LOCAL NEURONAL DYNAMICS AS TEMPORAL INDUCTIVE BIASES FOR SPIKING TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neurons maintain internal states that evolve over time, providing an intrinsic source of temporal dynamics for spiking neural networks. However, existing spiking Transformers often underutilize this intrinsic temporal structure, either treating timesteps independently or introducing explicit temporal interactions through temporal attention. In this work, we investigate whether local neuronal dynamics can serve as an effective temporal inductive bias for spiking Transformers. We propose SNN-native Temporal Modeling (SNTM), a framework that incorporates temporal information at two stages of spiking computation. Multi-threshold Spiking Patch Splitting (MT-SPS) enriches spike representations by aggregating responses from multiple fixed firing thresholds during the initial spike projection. Event-driven Value Modulation (EDVM) captures local temporal variations through the current spike activity and the first-order difference between adjacent value features, and injects these cues into the value pathway. Across four spiking Transformer backbones and six static or event-based benchmarks, SNTM improves the corresponding baselines. On ImageNet, it improves SDT, QKFormer, and Spikformer by 1.75, 1.46, and 2.07 accuracy points, respectively. Mechanism studies on CIFAR10-DVS show that disrupting temporal adjacency in EDVM reduces accuracy from 78.23% to 76.47%, whereas reversing the temporal reference has little effect, consistent with EDVM relying on local temporal contrast rather than temporal direction. SNTM adds only three learned scalars per enabled attention block and at most 0.09G MACs in our measured settings. These results demonstrate that local neuronal dynamics provide a practical and low-overhead temporal inductive bias for spiking Transformers.

1 INTRODUCTION

Spiking neural networks (SNNs) process information through stateful neurons and sparse spike events (Maass, 1997; Roy et al., 2019; Merolla et al., 2014; Davies et al., 2018), providing a computational paradigm fundamentally different from conventional artificial neural networks. Recent spiking Transformers combine this computation paradigm with Transformer architectures (Vaswani et al., 2017), achieving competitive performance while adapting attention mechanisms to spike-based representations (Zhou et al., 2022; Yao et al., 2023; Zhou et al., 2024; Shi et al., 2024). Despite these advances, effectively exploiting temporal information remains a central challenge in spiking Transformers. Existing approaches typically handle the timestep dimension in two ways: they apply the same feature transformation independently at each timestep, or introduce explicit temporal interactions through spatio-temporal attention (Lee et al., 2025; Arnab et al., 2021; Bertasius et al., 2021; Wu et al., 2023). The former does not explicitly exchange information across timesteps, whereas the latter introduces additional temporal computation coupled with the attention mechanism.

SNNs provide another perspective for temporal modeling. Unlike conventional neural networks, spiking neurons maintain internal states that evolve through historical inputs, membrane dynamics, thresholding, and reset mechanisms (Gerstner & Kistler, 2002; Maass, 2001; Fang et al., 2020). As a result, temporal variations can naturally emerge from neuronal responses even when identical inputs are repeatedly presented over multiple simulation steps, while event-based streams further contain dynamic changes from asynchronous signals. However, existing spiking Transformer designs rarely make these locally evolving neuronal responses an explicit source of temporal information. This ob-

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.

servation motivates a different question: can local neuronal dynamics serve as an effective temporal inductive bias for spiking Transformers?

We explore this question through **SNN-native Temporal Modeling (SNTM)**, a framework that incorporates temporal information into spiking Transformer computation by leveraging intrinsic neuronal dynamics. SNTM introduces temporal modeling at two stages where spike dynamics naturally emerge. At the spike representation stage, Multi-threshold Spiking Patch Splitting (MT-SPS) applies multiple fixed firing thresholds to a shared projected feature and aggregates their responses, enriching spike representations through threshold-dependent dynamics. At the feature aggregation stage, Event-driven Value Modulation (EDVM) extracts local temporal variations from the current spike activity and the first-order difference between adjacent value features, and uses the resulting event signal to modulate the value pathway. Together, MT-SPS and EDVM expose temporal information already present in spiking computation and provide a lightweight temporal bias while preserving the original Transformer structure.

We evaluate SNTM across four spiking Transformer backbones and six benchmarks, including static-image datasets (CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky et al., 2009), and ImageNet (Deng et al., 2009)) and neuromorphic event datasets (CIFAR10-DVS (Li et al., 2017), N-Caltech101 (Orchard et al., 2015), and DVS128 Gesture (Amir et al., 2017)). SNTM consistently improves the corresponding baselines across different architectures and data modalities. Further mechanism analyses show that the improvement of EDVM mainly originates from local temporal contrast between adjacent features. These results demonstrate that intrinsic neuronal dynamics provide an effective and efficient temporal inductive bias for spiking Transformers.

Our contributions are summarized as follows:

- We revisit temporal modeling in spiking Transformers from the perspective of intrinsic neuronal dynamics and propose local neuronal dynamics as a temporal inductive bias.
- We introduce SNN-native Temporal Modeling (SNTM), consisting of MT-SPS and EDVM, which incorporate temporal information at the spike representation and feature aggregation stages with minimal modification to the backbone.
- We conduct extensive evaluations across four spiking Transformer backbones and six static or event-based benchmarks, together with mechanism analyses that characterize how local temporal dynamics contribute to performance improvements.

2 Related Work

2.1 Neuronal Dynamics in Spiking Networks

Spiking neurons process information through internal state evolution, where inputs are integrated into membrane states and converted into spike events through thresholding and reset mechanisms (Gerstner & Kistler, 2002; Maass, 2001). This evolving internal state provides SNNs with an intrinsic source of temporal dynamics beyond conventional neural networks. Recent studies have explored neuronal dynamics from various perspectives, including membrane-potential propagation, learnable time constants, adaptive neuronal thresholds, and improved temporal credit assignment (Fang et al., 2020; Bellec et al., 2018; Ding et al., 2022; Hirano et al., 2026). These studies demonstrate that spiking neurons are not merely pointwise activation functions; their evolving internal states provide a natural source of temporal information.

2.2 Spiking Transformers

Recent works have extended Transformer architectures to spiking neural networks in order to combine the representational power of attention mechanisms with the efficiency of event-driven computation. Representative models include Spikformer (Zhou et al., 2022), SDT (Yao et al., 2023), QKFormer (Zhou et al., 2024), and SpikingResformer (Shi et al., 2024), which mainly redesign attention to better match spike-based computation. Despite these advances, most existing spiking Transformers primarily exploit the efficiency of spike-based computation, focusing on reducing computational cost through sparse activations and event-driven processing. However, another fun-

damental property of SNNs, namely their intrinsic temporal dynamics, has not been fully utilized for temporal modeling.

