**	**81.91**
GSC	SNN with SFA (Salaj et al., 2021)	Y	4.3M	91.21
	ALIF (Yin et al., 2021)	Y	0.22M	92.10
	TC-LIF (Zhang et al., 2024)	Y	0.19M	94.84
	SRC-SNN (ours)	Y	**0.088M**	**96.18**
	ASRC-SNN (ours)	Y	**0.089M**	**96.29**

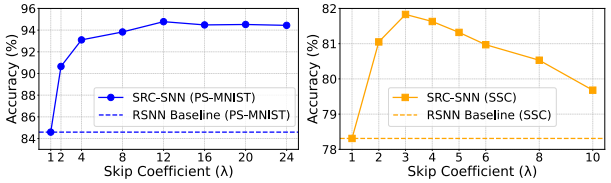
model, thereby creating more complex temporal dependencies compared to S-MNIST.

The SSC is a spike-based speech classification benchmark derived from Google Speech Commands v0.02, which contains 35 classes, proposed in (Cramer et al., 2020). The original waveform data have been converted into spike trains across 700 input channels. The dataset is divided into training, validation, and test splits, consisting of 75,466, 9,981 and 20,382 examples, respectively. The data were further processed with a discrete time scale of 5.6 ms to obtain a sequence length of 250 with zero right-padding. Additionally, the number of input neurons was reduced from 700 to 140 by binning every 5 neurons.

The GSC dataset consists of 64,727 audio files, which are divided into training, validation and test sets, containing 51,093, 6,799 and 3,081 samples, respectively, proposed in (Warden, 2018). Our data preprocessing approach follows the TC-LIF procedure (Zhang et al., 2024) , wherein the audio signals are first transformed into Mel-spectrograms and then converted to decibel units (dB).

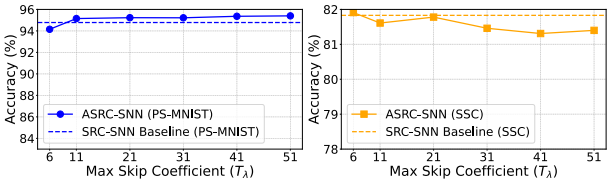
4.2. Resluts

Table.1 compares our two proposed methods, SRC-SNN and ASRC-SNN, with previous works in the field of SNNs on four benchmark datasets (S-MNIST, PS-MNIST, SSC and GSC) in terms of accuracy, model size and whether recurrent connections were used. SRC-SNN outperforms the previous state-of-the-art accuracy on the GSC and SSC benchmark datasets, while significantly reducing the number of parameters. ASRC-SNN further improves upon SRC-SNN, achieving state-of-the-art performance on S-MNIST, SSC and GSC. On PS-MNIST dataset, ASRC-SNN surpasses other approaches that use recurrent structures.



(a) Varying λ on PS-MNIST. (b) Varying λ on SSC.

Figure 2: The impact of varying skip coefficient λ on the performance of SRC-SNN across the PS-MNIST and SSC datasets. (a) and (b) correspond to the results on the PS-MNIST and SSC datasets, respectively.



(a) Varying T_λ on PS-MNIST. (b) Varying T_λ on SSC.

Figure 3: The impact of varying max skip coefficient T_λ on the performance of ASRC-SNN across the PS-MNIST and SSC datasets. (a) and (b) correspond to the results on the PS-MNIST and SSC datasets, respectively.

4.3. Ablation and Analysis

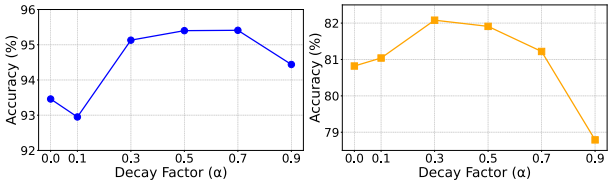
In this section, we conduct controlled experiments to investigate the effectiveness of the SRC and ASRC methods, analyzing the impact of the skip coefficient on SRC-SNN and the maximum skip coefficient on ASRC-SNN using relatively complex SSC and PS-MNIST datasets.

