TDFormer: A Top-Down Attention-Controlled Spiking Transformer

Anonymous Author(s)
Affiliation
Address
email

Abstract

Traditional spiking neural networks (SNNs) can be viewed as a combination of multiple subnetworks with each running for one time step, where the parameters are shared, and the membrane potential serves as the only information link between them. However, the implicit nature of the membrane potential limits its ability to effectively represent temporal information. As a result, each time step cannot fully leverage information from previous time steps, seriously limiting the model's performance. Inspired by the top-down mechanism in the brain, we introduce TDFormer, a novel model with a top-down feedback structure that functions hierarchically and leverages high-order representations from earlier time steps to modulate the processing of low-order information at later stages. The feedback structure plays a role from two perspectives: 1) During forward propagation, our model increases the mutual information across time steps, indicating that richer temporal information is being transmitted and integrated in different time steps. 2) During backward propagation, we theoretically prove that the feedback structure alleviates the problem of vanishing gradients along the time dimension. We find that these mechanisms together significantly and consistently improve the model performance on multiple datasets. In particular, our model achieves state-of-the-art performance on ImageNet with an accuracy of 86.83%.

1 Introduction

Spiking Neural Networks (SNNs) are more energy-efficient and biologically plausible than traditional artificial neural networks (ANNs) [1]. Transformer-based SNNs combine the architectural advantages of Transformers with the energy efficiency of SNNs, resulting in a powerful and efficient models that have attracted increasing research interest in recent years [2, 3, 4, 5, 6]. However, there is still a big performance gap between existing SNNs and ANNs. This is because SNNs represent information using binary spike activations, whereas ANNs use floating-point numbers, resulting in reduced representational capacity and degraded performance. Moreover, the non-differentiability of spikes hinders effective training with gradient-based methods.

In traditional SNNs, a common approach to increase representational capacity is to expand the time step T. However, SNNs trained with direct coding and standard learning methods [7] lack structural mechanisms for temporal adaptation. Temporal information is solely conveyed through membrane potential dynamics, while the network architecture, parameters, and inputs remain fixed across time steps. This reliance on membrane dynamics imposes two fundamental limitations. First, temporal information can only be expressed when spikes are fired, yet firing rates are typically low across layers, restricting the bandwidth of information flow. Moreover, the cumulative nature of membrane potentials leads to loss of temporal detail, as earlier spike patterns are summed. Second, temporal gradients must propagate solely through membrane potentials, which can result in vanishing

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

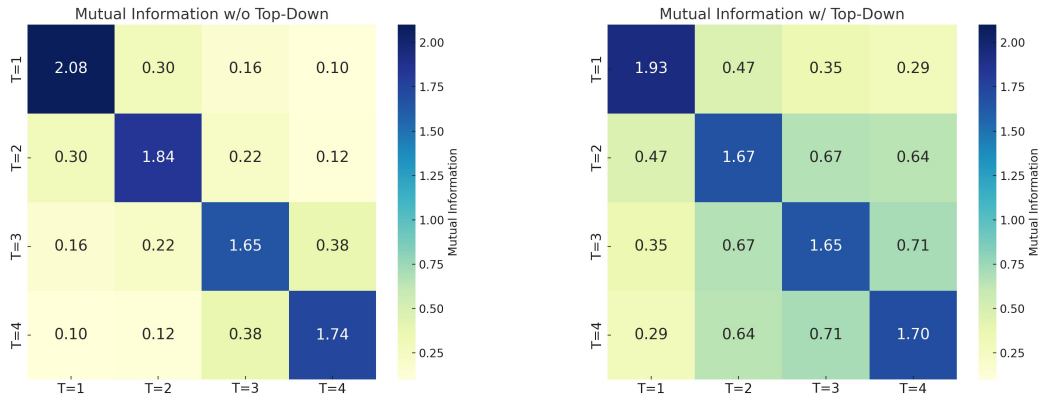

Figure 1: Visualization of mutual information matrices of features across time steps on ImageNet. The left panel shows the baseline model; the right panel shows the model incorporating feedback connections. A higher level of mutual information suggests that the model captures more consistent and temporally dependent features across time steps

gradients[8, 9]. We further confirm these limitations through temporal correlation analysis shown in Figure 1, which demonstrates the limited representational capacity of membrane potentials, and theoretical derivation in appendix B.3.

Previous work has been done to enhance the ability of SNNs to represent temporal information, e.g., by initializing the membrane potential and altering the surrogate gradients and dynamics equations [10, 11, 12]. Furthermore, some approaches have incorporated the dimension of time into attention mechanisms, resulting in time complexity that scales linearly with the number of simulation time steps [13]. However, structural mechanisms to facilitate information flow across multiple time steps remain largely unexplored. We argue that adding connections between different time steps has the following two benefits: First, in forward propagation, such connections help the model better leverage features from previous time steps. Second, in backpropagation, structural connections support gradient flow and help mitigate vanishing gradients caused by the membrane potential dynamics.

While traditional SNNs rely on bottom-up signal propagation, top-down mechanisms are prevalent in the brain, especially between the prefrontal and visual cortices [14, 15, 16, 17], as shown in Figure 2. These mechanisms are fundamental to how the brain incrementally acquires visual information over time, with higher-level cognitive processes guiding the extraction of lower-level sensory features, and prior knowledge informing the interpretation and refinement of new sensory input. Inspired by top-down mechanisms, we introduce TDFormer, a Transformer-based SNN architecture that incorporates a top-down feedback structure to improve temporal information utilization. Our main contributions can be summarized as follows:

- We identify structural limitations in traditional SNNs, showing that features across time steps exhibit weak mutual information, indicating insufficient temporal integration and utilization.
- We propose TDFormer, a Transformer-based SNN with a novel top-down feedback structure. We show that the proposed structure improves temporal information utilization, and provide theoretical analysis showing it mitigates vanishing gradients along the temporal dimension.
- We demonstrate state-of-the-art performance across multiple benchmarks with minimal energy overhead, achieving ANN-level accuracy on ImageNet while preserving the efficiency of SNNs.

2 Related Works

2.1 Transformer-based SNNs

Spikformer [2] presented the first Transformer architecture based on SNNs, laying the groundwork for spike-based self-attention mechanisms. Spike-driven TransformerV1 [5] introduced a spike-driven

mechanism to effectively process discrete-time spike signals and employed stacked transformer layers to capture complex spatiotemporal features. Built on [5], Spike-driven TransformerV2 [6] enhanced the spike-driven mechanism and added dynamic weight adjustment to improve adaptability and accuracy in processing spike data. SpikformerV2 [18] was specifically optimized for high-resolution image recognition tasks, incorporating an improved spike encoding method and a multi-layer self-attention mechanism. SpikeGPT [19] proposed an innovative combination of generative pre-trained Transformers with SNNs. SGLFormer [20] enhanced feature representations by effectively capturing both global context and local details.

2.2 Models with Top-Down Mechanisms

Unlike bottom-up processes that are driven by sensory stimuli, top-down attention is governed by higher cognitive processes such as goals, previous experience, or prior knowledge[21]. This mechanism progressively acquires information by guiding the focus of attention to specific regions or features of the visual scene. It can be seen as a feedback loop where higher-level areas provide signals that modulate the processing of lower-level sensory inputs, ensuring that the most relevant information is prioritized.

Many works have explored top-down attention mechanisms to improve model performance in traditional ANNs. For example, Zheng et al. [21] proposed FBTP-NN, which integrates bottom-up and top-down pathways to enhance visual object recognition, where top-down expectations modulate neuron activity in lower layers [21]. Similarly, Anderson et al. introduced a model combining bottom-up and top-down attention for image captioning and visual question answering, where top-down attention weights features based on task context [22]. Shi et al. introduced a top-down mechanism for Visual Question Answering (VQA), where high-level cognitive hypotheses influence the focus on relevant scene parts [23]. Finally, Abel and Ullman proposed a network that combines back-propagation with top-down attention to adjust gradient distribution and focus on important features [24].

3 Preliminaries

3.1 The Spiking Neuron

The fundamental distinction between SNNs and ANNs lies in their neuronal activation mechanisms. Drawing on established research [2, 4, 5, 3], we select the Leaky Integrate-and-Fire (LIF) [25] neuron model as our primary spike activation unit. LIF neuron dynamics can be formulated by:

$$V[t] = H[t](1 - S[t]) + V_{\text{reset}}S[t], \tag{1}$$

$$H[t] = V[t-1] + \frac{1}{\tau}(X[t] - (V[t-1] - V_{\text{reset}})), \tag{2}$$

$$S[t] = \Theta(H[t] - V_{\text{th}}), \tag{3}$$

where V_{reset} is the reset potential. When a spike is generated, $S[t] = 1$, the membrane potential $V[t]$ is reset to V_{reset}; otherwise, it remains at the hidden membrane potential $H[t]$. Moreover, τ represents the membrane time constant, and the input current $X[t]$ is decay-integrated into $H[t]$.

3.2 Spike-Based Self-Attention Mechanisms

A critical challenge in designing spike-based self-attention is eliminating floating-point matrix multiplication in Vanilla Self-Attention (VSA) [26], which is crucial for utilizing the additive processing characteristics of SNNs.

Spiking Self-Attention (SSA) Zhou et al. [2] first leveraged spike dynamics to replace the softmax operation in VSA, thereby avoiding costly exponential and division calculations, and reducing energy consumption. The process of SSA is as follows:

$$I_s = \mathcal{SN}(BN(XW_I)), I \in \{Q, K, V\}, \tag{4}$$

$$\text{SSA}(Q_s, K_s, V_s) = \mathcal{SN}(Q_s K_s^\top V_s * s), \tag{5}$$

where $W \in \mathbb{R}^{T\times N\times D}$ denotes a learnable weight matrix, I_s represents the spiking representations of query Q_s, key K_s, and value V_s. Here, $\mathcal{SN}(\cdot)$ denotes the LIF neuron, and s is a scaling factor.

Spike-Driven Self-Attention (SDSA) Yao et al. [5, 6] improved the SSA mechanism by replacing the matrix multiplication with the Hadamard product and computing the attention via column-wise summation, effectively utilizing the additive properties of SNNs. The first version of SDSA [5] is as follows:

$$\mathrm{SDSA}_1(Q_s, K_s, V_s) = Q_s \otimes \mathcal{SN}(\mathrm{SUM}_c(K_s \otimes V_s)), \tag{6}$$

where $\otimes$ denotes the Hadamard product, $\mathrm{SUM}_c(\cdot)$ represents the column-wise summation. Furthermore, the second version of SDSA [6] is described as follows:

$$\mathrm{SDSA}_2(Q_s, K_s, V_s) = \mathcal{SN}_s((Q_s K_s^\top)V_s), \tag{7}$$

where $\mathcal{SN}_s$ denotes a spiking neuron with a threshold of $s \cdot V_{\text{th}}$. **Q-K Attention** (QKA) The work in [3] reduces the computational complexity from quadratic to linear by utilizing only the query and key. QKA can be further divided into two variants: Q-K Token Attention (QKTA) and Q-K Channel Attention (QKCA). The formulations for QKTA and QKCA are provided in Equations 8 and 9, respectively:

$$\mathrm{QKTA}(Q_s, K_s) = \mathcal{SN}(\sum_{i=0}^{D} Q_s(i,j)) \otimes K_s, \tag{8}$$

$$\mathrm{QKCA}(Q_s, K_s) = \mathcal{SN}(\sum_{j=0}^{N} Q_s(i,j)) \otimes K_s, \tag{9}$$

where N denotes the token number, D represents the channel number.

4 Method

In this section, we introduce TDFormer, a Transformer-based SNN model featuring a top-down feedback structure. We describe its architecture, including the division into sub-networks for feedback processing. We theoretically show that the attention module prior to the LIF neuron in the feedback pathway exhibits lower variance compared to SSA and QKTA, and we provide guidance for hyperparameter selection. Finally, we introduce the training loss and inference process. Detailed mathematical derivations are provided in appendix B.