2.3 Temporal Modeling in Transformers

Temporal modeling is an important problem in Transformer-based architectures, as it provides additional representational capacity beyond static feature extraction. A common strategy is to introduce temporal information through explicit interactions among timesteps using temporal or spatio-temporal attention. For example, video Transformers such as ViViT (Arnab et al., 2021) model temporal dependencies by jointly computing attention over tokens from multiple time steps. Similarly, spiking Transformer variants extend this idea by incorporating spatio-temporal attention mechanisms, such as STAtten (Lee et al., 2025), where features from different time steps interact to capture cross-time dependencies. While effective for learning temporal dependencies, these approaches construct temporal interactions as explicit operations within the attention mechanism. Such designs introduce additional cross-time computation and tightly couple temporal modeling with the attention formulation.

3 Method

3.1 Temporal Signals from Neuronal Dynamics

A discrete spiking neuron maintains an internal membrane state that evolves across simulation steps. An illustrative leaky integrate-and-fire update is

$$\mathbf{U}_t = \rho \mathbf{U}_{t-1} + \mathbf{I}_t - \theta \mathbf{S}_{t-1}, \qquad \mathbf{S}_t = \mathcal{H}(\mathbf{U}_t - \theta), \tag{1}$$

where $\mathbf{U}_t$ is the membrane potential, $\mathbf{I}_t$ denotes the current input, ρ is the decay factor, θ is the firing threshold, and $\mathcal{H}$ denotes the spike function. Exact reset conventions may differ across backbones, but the common property is that the neuronal response at the current timestep depends on its evolving internal state. Consequently, even when the same static image is repeatedly presented over multiple simulation steps, hidden neuronal responses need not remain identical. For event-based inputs, these intrinsic dynamics further interact with temporal variations already present in the input stream.

These observations suggest that spiking computation itself provides locally available temporal signals. Rather than introducing an additional temporal encoder, SNTM exploits such signals directly and uses them as a temporal inductive bias within the spiking Transformer.

3.2 Overview

Let $\mathbf{X} \in \mathbb{R}^{T\times B\times C\times H\times W}$ denote an input sequence with T simulation steps. For static datasets, the same image is repeated along the temporal dimension, and timestep-dependent responses arise from membrane integration, firing, and reset. For event-based datasets, different temporal bins may already contain distinct inputs, which are further transformed by the neuronal dynamics of each layer. In both cases, local temporal variations naturally emerge during spiking computation, but standard timestep-wise attention does not explicitly use adjacent-time variation to guide feature aggregation.

SNTM incorporates these temporal signals at two complementary stages, as illustrated in Fig. 1. At the spike-representation stage, Multi-threshold Spiking Patch Splitting (MT-SPS) applies several fixed firing thresholds to a shared feature projection, retaining threshold-dependent response information before the hierarchical backbone begins. At the feature-aggregation stage, Event-driven Value Modulation (EDVM) measures local variation between adjacent value features, combines it with the current spike activity, and uses the resulting event signal to modulate the native value pathway. In this way, MT-SPS enriches the initial spike representation, while EDVM conditions feature aggregation on local temporal variations observed within each attention block.

The design follows three principles. First, temporal cues are derived from native neuronal responses rather than an additional dense temporal encoder. Second, the backbone-specific query-key interaction is preserved so that the same formulation can be applied to different spiking attention mechanisms. Third, the original value feature is retained through a residual path, allowing the contribution of temporal modulation to adapt to the strength of the observed variation. EDVM therefore

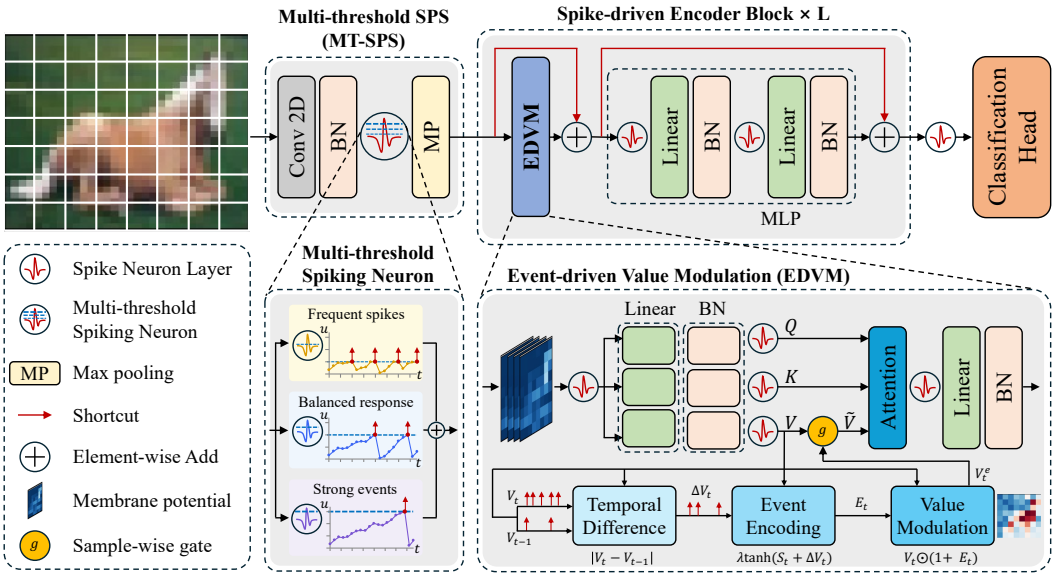


Figure 1: Overview of SNN-native Temporal Modeling (SNTM). MT-SPS enriches spike representations at the initial projection using multiple fixed firing thresholds, while EDVM incorporates local temporal variations into the value pathway through event-driven modulation.

introduces only local adjacent-time modulation while preserving the backbone's native attention structure.

3.3 Multi-threshold Spiking Patch Splitting

The initial spike projection converts pixels or event bins into the feature space processed by the spiking backbone. A standard spiking patch-splitting stem typically applies a single neuronal threshold after feature projection, mapping responses of different magnitudes to the same binary firing state once the threshold is crossed. Although efficient, this process discards part of the response diversity contained in the projected feature.

To retain richer neuronal responses at this stage, we propose Multi-threshold Spiking Patch Splitting (MT-SPS). Given the input sequence $\mathbf{X}$, we first obtain a shared projected feature

$$\mathbf{Z} = \mathcal{F}(\mathbf{X}), \tag{2}$$

where $\mathcal{F}(\cdot)$ denotes the backbone-specific convolutional or linear projection. The same feature is then evaluated by M spiking branches with different fixed firing thresholds:

$$\mathbf{S}^{(m)} = \mathrm{LIF}(\mathbf{Z}; \theta_m), \qquad m = 1, \ldots, M, \tag{3}$$

and their responses are aggregated as

$$\mathbf{S}_{\mathrm{MT}} = \sum_{m=1}^{M} \mathbf{S}^{(m)}. \tag{4}$$

In our implementation, we use $M = 3$ with $\theta_m \in \{1.0, 1.5, 2.5\}$. The low-threshold branch is more responsive to weaker accumulated evidence, the intermediate branch represents a moderate firing regime, and the high-threshold branch is activated by stronger neuronal responses. Since all branches share the same projected feature $\mathbf{Z}$, MT-SPS does not introduce separate convolutional or linear projections. Instead, the shared neuronal input is observed through multiple fixed firing regimes.

