SPATIO-TEMPORAL MARGIN REGULARIZATION FOR MITIGATING TEMPORAL ERROR ACCUMULATION IN SPIKING NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) exploit spatio-temporal spike dynamics for low-power computation and temporal modeling, yet early misfired spikes trigger cumulative temporal errors that amplify across iterative dynamics and degrade generalization. To address this issue, we propose the dual Spatio-Temporal Margin Regularization (STMR) framework, which mitigates the temporal error accumulation from both the perspectives of spike dynamics and output representation. Temporal margin regularization quantifies class-specific spike latency via weight mapping, encouraging target-correlated neurons to fire earlier than negative-correlated ones to suppress error generation at its source. Spatial margin regularization further constrains the per-timestep confidence margin between the target and the non-target risk class, with timestep-coordinated adaptive weights that focus on highly erroneous timesteps to balance regularization strength across the temporal dimension. Extensive experiments on 1D, 2D, and 3D benchmarks with spiking VGG, ResNet, and Transformer architectures demonstrate consistent performance gains. Notably, STMR achieves 87.1% accuracy on CIFAR10-DVS, and visual analysis confirms that it effectively advances target-class spike activation and narrows negative confidence margins. This work provides new insights into spike dynamics-aware optimization for SNNs.

1 INTRODUCTION

Spiking neural networks (SNNs) have attracted widespread attention due to their sparse binary spiking communication mechanism and temporal dynamics, and are extensively applied in various low-power computing and spatio-temporal tasks Roy et al. (2019); Yao et al. (2024); Shan et al. (2026). However, despite their efficiency and temporal advantages, SNNs typically underperform traditional artificial neural networks (ANNs). This is attributed to the limited information representation resulting from their binary spike sequences Guo et al. (2024). Consequently, previous work has focused on improving the performance of SNNs from various perspectives, such as designing spike-driven architectures Yao et al. (2025); Zhang et al. (2026), enhancing spiking neuron models Ding et al. (2025); Ma et al. (2026), and refining training strategies Yan et al. (2025a); Kang et al. (2026). However, few studies have delved deeply into the dynamical roots of SNNs to investigate whether other factors limit their performance.

This paper delves into the dynamics of spiking neurons and uncovers the potential problem of *temporal error accumulation*. Spiking neurons must iterate across multiple timesteps to stabilize objective representations. A well-functioning SNN activates target-class positively correlated neurons continuously Ding et al. (2026), generating dense positive spikes at early timesteps. It suppresses the firing of target-class negatively neurons through lateral inhibition Blakemore et al. (1970), ultimately achieving accurate decision-making. However, the unstable state in early timesteps is prone to misfiring negatively correlated spikes, which in turn negatively interfere with the critical target semantics Ding et al. (2026). If these errors are not adequately corrected in subsequent timesteps, they will continue to accumulate and degrade the SNN's overall generalization ability. Additionally, errors in the hidden layers will propagate and be amplified by iterative dynamics, ultimately manifesting in the output confidence distribution and resulting in negative confidence margins. As shown in Figure 1, high-quality SNN prediction behavior prioritizes the firing of target-related neurons.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

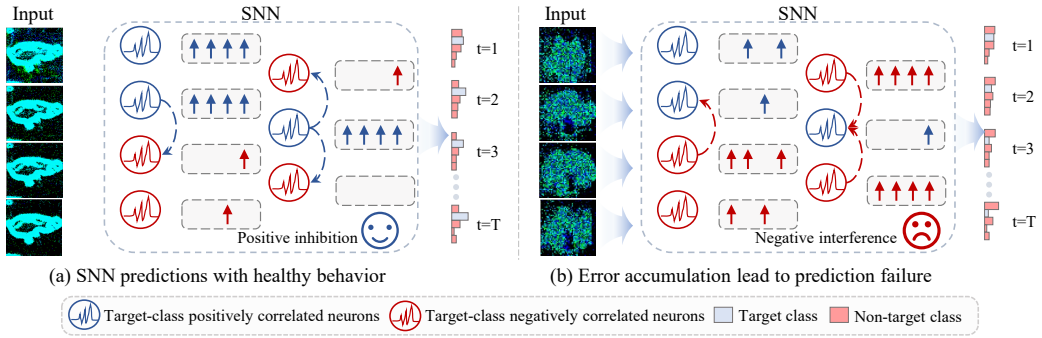

Figure 1: (a) Target-class positively correlated spikes are generated first, suppressing the firing of negative spikes; (b) Target-class negatively correlated spikes are generated first, interfering with the firing of positive spikes and causing prediction failures due to the temporal error accumulation.

However, misfired negatively correlated spikes introduce negative interference, resulting in significantly higher confidence in non-target classes than in the target class, thereby generating a negligible negative margin gap. Furthermore, the accumulation of errors causes the margin gap to increase progressively with each timestep, resulting in irreversible misdecisions. Since spiking neurons are the core building blocks of SNNs, generalization degradation caused by temporal error accumulation is a pervasive issue across various SNN architectures and tasks that cannot be mitigated by adapting existing broad-brush methods. Therefore, there is an urgent need for targeted solutions that can mitigate temporal error accumulation and improve generalization, thus unleashing the performance potential of SNNs.

To mitigate the impact of temporal error accumulation on the generalization ability of SNNs, we propose Spatio-Temporal Margin Regularization (STMR), which facilitates a positive margin between target and non-target classes during training. To prevent negative interference from misfired spikes in early timesteps, temporal margin regularization promotes the preferential firing of target-class positively correlated neurons, which positively inhibits negatively correlated spikes. However, it is impractical to distinguish precisely between positive and negative correlations in neurons since the final prediction depends on the nonlinear coupling of all neurons. Additionally, the discrete timing of spikes hinders the propagation of gradients in the regularization loss. To this end, we utilize weight mapping to quantify per-class contributions of spiking neurons, and design a differentiable class-specific spike latency metric to facilitate smooth gradient propagation. In this way, temporal margin regularization prevents misfiring spikes from the source of spike dynamics.

Furthermore, misfired spikes in the hidden layer propagate and amplify continuously. Ultimately, this manifests in the output confidence distribution, directly affecting the SNN's final predictions. Therefore, we propose spatial margin regularization to explicitly constrain the negative confidence margin between the target class and the non-target highest-risk class. Considering the differences between states at various timesteps in the SNN and the accumulation of errors over time, we designed timestep-coordinated weights, which focus on timesteps with significant errors and adaptively balance regularization strength across multiple timesteps. Therefore, STMR mitigates the impact of errors from the perspectives of neuronal spike dynamics and output representations, thereby enhancing SNNs' overall performance. Extensive experiments with spiking VGG, ResNet, and Transformer architectures on 1D, 2D, and 3D tasks and neuromorphic and static datasets demonstrate that STMR consistently delivers performance gains. Our contribution can be summarized as follows:

- We revealed that misfired spikes in SNNs during early timesteps trigger errors that accumulate over timestep, thereby impairing the generalization ability. Ultimately, these errors propagate and amplify, resulting in a negative confidence margin in the final prediction.
- We propose STMR, which promotes the preferential firing of target-class positively correlated spikes and suppresses the negative confidence margin from the perspectives of both neuronal spike dynamics and predictive output representation, thereby mitigating the influence of temporal error accumulation.
- Extensive experiments across multiple datasets and architectures confirm the effectiveness and versatility of STMR, demonstrating its ability to promote target-class spike firing and narrow the negative margin.

2 Related Work

2.1 Spiking Neural Network

SNNs transmit information between spiking neurons via biomimetic binary spikes, offering inherent hardware power-efficiency advantages over ANNs Roy et al. (2019); Yao et al. (2024). Furthermore, spiking neurons update their membrane potential dynamics iteratively over multiple timesteps. This enables SNNs to capture underlying temporal patterns Ponghiran & Roy (2022); Ding et al. (2024). Therefore, SNNs have been widely applied to various tasks, such as recognition Yao et al. (2025); Qiu et al. (2026), detection Luo et al. (2025); Mao et al. (2025), and decoding Mentzelopoulos et al. (2025); Yang et al. (2026), and are expected to enable lightweight, brain-inspired intelligence. However, discrete spike sequences have a lower information capacity than continuous-value activation coding. This makes SNNs' performance generally inferior to that of ANNs. To this end, previous studies have proposed spike-driven architectures Yao et al. (2025); Zhang et al. (2026), improved spiking neurons Ding et al. (2025); Ma et al. (2026), and optimized training algorithms Yan et al. (2025a); Kang et al. (2026) to improve the performance of SNNs.

Building upon the temporal dynamics of spiking neurons, this paper identifies another overlooked concern beyond binary information capacity: the temporal error accumulation can also lead to generalization degradation, thereby limiting the performance of SNNs. To address this issue, we propose STMR as a general-purpose regularization method that mitigates the influence of temporal error accumulation and unleashes the full potential of SNNs.