4.1 TDFormer Architecture

This work is based on three backbones: SpikformerV1 [2], Spike-driven TransformerV1 [5] and QKformer [3]. These can be summarized into a unified structure, as shown in Figure 2, which consists of L_c Conv-based SNN blocks, L_t Transformer-based SNN blocks, and a classification head (CH). Additionally, the Transformer-based SNN blocks incorporate spike-based self-attention modules and Multi-Layer Perceptron (MLP) modules.

Apart from the backbone structure, the TDFormer architecture specifically introduces a top-down pathway called TDAC that includes two modules: the control module (CM) and the processing module (PM), as shown in Figure 2.

Viewing traditional SNNs as a sequence of $T = 1$ sub-networks with shared parameters and temporal dynamics governed by membrane potentials, we propose two approaches to introducing the top-down pathway. The first adds recurrent feedback connections between these fine-grained $T = 1$ sub-networks, enabling temporal context to propagate backward through time. The second adopts a coarser temporal resolution by dividing a sequence (e.g., $T = 4$) into fewer segments (e.g., two $T = 2$ blocks). Importantly, the additional power overhead introduced by both schemes remains minimal. Detailed analysis of power consumption is provided in appendix C.1. Both approaches can be expressed in the following unified formulation:

$$H_1 = \mathrm{F_{tr}}\left(\mathrm{CM}\left(S_{bu}^{(1)}, \varnothing\right)\right) \qquad H_1 \in \{0,1\}^{T\times N\times C}, S_{bu}^{(1)} \in \{0,1\}^{T\times H\times W\times C} \tag{10}$$

$$S_{td}^{(1)} = \mathrm{PM}(H_1) \qquad S_{td}^{(1)} \in \{0,1\}^{T\times N\times C}, H_1 \in \{0,1\}^{T\times N\times C} \tag{11}$$

$$H_n = \mathrm{F_{tr}}\left(\mathrm{CM}\left(S_{bu}^{(n)}, S_{td}^{(n-1)}\right)\right) \qquad S_{bu}^{(n)} \in \{0,1\}^{T\times H\times W\times C}, n = 1\ldots N \tag{12}$$

$$S_{td}^{(n)} = \mathrm{PM}(H_n) \qquad S_{td}^{(n)} \in \{0,1\}^{T\times N\times C}, n = 1\ldots N \tag{13}$$

$$O_n = \mathrm{CH}(H_n) \qquad O_n \in \{0,1\}^{T\times L}, H_n \in \{0,1\}^{T\times N\times C}, n = 1\ldots N \tag{14}$$

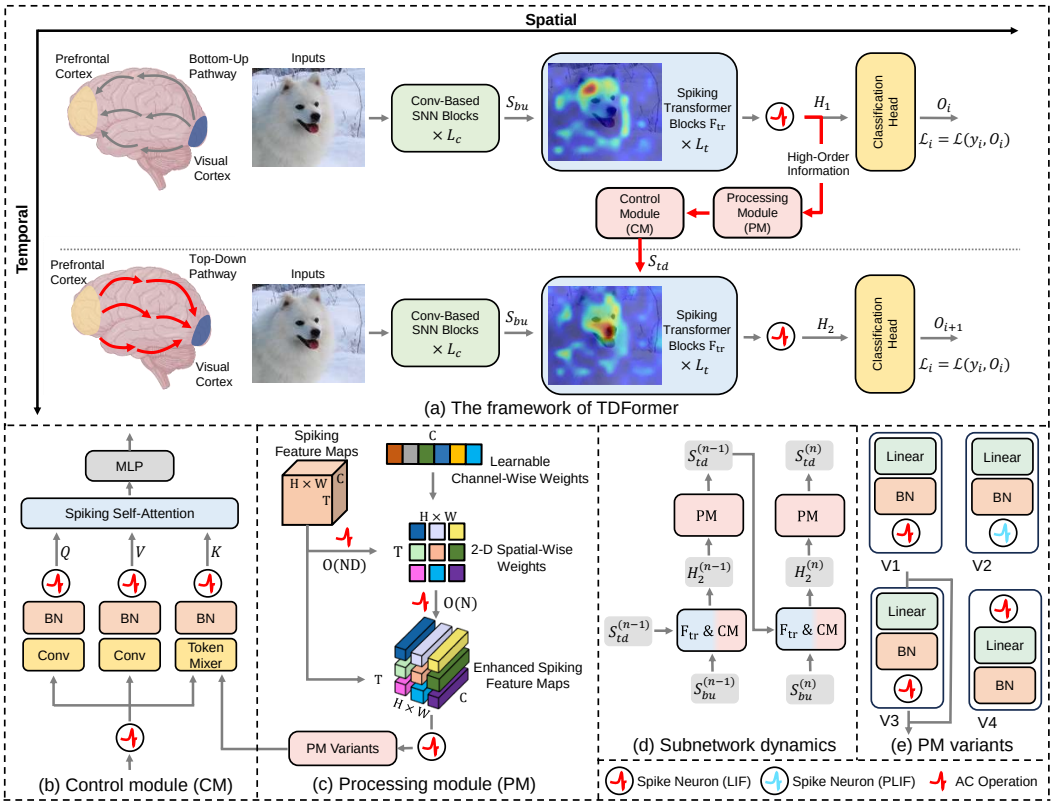

Figure 2: Overview of the TDFormer architecture. (a) Overall design inspired by top-down pathways in the brain, mimicking feedback from the prefrontal cortex to the visual cortex for temporal modulation in SNNs; (b) and (c) Detailed structures of the processing and control modules; (d) Information flow within the subnetwork, highlighting processing of feedback signals; (e) Four processing module variants, labeled v1–v4.

In the above formulation, $S_{bu}^{(n)}$ denotes the bottom-up input at time step n, while $S_{td}^{(n-1)}$ represents the top-down feedback from the previous step. CM is a control module that integrates bottom-up and top-down signals, and $\mathrm{F_{tr}}$ denotes the Transformer-based processing unit. The processing module PM generates the current feedback signal $S_{td}^{(n)}$ from the high-level representation H_n, and CH maps H_n to the final output O_n, where N denotes the number of sub-networks. The bottom part of Figure 2 illustrates the feedback information flow between sub-networks.

For the control module (CM), CM derives the query Q, key K, and value V vectors from the bottom-up information S_{bu} and the top-down information S_{td}. In more detail, S_{td} facilitates attention correction by controlling the attention map. The CM can be formulated as follows:

$$Q, K, V = CM(S_{bu}, S_{td}), \tag{15}$$

$$K = \mathcal{SN}(\mathrm{BN}(\mathrm{TokenMix}\,((S_{bu}, S_{td})))), \tag{16}$$

$$Q = \mathcal{SN}(\mathrm{BN}(\mathrm{Linear}(S_{bu}))), V = \mathcal{SN}(\mathrm{BN}(\mathrm{Linear}(S_{bu}))). \tag{17}$$

We choose concatenation along the channel dimension as the default token mixer, which allows us to combine the features of the current time step with those from previous time steps, and use the fused information to dynamically adjust the self-attention map. After passing through the CM, the query Q, key K and value V vectors are fed into the self-attention module to obtain the top-down attention map. To prevent the fusion of top-down information from altering the distribution of K in the self-attention computation, we first normalize the combined features, and then apply spike discretization before computing self-attention. Ablation studies on different CM variants are provided in the appendix C.2.

The processing module (PM) PM includes both channel-wise token mixer and spatial-wise token mixer [27]. The feature enhancement component enhances the original spiking feature maps $\mathbf{X}$ by learning channel-wise $\mathbf{W}_c$ and computing spatial-wise attention maps $\mathbf{M}_{\text{spatial}}$. This attention mechanism requires very few parameters and has a time complexity of $O(ND)$. This operation is represented as:

$$\mathbf{M}_{\text{spatial}}(t,n) = \sum_{c=1}^{C} \mathbf{W}_c \cdot \mathbf{X}_{t,n,c}, \tag{18}$$

$$\mathbf{M}_{\text{spatial}} = \text{clamp}\left(\mathbf{M}_{\text{spatial}}, b, a\right). \tag{19}$$

where $\mathbf{X}_{t,n,c}$ represents the spiking activation at time t, spatial position n (corresponding to the 2D coordinate (h, w) in the feature map), and channel c. Here, a and b are hyperparameters. We theoretically derive their effects on the PM output, and the details are given in appendix B.2. The spatial attention map $\mathbf{M}_{\text{spatial}}$ weights the spiking feature map $\mathbf{X}$ via element-wise multiplication, with broadcasting over the channel dimension:

$$\mathbf{O} = \mathcal{SN}(\mathbf{X} \odot \mathbf{M}_{\text{spatial}}). \tag{20}$$

The attention embedding spaces are different across layers, and we aim to use a PM variants to align the top-down information with the embedding spaces of different layers. We explored four PM variants that serve as the channel-wise token mixer, which are illustrated in Figure 2.

We introduce a clamp operation in the attention module to enforce a strict upper bound on the variance of the attention map which is formally stated in Proposition 4.1. Excessive variance can lead to gradient vanishing, as gradients in spiking neurons are only generated near the firing threshold of the membrane potential. Outside this narrow region, the gradient tends to vanish. Furthermore, high variance may introduce outliers, resulting in significant quantization errors during spike generation. The effect of the clamp operation on the gradient is shown in the Figure appendix C.2.

Proposition 4.1. *The upper bound $\overline{Var}(Y_{tnc})$ for the $\mathbf{X} \odot \mathbf{M}_{spatial}$ is given as follows:*

$$\overline{Var}(Y_{tnc}) = \begin{cases} a^2(f^2 - f + \frac{1}{2}) + ab(1-2f) + \frac{b^2}{2}, & \text{if } 0 \le f \le \frac{a+b}{2a}, \\ \frac{a^2+2ab+b^2-4fab}{4}, & \text{if } \frac{a+b}{2a} \le f \le 1, \end{cases} \tag{21}$$

where we assume each $\mathbf{X}_{t,n,c}$ is independent random variable $X_{tnc} \sim$ Bernoulli(f), with f as the firing rate.

Additionally, the clamp operation eliminates the need for scaling operations in attention mechanisms (e.g., QK product scaling), simplifying computations, reducing complexity, and improving energy efficiency in hardware implementations. The detailed proofs of this proposition are provided in appendix B.1.

4.2 Loss Function

The loss of the TDFormer can be formulated as follows:

$$\mathcal{L}_{\text{TDFormer}} = \sum_{n=1}^{N} \alpha_n \mathcal{L}(y, O_n), \quad \sum_{n=1}^{N} \alpha_n = 1, \quad 0 \le \alpha_n \le 1. \tag{22}$$

Here, α_n are hyperparameters. To maintain the overall loss scale, we apply a weighted average over the losses from all N stages, assigning a larger weight to the final output loss. This is because we believe that the receptive field in the temporal dimension increases as time progresses. Since the earlier stages lack feedback from future steps, their outputs are less accurate and thus subject to weaker supervision. By contrast, the final stage benefits from a larger temporal receptive field due to feedback, making its output more reliable. Therefore, during testing, only the output from the last sub-network is used for evaluation.

4.3 Top-down feedback enhances temporal dependency

Top-down feedback enhances temporal dependency from two perspectives. First, from the forward propagation perspective, we compute the mutual information matrix between features at different time

Table 1: Comparison with the baseline and previous work on ImageNet. The result in bold indicates superior performance compared to the baseline. SOTA is marked with *, previous SOTA with #. The default PM variant is v1.