The summation in Eq. (4) produces an integer-valued response in $\{0, \ldots, M\}$, reflecting how many threshold-specific neuronal branches are active at each location and timestep. Compared with a

single-threshold spike representation, this aggregation preserves response diversity across different firing regimes. Since all branches operate on the same projected feature, the resulting diversity arises from neuronal thresholds rather than different spatial receptive fields or independently learned projections.

MT-SPS is applied only once at the earliest spike-projection stage. This placement makes the enriched spike representation available to subsequent layers without repeatedly increasing activation precision throughout the network. It introduces additional fixed-threshold neuron evaluations but no learned projection parameters. Backbone-specific insertion points and implementation details are provided in Appendices A and B.

3.4 Event-driven Value Modulation

Within an attention block, the query and key determine the interaction pattern, while the value pathway carries the content to be aggregated. To incorporate temporal information without changing the backbone-specific interaction rule, EDVM operates only on the value/content pathway. This choice allows locally varying neuronal responses to influence feature aggregation while preserving the native query-key computation.

Let $\mathbf{V}_t$ denote the native value-path feature at timestep t, and let $\mathbf{S}_t$ denote the spike activation associated with the corresponding value/content pathway. Depending on the backbone, spatial dimensions may be represented explicitly or flattened into tokens; all EDVM operations remain elementwise and shape preserving. The exact source of $\mathbf{S}_t$ follows the backbone-specific value pathway and is detailed in Appendices A and B. EDVM constructs the modulation signal in three stages: local temporal difference, event encoding, and sample-wise residual mixing.

Temporal Difference. We first compute the magnitude of the variation between adjacent value features:

$$\Delta\mathbf{V}_t = \begin{cases} \mathbf{0}, & t = 1, \\ |\mathbf{V}_t - \mathbf{V}_{t-1}|, & t > 1. \end{cases} \tag{5}$$

This first-order difference provides a local measure of how the value representation changes between neighboring timesteps. The zero boundary at $t = 1$ avoids introducing an artificial difference from an undefined preceding feature. The absolute value captures the magnitude of local temporal variation without explicitly encoding its direction.

Event Encoding and Value Modulation. The local difference is combined with the current spike response to construct a bounded event descriptor:

$$\mathbf{E}_t = \lambda \tanh\left(\mathbf{S}_t + \Delta\mathbf{V}_t\right), \tag{6}$$

where $\lambda = 1.5$ controls the modulation magnitude and $\tanh(\cdot)$ bounds the response. The event descriptor is then used to obtain an enhanced value feature:

$$\mathbf{V}_t^e = \mathbf{V}_t \odot (1 + \mathbf{E}_t). \tag{7}$$

The two terms in Eq. (6) provide complementary information. $\mathbf{S}_t$ represents the neuronal activity at the current timestep, while $\Delta\mathbf{V}_t$ identifies features exhibiting local temporal variation. Their combination emphasizes value features that are both active and temporally changing. The bounded nonlinear mapping prevents large differences from producing unbounded modulation, while the residual factor $1 + \mathbf{E}_t$ keeps the native feature as the reference representation.

Sample-wise Gate and Residual Mixing. The relevance of local temporal variation can differ across samples. We therefore summarize the normalized variation magnitude for sample b as

$$r_b = \frac{\mathrm{Mean}(\Delta\mathbf{V}_{:,b})}{\mathrm{Mean}(|\mathbf{V}_{:,b}|) + \epsilon}, \tag{8}$$

where the mean is taken over timesteps, tokens, and channels, and $\epsilon = 10^{-6}$ ensures numerical stability. We parameterize the modulation strength as

$$\alpha = \sigma(a), \qquad \gamma = \mathrm{softplus}(\eta), \qquad g_b = \sigma\left(\gamma(r_b - \tau)\right), \tag{9}$$

where a, η, and τ are learned scalars.

Table 1: Top-1 accuracy (%) across static-image and event-based benchmarks, with ImageNet serving as the large-scale evaluation. Unmarked results on the five smaller benchmarks are our three-run means; their standard deviations and individual runs are reported in Appendices D and E. † denotes results quoted from the corresponding papers, and ‡ denotes our single-run ImageNet results. Static datasets use $T = 4$ and event datasets use $T = 16$.

Backbone	Temporal module	CIFAR10	CIFAR100	CIFAR10-DVS	N-Caltech101	Gesture	ImageNet
SDT (Yao et al., 2023)	None	95.14	77.82	76.77	80.88	96.87	74.57†
	STAtten (Lee et al., 2025)	95.08	78.43	77.27	81.00	–	76.18‡
	SNTM (ours)	**95.47**	**78.85**	**78.40**	**82.38**	**97.21**	**76.32‡**
QKFormer (Zhou et al., 2024)	None	95.12†	79.79†	82.90†	83.58†	98.60†	82.04†
	STAtten (Lee et al., 2025)	95.35†	80.20†	83.90†	84.25†	–	–
	SNTM (ours)	**96.23**	**81.16**	**84.03**	**84.42**	**98.97**	**83.50‡**
SpikingResformer (Shi et al., 2024)	None	95.03†	77.16†	78.80†	81.29†	98.60†	–
	STAtten (Lee et al., 2025)	95.26†	77.90†	80.60†	81.95†	–	–
	SNTM (ours)	**96.13**	**78.79**	**81.20**	**82.15**	**98.70**	–
Spikformer (Zhou et al., 2022)	None	93.99†	75.06†	80.90†	–	98.30†	73.38†
	STAtten (Lee et al., 2025)	94.36†	75.85†	–	–	–	–
	SNTM (ours)	**94.58**	**77.29**	**81.20**	**81.30**	**98.60**	**75.45‡**

The final modulated value is

$$\widetilde{\mathbf{V}}_{:,b} = \mathbf{V}_{:,b} + \alpha g_b \left(\mathbf{V}^{e}_{:,b} - \mathbf{V}_{:,b} \right). \tag{10}$$

Here, r_b normalizes temporal variation by the average magnitude of the native value feature. Since $\gamma = \mathrm{softplus}(\eta) > 0$, the gate g_b increases monotonically with r_b, allowing samples with stronger normalized temporal variation to receive larger modulation. The learned parameter τ determines the center of this transition, while γ controls its sharpness. The coefficient α bounds the overall strength of the event-driven residual, and the outer residual path in Eq. (10) preserves the native value feature.