4.3.1. Skip Coefficient on SRC-SNN

The results presented in Figure.2 indicate that when the skip recurrent coefficient λ of SRC-SNN exceeds 1, the model performance improves dramatically. This suggests that SRC-SNN demonstrates a significant improvement in long-term temporal modeling ability compared to the vanilla RSNN. However, it can be observed that the preferred values of λ differ significantly between different datasets, with the preferred values for the SSC dataset being 3, 4 and 5, while those for PS-MNIST being 12, 16, 20 and 24. On the one hand, the optimal values λ vary greatly between datasets; on the other hand, the sensitivity of the results to the setting of λ also differs between datasets, with the SSC dataset being particularly sensitive to λ. These factors complicate the hyparameter search process in SRC-SNN.

4.3.2. Max Skip Coefficient on ASRC-SNN

To validate the robustness of ASRC-SNN, the range and variation of T_λ in the experiments of this part are wider and more extensive compared to the values of λ in 4.3.2. As observed in Figure.3, once T_λ exceeds a certain value, further increases in T_λ result in slight performance fluctuations, with performance approaching the optimal level. This demonstrates the robustness of ASRC-SNN. Furthermore, the optimal performance of ASRC-SNN exceeds the best performance of the SRC-SNN model on both datasets. Notably, on the PS-MNIST dataset with longer sequences, the performance of ASRC-SNN consistently outperforms the best performance of SRC-SNN when T_λ is set to 11 or higher. In summary, ASRC-SNN demonstrates superior temporal modeling capabilities and robustness compared to SRC-SNN.



(a) Varying α on PS-MNIST. (b) Varying α on SSC.

Figure 4: The impact of the membrane potential decay factor α in LIF neurons on the performance of ASRC-SNN across the PS-MNIST and SSC datasets. (a) and (b) correspond to the results on the PS-MNIST and SSC datasets, respectively.

4.4. The Impact of LIF neurons on ASRC-SNN

In this section, we explore the impact of LIF neurons on the performance of ASRC-SNN. As shown in Figure.4, when the membrane potential decay factor $\alpha = 0$, the LIF neuron degenerates into Heaviside function, leading to degraded performance compared to when α is within a reasonable range. This indicates that LIF neurons, in combination with skip recurrent connections, play a collaborative role in temporal modeling. When α is close to 0 or 1, the performance of the model is poor and there is no clear pattern to determine the optimal value of α. $\alpha = 0.5$ is a suitable choice, as it shows good performance on both datasets. Additionally, we conducted experiments where LIF neurons in ASRC-SNN were replaced with PLIF (Fang et al., 2021) or GLIF (Yao et al., 2022) neurons, and the results show no performance improvement with these substitutions.

5. Discussion

In this paper, we conduct a gradient analysis along the temporal dimension by treating recurrent structures and neurons within a unified framework. We identify the issues of vanishing and exploding gradients. To address the challenging problem of vanishing gradients, we introduce skip recurrent connections that directly establish long-range temporal dependencies, replacing vanilla recurrent connections.

Compared to vanilla RSNN, SRC-SNN shows significant performance improvements in temporal benchmark datasets. However, SRC still has some limitations: first, the uniform span of skip connections across layers in the SRC-SNN constrains the network's temporal modeling ability; second, the hyperparameter tuning process is complex. To address these limitations, we introduce ASRC-SNN, a novel mechanism designed to adaptively learn skip spans at each layer of the network. ASRC utilizes a temperature-scaled Softmax kernel to assign weights to skip connections with different temporal spans, promoting competition among them. The intensity of the competition is regulated by the temperature parameter, which is gradually decreased during training to encourage the weights to converge to a single discrete position. Experiments show that ASRC-SNN outperforms SRC-SNN in terms of both temporal modeling ability and robustness. Finally, our experiments demonstrate that LIF neurons and skip recurrent connections in ASRC-SNN work synergistically in the task of temporal modeling.