2.2 Generalization Optimization in SNNs

Existing methods have been developed to improve the generalization ability of SNNs by preventing overfitting through data augmentation or by applying regularization loss during training. For example, NDA Li et al. (2022) performs spatial transformations on neuromorphic data, such as flipping and rotating, while EventDrop Gu et al. (2021) randomly drops events from the event stream. However, data augmentation only introduces variations to the data, rather than optimizing it specifically for the properties of SNNs and resulting in limited benefits. In terms of regularization training, Yu et al. (2025) applies entropy regularization during SNN distillation, Guo et al. (2023) incorporates membrane potential regularization to prevent excessive polarization, and TMC Yan et al. (2025b) calibrates the SNN by suppressing confidence scores through regularization. Different from existing regularization methods that either focus on membrane potential smoothing or output confidence calibration, STMR specifically targets the temporal error accumulation nature of SNNs and proposes STMR for dual margin regularization from both spike timing and output confidence perspectives.

3 Background and Motivation

3.1 Dynamics of Spiking Neurons

Spiking neurons simulate the continuous state-updating process of biological neurons by generating binary spiking signals through the iterative evolution of spiking dynamics over T timesteps. A spiking neuron maintains a membrane potential state H at timestep $t = \{1, 2, \cdots, T\}$. This state is updated when the neuron receives an input current I from postsynaptic neurons. Taking the commonly used leaky integrate-and-fire (LIF) neuron model Wu et al. (2018) as an example, its membrane potential update equation is:

$$H^l_{i,t} = (1 - \frac{1}{\tau})H^l_{i,t-1} + I^l_{i,t}, \tag{1}$$

where l and i represent the layer and neuron indices, respectively, and τ represents the membrane potential time constant, which controls the continuous leakage of the membrane potential to the resting state when no input is received ($I = 0$). When the membrane potential reaches the firing threshold ϑ, the spike neuron generates a spike signal ($S = 1$) and resets:

$$S^l_{i,t} = \begin{cases} 1, & H^l_{i,t} \geq \vartheta \\ 0, & H^l_{i,t} < \vartheta \end{cases}, \tag{2}$$

$$H^l_{i,t} = H^l_{i,t} - S^l_{i,t}\vartheta. \tag{3}$$

Then, the spiking neurons iterate to timestep $t+1$ and replicate the above process. Accordingly, the SNN generates a total of T timestep predictions $\{o_1, o_2, \cdots, o_T\}$ and typically averages them to produce its final output $O_T = \frac{1}{T}\sum_{t=1}^{T} o_t$.

3.2 Temporal Error Accumulation

Although the iterative membrane potential states in spiking neural dynamics enable SNNs to capture temporal patterns, this time-dependence property also introduces potential risks of temporal error accumulation.

For each timestep t, the membrane potential $H_{i,t}^l$ and spike output $S_{i,t}^l$ are directly determined by the state $H_{i,t-1}^l$ inherited from the previous timestep. Therefore, any inaccuracy in spike firing at timestep $t-1$ propagates forward, altering the membrane potential at timestep t and affecting subsequent spike events. This recursive dependency means that small errors in spike firing at early timesteps do not disappear. Rather, they accumulate and amplify throughout temporal evolution. Furthermore, previous studies have shown that SNNs are generally unstable during the initial timesteps Ding et al. (2025), making early errors inevitable. Moreover, it has been shown that lateral inhibition Blakemore et al. (1970); She et al. (2021); Liu et al. (2026a) occurs between neurons, where the firing of one neuron inhibits the firing of other neurons. This exacerbates the effects of misfired spikes, especially when target-class negatively correlated neurons inappropriately inhibit positively correlated spikes, ultimately leading to prediction failure.

Beyond the level of individual spiking neurons, spike errors in the hidden layers continue to propagate and amplify through iterative dynamics. These errors ultimately manifest in the output confidence distribution, causing confidence in non-target classes to outweigh confidence in the target class. Additionally, the negative confidence margins vary across timesteps due to the cumulative nature of temporal errors. However, standard training methods only optimize the final time-averaged output O_T without explicitly constraining these two levels. This motivates us to propose STMR, which regularizes both the firing times of spiking neurons and the negative confidence margin. This reduces sources of error and explicitly limits the impact of errors on the manifested results, thereby fully mitigating temporal error accumulation.

4 Spatio-Temporal Margin Regularization

To mitigate the negative impacts of temporal error accumulation in SNNs, STMR first prioritizes the firing of target-class positively correlated spikes in the temporal dimension. This prevents errors caused by misfired spikes in the early timesteps from the outset, thereby reducing the risk of cumulative error over time. Furthermore, since degradation in generalization is reflected by differences in inter-class confidence in the SNN's final output, STMA promotes the target class's confidence advantage through confidence spatial margin regularization and adjusts the regularization strength with timestep-coordinated weights, thereby improving overall performance. Figure 2 illustrates the overview of the STMR framework.

4.1 Temporal Margin Regularization

The error essentially arises from the firing of target-class negatively correlated neurons before target-class positively correlated neurons, resulting in negative inhibition and generalization degradation. Therefore, directly regulating the firing timing of spiking neuron can prevent erroneous spikes at their source. Based on this insight, temporal margin regularization aims to encourage target-class positively correlated neurons to fire preferentially, thereby forming a positive temporal margin of spikes.

However, it is impractical to determine whether a spiking neuron in the hidden layer belongs to the target class, because the final prediction depends on the nonlinear coupling of all neurons. This makes it challenging to determine whether the firing of a spiking neuron is positively or negatively correlated with the target class. To address this challenge, rather than rigidly assigning class labels to each neuron, we perform soft contribution allocation based on classifier weights and calculate class relevance by evaluating the final contribution of each neuron to each class.

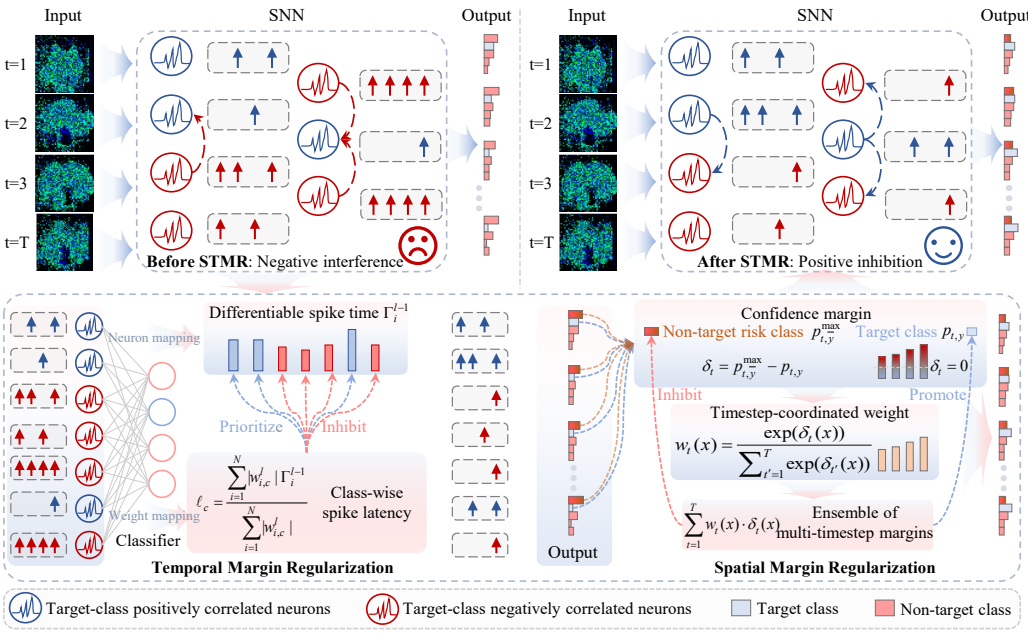

Figure 2: Overview of Spatio-Temporal Margin Regularization (STMR). Temporal margin regularization calculates category-specific, differentiable spike latencies via weight mapping and prioritizes the firing of target-class positively correlated neurons, thereby preventing misfiring at the source of neural dynamics. Spatial margin regularization suppresses negative confidence margins resulting from error propagation, and timestep-coordinated weights balance regularization strength across timesteps to further enhance generalization ability.