The modulated value is then passed to the backbone's original attention operator:

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \widetilde{\mathbf{V}}). \tag{11}$$

Thus, EDVM injects local temporal information through the value pathway while leaving the native query-key interaction unchanged. Only three scalars (a, η, τ) are learned for each enabled attention block; the event descriptor itself is computed directly from existing activations. The same formulation is used across all evaluated backbones, with only the native definition and layout of the value/content tensor adapted to each architecture. Detailed tensor mappings are provided in Appendix B.

Computational overhead. Let T, N, and D denote the numbers of timesteps, tokens, and feature dimensions, respectively. EDVM involves only elementwise differencing, pooling, gating, and modulation, introducing an additional $\mathcal{O}(TND)$ cost without changing the backbone's native attention operator. It therefore preserves the asymptotic order of linear spike-driven attention with complexity $\mathcal{O}(TND)$, while contributing only a lower-order $\mathcal{O}(TND)$ term to backbones with $\mathcal{O}(TND^2)$ attention. EDVM operates only between adjacent timesteps and does not construct pairwise temporal affinity matrices. MT-SPS similarly reuses the original feature projection and adds only M fixed-threshold neuron evaluations at the initial spike-projection stage. Measured parameter counts and MACs for the instantiated models are reported in Section 4.4.

4 Experiments

4.1 Experimental Setup

We evaluate SNTM on two small-scale static-image benchmarks, CIFAR10 and CIFAR100 (Krizhevsky et al., 2009); three neuromorphic event-based benchmarks, CIFAR10-DVS (Li et al., 2017), N-Caltech101 (Orchard et al., 2015), and DVS128 Gesture (Amir et al., 2017); and the large-scale static-image benchmark ImageNet (Deng et al., 2009). Static images are repeated for

Table 2: Component and mechanism analysis with SDT. Top-1 accuracy (%) is reported as mean±standard deviation over three runs. Spike-only and Difference-only isolate the two signals used by EDVM with MT-SPS disabled. Shuffle and reverse are test-time interventions on the complete-EDVM checkpoints that modify only the temporal reference used by the difference branch.

Variant	MT-SPS	Spike response	Difference	Temporal reference	CIFAR10	CIFAR10-DVS
SDT	×	×	×	–	95.14±0.21	76.77±0.46
+ MT-SPS	✓	×	×	–	95.20±0.14	77.70±0.46
+ EDVM	×	✓	✓	adjacent	95.32±0.11	78.23±0.32
Spike-only	×	✓	×	–	95.21±0.05	77.53±0.21
Difference-only	×	×	✓	adjacent	95.19±0.51	78.17±0.45
Shuffle	×	✓	✓	shuffled	–	76.47±0.40
Reverse	×	✓	✓	reversed	–	78.17±0.25
Full SNTM	✓	✓	✓	adjacent	**95.47±0.30**	**78.40±0.30**

Table 3: Parameters and MACs on CIFAR10-sized inputs (32×32, $T = 4$). Parameter totals are rounded to 0.01M; EDVM's three learned scalars per enabled block fall below this precision.

Method	Params (M)	MACs (G)
Spikformer (Zhou et al., 2022)	9.33	359.09
+ SNTM	9.33	359.18
SDT (Yao et al., 2023)	10.23	316.17
+ STAtten (Lee et al., 2025)	10.23	318.31
+ SNTM	10.23	316.19
SpikingResformer (Shi et al., 2024)	10.79	614.79
+ SNTM	10.79	614.88
QKFormer (Zhou et al., 2024)	6.71	707.42
+ SNTM	6.71	707.45

$T = 4$ simulation steps, so timestep-dependent features can arise from the internal state evolution of spiking neurons. Event-based datasets use $T = 16$, with CIFAR10-DVS and N-Caltech101 processed at 64×64 resolution and DVS128 Gesture at 128×128.

To examine whether SNTM depends on a particular attention formulation, we integrate it into four representative spiking Transformer backbones: SDT (Yao et al., 2023), QKFormer (Zhou et al., 2024), SpikingResformer (Shi et al., 2024), and Spikformer (Zhou et al., 2022). MT-SPS is applied once at the earliest spike projection, while EDVM is inserted into the value/content pathway of the selected attention blocks. Backbone depth, width, timestep configuration, and query-key computation are otherwise unchanged. Backbone-specific insertion locations, optimization settings, and preprocessing details are provided in Appendix B and C.

For the five smaller benchmarks, we use SDT as the primary controlled setting, where the baseline, STAtten, and SNTM are trained within the same codebase under matched training settings whenever the corresponding variant is available. For the other three backbones on these benchmarks, baseline and STAtten entries marked with † in Table 1 are taken from the corresponding papers, while the SNTM models are newly trained. These experiments assess the applicability of SNTM across different spiking attention formulations. All newly trained results on the five smaller benchmarks are averaged over three runs. Table 1 reports the means, while standard deviations and individual runs are provided in Appendices D and E.

For ImageNet, we evaluate SNTM with SDT, QKFormer, and Spikformer to assess its applicability at a larger data scale. The SDT-based experiments follow the training configuration of STAtten (Lee et al., 2025), whereas QKFormer and Spikformer follow the training configurations of their respective backbones. We retain the corresponding backbone-specific training and evaluation settings when incorporating SNTM. Results taken from prior publications are marked with †, while our newly trained ImageNet results are reported as single runs and marked with ‡.

4.2 Main Results

Controlled comparison on SDT. The SDT experiments provide the most direct comparison because the baseline and temporal variants share the same training pipeline. As shown in Table 1, SNTM improves the SDT baseline across all five smaller benchmarks. On the event-based benchmarks, CIFAR10-DVS increases from 76.77 to 78.40, while N-Caltech101 improves from 80.88 to 82.38. DVS128 Gesture increases from 96.87 to 97.21. SNTM also exceeds SDT+STAtten on all datasets for which both variants are available. The same trend is observed on static-image inputs. CIFAR100 improves from 77.82 to 78.85, and CIFAR10 from 95.14 to 95.47. Because the external image is identical at every simulation step, these improvements are consistent with SNTM exploiting timestep-dependent features generated by intrinsic neuronal dynamics, without requiring externally varying event inputs.