Finally, we have some thoughts for future research directions:

- The essence of ASRC lies in learning a discrete position along the temporal dimension, with the potential to extend this method to learning a discrete position in both time and space. To learn multiple positions, the ASRC method can be applied repeatedly. A promising application of this approach is the learning of non-zero positions in dilated convolution kernels (Yu, 2015), similar to (Khalfaoui-Hassani et al., 2021).

- We have observed that as the output of the Softmax function approaches Hardmax, the variance of the output becomes larger. Based on this observation, we plan to investigate a new approach for implementing ASRC: building upon the ASRC in this paper, by discarding the temperature parameter in the Softmax kernel function and incorporating the variance of the Softmax kernel output as part of the loss function.

- Inspired by the sharp weight distribution of the Softmax kernel during the intermediate phase of ASRC-SNN training, we will explore the possibility of adaptive multi-skip recurrent connections, considering both parameter-shared and parameter-independent versions.

References

Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., Imam, N., Nakamura, Y., Datta, P., Nam, G.-J., et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE transactions on computer-aided design of integrated circuits and systems*, 34(10):1537–1557, 2015.

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7243–7252, 2017.

Baronig, M., Ferrand, R., Sabathiel, S., and Legenstein, R. Advancing spatio-temporal processing in spiking neural networks through adaptation. *arXiv preprint arXiv:2408.07517*, 2024.

Bittar, A. and Garner, P. N. A surrogate gradient spiking baseline for speech command recognition. *Frontiers in Neuroscience*, 16:865897, 2022.

Chen, X., Wu, J., Ma, C., Yan, Y., Wu, Y., and Tan, K. C. Pmsn: A parallel multi-compartment spiking neuron for multi-scale temporal processing. *arXiv preprint arXiv:2408.14917*, 2024.

Cramer, B., Stradmann, Y., Schemmel, J., and Zenke, F. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757, 2020.

Dampfhoffer, M., Mesquida, T., Valentian, A., and Anghel, L. Investigating current-based and gating approaches for accurate and energy-efficient spiking recurrent neural networks. In *International Conference on Artificial Neural Networks*, pp. 359–370. Springer, 2022.

Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., Dimou, G., Joshi, P., Imam, N., Jain, S., et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99, 2018.

Deng, S., Li, Y., Zhang, S., and Gu, S. Temporal efficient training of spiking neural network via gradient reweighting. *arXiv preprint arXiv:2202.11946*, 2022.

Deng, S., Lin, H., Li, Y., and Gu, S. Surrogate module learning: Reduce the gradient error accumulation in training spiking neural networks. In *International Conference on Machine Learning*, pp. 7645–7657. PMLR, 2023.

Ding, J., Yu, Z., Tian, Y., and Huang, T. Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks. *arXiv preprint arXiv:2105.11654*, 2021.

Elman, J. L. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., and Tian, Y. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2661–2671, 2021.

Fang, W., Yu, Z., Zhou, Z., Chen, D., Chen, Y., Ma, Z., Masquelier, T., and Tian, Y. Parallel spiking neurons with high efficiency and ability to learn long-term dependencies. *Advances in Neural Information Processing Systems*, 36, 2024.

Gerstner, W. and Kistler, W. M. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Hammouamri, I., Khalfaoui-Hassani, I., and Masquelier, T. Learning delays in spiking neural networks using dilated convolutions with learnable spacings. In *The Twelfth International Conference on Learning Representations*, 2024.

Han, B., Srinivasan, G., and Roy, K. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13558–13567, 2020.

Henaff, M., Szlam, A., and LeCun, Y. Recurrent orthogonal networks and long-memory tasks. In *International Conference on Machine Learning*, pp. 2034–2042. PMLR, 2016.

Higuchi, S., Kairat, S., Bohté, S. M., and Otte, S. Balanced resonate-and-fire neurons. *arXiv preprint arXiv:2402.14603*, 2024.