Methods	Spike	Architecture	ImageNet			
			Time Step	Power (mJ)	Param (M)	Acc (%)
ViT [28]	✗	ViT-B/16(384^2)	1	254.84	86.59	77.90
DeiT [29]	✗	DeiT-B(384^2)	1	254.84	86.59	83.10
Swin [30]	✗	Swin Transformer-B(384^2)	1	216.20	87.77	84.50
Spikingformer [4]	✓	Spikingformer-8-768	4	13.68	66.34	75.85
SpikformerV1 [2]	✓	Spikformer-8-512	4	11.58	29.68	73.38
	✓	Spikformer-8-768	4	21.48	66.34	74.81
SDTV2 [6]	✓	Meta-SpikeFormer-8-384	4	32.80	31.30	77.20
	✓	Meta-SpikeFormer-8-512	4	52.40	55.40	80.00
E-Spikeformer [31]	✓	E-Spikeformer	8	30.90	83.00	84.00
	✓	E-Spikeformer	8	54.70	173.00	85.10
	✓	E-Spikeformer	8	-	173.00	86.20 #
QKFormer [3]	✓	HST-10-768 (224^2)	4	38.91	64.96	84.22
	✓	HST-10-768 (288^2)	4	64.27	64.96	85.20
	✓	HST-10-768 (384^2)	4	113.64	64.96	85.65
TDFormer	✓	HST-10-768 (224^2)	4	38.93	65.55	**85.37(+1.15)**
	✓	HST-10-768 (288^2)	4	64.39	65.55	**86.29(+1.09)**
	✓	HST-10-768 (224^2)	4	39.10	69.09	**85.57(+1.35)**
	✓	HST-10-768 (288^2)	4	64.45	69.09	**86.43 (+1.23)**
	✓	HST-10-768 (384^2)	4	113.79	69.09	**86.83 (+1.18)***

steps, as shown in Figure 1. Second, from the backward propagation perspective, we demonstrate that introducing top-down feedback helps alleviate the problem of vanishing gradients along the temporal dimension. We present the following theorem:

Definition 4.2. $\epsilon^l(t)$ is defined as the sensitivity of the membrane potential $\mathbf{H}^l(t+1)$ to its previous state $\mathbf{H}^l(t)$, and is computed as:

$$\epsilon^l(t) \equiv \frac{\partial \mathbf{H}^l(t+1)}{\partial \mathbf{H}^l(t)} + \frac{\partial \mathbf{H}^l(t+1)}{\partial \mathbf{S}^l(t)} \frac{\partial \mathbf{S}^l(t)}{\partial \mathbf{H}^l(t)}, \tag{23}$$

where l indexes the layer.

Theorem 4.3. *We adopt the rectangular function as the surrogate gradient, following the setting used in previous studies[8, 9, 12]. For a conventional SNN, the sensitivity of the membrane potential is expressed as follows:*

$$\epsilon^l(t)_{jj} = \begin{cases} 0, & \frac{1}{2}\vartheta < H_j^l(t) < \frac{3}{2}\vartheta, \\ 1 - \frac{1}{\tau}, & \textit{otherwise}\ . \end{cases} \tag{24}$$

For SNN with top-down feedback structure, the sensitivity of the membrane potential can be expressed as:

$$\epsilon^l(t)_{jj} = \begin{cases} \frac{\partial \varphi_\theta(\mathbf{S}^l(t))}{\partial \mathbf{S}^l(t)}, & \frac{1}{2}\vartheta < H_j^l(t) < \frac{3}{2}\vartheta, \\ 1 - \frac{1}{\tau}, & \textit{otherwise}\ . \end{cases} \tag{25}$$

where ϑ is the spike threshold, τ is a time constant and φ_θ is a differentiable feedback function parameterized by θ.

According to Equation 24, $\epsilon^l(t)$ becomes zero within an easily-reached interval, and outside that interval, it is upper-bounded by a small value $1 - \frac{1}{\tau}$, since τ is typically close to 1 in practice[32, 33, 34, 9]. In contrast, our method allows non-zero gradients within this interval, and the $\frac{\partial \varphi_\theta(\mathbf{S}^l(t))}{\partial \mathbf{S}^l(t)}$ can

Table 2: Comparison with the baselines and previous work on static datasets: CIFAR-10 and CIFAR-100. Conventions align with those in Table 1. The default PM variant is v1.

Methods [Architecture]	Time Step	CIFAR-10 Acc (%)	CIFAR-100 Acc (%)
STBP-tdBN [33] [ResNet-19]	4	92.92	70.86
TET [32] [ResNet-19]	4	94.44	74.47
SDTV1[5][SDT-2-512]	4	95.60	78.40
QKformer [3] [HST-4-384]	4	96.18 #	81.15 #
SpikformerV1 [2] [Spikformer-4-384]	2	93.59	76.28
	4	95.19	77.86
SpikformerV1(ours)[Spikformer-4-384]	2	93.65	75.29
	4	94.73	77.88
TDFormer[Spikformer-4-384]	2	**94.17 (+0.52)**	**75.79 (+0.50)**
	4	**95.11 (+0.38)**	**77.99 (+0.11)**
SDTV1(ours)[SDT-2-256]	4	94.47	76.05
SDTV1(ours)[SDT-2-512]	4	95.78	79.15
TDFormer[SDT-2-256]	4	**94.61 (+0.14)**	**76.23 (+0.18)**
TDFormer[SDT-2-512]	4	**96.07 (+0.29)**	**79.67 (+0.52)**
TDFormer [HST-4-384]	4	**96.51 (+0.33)***	**81.45 (+0.30)***

exceed $1 - \frac{1}{\tau}$. This property helps to alleviate the vanishing gradient problem along the temporal dimension. The detailed proof is provided in the appendix B.3.

5 Experiments

We evaluate our models on several datasets: CIFAR-10 [35], CIFAR-100 [35], CIFAR10-DVS [36], DVS128 Gesture [37], ImageNet [38], CIFAR-10C [39] and ImageNet-C [39]. For the smaller datasets, we employ the feedback pathway on SpikformerV1 [2] , Spike-driven TransformerV1 [5] and QKformer[3], experimenting with different configurations tailored to each dataset. For the large-scale datasets, we utilize QKformer[3] as baselines. Specific implementation details are provided in appendix A.

5.1 Experiments on ImageNet

Table 1 presents the results for the large-scale dataset ImageNet. The incorporation of top-down feedback structure has demonstrated significant improvements on E-spikformer, which is the previous SOTA model of SNNs. Notably, compared to QKFormer, increasing the model size by merely 0.02 million parameters and 0.59 millijoules of power consumption leads to a significant gain of 1.15% in top-1 accuracy on the ImageNet dataset. Our model sets a new SOTA performance in the SNN field. This milestone lays a solid foundation for advancing SNNs toward large-scale networks, further bridging the gap between SNNs and traditional deep learning models. Furthermore, we calculate the power of TDFormer following the method in [3], as detailed in Table 1. TDFormer results in a slight increase in energy consumption due to the feedback structure, but it achieves superior performance with minimal additional power usage. The detailed calculation of power consumption is provided in the appendix C.1.

5.2 Experiments on Neuromorphic and CIFAR Datasets

Table 3 presents the results for the neuromorphic datasets CIFAR10-DVS and DVS128 Gesture. Our proposed TDFormer consistently outperforms the baselines across all experiments, except for the Spiking Transformer-2-256 at a time step of 10. Furthermore, we achieve SOTA results, with an accuracy of 85.83% on CIFAR10-DVS using the HST-2-256 (V1), marking a notable improvement

Table 3: Comparison with the baselines and previous work on the Neuromorphic Dataset. Conventions align with those in Table 1. The default PM variant is v1.

Methods [Architecture]	CIFAR10-DVS		DVS128 Gesture	
	Time Step	Acc (%)	Time Step	Acc (%)
STBP-tdBN [33] [ResNet-19]	10	67.80	40	96.90
DSR [40] [VGG-11]	10	77.30	-	-
SDTV1 [5][SDT-2-256]	16	80.00	16	99.30 #
SpikformerV1 [2] [Spikformer-2-256]	10	78.90	10	96.90
	16	80.90	16	98.30
Spikingformer [4] [Spikingformer-2-256]	10	79.90	10	96.20
	16	81.30	16	98.30
Qkformer [3] [HST-2-256]	16	84.00 #	16	98.60
SpikformerV1(ours) [Spikformer-2-256]	10	78.08	-	-
	16	79.40	-	-
TDFormer [Spikformer-2-256]	10	**78.90 (+0.82)**	-	-
	16	**81.70 (+2.30)**	-	-
SDTV1(ours) [SDT-2-256]	10	75.22	10	96.79
	16	77.07	16	97.98
TDFormer[SDT-2-256]	10	75.05 (-0.17)	10	**96.92 (+0.13)**
	16	**77.45 (+0.38)**	16	**99.65 (+1.67)***
TDFormer[HST-2-256]	16	**85.83 (+1.83)***	16	**98.96 (+0.36)**

of 1.83% compared to the previous SOTA model, QKformer. We also achieve 99.65% accuracy on DVS128 Gesture using the Spiking Transformer-2-256 (V1) at 16 time steps.

In addition, the results for the static datasets CIFAR-10 and CIFAR-100 are summarized in Table 2. Compared to the baselines, the proposed TDFormer consistently demonstrates significant performance improvements across all experiments, with the exception of Spikformer-4-384 (V1) at time step 6. Furthermore, we achieve the SOTA performance, attaining 96.51% accuracy on CIFAR-10 and 81.45% on CIFAR-100 using the HST-2-256 (V1) at a time step of 4.

5.3 Model Generalization Analysis

As reported in Table 5, we report results averaged over five random seeds for reliability. Our model consistently improves performance across time steps and depths. To assess robustness, we evaluate on the CIFAR-10C dataset with 15 corruption types. As shown in Table 7, the model equipped with the TDAC module consistently achieves higher accuracy under various distortion settings.

Moreover, we provide a visualization analysis of the TDFormer attention modules on CIFAR-10C and ImageNet-C. The specific results can be seen in Figure 4 and Figure 5 of the appendix C. We find that after adding the TDAC module, the model focuses more on the targets and their surrounding areas. This indicates that TDAC can filter noise and irrelevant information, allowing the model to focus more on task-related information.

6 Conclusion

In this study, we propose TDFormer, which integrates an adaptive top-down feedback structure into Transformer-based SNNs, addressing a key limitation of temporal information utilization in existing models by incorporating biological top-down mechanisms. The TDFormer model outperforms traditional Transformer-based SNNs, achieving SOTA performance across all evaluated datasets. Our work suggests that the top-down feedback structure could be a valuable component for Transformer-based SNNs and offers insights for future research into more advanced, biologically inspired neural architectures that better mimic human cognition.

References

[1] Kai Malcolm and Josue Casco-Rodriguez. A comprehensive review of spiking neural networks: Interpretation, optimization, efficiency, and best practices. *arXiv preprint arXiv:2303.10780*, 2023.

[2] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng YAN, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

[3] 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. *arXiv preprint arXiv:2403.16552*, 2024.

[4] Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Zhengyu Ma, Han Zhang, Huihui Zhou, and Yonghong Tian. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv preprint arXiv:2304.11954*, 2023.

[5] Man Yao, JiaKui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo XU, and Guoqi Li. Spike-driven transformer. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[6] Man Yao, JiaKui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo XU, and Guoqi Li. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. In *The Twelfth International Conference on Learning Representations*, 2024.

[7] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 1311–1318, 2019.

[8] Yongqi Ding, Lin Zuo, Mengmeng Jing, Pei He, and Hanpu Deng. Rethinking spiking neural networks from an ensemble learning perspective. *arXiv preprint arXiv:2502.14218*, 2025.

[9] Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Towards memory-and time-efficient backpropagation for training spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6166–6176, 2023.

[10] Hangchi Shen, Qian Zheng, Huamin Wang, and Gang Pan. Rethinking the membrane dynamics and optimization objectives of spiking neural networks. *Advances in Neural Information Processing Systems*, 37:92697–92720, 2024.

[11] 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*.

[12] 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. *arXiv preprint arXiv:2402.04663*, 2024.