Architectural generality. SNTM also produces positive accuracy changes across the other three attention formulations. Relative to their reported native baselines, QKFormer improves by 1.11 points on CIFAR10, 1.37 on CIFAR100, 1.13 on CIFAR10-DVS, and 0.84 on N-Caltech101. SpikingResformer gains 1.63 points on CIFAR100 and 2.40 points on CIFAR10-DVS, while Spikformer gains 2.23 points on CIFAR100. Improvements are also observed on DVS128 Gesture. The magnitude of the gain varies across architectures and datasets, but the consistent direction supports the architectural generality of SNTM across substantially different spiking attention mechanisms. In particular, EDVM can be inserted into their native value/content pathways without redefining the corresponding query-key interactions.

Scaling to ImageNet. The ImageNet experiments evaluate whether the same temporal bias remains effective beyond small-scale benchmarks. SNTM improves SDT from 74.57% to 76.32%, QKFormer from 82.04% to 83.50%, and Spikformer from 73.38% to 75.45%, corresponding to gains of 1.75, 1.46, and 2.07 accuracy points, respectively. These results provide positive evidence that SNTM remains effective at ImageNet scale across three distinct spiking Transformer architectures.

4.3 Mechanism Analysis

Table 2 examines both component contributions and the temporal signals used by EDVM. All trained variants use SDT and are evaluated with three random seeds. Shuffle and reverse are test-time interventions evaluated on the three corresponding complete-EDVM checkpoints. MT-SPS is disabled in the Spike-only, Difference-only, shuffle, and reverse variants so that EDVM can be analyzed independently.

Component contributions. On CIFAR10-DVS, MT-SPS improves the SDT baseline from $76.77_{\pm 0.46}$ to $77.70_{\pm 0.46}$, while EDVM alone reaches $78.23_{\pm 0.32}$. Combining the two components gives the highest mean accuracy of $78.40_{\pm 0.30}$, although the margin over EDVM alone is modest. On CIFAR10, MT-SPS and EDVM reach $95.20_{\pm 0.14}$ and $95.32_{\pm 0.11}$, respectively, and the full SNTM reaches $95.47_{\pm 0.30}$. The larger gains on CIFAR10-DVS are consistent with the richer temporal variation available in event streams, whereas static inputs rely only on dynamics generated internally by the spiking neurons.

Source of the EDVM signal. We next separate the two terms in the event descriptor of Eq. (6). Spike-only retains the current spike response $\mathbf{S}_t$ while removing the adjacent difference, whereas Difference-only retains $\Delta\mathbf{V}_t$ without the current-response term. On CIFAR10-DVS, Spike-only reaches $77.53_{\pm 0.21}$, while Difference-only reaches $78.17_{\pm 0.45}$, only 0.06 points below complete EDVM at $78.23_{\pm 0.32}$. The adjacent value difference is therefore the dominant measured signal in this setting. Complete EDVM retains the current spike response as an additional content-dependent cue and achieves the highest mean, although its 0.06-point advantage over Difference-only is small relative to the observed run-to-run variation. On CIFAR10, the three variants are more closely grouped: Spike-only, Difference-only, and complete EDVM achieve $95.21_{\pm 0.05}$, $95.19_{\pm 0.51}$, and $95.32_{\pm 0.11}$, respectively. This weaker separation is consistent with the smaller temporal variation generated from repeated static inputs.

Temporal adjacency and direction. To test whether EDVM depends on the correct local temporal relationship, we modify only the temporal reference used in Eq. (5) at test time, while keeping the original value sequence and backbone states unchanged. For Shuffle, the reference feature paired with $\mathbf{V}_t$ is randomly sampled from another timestep instead of using the adjacent predecessor $\mathbf{V}_{t-1}$. For Reverse, the reference is changed from the previous timestep $\mathbf{V}_{t-1}$ to the next timestep $\mathbf{V}_{t+1}$ whenever available, with the boundary handled accordingly. Shuffling reduces CIFAR10-DVS accuracy from 78.23$\pm$0.32 to 76.47$\pm$0.40, showing that correctly paired adjacent features are important for the observed gain. Reversing the reference direction yields 78.17$\pm$0.25, close to the original EDVM result. This behavior is consistent with the absolute difference in Eq. (5), which captures the magnitude of local temporal variation without explicitly encoding its direction. Together, these controls support the interpretation of EDVM as a local temporal-contrast mechanism rather than a directional or long-range temporal model.

4.4 Efficiency Analysis

Table 3 reports implementation-level parameter counts and MACs. Across all four backbones, SNTM introduces only marginal computational overhead. The MAC increase ranges from 0.02G to 0.09G, while the rounded parameter totals remain unchanged. For example, SDT increases from 316.17G to 316.19G MACs with SNTM, compared with 318.31G for SDT+STAtten. These measurements are consistent with the $\mathcal{O}(TND)$ overhead derived in the Method section. EDVM adds only three learned scalars (a, η, τ) per enabled attention block, while MT-SPS reuses the original feature projection with fixed firing thresholds. The additional parameter count is therefore negligible at the 0.01M reporting precision. Overall, SNTM provides temporal modeling with minimal parameter and arithmetic overhead across the evaluated backbones. Since practical deployment cost also depends on factors such as activation precision and hardware implementation, we report MACs here as an implementation-level measure of computational overhead.

4.5 Limitations

SNTM focuses on local, first-order temporal variation, while longer-range and direction-sensitive temporal structures remain to be explored. Our ImageNet results are based on single runs, and the practical efficiency of MT-SPS may depend on deployment-specific factors such as activation precision and hardware implementation. The fixed threshold set and backbone-specific insertion locations used in this work may also require further adaptation for substantially different spiking architectures or longer-horizon tasks.

5 Conclusion

We revisited temporal modeling in spiking Transformers by treating intrinsic neuronal dynamics as a temporal inductive bias. Based on this perspective, we proposed SNN-native Temporal Modeling (SNTM), which incorporates temporal information at both the initial spike-representation stage and the attention-based feature-aggregation stage through MT-SPS and EDVM. Across four spiking Transformer backbones and six static-image and event-based benchmarks, SNTM consistently improves the corresponding baselines with limited computational overhead, including gains of 1.75, 1.46, and 2.07 accuracy points on ImageNet for SDT, QKFormer, and Spikformer, respectively. Controlled mechanism analyses further show that the effectiveness of EDVM depends strongly on correctly paired local temporal variations, supporting the role of neuronal dynamics as a useful temporal signal. These results suggest that temporal modeling in spiking Transformers need not rely exclusively on explicit cross-time attention, and that the intrinsic dynamics of spiking neurons provide a complementary and lightweight source of temporal structure.

AI Use Statement

Generative AI tools were used to assist with language editing and presentation, including improvements to grammar, clarity, concision, and organization. They were not used to formulate the research question, design the proposed method, conduct experiments, analyze experimental results, or determine the scientific conclusions of this work. All AI-assisted revisions were reviewed and, where

necessary, modified by the authors to ensure consistency with the intended technical content, experimental evidence, and cited literature. The authors take full responsibility for the final content of the paper.