Huang, Y., Lin, X., Ren, H., Fu, H., Zhou, Y., Liu, Z., Pan, B., and Cheng, B. Clif: Complementary leaky integrate-and-fire neuron for spiking neural networks. *arXiv preprint arXiv:2402.04663*, 2024.

Khalfaoui-Hassani, I., Pellegrini, T., and Masquelier, T. Dilated convolution with learnable spacings. *arXiv preprint arXiv:2112.03740*, 2021.

Leñero-Bardallo, J. A., Serrano-Gotarredona, T., and Linares-Barranco, B. A 3.6 μs latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits*, 46(6):1443–1455, 2011.

Li, H., Liu, H., Ji, X., Li, G., and Shi, L. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:309, 2017.

Liu, Z., Datta, G., Li, A., and Beerel, P. A. Lmuformer: Low complexity yet powerful spiking model with legendre memory units. *arXiv preprint arXiv:2402.04882*, 2024.

Ma, G., Yan, R., and Tang, H. Exploiting noise as a resource for computation and learning in spiking neural networks. *Patterns*, 2023. doi: doi.org/10.1016/j.patter.2023.100831.

Neftci, E. O., Mostafa, H., and Zenke, F. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Nunes, J. D., Carvalho, M., Carneiro, D., and Cardoso, J. S. Spiking neural networks: A survey. *IEEE Access*, 10: 60738–60764, 2022.

Pascanu, R. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*, 2013.

Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, S., Wang, G., Zou, Z., Wu, Z., He, W., et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

Roy, K., Jaiswal, A., and Panda, P. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M., and Liu, S.-C. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017.

Sadovsky, E., Jakubec, M., and Jarina, R. Speech command recognition based on convolutional spiking neural networks. In *2023 33rd International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 1–5. IEEE, 2023.

Salaj, D., Subramoney, A., Kraisnikovic, C., Bellec, G., Legenstein, R., and Maass, W. Spike frequency adaptation supports network computations on temporally dispersed information. *Elife*, 10:e65459, 2021.

Shen, S., Wang, C., Huang, R., Zhong, Y., Guo, Q., Lu, Z., Zhang, J., and Leng, L. Spikingssms: Learning long sequences with sparse and parallel spiking state space models. *arXiv preprint arXiv:2408.14909*, 2024.

Stan, M.-I. and Rhodes, O. Learning long sequences in spiking neural networks. *Scientific Reports*, 14(1):21957, 2024.

Wang, L. and Yu, Z. Autaptic synaptic circuit enhances spatio-temporal predictive learning of spiking neural networks. *arXiv preprint arXiv:2406.00405*, 2024.

Wang, Z., Jiang, R., Lian, S., Yan, R., and Tang, H. Adaptive smoothing gradient learning for spiking neural networks. In *International Conference on Machine Learning*, pp. 35798–35816. PMLR, 2023.

Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Yao, M., Gao, H., Zhao, G., Wang, D., Lin, Y., Yang, Z., and Li, G. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10221–10230, 2021.

Yao, M., Hu, J., Zhou, Z., Yuan, L., Tian, Y., Xu, B., and Li, G. Spike-driven transformer. *Advances in neural information processing systems*, 36, 2024.

Yao, X., Li, F., Mo, Z., and Cheng, J. Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks. *Advances in Neural Information Processing Systems*, 35:32160–32171, 2022.

Yin, B., Corradi, F., and Bohté, S. M. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10): 905–913, 2021.

Yu, F. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Zhang, S., Wu, Y., Che, T., Lin, Z., Memisevic, R., Salakhutdinov, R. R., and Bengio, Y. Architectural complexity measures of recurrent neural networks. *Advances in neural information processing systems*, 29, 2016.

Zhang, S., Yang, Q., Ma, C., Wu, J., Li, H., and Tan, K. C. Tc-lif: A two-compartment spiking neuron model for long-term sequential modelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16838–16847, 2024.