[13] Donghyun Lee, Yuhang Li, Youngeun Kim, Shiting Xiao, and Priyadarshini Panda. Spiking transformer with spatial-temporal attention. *arXiv preprint arXiv:2409.19764*, 2024.

[14] Charles D Gilbert and Wu Li. Top-down influences on visual processing. *Nature reviews neuroscience*, 14(5):350–363, 2013.

[15] Timothy J Buschman and Earl K Miller. Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *science*, 315(5820):1860–1862, 2007.

[16] John H Reynolds and David J Heeger. The normalization model of attention. *Neuron*, 61(2):168–185, 2009.

[17] Maurizio Corbetta, Erbil Akbudak, Thomas E Conturo, Abraham Z Snyder, John M Ollinger, Heather A Drury, Martin R Linenweber, Steven E Petersen, Marcus E Raichle, David C Van Essen, et al. A common network of functional areas for attention and eye movements. *Neuron*, 21(4):761–773, 1998.

[18] Zhaokun Zhou, Kaiwei Che, Wei Fang, Keyu Tian, Yuesheng Zhu, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer v2: Join the high accuracy club on imagenet with an snn ticket. *arXiv preprint arXiv:2401.02020*, 2024.

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

[20] Han Zhang, Chenlin Zhou, Liutao Yu, Liwei Huang, Zhengyu Ma, Xiaopeng Fan, Huihui Zhou, and Yonghong Tian. Sglformer: Spiking global-local-fusion transformer with high performance. *Frontiers in Neuroscience*, 18:1371290, 2024.

[21] Yuhua Zheng, Yan Meng, and Yaochu Jin. Object recognition using a bio-inspired neuron model with bottom-up and top-down pathways. *Neurocomputing*, 74(17):3158–3169, 2011.

[22] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[23] Baifeng Shi, Trevor Darrell, and Xin Wang. Top-down visual attention from analysis by synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2102–2112, 2023.

[24] Roy Abel and Shimon Ullman. Top-down network combines back-propagation with attention. *arXiv preprint arXiv:2306.02415*, 2023.

[25] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[26] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[27] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022.

[28] 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.

[29] 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*, pages 10347–10357. PMLR, 2021.

[30] 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*, pages 10012–10022, 2021.

[31] 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*, 2025.

[32] 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.

[33] 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*, volume 35, pages 11062–11070, 2021.

[34] Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 326–335, 2022.

[35] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[36] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:309, 2017.

[37] 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*, pages 7243–7252, 2017.

[38] 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*, pages 248–255. Ieee, 2009.

[39] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[40] 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*, pages 12444–12453, 2022.

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[42] 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 *European Conference on Computer Vision*, pages 253–272. Springer, 2024.

[43] Youngeun Kim, Joshua Chough, and Priyadarshini Panda. Beyond classification: Directly training spiking neural networks for semantic segmentation. *Neuromorphic Computing and Engineering*, 2(4):044015, 2022.

[44] Changze Lv, Jianhan Xu, and Xiaoqing Zheng. Spiking convolutional neural networks for text classification. *arXiv preprint arXiv:2406.19230*, 2024.

A Implementation Details

A.1 Training Protocols

We adopted the following training protocols:

- **Spike Generation**: We used a rate-based method for spike generation [2].
- **Data Augmentation and Training Duration**: SpikformerV1 experiments followed [2], while Spike-driven TransformerV1 experiments followed [5], furthermore QKformer experiments followed the experimental setting in and [3].
- **Optimization**: We employed AdamW [41] as the optimizer for our experiments. The learning rate was set to 3×10^{-4} for the Spike-driven TransformerV1. For SpikformerV1, we used a learning rate of 5×10^{-4} on static datasets and 1×10^{-3} on neuromorphic datasets. Additionally, we utilized a cosine learning rate scheduler to adjust the learning rate dynamically during training. Specifically, for QKformer, we fine-tuned the pretrained network with a base learning rate of 2×10^{-5} for 15 epochs, due to the high cost of direct training on ImageNet using 4 time steps.
- **Batch Size**: The batch sizes for different datasets and models are specified in Table 4.

Table 4: Batch sizes for different datasets and models.

Dataset	Model	Batch Size
CIFAR-10 and CIFAR-100	SpikeformerV1	128
	Spike-driven TransformerV1	64
CIFAR10-DVS and DVS128 Gesture	SpikeformerV1	16
	Spike-driven TransformerV1	16
ImageNet	QKformer	57

A.2 Datasets

Our experiments evaluated the performance and robustness of the TDFormer model using the following datasets:

- **CIFAR-10:** This dataset contains 60,000 32×32 color images divided into 10 classes [35].
- **CIFAR-100:** This dataset is similar to CIFAR-10 but includes 100 classes, providing a more challenging classification task [35].
- **CIFAR10-DVS:** This is an event-based version of the CIFAR-10 dataset [36].
- **DVS128 Gesture:** This is an event-based dataset for gesture recognition with 11 classes [37].
- **ImageNet:** This large-scale dataset contains over 1.2 million images divided into 1,000 classes [38].
- **CIFAR-10C:** This is a corrupted version of CIFAR-10 with 19 common distortion types, used to assess robustness [39].
- **ImageNet-C:** This dataset is a corrupted version of ImageNet, designed similarly to CIFAR-10C [39].

A.3 Computational Environment

A.3.1 Software Setup

We utilized PyTorch version 2.0.1 with CUDA 11.8 support and SpikingJelly version 0.0.0.0.12 as the primary software tools.

A.3.2 Hardware Setup.

For the smaller dataset experiments, we utilized the following configuration:

- **Hardware Used:** NVIDIA L40S and L40 GPUs.
- **Configuration:** Single-GPU for each experiment.
- **Memory Capacity:** Each GPU is equipped with 42 GB of memory.

For the large-scale dataset (ImageNet) experiments, we employed the following setup:

- **Hardware Used:** NVIDIA H20 GPUs.
- **Configuration:** Eight-GPU for each experiment.
- **Memory Capacity:** Each GPU provides 96 GB of memory.

A.4 Random Seed

To ensure the comparability of the results, we selected the same random seeds as those in the baseline paper. To ensure robustness, we also conducted experiments with random seeds 0, 42, 2024, 3407 and 114514, averaging the results. Detailed results are presented in Table 5.

B Mathematical Derivations

B.1 Detailed proofs of the upper bound on PM output variance

Proof. We assume that each $\mathbf{M}_{\text{spatial}}(t, n)$ is an independent random variable M_{tn}. Given that $b \leq M_{tn} \leq a$, it follows that $b \leq \mathbb{E}[M_{tn}] \leq a$. Furthermore, when $X_{tnc} \neq 0$, we have:

$$(X_{tnc}M_{tn} - b)(a - X_{tnc}M_{tn}) \geq 0, \tag{26}$$

which expands to:

$$-(X_{tnc}M_{tn})^2 + (a + b)(X_{tnc}M_{tn}) - ab \geq 0. \tag{27}$$

Taking the expectation on both sides yields:

$$\mathbb{E}\left[(X_{tnc}M_{tn})^2\right] \leq (a + b)\mathbb{E}\left[X_{tnc}M_{tn}\right] - ab. \tag{28}$$

Using the Law of Total Variance, we can decompose the variance of Y_{tnc} as:

$$\text{Var}(Y_{tnc}) = \mathbb{E}[\text{Var}(Y_{tnc}|X_{tnc})] + \text{Var}(\mathbb{E}[Y_{tnc}|X_{tnc}]). \tag{29}$$

For the first term, the expectation of the conditional variance can be expressed as:

$$\mathbb{E}[\text{Var}(Y_{tnc}|X_{tnc})] = f \cdot \text{Var}(Y_{tnc}|X_{tnc} = 1) + (1 - f) \cdot \text{Var}(Y_{tnc}|X_{tnc} = 0). \tag{30}$$

For the second term, the variance of the conditional expectation can be expanded as:

$$\text{Var}(\mathbb{E}[Y_{tnc}|X_{tnc}]) = \mathbb{E}[\mathbb{E}[Y_{tnc}|X_{tnc}]^2] - \mathbb{E}[\mathbb{E}[Y_{tnc}|X_{tnc}]]^2. \tag{31}$$

By substituting the conditional probabilities, we have:

$$\text{Var}(\mathbb{E}[Y_{tnc}|X_{tnc}]) = f \cdot \mathbb{E}[Y_{tnc}|X_{tnc} = 1]^2 - f^2 \cdot \mathbb{E}[Y_{tnc}|X_{tnc} = 1]^2. \tag{32}$$

Combining the two terms, the total variance becomes:

$$\text{Var}(Y_{tnc}) = f \cdot \text{Var}(Y_{tnc}|X_{tnc} = 1) + (f - f^2) \cdot \mathbb{E}[Y_{tnc}|X_{tnc} = 1]^2. \tag{33}$$

From Equation 32, we define $\mathbb{E}[Y_{tnc}|X_{tnc} = 1] = \mu$. Substituting this definition, the variance can be rewritten as:

$$\text{Var}(Y_{tnc}) = f \cdot (\mathbb{E}[Y_{tnc}^2|X_{tnc} = 1] - \mu^2) + (f - f^2) \cdot \mu^2. \tag{34}$$

Using the constraints $b \leq M_{tn} \leq a$, we have the following bound for $\text{Var}(Y_{tnc}|X_{tnc} = 1)$:

$$\text{Var}(Y_{tnc}|X_{tnc} = 1) \leq (a + b)\mu - ab - \mu^2. \tag{35}$$

By substituting this into the total variance expression, the upper bound of $\text{Var}(Y_{tnc})$ becomes:

$$\begin{aligned}\text{Var}(Y_{tnc}) &\leq f \cdot ((a+b)\mu - ab - \mu^2) + (f - f^2) \cdot \mu^2 \\ &\leq -f^2 \cdot \left(\mu - \frac{a+b}{2f}\right)^2 + \frac{a^2 + 2ab + b^2 - 4fab}{4}.\end{aligned} \tag{36}$$

Next, we will prove that this upper bound can be achieved with equality under specific conditions.

Case 1: When $\frac{a+b}{2a} \leq f \leq 1$, we assume that:

$$\mathbb{E}[Y_{tnc}|X_{tnc} = 1] = \frac{a+b}{2f}, \quad M_{tn} = a \text{ or } b. \tag{37}$$

Here, M_{tn} is a binary random variable, taking the value a with probability p and the value b with probability $1-p$. Using this assumption, we can express the conditional expectation $\mathbb{E}[Y_{tnc}|X_{tnc} = 1]$ as:

$$\mathbb{E}[Y_{tnc}|X_{tnc} = 1] = pa + (1-p)b. \tag{38}$$

Substituting $\mathbb{E}[Y_{tnc}|X_{tnc} = 1] = \frac{a+b}{2f}$ into the above equation, we solve for p:

$$pa + (1-p)b = \frac{a+b}{2f} \Rightarrow p = \frac{a+b-2bf}{2f(a-b)}. \tag{39}$$

The variance of Y_{tnc} under this distribution is maximized when M_{tn} follows this binary distribution. Substituting p into the variance formula, the maximum variance is given by:

$$\max(\text{Var}(Y_{tnc})) = \frac{a^2 + 2ab + b^2 - 4fab}{4}. \tag{40}$$

Case 2: When $0 \leq f \leq \frac{a+b}{2a}$, the upper bound is achieved when $M_{tn} = a$. In this scenario, M_{tn} is deterministic, and therefore:

$$Y_{tnc} = X_{tnc}M_{tn} = X_{tnc}a, \quad \mathbb{E}[Y_{tnc}|X_{tnc} = 1] = a. \tag{41}$$

Substituting this into the variance formula, the maximum variance simplifies to:

$$\max(\text{Var}(Y_{tnc})) = a^2(f^2 - f + 1/2) + ab(1-2f) + b^2/2. \tag{42}$$

The proof is now complete. □

We observe that both SSA and QKTA exhibit significantly larger variance compared to our proposed attention mechanism. Their variances are expressed as follows:

Variance of QKTA:

$$\text{Var}(\text{QKTA}) = df_Q(1 - f_Q), \tag{43}$$

where d is the feature dimension, and f_Q represents the firing rate of the query.