Let $y \in \{1, 2, \cdots, C\}$ denote the class label corresponding to sample x, where C is the total number of classes. In an l-layer SNN, the final layer serves as the classifier and the firing times of the $l-1$th layer of spiking neurons are regularized. Since the classifier weight $w_{i,c}^l$ represents the contribution of neuron i to class c, we map the absolute values of the classifier weights $|w_{i,c}^l|$ to obtain the class-specific spike latency metric ℓ_c:

$$\ell_c = \frac{\sum_{i=1}^{N} |w_{i,c}^l| \Gamma_i^{l-1}}{\sum_{i=1}^{N} |w_{i,c}^l|}, \tag{4}$$

where N represents the number of spiking neurons and Γ_i^{l-1} represents the firing time of neuron i. Eq. (4) quantifies the spike timing for each class, transforming the temporal margin regularization objective into the minimization of the spike latency ℓ_y for the target class y and thereby encouraging the priority firing of spikes for that class. However, the exact spike times are discrete, and their non-differentiable properties hinder the employment of the spike latency metric ℓ_c in backpropagation. To address this issue, we use the average spike firing time as a differentiable proxy equivalent, expressed as:

$$\Gamma_i^{l-1} = \frac{\sum_{t=1}^{T} t\, S_{i,t}^{l-1}}{\sum_{t=1}^{T} S_{i,t}^{l-1}}. \tag{5}$$

Furthermore, we consider the non-target class with the smallest spike latency $\ell_{\bar{y}}^{\min} = \min_{c \neq y} \ell_c$ to be the source of errors with the highest risk, as it is most likely to trigger negative spikes before the target class. Therefore, temporal margin regularization enables the spike latency for the target class to be lower than that for the non-target class with the highest risk:

$$\mathcal{L}_{\text{TMR}} = \frac{1}{|\mathcal{M}|} \sum_{x \in \mathcal{M}} \ell_y(x) - \ell_{\bar{y}}^{\min}(x), \tag{6}$$

where $\mathcal{M}$ is the set of misclassified samples in a batch and $|\cdot|$ is the cardinality of the set. This prompts the target class positively correlated spiking neurons to fire earlier and suppresses the negatively correlated spikes. Consequently, misfired spikes in early timesteps are reduced, and error accumulation at its source is alleviated.

4.2 SPATIAL MARGIN REGULARIZATION

Although temporal margin regularization $\mathcal{L}_{\text{TMR}}$ alleviates error accumulation at the source by regulating relative spike timing across classes, it cannot completely eliminate temporal errors. Residual spike errors continue to propagate and amplify through iterative membrane potential updates, and eventually manifest themselves in the per-timestep output confidence distribution. Figure 4 illustrates the evolution of average confidence in misclassified samples for VGGSNN on CIFAR10-DVS. For the vanilla SNN, confidence in the incorrect class is significantly higher than in the target class, and this negative confidence margin continues to increase over timesteps. This suggests that the evolution of anomalous confidence must be explicitly constrained.

Spatial margin regularization explicitly constrains the confidence margin between the target class and non-target classes on a per-timestep basis, directly acting on the SNN's final prediction to ensure overall performance. At timestep t, the target class confidence $p_{t,y} = p_t[y]$ is first obtained. Then, the non-target class with the highest confidence is selected as the class posing the greatest risk. This non-target risk class is suppressed via margin regularization to prevent an exaggerated gap from forming between it and the target class:

$$p_{t,\bar{y}}^{\max} = \max_{c \neq y} p_t[c], \qquad \delta_t = p_{t,\bar{y}}^{\max} - p_{t,y}, \tag{7}$$

where $p_{t,\bar{y}}^{\max}$ represents the confidence that the non-target risk class will be suppressed and δ_t represents the risk confidence margin. Going forward, the spatial margin regularization loss across T timesteps can be defined as an ensemble of risk confidence margins at each timestep:

$$\ell_{\text{SMR}}(x) = \sum_{t=1}^{T} w_t(x) \cdot \delta_t(x), \quad \mathcal{L}_{\text{SMR}} = \frac{1}{|\mathcal{M}|} \sum_{x \in \mathcal{M}} \ell_{\text{SMR}}(x), \tag{8}$$

where $w_t(x)$ denotes the regularization weight of the risk confidence margin at timestep t. Due to the time-varying nature of SNNs and the cumulative nature of errors, the severity of the negative confidence margin fluctuates with each timestep. For example, early timesteps tend to have fewer errors due to limited action potential accumulation, whereas later timesteps are subject to more severe cumulative errors. Employing a uniform regularization strength across all timesteps would result in over-regularization at timesteps with low confidence margins, leading to overfitting. Alternatively, it would result in under-regularization at timesteps with high confidence margins, failing to effectively suppress cumulative error. To address this issue, we designed timestep-coordinated weights that adaptively allocate regularization strength based on the confidence margin at each timestep. Timesteps with larger negative confidence margins receive higher weights, which prioritizes the suppression of severe cumulative errors. Meanwhile, timesteps with smaller margins receive lower weights, which prevents excessive constraints on well-learned representations:

$$w_t(x) = \frac{\exp(\delta_t(x))}{\sum_{t'=1}^{T} \exp(\delta_{t'}(x))}. \tag{9}$$

In this way, spatial margin regularization balances the unique margin requirements of the SNN for each sample at every timestep, paying greater attention to moments of underperformance to sufficiently boost the "weakest link in the bucket." Meanwhile, slight regularization weights are applied when the risk confidence margin is small to prevent overfitting.

4.3 STMR TRAINING OBJECTIVES

In vanilla SNN training, the cross-entropy loss $\mathcal{L}_{\text{CE}}$ is computed for the final output O_T. STMR additionally computes temporal margin regularization loss $\mathcal{L}_{\text{TMR}}$ and spatial margin regularization loss $\mathcal{L}_{\text{SMR}}$, controlling the regularization strength with separate balancing coefficients λ_{TMR} and λ_{SMR}:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{TMR}} \mathcal{L}_{\text{TMR}} + \lambda_{\text{SMR}}\, \mathcal{L}_{\text{SMR}}. \tag{10}$$

The two regularization terms in STMR serve complementary functions in addressing temporal error accumulation at separate levels. Temporal margin regularization $\mathcal{L}_{\text{TMR}}$ suppresses error generation at the spike source, and spatial margin regularization $\mathcal{L}_{\text{SMR}}$ limits error manifestation at the final output representation. Notably, STMR regularization is applied only to misclassified samples in each mini-batch. From a bias-variance perspective, correctly classified samples have satisfactory

timing and confidence margins. Applying regularization to these samples would introduce unnecessary inductive bias and hinder generalization. By only correcting misclassified samples, STMR repairs error-prone representations without disrupting well-learned patterns. Appendix B provides theoretical analyses from the perspectives of output margin and temporal margin that further demonstrate STMR's ability to enhance the generalization capacity of SNNs. Algorithm 1 illustrates the complete framework for training an SNN with STMR regularization.

5 Experiments

To verify the generalizability and versatility of the proposed STMR, we conducted experiments on the 1D neuromorphic speech dataset SHD Cramer et al. (2022), the 2D neuromorphic vision datasets CIFAR10-DVS Li et al. (2017), DVS-Gesture Amir et al. (2017), and N-Caltech101 Orchard et al. (2015), the 2D static datasets CIFAR10 and CIFAR100, and the 3D point cloud dataset ModelNet40 Wu et al. (2015). These experiments cover timesteps ranging from low ($T = 4$) to high ($T = 100$) to verify the generalizability of STMR across different time scales. Please refer to the **Appendix** D for detailed settings.

5.1 Ablation Study

To validate the effectiveness and generalizability of STMR, we conducted ablation studies with the spiking VGG-9, ResNet-18, and QKFormer architectures on CIFAR10-DVS and DVS-Gesture. In the ablation studies, the timestep $T = 4$. As shown in Table 1, STMR achieves consistent performance gains, improving the accuracy of DVS-Gesture by up to 4.52%. This suggests that temporal error accumulation is a widespread problem across architectures and datasets, and that STMR can be used as a versatile mitigation strategy.

Table 1: Ablation study results across architectures and datasets (%).

Dataset	Method	VGG-9	ResNet-18	QKFormer
CIFAR10-DVS	Baseline	72.9	66.1	81.2
	+$\mathcal{L}_{\text{TMR}}$	$75.1_{+2.2}$	$67.0_{+0.9}$	$82.3_{+1.1}$
	+$\mathcal{L}_{\text{SMR}}$	$74.8_{+1.9}$	$67.2_{+1.1}$	$82.2_{+1.0}$
	+Both	$\mathbf{75.9}_{+3.0}$	$\mathbf{68.1}_{+2.0}$	$\mathbf{82.8}_{+1.6}$
DVS-Gesture	Baseline	87.15	81.59	93.75
	+$\mathcal{L}_{\text{TMR}}$	$89.93_{+2.78}$	$83.33_{+1.74}$	$95.49_{+1.74}$
	+$\mathcal{L}_{\text{SMR}}$	$89.58_{+2.43}$	$82.64_{+1.05}$	$95.14_{+1.39}$
	+Both	$\mathbf{91.67}_{+4.52}$	$\mathbf{84.03}_{+2.44}$	$\mathbf{95.83}_{+2.08}$

Table 2 further evaluates STMR's compatibility by integrating it with other methods on the DVS-Gesture dataset. We combined STMR with the optimized spiking neuron models, PLIF Fang et al. (2021) and CLIF Huang et al. (2024); the SNN-oriented BN layer TAB Jiang et al. (2024); and the SLT Anumasa et al. (2024) training strategy. The results show that STMR improves the performance of these methods, further demonstrating its compatibility.