Zhou, C., Zhang, H., Zhou, Z., Yu, L., Huang, L., Fan, X., Yuan, L., Ma, Z., Zhou, H., and Tian, Y. QKFormer: Hierarchical spiking transformer using q-k attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Zhou, Z., Zhu, Y., He, C., Wang, Y., YAN, S., Tian, Y., and Yuan, L. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

A. Training configuration

In our experiments, we use simple three-layer fully connected networks combined with the recurrent structure. For the PS-MNIST and S-MNIST datasets, we employed the AdamW optimizer, while for the SSC and GSC datasets, we use the Adam optimizer. Across all datasets, we apply the OneCycle learning rate scheduler, with the learning rate for the softmax kernel set to 100 times the global learning rate. More detailed hyparameter configurations can be found in Table.2. In addition, the hyperparameter configurations corresponding to the best performance of our models on different datasets are provided in Table.3.

Table 2: Hyperparameters used in different tasks.

Dataset	Learning Rate	Softmax kernel learning rate	Weight Decay	Dropout	Batchsize	Epochs
S-MNIST	0.001	0.1	0.01	0	256	200
PS-MNIST	0.001	0.1	0.01	0	256	200
SSC	0.001	0.1	0	0.1	128	100
GSC	0.0025	0.25	0	0.1	128	100

Table 3: Hyperparameters used in best models.

Dataset	Model	Parameter	Hidden Size	λ	T_λ
S-MNIST	SRC-SNN	0.063M	[64, 128, 128]	16	-
	SRC-SNN	0.15M	[64, 212, 212]	12	-
	ASRC-SNN	0.063M	[64, 128, 128]	-	41
PS-MNIST	SRC-SNN	0.063M	[64, 128, 128]	12	-
	SRC-SNN	0.15M	[64, 212, 212]	16	-
	ASRC-SNN	0.063M	[64, 128, 128]	-	51
	ASRC-SNN	0.15M	[64, 212, 212]	-	31
SSC	SRC-SNN	0.37M	[256, 256, 256]	3	-
	ASRC-SNN	0.37M	[256, 256, 256]	-	6
GSC	SRC-SNN	0.088M	[128, 128, 128]	4	-
	ASRC-SNN	0.089M	[128, 128, 128]	-	21

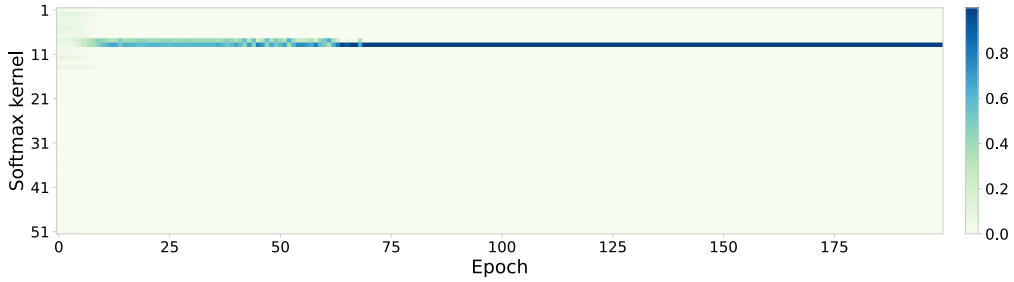
B. Some information related to the experiments

Table.4 presents the final convergence values of the skip connection coefficients across different layers of ASRC-SNN as T_λ increases.