Variance of SSA:

$$\begin{aligned}\text{Var}(\text{SSA}) = Nd\Big(&f_Q f_K f_V (1-f_Q)(1-f_K)(1-f_V) \\ &+ f_Q f_K f_V^2 (1-f_Q)(1-f_K) \\ &+ f_Q f_K^2 f_V (1-f_Q)(1-f_V) \\ &+ f_Q^2 f_K f_V (1-f_K)(1-f_V) \\ &+ f_Q f_K^2 f_V^2 (1-f_Q) \\ &+ f_Q^2 f_K f_V^2 (1-f_K) \\ &+ f_Q^2 f_K^2 f_V (1-f_V)\Big),\end{aligned} \tag{44}$$

where N is the number of spatial locations, d is the feature dimension, and f_Q, f_K, f_V are the firing rates of the query, key, and value.

Comparison with Our Attention Mechanism: The variance of QKTA scales linearly with d. By contrast, the variance of SSA grows with both N and d, resulting in significantly larger values compared to QKTA. Our proposed attention mechanism is particularly effective in scenarios with large spatial (N) and feature (d) dimensions. The strict upper bound on output variance ensures numerical stability, preventing vanishing during training. Additionally, this upper bound eliminates the need for traditional scaling operations (e.g., scaling factors in QK products), simplifying computations, reducing complexity, and enhancing energy efficiency.

B.2 The mathematical properties of hyperparameters

Next, we will analyze the expectation and variance of the PM and propose an appropriate selection of hyperparameters to ensure output stability.

Lemma B.1. *if the set* $\{c \in \mathbb{N} : w_c = 0\}$ *is finite and* $\exists\, m, M > 0, \forall\, c \in \mathbb{N},\ m \leq |w_c| \leq M$*, then:*

$$w_c' = \lim_{C\to\infty} \frac{w_c}{\sqrt{\sum_{c=1}^{C} w_c^2}} = 0 \tag{45}$$

Proof. We begin by defining the normalized weight:

$$w_c' = \frac{w_c}{\sqrt{\sum_{c=1}^{C} w_c^2}}. \tag{46}$$

By assumption, there are k terms where $w_c = 0$, and for the remaining $C - k$ terms, the weights satisfy:

$$m^2 \leq w_c^2 \leq M^2 \quad \text{for all } c. \tag{47}$$

Thus, the sum of squares of the weights is bounded as follows:

$$(C-k)m^2 \leq \sum_{c=1}^{C} w_c^2 \leq (C-k)M^2. \tag{48}$$

Taking the square root, we find that the denominator grows as:

$$\sqrt{\sum_{c=1}^{C} w_c^2} \geq \sqrt{(C-k)m^2} \sim O(\sqrt{C}). \tag{49}$$

Using the bound $|w_c| \leq M$, the normalized weight w_c' satisfies:

$$|w_c'| = \frac{|w_c|}{\sqrt{\sum_{c=1}^{C} w_c^2}} \leq \frac{M}{\sqrt{\sum_{c=1}^{C} w_c^2}} \leq \frac{M}{\sqrt{(C-k)m^2}}. \tag{50}$$

To ensure $|w_c'| < \epsilon$ for a given $\epsilon > 0$, it suffices to require:

$$\frac{M}{\sqrt{(C-k)m^2}} < \epsilon. \tag{51}$$

Rearranging, this condition can be rewritten as:

$$C \geq \frac{M^2}{m^2\epsilon^2} + k. \tag{52}$$

As $C \to \infty$, the condition $C \geq \frac{M^2}{m^2\epsilon^2} + k$ is always satisfied. Thus, for any $\epsilon > 0$, we have $|w_c'| < \epsilon$, which implies:

$$\lim_{C\to\infty} w_c' = 0. \tag{53}$$

The proof is complete. □

Lemma B.2. *We assume that the features across different channels are independent and identically distributed (i.i.d.). When the number of channels C is large, we have:*

$$M_{tn} \sim \mathcal{N}\left(\sum_{c=1}^{C} w'_c f_r, \sum_{c=1}^{C} w'^2_c f_r(1-f_r)\right), \quad C \to \infty, \tag{54}$$

$$M_{tn} = \sum_{c=1}^{C} x_{tnc} w'_c. \tag{55}$$

where $x \in X$, $x \sim$ Bernoulli(f_r), f_r represents the firing rate (the probability of $x_{tnc} = 1$).

Proof. To prove this lemma, we use the characteristic function method. The characteristic function of a Bernoulli random variable x_{tnc} is given by:

$$\Phi_{x_{tnc}}(t) = \mathbb{E}\left[e^{itx_{tnc}}\right] = f_r e^{it} + (1-f_r). \tag{56}$$

For the weighted variable $w'_c x_{tnc}$, its characteristic function is:

$$\Phi_{w'_c x_{tnc}}(t) = \mathbb{E}\left[e^{itw'_c x_{tnc}}\right] = f_r e^{itw'_c} + (1-f_r). \tag{57}$$

Since the features across channels are independent, the characteristic function of M_{tn} is:

$$\Phi_{M_{tn}}(t) = \prod_{c=1}^{C} \Phi_{w'_c x_{tnc}}(t). \tag{58}$$

Substituting the expression for $\Phi_{w'_c x_{tnc}}(t)$:

$$\Phi_{M_{tn}}(t) = \prod_{c=1}^{C} \left(f_r e^{itw'_c} + (1-f_r)\right). \tag{59}$$

$$\begin{aligned} f_r e^{itw'_c} + (1-f_r) &= f_r\left(1 + itw'_c - \frac{1}{2}t^2 w'^2_c + o(w'^2_c)\right) + (1-f_r) \\ &\approx 1 + f_r(itw'_c - \frac{1}{2}t^2 w'^2_c). \end{aligned} \tag{60}$$

Thus, the characteristic function becomes:

$$\Phi_{M_{tn}}(t) \approx \prod_{c=1}^{C} \left(1 + f_r(itw'_c - \frac{1}{2}t^2 w'^2_c)\right). \tag{61}$$

Taking the logarithm to simplify the product into a sum:

$$\begin{aligned} \ln \Phi_{M_{tn}}(t) &= \sum_{c=1}^{C} \ln\left(1 + f_r(itw'_c - \frac{1}{2}t^2 w'^2_c)\right) \\ &= \sum_{c=1}^{C} f_r itw'_c - \frac{1}{2}t^2 w'^2_c f_r + \frac{1}{2}t^2 w'^2_c f_r^2 + O(w'^2_c), \end{aligned} \tag{62}$$

where we used $\ln(1+x) = x - \frac{1}{2}x^2 + O(x^2)$ for small x.

Separating terms, we get:

$$\ln \Phi_{M_{tn}}(t) \approx it \sum_{c=1}^{C} w'_c f_r - \frac{1}{2}t^2 \sum_{c=1}^{C} w'^2_c f_r(1-f_r). \tag{63}$$

Exponentiating the logarithm gives:

$$\Phi_{M_{tn}}(t) = \exp\left(it \sum_{c=1}^{C} w'_c f_r - \frac{1}{2} t^2 \sum_{c=1}^{C} w'^2_c f_r (1 - f_r) \right). \tag{64}$$

This is the characteristic function of a normal distribution with:

$$\text{Mean:} \quad \mu = \sum_{c=1}^{C} w'_c f_r, \tag{65}$$

$$\text{Variance:} \quad \sigma^2 = \sum_{c=1}^{C} w'^2_c f_r (1 - f_r). \tag{66}$$

Since the characteristic function corresponds to a normal distribution, we conclude:

$$M_{tn} \sim \mathcal{N}\left(\sum_{c=1}^{C} w'_c f_r, \sum_{c=1}^{C} w'^2_c f_r (1 - f_r) \right). \tag{67}$$

The proof is complete. □

Lemma B.3. *The distributions of X_{tnc} and M_{tn} can be considered independent when the number of channels C is large. Specifically, for all $t_1, t_2 \in \mathbb{R}$, we have:*

$$\phi_{M_{tn}, X_{tnc}}(t_1, t_2) = \phi_{M_{tn}}(t_1) \cdot \phi_{X_{tnc}}(t_2), \quad C \to \infty, \tag{68}$$

where $\phi_X(t)$ represents the characteristic function of X.

Proof. The joint characteristic function of M_{tn} and X_{tnc} is given by:

$$\begin{aligned} \phi_{M_{tn}, X_{tnc}}(t_1, t_2) &= \mathbb{E}\left[e^{(it_1 M_{tn} + it_2 X_{tnc})} \right] \\ &= \mathbb{E}\left[e^{\left(it_1 \sum_c w'_c X_{tnc} + it_2 X_{tnc}\right)} \right]. \end{aligned} \tag{69}$$

Separating X_{tnc} and the sum $\sum_{i \neq c} w'_i X_{tni}$, we rewrite:

$$\begin{aligned} \phi_{M_{tn}, X_{tnc}}(t_1, t_2) &= \mathbb{E}\left[e^{\left(it_1 \sum_{i \neq c} w'_i X_{tni} + iX_{tnc}(t_2 + t_1 w'_c)\right)} \right] \\ &= \mathbb{E}\left[e^{\left(it_1 \sum_{i \neq c} w'_i X_{tni}\right)} \right] \cdot \mathbb{E}\left[e^{\left(iX_{tnc}(t_2 + t_1 w'_c)\right)} \right]. \end{aligned} \tag{70}$$

Using the independence of X_{tni} across channels:

$$\phi_{M_{tn}, X_{tnc}}(t_1, t_2) = \prod_{i \neq c} \mathbb{E}\left[e^{\left(it_1 w'_i X_{tni}\right)} \right] \cdot \mathbb{E}\left[e^{\left(iX_{tnc}(t_2 + t_1 w'_c)\right)} \right]. \tag{71}$$

Substituting the characteristic function of Bernoulli random variables $X_{tnc} \sim \text{Bernoulli}(f)$:

$$\mathbb{E}\left[e^{itX_{tnc}} \right] = (1 - f) + f e^{it}. \tag{72}$$

Thus:

$$\phi_{M_{tn}, X_{tnc}}(t_1, t_2) = \prod_{i \neq c} \left[(1 - f) + f e^{it_1 w'_i} \right] \cdot \left[(1 - f) + f e^{i(t_2 + t_1 w'_c)} \right]. \tag{73}$$

Using Lemma B.2, for small w'_c, we apply the Taylor expansion to approximate each term:

$$(1 - f) + f e^{it_1 w'_i} \approx 1 + f(it_1 w'_i), \tag{74}$$

$$(1 - f) + f e^{i(t_2 + t_1 w'_c)} \approx (1 - f) + f e^{it_2}. \tag{75}$$

Substituting back:

$$\phi_{M_{tn},X_{tnc}}(t_1,t_2) \approx \prod_{i\neq c}(1+fit_1w_i')\cdot\left[(1-f)+fe^{it_2}\right]. \tag{76}$$

Using Equation 59, Equation 72 and Taylor expansion, the product of the characteristic functions for the two distributions is:

$$\begin{aligned}
\phi_{X_{tnc}}(t_2)\phi_{M_{tn}}(t_1) &= (1-f+fe^{it_2})\prod_{i=1}^{C}(1-f+fe^{it_1w_i'}) \\
&= (1-f+fe^{it_2})\prod_{i=1}^{C}(1+fit_1w_i') \\
&= (1-f+fe^{it_2})(1+fit_1w_c')\prod_{i\neq c}(1+fit_1w_i') \\
&= (1-f+fe^{it_2})\prod_{i\neq c}(1+fit_1w_i') \\
&= \phi_{M_{tn},X_{tnc}}(t_1,t_2)
\end{aligned} \tag{77}$$