Table 2: Compatibility experiments of STMR on DVS-Gesture (%).

Type	Method	Acc. (%)	+ STMR
Spiking neuron	PLIF Fang et al. (2021)	87.85	$\mathbf{90.28}_{+2.43}$
Spiking neuron	CLIF Huang et al. (2024)	89.58	$\mathbf{91.32}_{+1.74}$
BN Layer	TAB Jiang et al. (2024)	87.50	$\mathbf{89.93}_{+2.43}$
Training	SLT Anumasa et al. (2024)	88.19	$\mathbf{88.89}_{+0.70}$

Additionally, we investigated how the regularization coefficients λ_{TMR} and λ_{SMR} influence performance with the VGG-9 and QKFormer architectures on DVS-Gesture. As shown in Figure 5, varying λ_{TMR} and λ_{SMR} within the range $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$ leads to slight fluctuations in performance. However, STMR consistently yields performance gains. This demonstrates that STMR is relatively robust across different hyperparameters.

5.2 Comparison with Other Methods

2D neuromorphic and static object recognition. As shown in Table 3, STMR with VGGSNN architecture achieved an accuracy of 87.1% on CIFAR10-DVS, outperforming other methods. On DVS-Gesture, STMR reached an accuracy of 95.83% with only 4 timesteps. Similar results were observed on N-Caltech101, where STMR achieved a precision of 94.48%, demonstrating its generalizability across architectures and datasets. Additionally, Table 8 shows the comparative results for the static CIFAR-10 and CIFAR-100 datasets, for which STMR achieved accuracies of 96.61% and 81.44%, respectively.

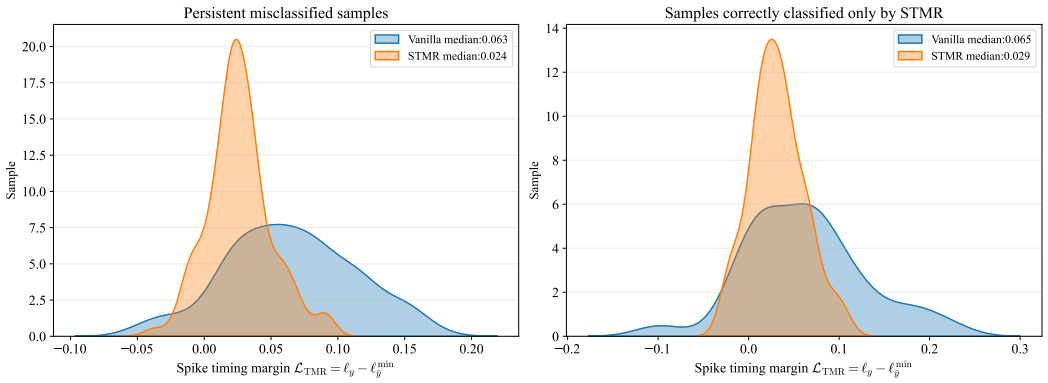

Figure 3: Spike time margin distributions of VGGSNN on CIFAR10-DVS. STMR significantly reduces the spike time margin, which mitigates the influence of misfired target-class negatively correlated spikes.

Table 3: Comparative results on 2D neuromorphic datasets. *: Open-source implementation.

Dataset	Method	Architecture	$T \downarrow$	Accuracy ↑ (%)
CIFAR10-DVS	PIT^{ICLR} Yan et al. (2026)	VGG11	10	80.5
	DeepTAGE^{ICLR} Liu et al. (2025)	VGG-11	10	81.23
	SLT^{AAAI} Anumasa et al. (2024)	VGGSNN	10	81.46
	Stable Spike^{CVPR} Ding et al. (2026)	VGGSNN	10	83.7
	DS-ATGO^{AAAI} Jiang et al. (2026)	VGGSNN	10	83.7
	QKFormer^{NeurIPS} Zhou et al. (2024a)	QKFormer	16	84.0
	AdaS^{ICML} Zhou et al. (2026)	QKFormer	16	85.1
	STMR (Ours)	QKFormer	16	**85.3**
		VGGSNN	10	**87.1**
DVS-Gesture	QKFormer^{NeurIPS} Zhou et al. (2024a)	QKFormer	16	98.60
	SpikingResformer^{CVPR} Shi et al. (2024)	SpikingResformer	16	98.60
	SEMM^{NeurIPS} Zhou et al. (2024b)	Spikingformer	10	96.88
	MPS^{ICLR} Ding et al. (2025)	SpikingResformer	5	94.44
	QKFormer*^{NeurIPS} Zhou et al. (2024a)	QKFormer	4	93.75
	Stable Spike^{CVPR} Ding et al. (2026)	QKFormer	4	95.49
	STMR (Ours)	QKFormer	4	**95.83**
			16	**98.61**
N-Caltech101	TKS$^{IEEE\ TAI}$ Dong et al. (2024)	VGGSNN	10	84.10
	IMP^{NeurIPS} Shen et al. (2024)	VGGSNN	10	85.01
	TMC^{ICML} Yan et al. (2025b)	VGGSNN	10	86.03
	Knowledge-Transfer*^{AAAI} He et al. (2024)	VGGSNN	10	93.18
	MPS^{ICLR} Ding et al. (2025)	VGGSNN	10	93.68
	Stable Spike^{CVPR} Ding et al. (2026)	VGGSNN	10	94.25
	STMR (Ours)	VGGSNN	10	**94.48**

Table 4: Comparative results on 1D neuromorphic dataset SHD. *: Open-source implementation.

Method	Parameter (M) ↓	$T \downarrow$	Accuracy ↑ (%)
DH-SFNN^{NC} Zheng et al. (2024)	0.05	1000	92.10
DCLS-Delays^{ICLR} Hammouamri et al. (2024)	0.20	100	95.07
SpikeSCR^{NN} Wang et al. (2025a)	0.26	100	95.60
SE-adLIF^{NC} Baronig et al. (2025)	0.45	250	95.81
Event-SSM^{ICONS} Schöne et al. (2024)	0.40	-	95.90
PMSN$^{IEEE\ TNNLS}$ Chen et al. (2026)	0.20	250	95.10
SpikCommander*^{AAAI} Wang et al. (2026)	0.19	100	95.06
STMR (Ours)	0.19	100	**96.27**

1D and 3D tasks. Table 4 and Table 5 present the results on the 1D speech dataset SHD and the 3D point cloud dataset ModelNet40, where STMR employs the SpikCommander Wang et al. (2026) and SDT Lu et al. (2026) architectures, respectively. STMR achieved an accuracy of 96.27% on SHD, outperforming the original model by 1.21% without any additional computational cost. On ModelNet40, STMR achieved an overall accuracy (OA) of 92.47% and a mean accuracy (mAcc) of 89.99%, also demonstrating competitive performance. These competitive experimental results across 1D, 2D, and 3D tasks demonstrate the generalizability and versatility of STMR.

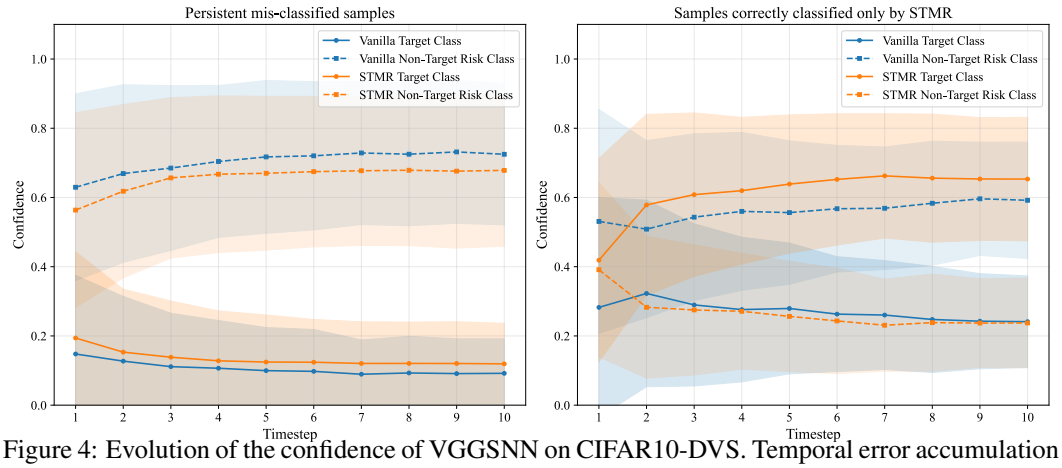

Figure 4: Evolution of the confidence of VGGSNN on CIFAR10-DVS. Temporal error accumulation causes the vanilla SNN to exhibit a significant negative confidence margin between the target class and the non-target risk class, which continues to widen with each timestep. STMR regularizes the spike time and spatial confidence margins, which mitigates error accumulation and converts incorrect predictions into correct ones.