Reproducibility Statement

The main paper describes the formulation of SNTM, its integration into the evaluated spiking Transformer backbones, and the experimental protocol. The appendices provide the exact MT-SPS thresholds, EDVM hyperparameters and parameter initializations, backbone-specific insertion locations, dataset and training configurations, hardware settings, result provenance, individual-run results, and the procedures used to reproduce the reported experiments. Together, these materials provide the implementation and evaluation details necessary to reproduce the results presented in this work.

References

Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, 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.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems*, 2018.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is Space-Time attention all you need for video understanding? In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, Yuyun Liao, Chit-Kwan Lin, Andrew Lines, Ruokun Liu, Deepak Mathaikutty, Steven McCoy, Arnab Paul, Jonathan Tse, Guruguhanathan Venkataramanan, Yi-Hsin Weng, Andreas Wild, Yoonseok Yang, and Hong Wang. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022.

Jianchuan Ding, Bo Dong, Felix Heide, Yufei Ding, Yunduo Zhou, Baocai Yin, and Xin Yang. Biologically inspired dynamic thresholds for spiking neural networks. *Advances in Neural Information Processing Systems*, 2022.

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

Haowen Fang, Amar Shrestha, Ziyi Zhao, and Qinru Qiu. Exploiting neuron and synapse filter dynamics in spatial temporal learning of deep spiking neural network. *arXiv preprint arXiv:2003.02944*, 2020.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in neural information processing systems*, 2021a.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021b.

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

Takaya Hirano, Kyo Kutsuzawa, Dai Owaki, and Mitsuhiro Hayashibe. Enhancing time scalability of spiking neural networks with dynamic time constants. *Scientific Reports*, 2026.

Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing spiking neural networks toward deep residual learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

Youngeun Kim, Yuhang Li, Hyoungseob Park, Yeshwanth Venkatesha, and Priyadarshini Panda. Neural architecture search for spiking neural networks. In *European conference on computer vision*, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Donghyun Lee, Ruokai Yin, Youngeun Kim, Abhishek Moitra, Yuhang Li, and Priyadarshini Panda. Tt-snn: tensor train decomposition for efficient spiking neural network training. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2024.

Donghyun Lee, Yuhang Li, Youngeun Kim, Shiting Xiao, and Priyadarshini Panda. Spiking transformer with spatial-temporal attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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

Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in neural information processing systems*, 2021.

Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks. In *European Conference on Computer Vision*, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 1997.

Wolfgang Maass. On the relevance of time in neural computation and learning. *Theoretical Computer Science*, 2001.

Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Training high-performance low-latency spiking neural networks by differentiation on spike representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

Paul A. Merolla, John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy, Jun Sawada, Filipp Akopyan, Bryan L. Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, Bernard Brezzo, Ivan Vo, Steven K. Esser, Rathinakumar Appuswamy, Brian Taba, Arnon Amir, Myron D. Flickner, William P. Risk, Rajit Manohar, and Dharmendra S. Modha. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 2014.

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

Nitin Rathi and Kaushik Roy. Diet-snn: Direct input encoding with leakage and threshold optimization in deep spiking neural networks. *arXiv preprint arXiv:2008.03658*, 2020.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 2019.

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*, 2024.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Xiyan Wu, Yong Song, Ya Zhou, Yurong Jiang, Yashuo Bai, Xinyi Li, and Xin Yang. Stca-snn: self-attention-based temporal-channel joint attention for spiking neural networks. *Frontiers in Neuroscience*, 2023.

Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *Advances in neural information processing systems*, 2023.

Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2021.

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 qk attention. *Advances in Neural Information Processing Systems*, 37:13074–13098, 2024.

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

Technical Appendices

A Exact Instantiation of SNTM

This section specifies the instantiation used in all experiments. It follows the notation of the main paper and records the fixed hyperparameters and parameter initializations needed for reproduction.

A.1 Multi-threshold Spiking Patch Splitting

Given a projected feature tensor $\mathbf{Z} \in \mathbb{R}^{T\times B\times C\times H\times W}$, MT-SPS evaluates the same feature using M spiking branches with different firing thresholds:

$$\mathbf{S}^{(m)} = \mathrm{LIF}(\mathbf{Z}; \theta_m), \qquad m = 1, \dots, M, \tag{12}$$

and aggregates the responses by

$$\mathbf{S}_{\mathrm{MT}} = \sum_{m=1}^{M} \mathbf{S}^{(m)}. \tag{13}$$

We use $M = 3$ and $\theta_m \in \{1.0, 1.5, 2.5\}$. All branches share the same upstream convolutional or linear projection and differ only in their fixed firing thresholds. Their remaining neuron settings, including the membrane time constant and reset behavior, follow the defaults of the corresponding backbone. Consequently, MT-SPS introduces no branch-specific learned projection parameters. Its summed output lies in $\{0, \dots, M\}$ and is used only at the earliest spike-projection stage.

A.2 Event-driven Value Modulation

Let $\mathbf{V}_t$ be the native value/content feature and $\mathbf{S}_t$ the associated spike response at timestep t. EDVM first computes

$$\Delta\mathbf{V}_t = \begin{cases} \mathbf{0}, & t = 1, \\ |\mathbf{V}_t - \mathbf{V}_{t-1}|, & t > 1, \end{cases} \tag{14}$$

followed by the bounded event descriptor and event-enhanced value

$$\mathbf{E}_t = \lambda \tanh(\mathbf{S}_t + \Delta\mathbf{V}_t), \qquad \mathbf{V}^e_t = \mathbf{V}_t \odot (1 + \mathbf{E}_t). \tag{15}$$

We set $\lambda = 1.5$. For each sample b, the normalized temporal-variation magnitude is

$$r_b = \frac{\mathrm{Mean}(\Delta\mathbf{V}_{:,b})}{\mathrm{Mean}(|\mathbf{V}_{:,b}|) + \epsilon}, \qquad \epsilon = 10^{-6}. \tag{16}$$

The residual strength and sample-wise gate are parameterized as

$$\alpha = \sigma(a), \qquad \gamma = \mathrm{softplus}(\eta), \qquad g_b = \sigma(\gamma(r_b - \tau)), \tag{17}$$

and the final value is

$$\widetilde{\mathbf{V}}_{:,b} = \mathbf{V}_{:,b} + \alpha g_b \left(\mathbf{V}^e_{:,b} - \mathbf{V}_{:,b}\right). \tag{18}$$

Thus, EDVM uses exactly three learned scalars (a, η, τ) in each enabled attention block. We initialize them as

$$a = -0.5, \qquad \eta = 2.0, \qquad \tau = 0.15. \tag{19}$$

The sample gate g_b is broadcast over timesteps, tokens or spatial locations, and channels. All other EDVM operations are elementwise and shape preserving.