Thus, the joint characteristic function factorizes into the product of the marginal characteristic functions, which demonstrates that M_{tn} and X_{tnc} are asymptotically independent as $C\to\infty$. □

Proposition B.4. *If $b\approx 0$, $a\geq 1$, and the firing rate f is relatively small value, the PM output Y_{tnc} satisfies:*

$$\mathbb{E}(Y_{tnc}) \approx \sqrt{\frac{f(1-f)}{2\pi}}\,\mathbb{E}(X_{tnc}) \tag{78}$$

$$\mathit{Var}(Y_{tnc}) \approx \frac{f(\pi-f)}{2\pi}\,\mathit{Var}(X_{tnc}) \tag{79}$$

Proof. For convenience, we denote:

$$\mu=\sum_{c=1}^{C}w_c'f,\quad \sigma^2=\sum_{c=1}^{C}w_c'^2f(1-f)=f(1-f),\quad M_{tn}'=\text{clamp}(M_{tn},b,a). \tag{80}$$

According to Lemma B.2, the input distribution satisfies:

$$M_{tn}\sim\mathcal{N}(\mu,\sigma^2). \tag{81}$$

The expectation of the clamped variable M_{tnc}' is:

$$\begin{aligned}
\mathbb{E}(M_{tn}') &= \int_{-\infty}^{\infty}xf(x)dx \\
&= \frac{1}{\sqrt{2\pi\sigma^2}}\int_0^a x\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)dx+\frac{a}{\sqrt{2\pi\sigma^2}}\int_a^{\infty}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)dx.
\end{aligned} \tag{82}$$

For the first term, let $t=(x-\mu)^2$, if $\mu\approx 0$, then:

$$\begin{aligned}
&\frac{1}{\sqrt{2\pi\sigma^2}}\int_0^a x\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)dx \\
&= \frac{1}{2\sqrt{2\pi\sigma^2}}\int_{\mu^2}^{(a-\mu)^2}\exp\left(-\frac{t}{2\sigma^2}\right)dt+\frac{\mu}{\sqrt{2\pi\sigma^2}}\int_0^a\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)dx \\
&= \frac{-\sigma}{\sqrt{2\pi\sigma}}\left[\exp\left(-\frac{t}{2\sigma^2}\right)\right]_{\mu^2}^{(a-\mu)^2}+\mu\left(\Phi\left(\frac{a-\mu}{\sigma}\right)-\Phi\left(\frac{-\mu}{\sigma}\right)\right) \\
&\approx \frac{\sigma}{\sqrt{2\pi}}\left(1-\exp\left(-\frac{a^2}{2\sigma^2}\right)\right).
\end{aligned} \tag{83}$$

where $\Phi(x)$ is the CDF of the standard normal distribution. The second term in the expectation is straightforward:

$$\frac{a}{\sqrt{2\pi\sigma^2}}\int_a^\infty \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)dx = \frac{a}{\sqrt{2\pi\sigma^2}}\int_{a-\mu}^\infty \exp\left(-\frac{t^2}{2\sigma^2}\right)dt, \tag{84}$$

Using the cumulative distribution function (CDF) again:

$$\begin{aligned}\frac{a}{\sqrt{2\pi\sigma^2}}\int_{a-\mu}^\infty \exp\left(-\frac{t^2}{2\sigma^2}\right)dt &= a\left(1-\Phi\left(\frac{a-\mu}{\sigma}\right)\right)\\ &\approx a\left(1-\Phi\left(\frac{a}{\sigma}\right)\right)\end{aligned} \tag{85}$$

The $\Phi(\frac{a}{\sigma})$ and $\exp(-\frac{a^2}{2\sigma^2})$ function decay rapidly as σ decreases. Now, combining the results from the two integrals, we have:

$$\begin{aligned}\mathbb{E}(M'_{tn}) &= \frac{\sigma}{\sqrt{2\pi}} - \frac{\sigma}{\sqrt{2\pi}}\exp\left(-\frac{a^2}{2\sigma^2}\right) + a\left(1-\Phi\left(\frac{a-\mu}{\sigma}\right)\right)\\ &\approx \frac{\sigma}{\sqrt{2\pi}}\end{aligned} \tag{86}$$

Based on B.3, we calculate the expectation and variance of M'^2_{tn}:

$$\begin{aligned}\mathbb{E}(M'^2_{tn}) &= \int_{-\infty}^\infty x^2 f(x)dx\\ &= \frac{1}{\sqrt{2\pi\sigma^2}}\int_0^a x^2\exp\left(-\frac{x^2}{2\sigma^2}\right)dx + a^2\cdot\int_a^\infty f(x)dx.\end{aligned} \tag{87}$$

We calculate the first term using integration by parts. Let:

$$u = x,\quad dv = x\exp\left(-\frac{x^2}{2\sigma^2}\right)dx,\quad du = dx,\quad v = -\sigma^2\exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{88}$$

Then:

$$\begin{aligned}&\frac{1}{\sqrt{2\pi\sigma^2}}\int_0^a x^2\exp\left(-\frac{x^2}{2\sigma^2}\right)dx\\ &= \frac{1}{\sqrt{2\pi\sigma^2}}\left(\left[-\sigma^2 x\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]_0^a + \sigma^2\int_0^a\exp\left(-\frac{x^2}{2\sigma^2}\right)dx\right)\\ &= \frac{1}{\sqrt{2\pi\sigma^2}}\left(-\sigma^2 a\exp\left(-\frac{a^2}{2\sigma^2}\right) + \sigma^2\int_0^a\exp\left(-\frac{x^2}{2\sigma^2}\right)dx\right).\end{aligned} \tag{89}$$

The remaining integral is a standard normal distribution integral:

$$\frac{\sigma^2}{\sqrt{2\pi\sigma^2}}\int_0^a\exp\left(-\frac{x^2}{2\sigma^2}\right)dx = \sigma^2\left(\Phi\left(\frac{a}{\sigma}\right)-\frac{1}{2}\right), \tag{90}$$

where $\Phi(x)$ is the CDF of the standard normal distribution.

Substituting (90) into (89):

$$\frac{1}{\sqrt{2\pi\sigma^2}}\int_0^a x^2\exp\left(-\frac{x^2}{2\sigma^2}\right)dx = \frac{-a\sigma}{\sqrt{2\pi}}\exp\left(-\frac{a^2}{2\sigma^2}\right) + \sigma^2\left(\Phi\left(\frac{a}{\sigma}\right)-\frac{1}{2}\right). \tag{91}$$

The second term is the tail of the normal distribution:

$$\int_a^\infty f(x)dx = \Phi\left(-\frac{a}{\sigma}\right), \tag{92}$$

we have:

$$a^2\cdot\int_a^\infty f(x)dx = a^2\Phi\left(-\frac{a}{\sigma}\right). \tag{93}$$

Combining (91) and (93) into (87), we get:

$$\begin{aligned}\mathbb{E}(M_{tn}'^2) &= \frac{-a\sigma}{\sqrt{2\pi}}\exp\left(-\frac{a^2}{2\sigma^2}\right) + \sigma^2\left(\Phi\left(\frac{a}{\sigma}\right) - \frac{1}{2}\right) + a^2\Phi\left(-\frac{a}{\sigma}\right) \\ &\approx \frac{\sigma^2}{2}.\end{aligned} \tag{94}$$

Since $\Phi\left(-\frac{a}{\sigma}\right)$ is exponentially small for moderate a, the term $a^2\Phi\left(-\frac{a}{\sigma}\right)$ is negligible compared to leading terms and is often omitted for simplicity.

Using $\text{Var}(M_{tn}') = \mathbb{E}(M_{tn}'^2) - \mathbb{E}(M_{tn}')^2$, we calculate:

$$\begin{aligned}\text{Var}(M_{tn}') &\approx \frac{\sigma^2}{2} - \left[\frac{\sigma}{\sqrt{2\pi}}\left(1 - \exp\left(-\frac{a^2}{2\sigma^2}\right)\right)\right]^2 \\ &\approx \frac{\pi - 1}{2\pi}\sigma^2 \\ &= \frac{\pi - 1}{2\pi}f(1-f).\end{aligned} \tag{95}$$

Given that $Y_{tnc} = M_{tnc}' \cdot X_{tnc}$, and based on Lemma B.3 that the distributions of $X_{tnc}$and M_{tn}'can be considered independent, the expectation of Y_{tnc} is:

$$\begin{aligned}\mathbb{E}(Y_{tnc}) &= \mathbb{E}(M_{tn}') \cdot \mathbb{E}(X_{tnc}) \\ &\approx \sqrt{\frac{f(1-f)}{2\pi}}\mathbb{E}(X_{tnc}).\end{aligned} \tag{96}$$

The variance of Y_{tnc} is computed as:

$$\begin{aligned}\text{Var}(Y_{tnc}) &= \text{Var}(M_{tn}') \cdot \text{Var}(X_{tnc}) + \text{Var}(M_{tn}') \cdot \mathbb{E}(X_{tnc})^2 + \text{Var}(X_{tnc}) \cdot \mathbb{E}[M_{tn}']^2 \\ &= \frac{f(\pi - f)}{2\pi}f(1-f) \\ &\approx \frac{f(\pi - f)}{2\pi}\text{Var}(X_{tnc}).\end{aligned} \tag{97}$$

Thus, the proposition is proven:

$$\mathbb{E}(Y_{tnc}) \approx \sqrt{\frac{f(1-f)}{2\pi}}\mathbb{E}(X_{tnc}), \quad \text{Var}(Y_{tnc}) \approx \frac{f(\pi - f)}{2\pi}\text{Var}(X_{tnc}). \tag{98}$$

□

In practice, we recommend setting the hyperparameters as follows: $b = 0$ and $a \in [1, 2]$. Setting $b = 0$ allows the processing module to completely eliminate certain features in the spatial domain. Furthermore, selecting $a \in [1, 2]$ enables the processing module to selectively enhance specific spatial features. This also ensures that both the mean and variance do not become too large or too small, maintaining the numerical stability.

B.3 Gradient Analysis

This section on the derivation of the traditional SNN network is mainly referenced from [40, 7, 8]. First, we derive the temporal gradient of the traditional SNN network, where the temporal gradient is primarily backpropagated through the membrane potential. Taking the vanilla LIF neuron as an example, we use the following form to analyze the gradient problem:

$$\mathbf{H}^l(t+1) = \left(1 - \frac{1}{\tau}\right)\left(\mathbf{H}^l(t) - \vartheta\mathbf{S}^l(t)\right) + \mathbf{W}^l\mathbf{S}^{l-1}(t+1), \tag{99}$$

The derivative of the loss with respect to the weights W_l is:

$$\nabla_{\mathbf{W}^l}\mathcal{L} = \sum_{t=0}^{T-1}\frac{\partial\mathcal{L}}{\partial\mathbf{H}^l(t)}^\top\mathbf{S}^{l-1}[t]^\top, l = L, L-1, \cdots, 1, \tag{100}$$

The gradient expression can be written as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{H}^l(t)} = \underbrace{\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{l+1}(t)} \frac{\partial \mathbf{H}^{l+1}(t)}{\partial \mathbf{S}^l(t)} \frac{\partial \mathbf{S}^l(t)}{\partial \mathbf{H}^l(t)}}_{Spatial\ \ Gradient} + \underbrace{\sum_{t'=t+1}^{T-1} \frac{\partial \mathcal{L}}{\partial \mathbf{H}^{l+1}(t')} \frac{\partial \mathbf{H}^{l+1}(t')}{\partial \mathbf{S}^l(t')} \frac{\partial \mathbf{S}^l(t')}{\partial \mathbf{H}^l(t')} \prod_{t''=1}^{t'-t} \epsilon^L(t'-t''), l < L,}_{Temporal\ \ Gradient} \tag{101}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{H}^l(t)} = \underbrace{\frac{\partial \mathcal{L}}{\partial \mathbf{S}^l(t)} \frac{\partial \mathbf{S}^l(t)}{\partial \mathbf{H}^l(t)}}_{Spatial\ \ Gradient} + \underbrace{\sum_{t'=t+1}^{T-1} \frac{\partial \mathcal{L}}{\partial \mathbf{S}^l(t')} \frac{\partial \mathbf{S}^l(t')}{\partial \mathbf{H}^l(t')} \prod_{t''=1}^{t'-t} \epsilon^L(t'-t''), l = L,}_{Temporal\ \ Gradient} \tag{102}$$

$\epsilon^L$is defined as the sensitivity of the membrane potential $H^l(t+1)$ with respect to $H^l(t)$ between adjacent timesteps.

$$\epsilon^l(t) \equiv \frac{\partial \mathbf{H}^l(t+1)}{\partial \mathbf{H}^l(t)} + \frac{\partial \mathbf{H}^l(t+1)}{\partial \mathbf{S}^l(t)} \frac{\partial \mathbf{S}^l(t)}{\partial \mathbf{H}^l(t)}. \tag{103}$$

If we use a simple rectangular function as a surrogate for the gradient.

$$\epsilon^l(t)_{jj} = \begin{cases} 0, & \frac{1}{2}\vartheta < H_j^l(t) < \frac{3}{2}\vartheta, \\ 1 - \frac{1}{\tau}, & \text{otherwise .} \end{cases} \tag{104}$$

From the above equation, it can be concluded that if the membrane potential approaches the threshold at any given timestep, the temporal gradient $\prod_{t''=1}^{t'-t} \epsilon^L(t'-t'')$ will vanish. This highlights a common issue with temporal gradients in the vanilla LIF model, which remains a problem even with short timesteps.