Table 5: Comparative results on the 3D ModelNet40 dataset. $T \times D$ denotes T timesteps, each with a maximum spike value of D. *: Implemented based on open-source code.

Method	Parameter (M) ↓	$T \times D$ ↓	OA(%) ↑	mAcc ↑(%)
Spiking PointNet Ren et al. (2023)$^{NeurIPS'23}$	3.50	4×1	88.61	84.20
P2SResLNet Wu et al. (2024)$^{AAAI'24}$	15.20	4×1	90.60	89.20
SPT Wu et al. (2025)$^{AAAI'25}$	9.64	4×1	91.43	89.39
E-3DSNN Qiu et al. (2025)$^{AAAI'25}$	3.27	1×4	91.70	88.40
SDT Lu et al. (2026)$^{ICLR'26}$	2.25	4×1	92.46	89.48
SDT* Lu et al. (2026)$^{ICLR'26}$	2.25	4×1	92.30	89.61
STMR (Ours)	2.25	4×1	**92.47**	**89.99**

5.3 Class-Specific Spike Latency Analysis

Figure 3 shows the temporal margins of the vanilla SNN and STMR for misclassified samples predicted by the vanilla VGGSNN on CIFAR10-DVS. The results show that STMR significantly reduced the spike time margin. The median for persistently misclassified samples decreased from 0.063 to 0.024. Meanwhile, the median for samples corrected by STMR decreased from 0.065 to 0.029. This reduces the number of misfired target-class negatively correlated spikes at the source of the spike dynamics, thereby preventing them from interfering with the target's semantics and mitigating the temporal cumulative effect of errors.

5.4 Analysis of Confidence Evolution Across Timesteps

Figure 4 illustrates the evolution of confidence for the target and non-target classes in the original misclassified samples from CIFAR10-DVS. The vanilla SNN has significantly less confidence in the target class than in the non-target risk classes, especially for difficult samples that the STMR also misclassifies. Additionally, the negative confidence margin gradually increases with each timestep, suggesting that cumulative errors are becoming more severe over time. By regularizing the spike time and spatial confidence margins, STMR significantly mitigates the negative confidence margins and can even convert moderately difficult samples from incorrect to correct classifications.

6 Conclusion

This paper reveals the problem of temporal error accumulation in SNNs and proposes STMR dual regularization to mitigate it from the perspectives of both neuronal spike dynamics and output distribution. On the one hand, STMR promotes the firing of target-related spikes from the perspective of spiking neuron dynamics while suppressing misfired spikes at their source. On the other hand, STMR explicitly constrains the error-affected output confidence distribution and balances the regularization strength through timestep-coordinated weights. Extensive experiments and analyses of 1D, 2D, and 3D tasks across spiking VGG, ResNet, and Transformer architectures demonstrate the effectiveness and versatility of the proposed method. This work offers a new perspective on analyzing the suboptimal performance of SNNs and provides new insights into unleashing their performance potential in a plug-and-play manner.

AI USE STATEMENT

For this study, we used generative AI tools only to polish some of the sentences. Then, we conducted a detailed review to ensure their correctness. This paper's core idea does not employ generative AI tools, nor was its core content generated using such tools. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Arnon Amir et al. A low power, fully event-based gesture recognition system. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7388–7397, 2017.

Srinivas Anumasa, Bhaskar Mukhoty, Velibor Bojkovic, Giulia De Masi, Huan Xiong, and Bin Gu. Enhancing training of spiking neural network with stochastic latency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10900–10908, 2024.

Maximilian Baronig, Romain Ferrand, Silvester Sabathiel, and Robert Legenstein. Advancing spatio-temporal processing through adaptation in spiking neural networks. *Nature Communications*, 16(1):5776, 2025.

Peter Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In Christopher J.C. Burges, Bernhard Schölkopf, and Alexander J. Smola (eds.), *Advances in Kernel Methods: Support Vector Learning*. The MIT Press, 12 1998. ISBN 9780262283199.

Colin Blakemore, Roger HS Carpenter, and Mark A Georgeson. Lateral inhibition between orientation detectors in the human visual system. *Nature*, 228(5266):37–39, 1970.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

Xinyi Chen, Jibin Wu, Chenxiang Ma, Yinsong Yan, Hanwen Liu, Yujie Wu, and Kay Chen Tan. Pmsn: A parallel multi-compartment spiking neuron for multiscale temporal processing. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2026.

Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. 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, 2022.

Yongqi Ding, Lin Zuo, Mengmeng Jing, Pei He, and Yongjun Xiao. Shrinking your timestep: Towards low-latency neuromorphic object recognition with spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11811–11819, 2024.

Yongqi Ding, Lin Zuo, Mengmeng Jing, Pei He, and Hanpu Deng. Rethinking spiking neural networks from an ensemble learning perspective. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yongqi Ding, Kunshan Yang, Linze Li, Yiyang Zhang, Mengmeng Jing, and Lin Zuo. Stable spike: Dual consistency optimization via bitwise AND operations for spiking neural networks. In *Conference on Computer Vision and Pattern Recognition 2026*, 2026.

Yiting Dong, Dongcheng Zhao, and Yi Zeng. Temporal knowledge sharing enable spiking neural network learning from past and future. *IEEE Transactions on Artificial Intelligence*, 5(7):3524–3534, 2024.

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 *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2641–2651, 2021.

Fuqiang Gu, Weicong Sng, Xuke Hu, and Fangwen Yu. Eventdrop: Data augmentation for event-based learning. In *30th International Joint Conference on Artificial Intelligence, IJCAI*, 2021.

Yufei Guo, Xiaode Liu, Yuanpei Chen, Liwen Zhang, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Rmp-loss: Regularizing membrane potential distribution for spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17391–17401, October 2023.

Yufei Guo, Yuanpei Chen, Xiaode Liu, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Ternary spike: Learning ternary spikes for spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12244–12252, 2024.

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

Xiang He, Dongcheng Zhao, Yang Li, Guobin Shen, Qingqun Kong, and Yi Zeng. An efficient knowledge transfer strategy for spiking neural networks from static to event domain. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 512–520, 2024.

Yifan Huang, Wei Fang, Zecheng Hao, Zhengyu Ma, and Yonghong Tian. Towards lossless memory-efficient training of spiking neural networks via gradient checkpointing and spike compression. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yulong Huang, Xiaopeng LIN, Hongwei Ren, Haotian FU, Yue Zhou, Zunchang LIU, biao pan, and Bojun Cheng. CLIF: Complementary leaky integrate-and-fire neuron for spiking neural networks. In *Forty-first International Conference on Machine Learning*, 2024.

Haiyan Jiang, Vincent Zoonekynd, Giulia De Masi, Bin Gu, and Huan Xiong. TAB: Temporal accumulated batch normalization in spiking neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

Jiaqiang Jiang, Wenfeng Xu, Jing Fan, and Rui Yan. Ds-atgo: dual-stage synergistic learning via forward adaptive threshold and backward gradient optimization for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1855–1863, 2026.

Yechan Kang, Yongjin Kweon, Mingyeong Seo, Sohee Park, Yeonguk jeon, Jongkil Park, Hyun Jae Jang, Jaewook Kim, YeonJoo Jeong, Suyoun Lee, and Seongsik Park. A2SG: Adaptive and asymmetric surrogate gradients for training deep spiking neural networks. In *Forty-third International Conference on Machine Learning*, 2026.

Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 11, 2017.

Yuhang Li, Youngeun Kim, Hyoungseob Park, et al. Neuromorphic data augmentation for training spiking neural networks. In *ECCV*, 2022.

Peiyu Liu, Jianhao Ding, and Zhaofei Yu. Training deep normalization-free spiking neural networks with lateral inhibition. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Wei Liu, Li Yang, Mingxuan Zhao, Shuxun Wang, Jin Gao, Wenjuan Li, Bing Li, and Weiming Hu. DeepTAGE: Deep temporal-aligned gradient enhancement for optimizing spiking neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Wei Liu, Li Yang, Yufei Wang, Han Xiao, Boyu Cai, and Weiming Hu. Temporal representation enhancement (tre): Learning to forget dominant patterns for enhanced temporal spiking features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 34480–34490, June 2026b.

Yijie Lu, Zhiyi Pan, Renrui Zhang, Yanhao Jia, Ronggang Wang, and Zhaokun Zhou. Spiking discrepancy transformer for point cloud analysis. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xinhao Luo, Man Yao, Yuhong Chou, Bo Xu, and Guoqi Li. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 253–272, Cham, 2025. Springer Nature Switzerland.