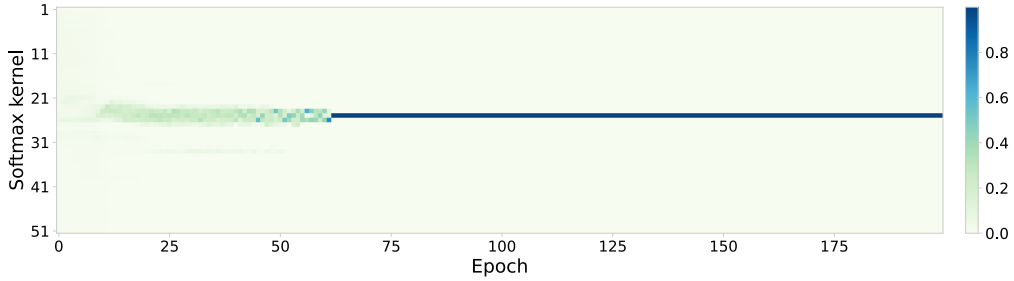
Table 4: The skip coefficients for each layer of ASRC-SNN after training

PS-MNIST		SSC	
T_λ	**Final Skip coefficients**	T_λ	**Final Skip coefficients**
6	[5, 6, 3]	6	[2, 6, 6]
11	[9, 10, 4]	11	[2, 7, 8]
21	[8, 12, 4]	21	[3, 8, 13]
31	[8, 14, 4]	31	[3, 9, 19]
41	[8, 13, 17]	41	[3, 10, 23]
51	[9, 25, 11]	51	[3, 10, 27]

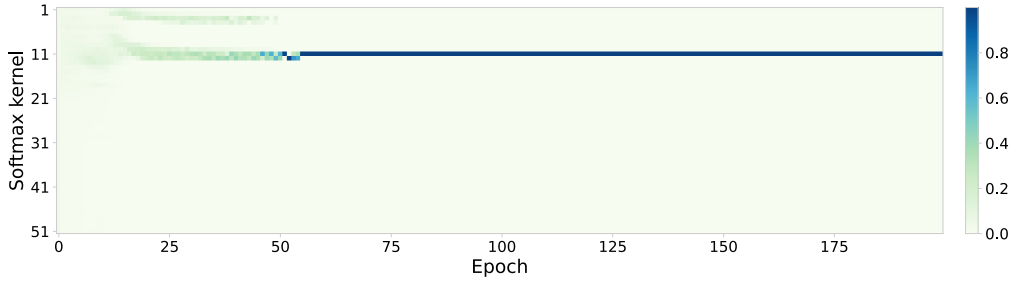
Figure.5 presents heatmaps of the weight variations of the Softmax kernels across different layers during ASRC-SNN training on PS-MNIST. Figure.6 shows the corresponding accuracy change plot.



(a) First layer



(b) Second layer



(c) Third layer

Figure 5: These plots present heatmaps of the weight distributions of the Softmax kernels across different layers during the training of ASRC-SNN. The x-axis represents the epochs, the y-axis represents time, and each kernel has a size of $T_\lambda = 51$. (a), (b), and (c) represent the first, second, and third layers, respectively.

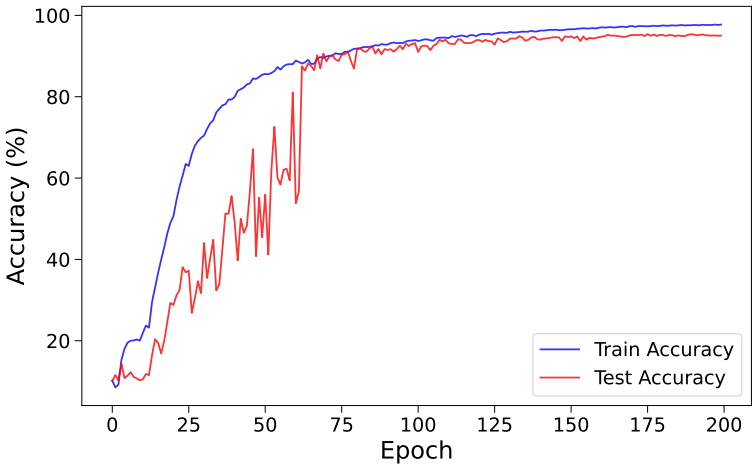


Figure 6: The accuracy change plot during the training of ASRC-SNN.