Next, we perform gradient analysis on neurons with a feedback structure. Assume the structure of the feedback is φ, which includes PM and CM.

$$\mathbf{H}^l(t+1) = \left(1 - \frac{1}{\tau}\right)\left(\mathbf{H}^l(t) - \vartheta \mathbf{S}^l(t)\right) + \mathbf{W}^l \mathbf{S}^{l-1}(t+1) + \varphi_\theta(\mathbf{S}^l(t)) \tag{105}$$

Following the above derivation, we similarly define the variable ϵ:

$$\epsilon^l(t) \equiv \frac{\partial \mathbf{H}^l(t+1)}{\partial \mathbf{H}^l(t)} + \frac{\partial \mathbf{H}^l(t+1)}{\partial \mathbf{S}^l(t)} \frac{\partial \mathbf{S}^l(t)}{\partial \mathbf{H}^l(t)} + \underbrace{\frac{\partial \mathbf{H}^l(t+1)}{\partial \varphi_\theta(\mathbf{S}^l(t))} \frac{\partial \varphi_\theta(\mathbf{S}^l(t))}{\partial \mathbf{S}^l(t)} \frac{\partial \mathbf{S}^l(t)}{\partial \mathbf{H}^l(t)}}_{Feedback\ \ \ gradient} \tag{106}$$

$$\epsilon^l(t) = \left(1 - \frac{1}{\tau}\right) - \left(1 - \frac{1}{\tau}\right)\vartheta \cdot \frac{\partial \mathbf{S}^l(t)}{\partial \mathbf{H}^l(t)} + \frac{\partial \varphi_\theta(\mathbf{S}^l(t))}{\partial \mathbf{S}^l(t)} \frac{\partial \mathbf{S}^l(t)}{\partial \mathbf{H}^l(t)} \tag{107}$$

Similarly we have:

$$\epsilon^l(t)_{jj} = \begin{cases} \frac{\partial \varphi_\theta(\mathbf{S}^l(t))}{\partial \mathbf{S}^l(t)}, & \frac{1}{2}\vartheta < H_j^l(t) < \frac{3}{2}\vartheta, \\ 1 - \frac{1}{\tau}, & \text{otherwise .} \end{cases} \tag{108}$$

Then, in training, $\frac{\partial \varphi_\theta(\mathbf{S}^l(t))}{\partial \mathbf{S}^l(t)}$ is not possible to be zero.

C Supplementary Results

C.1 Energy Consumption Calculation of TDFormer

This section is mainly referenced from [3]. We calculate the number of Synaptic Operations (SOPs) of spike before calculating theoretical energy consumption for TDFormer.

$$\mathrm{SOP} = f_r \times T \times \mathrm{FLOPs} \tag{109}$$

Table 5: Results averaged across seeds: 0, 42, 2024, 3407 and 114514. Bold results indicate superior performance compared to the baselines.

Methods	Dataset/Time Step	Architecture	Baseline	CM1+V1
SpikeformerV1	CIFAR-10/T = 2	Spikformer-2-384	**94.18±0.06**	94.07±0.07
	CIFAR-10/T = 4		94.84±0.14	**94.86±0.05**
	CIFAR-10/T = 2	Spikformer-4-384	93.65±0.23	**94.05±0.14**
	CIFAR-100/T =2		75.25±0.19	**75.99±0.12**
	CIFAR-10/T = 4		94.73±0.06	**95.13±0.07**
	CIFAR-100/T = 4		77.56±0.22	**77.60±0.26**
	CIFAR-10/T = 6		95.09±0.08	**95.16±0.14**
	CIFAR-100/T = 6		**78.21±0.22**	77.99±0.05
	CIFAR10-DVS/T = 10		78.08±0.70	**78.13±0.72**
	CIFAR10-DVS/T= 16		79.40±0.36	**80.20±0.75**
SDTV1	CIFAR-10/T = 4	Spiking Transformer-2-512	95.76±0.06	**95.92±0.02**
	CIFAR-100/T =4		79.15±0.14	**79.35±0.16**
	CIFAR-10/T = 4	Spiking Transformer-2-256	94.47±0.11	**94.64±0.04**
	CIFAR-100/T =4		76.15±0.13	**76.26±0.13**
	DVS128 Gesture/T=10		96.79±0.67	**96.92±0.29**
	DVS128 Gesture/T=16		97.98±0.59	**99.04±0.28**
	CIFAR10-DVS/T = 10		75.03±0.67	**75.05±0.11**
	CIFAR10-DVS/T = 16		77.07±0.19	**77.45±0.43**

where f_r is the firing rate of the block and T is the simulation time step of spiking neuron. FLOPS refers to floating point operations of block, which is the number of multiply-and-accumulate (MAC) operations and SOP is the number of spike-based accumulate (AC) operations.

$$E_{\text{TDFormer}} = E_{Baseline} + E_{AC} \times (\text{SOP}_{\text{PM}} + \text{SOP}_{\text{CM}}) \tag{110}$$

The channel-wise token mixer in TDFormer is highly power-efficient, consisting of only a linear layer, a LIF neuron, and a BN layer. The BN parameters can be fused into the linear layer via reparameterization, making its power consumption negligible. The linear layer maintains a constant channel dimension, resulting in much lower power usage than conventional MLPs. Furthermore, the spatial-wsie token mixer in PM has a time complexity of only $O(ND)$, which is much lower than the $O(N^2D)$ of SSA. In the CM module, although a token mixer is used, the firing rates in both PM and CM are very low. In our experiments, we observed that the firing rate in both modules remains around 0.05. As a result, the overall power overhead of TDFormer is marginal.

C.2 Additional Experiments and Visualizations

Table 6: Results of different TDFormer variants. The results in bold indicate superior performance compared to the baseline. The default configuration used in our work is indicated by *. CM1-CM3 denote different strategies for integrating top-down information with bottom-up features. CM1: S_{td} is fused into the computation of the attention map. CM2: S_{td} is fused into the value of self-attention. CM3:S_{td} is incorporated into the input of the attention module.

Model Type	SpikeformerV1 (Spikformer-4-384)			SDTV1 (Spiking Transformer-2-256)		
	Acc (%)	FLOPs (G)	Param (M)	Acc (%)	FLOPs (G)	Param (M)
Baseline	94.73	3.71	9.33	94.47	1.25	2.57
*CM1+V1	**95.14**	3.88	9.92	**94.77**	1.31	2.69
CM1+V2	**94.79**	3.88	9.92	**94.93**	1.31	2.69
CM1+V3	**94.90**	3,88	9.92	**94.61**	1.31	2.69
CM1+V4	**94.94**	3.88	9.92	**94.88**	1.31	2.69
CM2+V1	**94.88**	3.88	9.92	**94.73**	1.31	2.69
CM2+V2	**94.75**	3.88	9.92	**94.79**	1.31	2.69
CM2+V3	94.70	3.88	9.92	**94.75**	1.31	2.69
CM2+V4	**95.27**	3.88	9.92	**94.66**	1.31	2.69
CM3+V1	94.69	3.90	9.92	94.43	1.32	2.69
CM3+V2	**94.89**	3.90	9.92	**94.69**	1.32	2.69
CM3+V3	94.35	3.90	9.92	93.94	1.32	2.69
CM3+V4	**94.90**	3.90	9.92	**94.61**	1.32	2.69

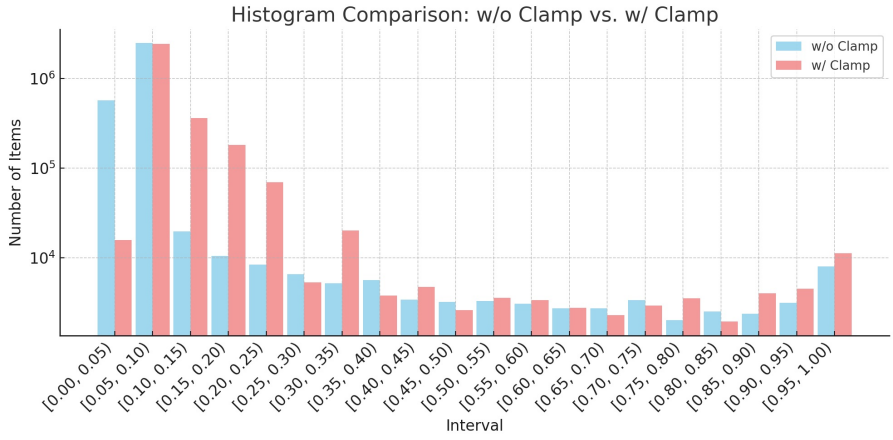

Figure 3: This is the histogram of the gradient of the surrogate function for LIF neurons in the attention module within the PM model. From the figure, we can see that the clamp operation ensures that the variance in the attention map does not become too large, thus preventing the vanishing gradient problem.