Jingyang Ma, Chongming Liu, Songting Li, and Douglas Zhou. Beyond linear processing: Dendritic bilinear integration in spiking neural networks. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ruixin Mao, Aoyu Shen, Lin Tang, and Jun Zhou. Crest: An efficient conjointly-trained spike-driven framework for event-based object detection exploiting spatiotemporal dynamics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 6072–6080, 2025.

Georgios Mentzelopoulos, Ioannis Asmanis, Konrad Kording, Eva L Dyer, Kostas Daniilidis, and Flavia Vitale. A scalable, causal, and energy efficient framework for neural decoding with spiking neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Garrick Orchard, Ajinkya Jayawant, Gregory K. Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9, 2015.

Wachirawit Ponghiran and Kaushik Roy. Spiking neural networks with improved inherent recurrence dynamics for sequential learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8001–8008, 2022.

Xuerui Qiu, Man Yao, Jieyuan Zhang, Yuhong Chou, Ning Qiao, Shibo Zhou, Bo Xu, and Guoqi Li. Efficient 3d recognition with event-driven spike sparse convolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20086–20094, 2025.

Xuerui Qiu, Shaowei Gu, Peixi Wu, JiaKui Hu, Yaozhi Wen, Yuqi Pan, Xinhao Luo, Bo XU, and Guoqi Li. SVL: Empowering spiking neural networks for efficient 3d open-world understanding. In *Forty-third International Conference on Machine Learning*, 2026.

Dayong Ren, Zhe Ma, Yuanpei Chen, Weihang Peng, Xiaode Liu, Yuhan Zhang, and Yufei Guo. Spiking pointnet: Spiking neural networks for point clouds. *Advances in Neural Information Processing Systems*, 36:41797–41808, 2023.

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

Mark Schöne, Neeraj Mohan Sushma, Jingyue Zhuge, Christian Mayr, Anand Subramoney, and David Kappel. Scalable event-by-event processing of neuromorphic sensory signals with deep state-space models. In *2024 International Conference on Neuromorphic Systems (ICONS)*, pp. 124–131, 2024.

Yimeng Shan, Zhenbang Ren, Haodi Wu, Wenjie Wei, Rui-Jie Zhu, Shuai Wang, Dehao Zhang, Yichen Xiao, Jieyuan Zhang, Kexin Shi, Jingzhinan Wang, Jason K. Eshraghian, Haicheng Qu, and Malu Zhang. Sdtrack: A baseline for event-based tracking via spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36245–36254, June 2026.

Xueyuan She, Saurabh Dash, Daehyun Kim, and Saibal Mukhopadhyay. A heterogeneous spiking neural network for unsupervised learning of spatiotemporal patterns. *Frontiers in Neuroscience*, Volume 14 - 2020, 2021.

Hangchi Shen, Qian Zheng, Huamin Wang, and Gang Pan. Rethinking the membrane dynamics and optimization objectives of spiking neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Xinyu Shi, Zecheng Hao, and Zhaofei Yu. Spikingresformer: Bridging resnet and vision transformer in spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5610–5619, June 2024.

Jiaqi Wang, Liutao Yu, Liwei Huang, Chenlin Zhou, Han Zhang, Zhenxi Song, Honghai Liu, Min Zhang, Zhengyu Ma, and Zhiguo Zhang. Efficient speech command recognition leveraging spiking neural networks and progressive time-scaled curriculum distillation. *Neural Networks*, pp. 108253, 2025a.

Jiaqi Wang, Liutao Yu, Xiongri Shen, Sihang Guo, Chenlin Zhou, Leilei Zhao, Yi Zhong, Zhiguo Zhang, and Zhengyu Ma. Spikcommander: A high-performance spiking transformer with multi-view learning for efficient speech command recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2119–2127, 2026.

Shuai Wang, Malu Zhang, Dehao Zhang, Ammar Belatreche, Yichen Xiao, Yu Liang, Yimeng Shan, Qian Sun, Enqi Zhang, and Yang Yang. Spiking vision transformer with saccadic attention. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Peixi Wu, Bosong Chai, Hebei Li, Menghua Zheng, Yansong Peng, Zeyu Wang, Xuan Nie, Yueyi Zhang, and Xiaoyan Sun. Spiking point transformer for point cloud classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21563–21571, 2025.

Qiaoyun Wu, Quanxiao Zhang, Chunyu Tan, Yun Zhou, and Changyin Sun. Point-to-spike residual learning for energy-efficient 3d point cloud classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 6092–6099, 2024.

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

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, June 2015.

Qi Xu, Yaxin Li, Jiangrong Shen, Jian K. Liu, Huajin Tang, and Gang Pan. Constructing deep spiking neural networks from artificial neural networks with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7886–7895, June 2023.

Jiaqi Yan, Changping Wang, De Ma, Huajin Tang, Qian Zheng, and Gang Pan. Training high performance spiking neural network by temporal model calibration. In *Forty-second International Conference on Machine Learning*, 2025a.

Jiaqi Yan, Changping Wang, De Ma, et al. Training high performance spiking neural network by temporal model calibration. In *ICML*, 2025b.

Yinsong Yan, Yujie Wu, and Jibin Wu. Advancing spatiotemporal representations in spiking neural networks via parametric invertible transformation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yuqi Yang, Tengjun Liu, Haiyan Zhang, Wang Ruixue, Xuchao Chen, Mingkang Li, Yansong Chua, Nenggan Zheng, and Shaomin Zhang. Pretraining with re-parametrized self-attention: Unlocking generalizationin SNN-based neural decoding across time, brains, and tasks. In *The Fourteenth International Conference on Learning Representations*, 2026.

Man Yao, Ole Richter, Guangshe Zhao, Ning Qiao, Yannan Xing, Dingheng Wang, Tianxiang Hu, Wei Fang, Tugba Demirci, Michele De Marchi, et al. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications*, 15(1):4464, 2024.

Man Yao, Xuerui Qiu, Tianxiang Hu, Jiakui Hu, Yuhong Chou, Keyu Tian, Jianxing Liao, Luziwei Leng, Bo Xu, and Guoqi Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4): 2973–2990, 2025.

Chengting Yu, Lei Liu, Gaoang Wang, Erping Li, and Aili Wang. Advancing training efficiency of deep spiking neural networks through rate-based backpropagation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Kairong Yu, Chengting Yu, Tianqing Zhang, Xiaochen Zhao, Shu Yang, Hongwei Wang, Qiang Zhang, and Qi Xu. Temporal separation with entropy regularization for knowledge distillation in spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8806–8816, June 2025.

Qiuyang Zhang, Jiujun Cheng, Qichao Mao, Cong Liu, Yu Fang, Yuhong Li, Mengying Ge, and Shangce Gao. Spiketrack: A spike-driven framework for efficient visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6802–6811, June 2026.

Hanle Zheng, Zhong Zheng, Rui Hu, Bo Xiao, Yujie Wu, Fangwen Yu, Xue Liu, Guoqi Li, and Lei Deng. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15(1):277, 2024.

Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. QKFormer: Hierarchical spiking transformer using q-k attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Zhaokun Zhou, Yijie Lu, Yanhao Jia, Kaiwei Che, Jun Niu, Liwei Huang, Xinyu Shi, Yuesheng Zhu, Guoqi Li, Zhaofei Yu, and Li Yuan. Spiking transformer with experts mixture. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b.

Zijian Zhou, Honglin Cao, Ammar Belatreche, Wenjie Wei, Yimeng Shan, Yu Liang, Yu Yang, Shuai Wang, Yalan Ye, Malu Zhang, Yang Yang, and Haizhou Li. Adas: Adaptive gradient descent for spiking transformers. In *Forty-third International Conference on Machine Learning*, 2026.

A Description of the Appendix

This appendix includes the following:

- Appendix B provides a theoretical analysis that demonstrates how STMR improves the generalization performance of SNNs.
- Appendix C illustrates the STMR training algorithm framework.
- Appendix D details the experimental setup, including the dataset, model architecture, and training strategy.
- Appendix E presents additional experimental results, including comparative results on CIFAR10/100 and sensitivity analysis experiments on hyperparameters.

B Theoretical Analysis

This section provides theoretical analysis from the perspectives of output margin generalization bounds and temporal dynamic stability to demonstrate that the proposed STMR method enhances the generalization ability of SNNs.

B.1 Confidence Margin and Generalization Bound

First, we establish a connection between the confidence margins at each timestep and the generalization performance of the final multi-timestep averaged classifier. For multi-class classification, classical margin-based generalization bounds Bartlett & Shawe-Taylor (1998); Koltchinskii & Panchenko (2002) state that with probability at least $1-\delta$, the expected 0–1 loss satisfies:

$$\mathcal{R}(h) \leq \mathcal{R}_{\text{emp}}(h) + \mathcal{O}\left(\sqrt{\frac{d+\log(1/\delta)}{n\gamma^2}}\right), \tag{11}$$

where d reflects the capacity of hypothesis class $\mathcal{H}$, n is the number of training samples, and $\gamma > 0$ denotes the classification margin (the minimum gap between the confidence of the target class and that of the strongest competing class). Larger γ is associated with tighter generalization gaps.