B Backbone Integration

We modify only the modules required by SNTM. MT-SPS is applied once at the earliest spike-projection stage, and EDVM is inserted into the native value/content pathway of the selected attention blocks. The query–key computation, backbone depth and width, and timestep configuration are otherwise unchanged. The backbone-specific mappings used in our implementations are summarized in Table 4.

Table 4: Backbone-specific integration of SNTM.

Backbone	**EDVM insertion**	**MT-SPS insertion**	**Implementation note**
SDT (Yao et al., 2023)	Value branch of the SDSA block before native spike-driven attention	First spike projection in the SPS stem	EDVM is enabled in the selected later attention blocks; the native query–key interaction is unchanged.
Spikformer (Zhou et al., 2022)	Value tokens before the native $Q(K^\top V)$ spike attention	First spike projection in the SPS stem	Multi-threshold encoding precedes subsequent token aggregation.
QKFormer (Zhou et al., 2024)	Value branch of the spiking self-attention block	First spike projection in the patch stem	EDVM remains inside the attention block, while MT-SPS is confined to the stem.
SpikingResformer (Shi et al., 2024)	Content/value tensor $\mathbf{Y}_2$ inside DSSA before multiplication with the attention map	Prologue LIF in the convolutional stem	EDVM modulates the DSSA content/value branch without redefining its attention map.

B.1 Compatibility with Different Attention Operators

EDVM is a value-modulation strategy rather than a replacement for the native attention operator. Formally, if a backbone computes $\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, SNTM supplies the modulated value and retains the original interaction rule:

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \widetilde{\mathbf{V}}). \tag{20}$$

For SDT (Yao et al., 2023), EDVM is applied before the native key–value aggregation. For Spikformer and QKFormer (Zhou et al., 2022; 2024), it is applied before the value interaction with the query–key pathway. For SpikingResformer (Shi et al., 2024), it modulates the DSSA content/value tensor before that tensor interacts with the attention map. In every case, the query and key branches are unchanged. This common interface allows EDVM to be inserted without redefining the backbone-specific attention formulation.

B.2 Temporal Dynamics under Static Inputs

For static-image experiments, the same image is repeated at every simulation step, but the internal neuronal responses are not generally constant. Membrane integration, thresholding, and reset produce timestep-dependent states and spike patterns even under identical external inputs. Consequently, $\mathbf{V}_t$ can vary with t and $\Delta\mathbf{V}_t$ can be nonzero after the first step. EDVM therefore measures and reweights temporal variation generated internally by the spiking computation rather than requiring external motion. MT-SPS complements this process by exposing the shared projected feature to several firing regimes. When normalized variation is weak, the sample-wise gate reduces the contribution of the event-enhanced residual.

C Training and Data Details

We follow the original training configuration of each backbone and modify only the SNTM components described above. For the controlled SDT comparison, SDT, SDT+STAtten, and SDT+SNTM are trained in the same codebase under matched settings. The SNTM variants of QKFormer, SpikingResformer, and Spikformer are newly trained using the training configurations of their respective backbones, whereas baseline or STAtten results marked with † are quoted from the corresponding publications. ImageNet results marked with ‡ are our single runs.

For CIFAR10 and CIFAR100, static images are repeated for $T = 4$ simulation steps. CIFAR10-DVS and N-Caltech101 use $T = 16$ with 64×64 inputs, while DVS128 Gesture uses $T = 16$ with 128×128 inputs. Data augmentation and preprocessing follow the corresponding backbone implementations.

Table 5 summarizes the general training configuration used for the SDT experiments. For QKFormer, SpikingResformer, and Spikformer, we follow the training settings of their original backbone implementations for fair comparison. The five smaller-benchmark experiments were conducted on NVIDIA RTX 4090 GPUs. ImageNet experiments were conducted on NVIDIA A100 GPUs. For SDT on ImageNet, we follow the training protocol of STAtten (Lee et al., 2025), using a global batch size of 32, a learning rate of 10^{-3}, and LAMB optimization for 210 epochs, including 10 cooldown epochs. QKFormer and Spikformer follow the ImageNet training configurations of their respective backbones.

Table 5: General training configuration for the SDT experiments. Other backbones follow their original training settings.

	CIFAR10/100	**Event datasets**	**ImageNet**
Timestep	4	16	4
Batch size	64	16	32
Learning rate	3×10^{-4}	10^{-2}	10^{-3}
Training epochs	210	210	210
Optimizer	AdamW	AdamW	LAMB
Hardware	RTX 4090	RTX 4090	A100

D Full Experimental Results

The compact table in the main paper emphasizes the controlled backbone-level comparison and therefore shows only three-run means on the five smaller benchmarks. Here we provide the complete results with uncertainty estimates and additional prior methods. Unless marked otherwise, values in Tables 6 and 7 are our mean$\pm$sample standard deviation over three runs. A dagger (†) marks a result quoted from the cited paper rather than rerun by us. A double dagger (‡) in Table 8 marks one of our single-run ImageNet results. Reported prior-work rows are included as broader context; because their training recipes and architectures can differ, the within-SDT reruns remain the primary controlled comparison.

Table 6: Top-1 accuracy (%) on CIFAR10 and CIFAR100.

Method	Architecture	Timestep	CIFAR10	CIFAR100
tdBN (Zheng et al., 2021)	ResNet19	4	92.92†	–
PLIF (Fang et al., 2021b)	ConvNet	8	93.50†	–
Dspike (Li et al., 2021)	ResNet18	6	94.25†	74.24†
DSR (Meng et al., 2022)	ResNet18	20	95.40†	78.50†
DIET-SNN (Rathi & Roy, 2020)	VGG16	5	92.70†	69.67†
SNASNet (Kim et al., 2022)	ConvNet	5	93.64†	73.04†
Spikformer (Zhou et al., 2022)	Spikformer-4-384	4	93.99†	75.06†
+ STAtten (Lee et al., 2025)	Spikformer-4-384	4	94.36†	75.85†
+ SNTM (ours)	Spikformer-4-384	4	**94.58**±0.04	**77.29**±0.25
SDT (reported) (Yao et al., 2023)	SDT-2-512	4	95.34†	78.32†
SDT (our rerun)	SDT-2-512	4	95.14±0.21	77.82±0.44
+ STAtten (Lee et al., 2025)	SDT-2-512	4	95.08±0.15	78.43±0.42
+ SNTM (ours)	SDT-2-512	4	**95.47**±0.30	**78.85**±0.03
SpikingResformer (Shi et al., 2024)	SpikingResformer-6-384	4	95.03†	77.16†
+ STAtten (Lee et al., 2025)	SpikingResformer-6-384	4	95.26†	77.90†
+ SNTM (ours)	SpikingResformer-6-384	4	**96.13**±0.08	**78.79**±0.32
QKFormer (Zhou et al., 2024)	QKFormer-4-384	4	95.12†	79.79†
+ STAtten (Lee et al., 2025)	QKFormer-4-384	4	95.35†	80.20†
+ SNTM (ours)	QKFormer-4-384	4	**96.23**±0.11	**81.16**±0.08

Table 7: Top-1 accuracy (%) on event-based benchmarks.