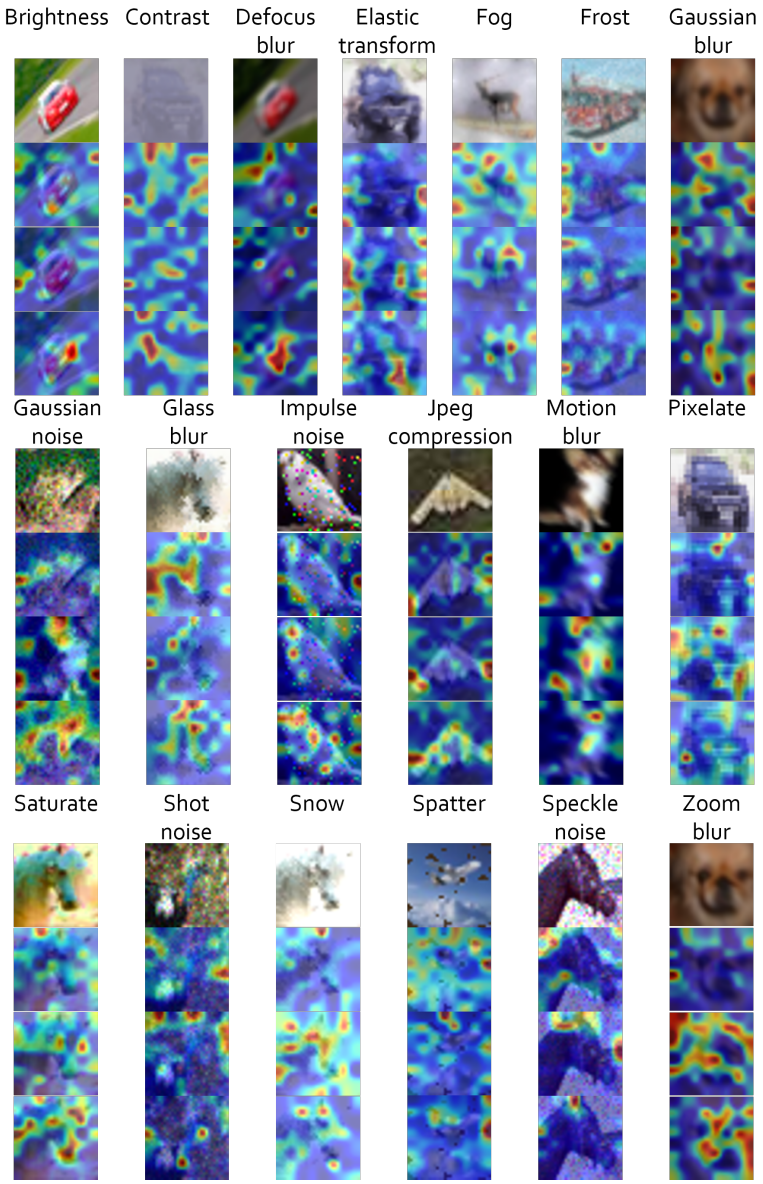

Figure 4: Visualization of CIFAR-10C. This figure showcases 19 columns corresponding to 19 different types of corruptions. Each column contains four images: the top image displays the original CIFAR-10C image; the second image shows the visualization result of the baseline model; the third image illustrates the first feedforward stage of the TDFormer model; the fourth image depicts the second feedforward stage of the TDFormer model, demonstrating the model's dynamic attention adjustments across stages.

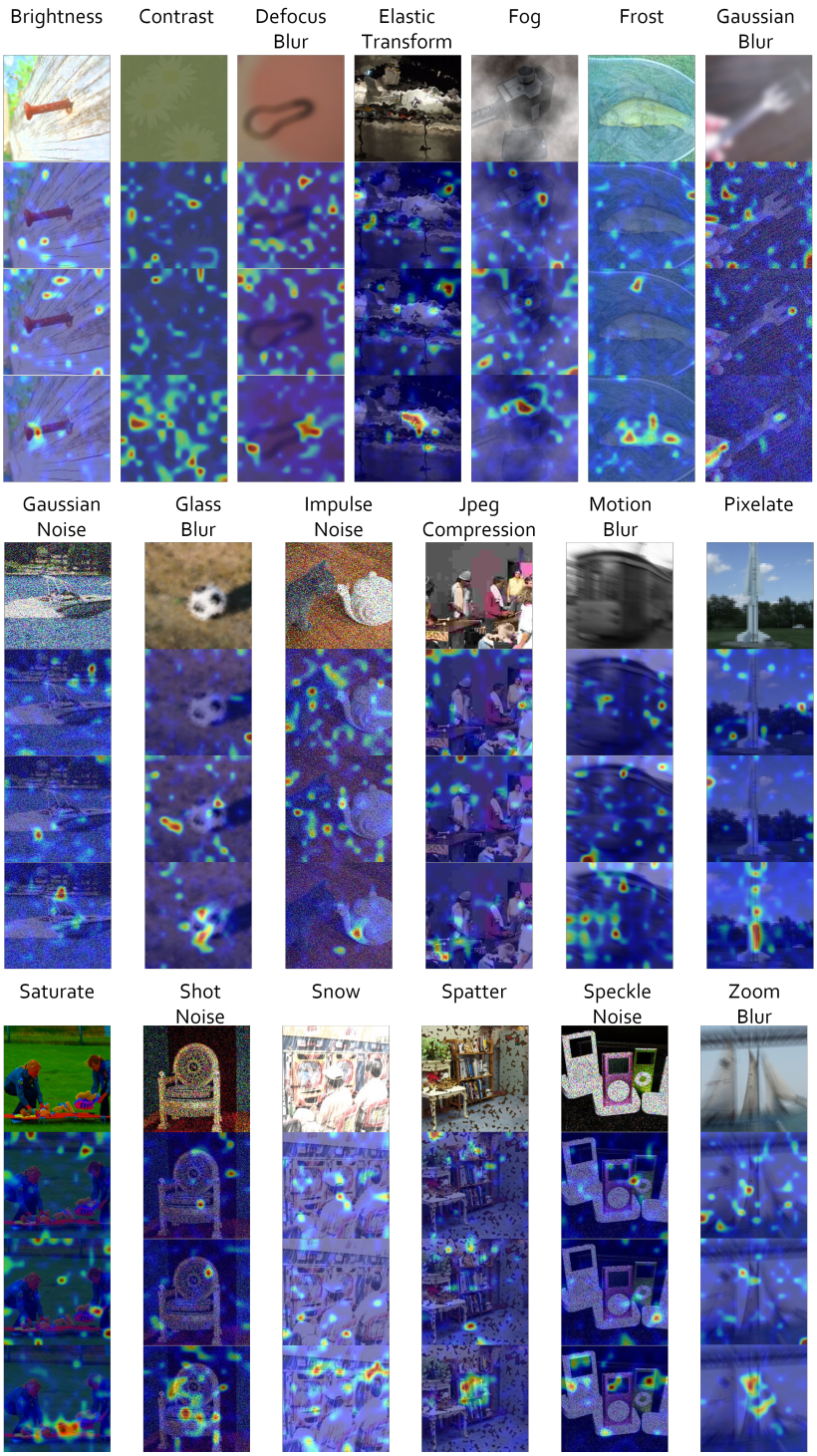

Figure 5: Visualization of ImageNet-C. This figure showcases 19 columns corresponding to 19 different types of corruptions. The layout and visualization style are similar to those shown in Figure 4.

Table 7: Robustness comparison on the CIFAR-10C dataset. The results in bold indicate superior performance compared to the baseline. Average performance across different distortion types is indicated by *.

Corruption Type	SpikformerV1 /TDFormer Acc (%)	Time Step	SpikformerV1 /TDFormer Acc (%)	Corruption Type
Brightness	91.32/91.27 (-0.05)	1	**76.23/76.97 (+0.74)**	Motion Blur
	91.87/91.94 (+0.06)	2	**77.00/78.30 (+1.30)**	
	93.14/93.29 (+0.15)	4	**79.44/80.01 (+0.57)**	
Contrast	**69.93/70.40 (+0.47)**	1	**79.31/79.51 (+0.20)**	Pixelate
	70.41/71.25 (+0.84)	2	78.70/78.67 (-0.03)	
	77.06/76.57 (-0.49)	4	**81.14/81.45 (+0.31)**	
Defocus Blur	**80.59/80.83 (+0.24)**	1	87.33/87.10 (-0.23)	Saturate
	81.39/82.15 (+0.76)	2	**88.30/88.44 (+0.14)**	
	82.88/82.75 (-0.13)	4	**90.58/90.60 (+0.02)**	
Elastic Transform	**84.00/84.05 (+0.05)**	1	**69.63/70.68 (+1.05)**	Shot Noise
	84.10/84.63 (+0.53)	2	**70.96/71.09 (+0.13)**	
	85.54/85.52 (-0.02)	4	**73.23/73.32 (+0.09)**	
Fog	**84.29/85.22 (+0.93)**	1	**84.47/84.71 (+0.24)**	Snow
	85.09/85.75 (+0.66)	2	**84.72/84.72 (+0.00)**	
	87.25/87.53 (+0.28)	4	**86.90/87.18 (+0.28)**	
Frost	**82.35/82.66 (+0.31)**	1	88.20/88.03 (-0.17)	Spatter
	83.04/83.27 (+0.23)	2	**87.58/87.71 (+0.13)**	
	85.46/85.70 (+0.24)	4	89.14/89.02 (-0.12)	
Gaussian Blur	**73.33/74.05 (+0.72)**	1	**71.77/72.66 (+0.89)**	Speckle Noise
	74.79/75.84 (+1.05)	2	72.66/72.64 (-0.02)	
	76.08/76.25 (+0.17)	4	**75.10/75.37 (+0.27)**	
Gaussian Noise	**61.35/62.71 (+1.36)**	1	**75.98/76.68 (+0.70)**	Zoom Blur
	63.05/62.71 (-0.34)	2	**77.60/78.75 (+1.15)**	
	64.34/64.89 (+0.55)	4	**78.68/79.14 (+0.46)**	
Impulse Noise	**67.84/68.10 (+0.26)**	1	**57.86/58.26 (+0.40)**	Glass Blur
	65.83/65.36 (-0.47)	2	56.09/55.81 (-0.28)	
	65.98/66.93 (+0.95)	4	**59.43/60.46 (+1.03)**	
JPEG Compression	**83.32/83.53 (+0.21)**	1	**78.11/78.55 (+0.44)**	* Avg
	83.93/84.00 (+0.07)	2	**78.52/78.84 (+0.32)**	
	84.60/84.76 (+0.16)	4	**80.53/80.78 (+0.25)**	

D Limitations, Future Work, and Broader Impacts

D.1 Limitations

Despite the promising enhancements introduced by our proposed TDFormer with top-down feedback structure for spiking neural networks, several limitations remain. First, the current feedback mechanism is specifically designed for Transformer-based architectures and may not be directly applicable to CNN-based SNNs, limiting its architectural generalizability. Second, our evaluation has so far been limited to image classification tasks, which may not fully reflect the method's effectiveness in other domains such as object detection[42], semantic segmentation[43], and NLP tasks[44].

D.2 Future Work

Future work could focus on generalizing the proposed TDFormer architecture to other network backbones, such as CNN-based spiking neural networks, thereby improving its architectural compatibility

and deployment flexibility. In addition, extending the evaluation of TDFormer to tasks such as object detection, semantic segmentation, and natural language processing would provide deeper insights into its generalization capacity across diverse domains and data modalities. Moreover, we observe that the proposed top-down feedback structure increases the diversity of spike patterns[10], which may contribute to the observed performance gains. Investigating the underlying relationship between spike diversity and task performance remains an important direction for future research.

D.3 Broader Impacts

This paper focuses on the fundamental research of spiking neural networks, introducing a top-down feedback structure that aims to enhance their performance. Generally, there are no negative societal impacts in this work.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:The abstract and introduction clearly state our contributions in the field of spiking neural networks, including the discovery of limitation caused by SNN dynamics and the inspired improvement methods.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We discussed the limitations of the proposed method in the appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides a complete proof of the proposed viewpoint and method.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The method section provides a detailed introduction to the method proposed in this paper, which can be reproduced by referring to the experiment section and submitted code.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The dataset used in this article is publicly available, and the code will be made public to ensure that others can reproduce the experimental results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The appendix of the paper provides detailed experimental settings.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper accurately presents error bars for the execution speed benchmark. Notably, our experiments involved comparing our method's optimal performance with other approaches

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The article provides the resource cost required for conducting experiments, further detailed information is provided in the code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research in this paper adheres to the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This paper focuses on the fundamental research of spiking neural networks, there are no negative societal impacts in this work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper focuses on the fundamental research of spiking neural networks, which does not involve the development or release of data or models that have a high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The creators or original owners of the assets (such as code, data, and models) used in this paper have been properly credited. Their contributions have been explicitly mentioned in an appropriate manner. Additionally, the license and terms of use for each asset have been explicitly stated and adhered to, including obtaining any necessary permissions or authorizations.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The experimental code will be made openly accessible, along with the necessary documents to facilitate reproducibility of the experimental results and utilization of the code for future work.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The LLM was only used for translation purposes and did not affect the core scientific methodology, analysis, or originality of the research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.