For SNNs, the final prediction is obtained by temporal averaging of per-timestep outputs:

$$O_T = \frac{1}{T}\sum_{t=1}^{T} o_t, \quad p_T = \text{softmax}(O_T). \tag{12}$$

Let $\delta_t = p_{t,\tilde{y}}^{\max} - p_{t,y}$ denote the negative confidence margin at timestep t, as defined in Eq. (7). The final margin γ_T is lower-bounded by the average of per-timestep margins:

$$\gamma_T \geq \frac{1}{T}\sum_{t=1}^{T}(-\delta_t). \tag{13}$$

That is, improving the confidence margin at any individual timestep improves the final margin, and timesteps with more negative δ_t contribute disproportionately to degrading the final margin.

Our timestep-coordinated weights $w_t(x) \propto \exp(\delta_t(x))$ (Eq. (9)) allocate more regularization strength to timesteps with larger negative margins. This is equivalent to minimizing the worst-case per-timestep margin violation rather than applying uniform smoothing across all timesteps. Compared with uniform regularization, this adaptive strategy achieves a larger improvement in the minimum margin under the same regularization budget, which is consistent with a tighter generalization bound under the margin-based framework above.

B.2 Temporal Margin and Dynamical Stability

We next analyze why advancing target-class spike timing suppresses temporal error accumulation from the dynamical perspective, which is the root cause of generalization degradation.

Recall the LIF membrane dynamics in Eq. (1)–(3). The input current $I_{i,t}^{l}$ is the weighted sum of spikes from the preceding layer. A well-documented property of SNN circuits is lateral inhibition (Blakemore et al., 1970), where the firing of one neuron suppresses competing neurons. This creates a positive-feedback loop for timing errors:

- If a target-negatively correlated neuron misfires early, it suppresses the membrane potential of target-positively correlated neurons.
- Weakened positive neurons in turn exert less reciprocal inhibition on negative neurons.
- The timing gap widens across timesteps, and small initial errors are amplified rather than dissipated by membrane leakage.

Define the temporal margin as

$$\Delta_{\text{lat}} = \ell_y - \ell_{\bar{y}}^{\min}, \tag{14}$$

where ℓ_y is the target-class equivalent latency and $\ell_{\bar{y}}^{\min}$ is the latency of the earliest-firing wrong class, consistent with Eq. (6). Negative Δ_{lat} means the target pathway fires first and establishes inhibitory dominance early.

For vanilla SNNs where $\Delta_{\text{lat}} > 0$ (the target-class positively correlated neuron fires first), an initial timing perturbation ε_0 at $t = 1$ grows approximately as

$$\varepsilon_t \approx \alpha^t \varepsilon_0, \quad \alpha > 1, \tag{15}$$

where $\alpha > 1$ is the error amplification factor driven by the positive-feedback loop. In contrast, when TMR pushes $\Delta_{\text{lat}} < 0$ (target pathway fires first), the target pathway establishes inhibitory dominance early, and the perturbation satisfies

$$\varepsilon_t \approx \alpha'^t \varepsilon_0, \quad \alpha' < 1. \tag{16}$$

That is, initial perturbations are suppressed rather than amplified over the temporal rollout.

This improvement in stability enhances generalization ability directly, allowing target spikes to suppress minor input noise or spike timing jitter that might otherwise be amplified into prediction errors. In other words, TMR improves the robustness of spike dynamics to input perturbations, a mechanism widely associated with better generalization Bousquet & Elisseeff (2002).

C STMR TRAINING FRAMEWORK

To facilitate an understanding of the detailed steps of the proposed STMR, Algorithm 1 shows the overall STMR framework for training SNNs.

Algorithm 1 Spatio-temporal margin regularization for SNNs

Input: input data x, label y.
Parameter: timestep T, regularization coefficients λ_{TMR} and λ_{SMR}.
Output: Trained SNN.

Initialize SNN $f(\cdot)$ with parameters θ
// Optimization over multiple iterations
for $i = 1, 2, \cdots, i_{train}$ iterations **do**
  for $t = 1, 2, \cdots, T$ **do**
    $o_t = f(x; \theta, t)$; // Calculate the output at timestep t
    $\delta_t(x) = p^{\max}_{t,\bar{y}}(x) - p_{t,y}(x)$; // Calculate the confidence margin by Eq. (7)
  end for
  $O_T = \frac{1}{T}\sum_{t=1}^{T} O_t$; // Calculate the average output
  $\mathcal{L}_{CE} = \text{Cross-entropy}(O_T, y)$; // Calculate the cross-entropy loss
  $w_t(x) = \frac{\exp(\delta_t(x))}{\sum_{t'=1}^{T}\exp(\delta_{t'}(x))}$; // Calculate the timestep coordination weights by Eq. (9)
  Calculate the spatial margin regularization loss $\mathcal{L}_{\text{SMR}}$ by Eq. (8)
  Calculate the class-specific spike latency metric ℓ_c by Eq. (4)
  Calculate the temporal margin regularization loss $\mathcal{L}_{\text{TMR}}$ by Eq. (6)
  Calculate total STMR loss by Eq. (10)
  Backpropagation and optimize model parameters θ;
end for
return Trained SNN.

Table 6: Structures of VGG-9, VGGSNN, ResNet-18, and ResNet-19, where AP, GAP, and FC denote average pooling, global average pooling, and fully connected layers, respectively.

Stage	VGG-9	VGGSNN	ResNet-18	ResNet-19
1	-	-	$\text{Conv}(3\times3@64)$	$\text{Conv}(3\times3@128)$
1	$\text{Conv}(3\times3@64)$ $\text{Conv}(3\times3@128)$	$\text{Conv}(3\times3@64)$ $\text{Conv}(3\times3@128)$	$\begin{pmatrix}\text{Conv}(3\times3@64)\\ \text{Conv}(3\times3@64)\end{pmatrix}\times2$	$\begin{pmatrix}\text{Conv}(3\times3@128)\\ \text{Conv}(3\times3@128)\end{pmatrix}\times3$
	AP(stride=2)	AP(stride=2)	-	-
2	$\text{Conv}(3\times3@256)$ $\text{Conv}(3\times3@256)$	$\text{Conv}(3\times3@256)$ $\text{Conv}(3\times3@256)$	$\begin{pmatrix}\text{Conv}(3\times3@128)\\ \text{Conv}(3\times3@128)\end{pmatrix}\times2$	$\begin{pmatrix}\text{Conv}(3\times3@256)\\ \text{Conv}(3\times3@256)\end{pmatrix}\times3$
	AP(stride=2)	AP(stride=2)	-	-
3	$\text{Conv}(3\times3@512)$ $\text{Conv}(3\times3@512)$	$\text{Conv}(3\times3@512)$ $\text{Conv}(3\times3@512)$	$\begin{pmatrix}\text{Conv}(3\times3@256)\\ \text{Conv}(3\times3@256)\end{pmatrix}\times2$	$\begin{pmatrix}\text{Conv}(3\times3@512)\\ \text{Conv}(3\times3@512)\end{pmatrix}\times2$
	AP(stride=2)	AP(stride=2)	-	
4	$\text{Conv}(3\times3@512)$ $\text{Conv}(3\times3@512)$	$\text{Conv}(3\times3@512)$ $\text{Conv}(3\times3@512)$	$\begin{pmatrix}\text{Conv}(3\times3@512)\\ \text{Conv}(3\times3@512)\end{pmatrix}\times2$	
	GAP,FC	AP,FC	GAP,FC	GAP,FC×2

D Experimental Details

D.1 2D Neuromorphic Vision Datasets.

We conducted experiments on three neuromorphic object recognition benchmark datasets: CIFAR10-DVS Li et al. (2017), DVS-Gesture Amir et al. (2017), and N-Caltech101 Orchard et al. (2015). The CIFAR10-DVS Li et al. (2017) dataset contains a total of 10,000 samples from ten categories. The DVS-Gesture Amir et al. (2017) dataset contains neuromorphic data for 11 hand gestures with 1176 training samples and 288 test samples. The spatial resolution of each sample in CIFAR10-DVS and DVS-Gesture is 128×128, which we downsampled to 48×48 and input to SNN, the standard input size for the community Ding et al. (2026); Zhou et al. (2024a). The N-Caltech101 Orchard et al. (2015) dataset contains 8,109 samples, categorized into 101 categories, with an original resolution of 180×240. We also downsampled the samples in N-Caltech101 to 48×48.