Method	Architecture	Timestep	CIFAR10-DVS	N-Caltech101	Gesture
tdBN (Zheng et al., 2021)	ResNet19	10	67.80†	–	96.90†
PLIF (Fang et al., 2021b)	ConvNet	20	74.80†	–	97.60†
Dspike (Li et al., 2021)	ResNet18	10	75.40†	–	–
DSR (Meng et al., 2022)	ResNet18	20	77.27†	–	–
SEW-ResNet (Fang et al., 2021a)	ConvNet	20	74.80†	–	–
TT-SNN (Lee et al., 2024)	ResNet34	6	–	77.80†	–
NDA (Li et al., 2022)	VGG11	10	79.60†	78.20†	–
Spikformer (Zhou et al., 2022)	Spikformer-2-256	16	80.90†	–	98.30†
+ SNTM (ours)	Spikformer-2-256	16	**81.20**±0.46	**81.30**±0.41	**98.60**±0.10
SDT (reported) (Yao et al., 2023)	SDT-2-256	16	77.30†	81.72†	99.30†
SDT (our rerun)	SDT-2-256	16	76.77±0.46	80.88±0.73	96.87±0.60
+ STAtten (Lee et al., 2025)	SDT-2-256	16	77.27±0.29	81.00±0.35	–
+ SNTM (ours)	SDT-2-256	16	**78.40**±0.30	**82.38**±0.79	**97.21**±0.60
SpikingResformer (Shi et al., 2024)	SpikingResformer-4-384	16	78.80†	81.29†	98.60†
+ STAtten (Lee et al., 2025)	SpikingResformer-4-384	16	80.60†	81.95†	–
+ SNTM (ours)	SpikingResformer-4-384	16	**81.20**±0.44	**82.15**±0.22	**98.70**±0.10
QKFormer (Zhou et al., 2024)	QKFormer-4-384	16	82.90†	83.58†	98.60†
+ STAtten (Lee et al., 2025)	QKFormer-4-384	16	83.90†	84.25†	–
+ SNTM (ours)	QKFormer-4-384	16	**84.03**±0.12	**84.42**±0.43	**98.97**±0.12

Table 8: Top-1 accuracy (%) on ImageNet.

Method	**Architecture**	**Timestep**	**Top-1 accuracy (%)**
Vision Transformer (Dosovitskiy et al., 2020)	ViT-B/16	1	77.90†
DeiT (Touvron et al., 2021)	DeiT-B	1	81.80†
Swin Transformer (Liu et al., 2021)	Swin Transformer-B	1	83.50†
TET (Deng et al., 2022)	SEW-ResNet34	4	68.00†
Spiking ResNet (Hu et al., 2021)	ResNet50	350	72.75†
tdBN (Zheng et al., 2021)	ResNet34	6	63.72†
SEW-ResNet (Fang et al., 2021a)	SEW-ResNet152	4	69.26†
MS-ResNet (Hu et al., 2024)	ResNet104	5	74.21†
Spikformer (Zhou et al., 2022)	Spikformer-8-512	4	73.38†
+ SNTM (ours)	Spikformer-8-512	4	**75.45**‡
SDT (Yao et al., 2023)	SDT-8-512	4	74.57†
+ STAtten (Lee et al., 2025)	SDT-8-512	4	76.18‡
+ SNTM (ours)	SDT-8-512	4	**76.32**‡
QKFormer (Zhou et al., 2024)	HST-10-512	4	82.04†
+ SNTM (ours)	HST-10-512	4	**83.50**‡

E INDIVIDUAL RUNS

Table 9 lists the three observations underlying every mean and sample standard deviation reported for our backbone comparisons on the five smaller benchmarks. A dash indicates that the corresponding experiment was not run.

Table 9: Top-1 accuracies (%) for individual runs (run 1 / run 2 / run 3) of our backbone experiments.

Method	CIFAR10	CIFAR100	CIFAR10-DVS	N-Caltech101	Gesture
SDT (our rerun)	95.34 / 95.15 / 94.92	78.32 / 77.50 / 77.64	77.30 / 76.50 / 76.50	81.72 / 80.41 / 80.52	97.56 / 96.52 / 96.52
SDT + STAtten	95.18 / 95.15 / 94.91	78.67 / 78.67 / 77.95	77.60 / 77.10 / 77.10	81.40 / 80.85 / 80.74	–
SDT + SNTM	95.47 / 95.77 / 95.18	78.87 / 78.87 / 78.82	78.70 / 78.10 / 78.40	83.04 / 82.60 / 81.50	97.56 / 96.52 / 97.56
QKFormer + SNTM	96.34 / 96.12 / 96.24	81.13 / 81.25 / 81.11	84.10 / 84.10 / 83.90	84.90 / 84.30 / 84.07	98.90 / 99.10 / 98.90
SpikingResformer + SNTM	96.21 / 96.14 / 96.05	78.42 / 79.02 / 78.93	81.70 / 80.90 / 81.00	82.30 / 82.25 / 81.90	98.70 / 98.80 / 98.60
Spikformer + SNTM	94.61 / 94.59 / 94.54	77.13 / 77.58 / 77.15	81.10 / 81.70 / 80.80	81.66 / 80.85 / 81.40	98.60 / 98.70 / 98.50

Table 10 provides the corresponding observations for the component and mechanism analysis. Shuffle and reverse are test-time interventions applied to the three complete-EDVM checkpoints; consequently, their entries correspond to the same three trained models under different temporal-reference rules.

Table 10: Top-1 accuracies (%) for individual runs (run 1 / run 2 / run 3) of the SDT ablations.

Variant	**CIFAR10**	**CIFAR10-DVS**
SDT	95.34 / 95.15 / 94.92	77.30 / 76.50 / 76.50
+ MT-SPS	95.35 / 95.17 / 95.08	78.10 / 77.80 / 77.20
+ EDVM	95.43 / 95.31 / 95.22	78.60 / 78.10 / 78.00
Spike-only	95.22 / 95.25 / 95.16	77.60 / 77.70 / 77.30
Difference-only	95.66 / 94.65 / 95.25	78.20 / 78.60 / 77.70
Shuffle	–	76.40 / 76.90 / 76.10
Reverse	–	78.40 / 77.90 / 78.20
Full SNTM	95.47 / 95.77 / 95.18	78.70 / 78.10 / 78.40