We conducted our experiments on an Ubuntu 20.04.5 operating system with NVIDIA 4090 GPUs. To validate the versatility of the proposed method, we conducted experiments using various architectures, including VGG-9, ResNet-18, VGGSNN, and QKFormer. For VGG-9 and ResNet-18, we use the same training strategy as in Ding et al. (2026). The model was trained for 100 epochs using a stochastic gradient descent optimizer. The initial learning rate was set to 0.1, and it was scaled down tenfold every 30 epochs. The batch size is 64, and the weight decay is 1e-3. In particular, for VGG-9 and ResNet-18, we do not use any data augmentation during training. The firing threshold of the spiking neuron was set to 1.0, and the membrane potential time constant was set to 2.0. In the training of the SNN, the rectangular surrogate gradient function is employed for backpropagation, consistent with Ding et al. (2026).

When using VGGSNN to recognize CIFAR10-DVS, we follow the training strategy described in Huang et al. (2026). When using the VGGSNN architecture on the N-Caltech101 dataset, we incorporate the knowledge transfer strategy He et al. (2024) and employ the same training strategy as in He et al. (2024). Similarly, when using the QKFormer architecture, we follow the training strategy outlined in the original paper Zhou et al. (2024a) to ensure the fairness of the experiments.

D.2 2D Static Vison Datasets.

For static images, we conducted experiments on the CIFAR10 and CIFAR100 datasets. Both the CIFAR-10 and CIFAR-100 datasets contain 60,000 32×32 images, 50,000 of which are in the training set and 10,000 of which are in the test set. For CIFAR10 and CIFAR100, we applied the ResNet-18 and ResNet-19 architectures. We used a stochastic gradient descent optimizer with a momentum of 0.9 to train for 300 epochs. The initial learning rate was set to 0.1, and we employed a cosine annealing learning rate strategy. The batch size is 128, and weight deacy is 5e-4. We em-

ployed the same training strategy as Ding et al. (2026). Table 6 illustrates the different architectures employed in the experiment.

D.3 1D Neuromorphic Speech Dataset.

The SHD dataset Cramer et al. (2022) is the benchmark for 1D neuromorphic speech recognition and includes 20 speech-digit categories. To ensure a fair comparison, we adopted the same preprocessing and training strategies for SHD as for Wang et al. (2026).

D.4 3D Point Cloud Dataset.

ModelNet40 Wu et al. (2015) is a benchmark dataset for 3D point cloud recognition. To ensure a fair comparison, we applied our proposed STMR method to the SDT Lu et al. (2026) architecture for ModelNet40 and followed its original training strategy.

D.5 Parameter Settings

Because temporal error accumulation varies depending on the dataset and architecture, we assigned different balancing coefficients to each one. See Table 7 for the specific settings. Figure 1 validates the influence of these two coefficients across the VGG-9 and QKFormer architectures in DVS-Gesture, showing that while they induce slight fluctuations in performance, STMR consistently yields performance gains. Therefore, the settings in Table 7 only represent the results used in this paper and are not the optimal configurations for these datasets. We recommend setting these two hyperparameters to relatively small values, such as 0.01, when using the STMR method proposed in this paper. Since STMR applies margin regularization to both the spike timing and the output confidence margin, setting the strength too high can easily lead to a decline in performance. However, it remains beneficial at an appropriate strength.

Dataset	Architecture	λ_{TMR}	λ_{SMR}
CIFAR10-DVS	VGG-9	0.01	0.02
CIFAR10-DVS	ResNet-18	0.2	0.2
CIFAR10-DVS	QKFormer	0.01	0.001
CIFAR10-DVS	VGGSNN	0.0005	0.01
DVS-Gesture	VGG-9	0.01	0.005
DVS-Gesture	ResNet-18	0.01	0.01
DVS-Gesture	QKFormer	0.1	0.1
SHD	SpikCommander	0.015	0.005
ModelNet-40	SDT	0.02	0.01

Table 7: Parameter settings across datasets and architectures.

Table 8: Comparative results on static CIFAR10/100.

	Method	Architecture	$T\downarrow$	Acc. (%) $\uparrow$
CIFAR10	QKFormer$^{NeurIPS'24}$ Zhou et al. (2024a)	QKFormer	4	96.18
	SNN-ViT$^{ICLR'25}$ Wang et al. (2025b)	SNN-ViT	4	96.10
	RateBP$^{NeurIPS'24}$ Yu et al. (2024)	ResNet-19	4	96.26
	DeepTAGE$^{ICLR'25}$ Liu et al. (2025)	ResNet-18	4	95.86
	SLT$^{AAAI'24}$ Anumasa et al. (2024)	ResNet-19	4	95.18
	RateBP$^{NeurIPS'24}$ Yu et al. (2024)	ResNet-18	4	95.61
	KDSNN$^{CVPR'23}$ Xu et al. (2023)	ResNet-18	4	93.41
	Stable Spike^{CVPR} Ding et al. (2026)	ResNet-18	4	95.70
	STMR (Ours)	ResNet-18	4	**95.72**
		ResNet-19	4	**96.61**
CIFAR100	QKFormer$^{NeurIPS'24}$ Zhou et al. (2024a)	QKFormer	4	81.15
	SNN-ViT$^{ICLR'25}$ Wang et al. (2025b)	SNN-ViT	4	80.10
	RateBP$^{NeurIPS'24}$ Yu et al. (2024)	ResNet-18	4	78.26
	DeepTAGE$^{ICLR'25}$ Liu et al. (2025)	ResNet-19	4	81.39
	RateBP$^{NeurIPS'24}$ Yu et al. (2024)	ResNet-19	4	80.71
	TMC$^{ICML'25}$ Yan et al. (2025b)	ResNet-19	4	77.52
	SLT$^{AAAI'24}$ Anumasa et al. (2024)	ResNet-19	4	75.01
	A^{2}SG^{ICML} Kang et al. (2026)	ResNet-19	4	81.05
	TRE^{CVPR} Liu et al. (2026b)	ResNet-19	4	81.27
	STMR (Ours)	ResNet-18	4	**78.39**
		ResNet-19	4	**81.44**

E Additional Experimental Results

E.1 Experimental Results on CIFAR-10 and CIFAR-100

Table 8 shows the comparative results of STMR on the static CIFAR-10 and CIFAR-100 datasets. On these datasets, STMR achieved accuracies of 96.61% and 81.44%, respectively. These results further validate STMR's cross-dataset and cross-architecture generality.

E.2 Results of Hyperparameter Sensitivity Analysis

We conducted experiments using the VGG-9 and QKFormer architectures on the DVS-Gesture dataset with four timesteps to investigate the effects of regularization coefficients λ_{TMR} and λ_{SMR} on STMR. We set the range of values for λ_{TMR} and λ_{SMR} to $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$. Figure 5 shows that changes in these two hyperparameters can cause performance fluctuations while STMR consistently outperforms the baseline. This indicates that STMR is relatively robust with respect to these hyperparameters.

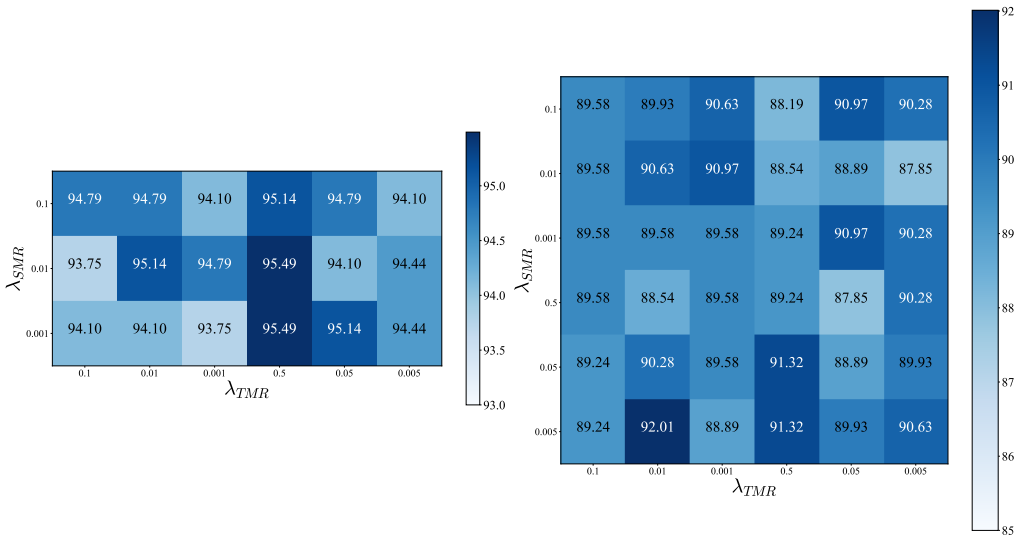

Figure 5: Sensitivity analysis results for coefficients λ_{TMR} and λ_{SMR} on DVS-Gesture. Left: VGG-9, Right: QKFormer. Varying the regularization coefficients λ_{TMR} and λ_{SMR} causes slight fluctuations in performance, but STMR consistently yields performance gains. This indicates that STMR is relatively robust with respect to hyperparameters.