STF: Shallow-Level Temporal Feedback to Enhance Spiking Transformers

Zeqi Zheng[1,2*], Zizheng Zhu[2,6*], Yingchao Yu[4,2], Yanchen Huang[2,3], Changze Lv[5], Junfeng Tang[1,2], Zhaofei Yu[7], Yaochu Jin[2†]
[1]Zhejiang University [2]Westlake University [3]Nanjing University [4]Donghua University
[5]Fudan University [6]University of Electronic Science and Technology of China [7]Peking University
{zhengzeqi, zhuzizheng, jinyaochu}@westlake.edu.cn

Abstract

Transformer-based Spiking Neural Networks (SNNs) suffer from a great performance gap compared to floating-point Artificial Neural Networks (ANNs) due to the binary nature of spike trains. Recent efforts have introduced deep-level feedback loops to transmit high-level semantic information to narrow this gap. However, these designs often span multiple deep layers, resulting in costly feature transformations, higher parameter overhead, increased energy consumption, and longer inference latency. To address this issue, we propose Shallow-level Temporal Feedback (STF), a lightweight plug-and-play module for the encoding layer, which consists of Temporal-Spatial Position Embedding (TSPE) and Temporal Feedback (TF). Extensive experiments show that STF consistently improves performance across various Transformer-based SNN backbones on static datasets, including CIFAR-10, CIFAR-100, and ImageNet-1K, under different spike timestep settings. Further analysis reveals that STF enhances the diversity of spike patterns, which is key to performance gain. Moreover, evaluations on adversarial robustness and temporal sensitivity confirm that STF outperforms direct coding and its variants, highlighting its potential as a new spike encoding scheme for static scenarios. Our code will be released upon acceptance.

1 Introduction

Spiking Neural Networks (SNNs), as the third generation of neural networks, are bio-inspired models that simulate information communication between biological neurons through discrete binary spike trains (Maass 1997). Their event-driven nature enables greater energy efficiency and biological plausibility, making them a promising alternative to traditional Artificial Neural Networks (ANNs) (Rathi et al. 2023). However, the binary nature of spike trains limits the network's representational capacity, leading to a performance gap remaining relative to floating-point ANNs (Eshraghian et al. 2023; Wang et al. 2025). Recent advances in training methods such as Spatio-Temporal Backpropagation (STBP) (Wu et al. 2018), Backpropagation Through Time (BPTT) (Werbos 1990), and the emergence of Transformer-based SNNs have significantly improved performance on various downstream tasks, including image classification (Zhou et al. 2023, 2024), semantic segmentation (Yao et al. 2023, 2025), and object detection (Luo et al. 2024; Yao et al. 2024). However, a noticeable performance gap still persists.

To further narrow this gap, researchers have incorporated biologically inspired mechanisms into SNN design. Inspired by theta rhythms and phase precession in the hippocampus, Zhang et al. improved spatio-temporal integration in SNNs (Zhang et al. 2025). MSD simulated bypass circuits in the optic nerve nucleus for multi-scale spike fusion (Li et al. 2025), while STNet leveraged joint spatio-temporal processing mechanisms observed in the brain to enhance high-dimensional data modeling (Duan et al. 2025). Despite their effectiveness, these approaches, based on convolutional neural networks (CNNs), often rely on the fine-grained design of specific neural mechanisms, which limits their scalability in complex tasks. By contrast, models such as SpiLiFormer (Zheng et al. 2025), TDFormer (Zhu et al. 2025b), and SpikingVTG (Bal et al. 2024), inspired by top-down regulation in the brain, introduced feedback loops into Transformer-based SNNs to transmit high-level semantic information without modifying the core architecture. Their strong empirical performance and architectural flexibility have drawn increasing research attention (Wang et al. 2023; Li et al. 2024).

However, as shown in Figure 1(a), existing deep-level feedback loops are explicitly constructed in the network's spatial domain, often being placed in deeper layers or spanning multiple hierarchical levels (Bal et al. 2024; Zheng et al. 2025). Although this design effectively improves model performance, it inevitably introduces two forward passes, leading to increased energy consumption and inference latency. Moreover, as the network depth increases, feature dimensionality rises substantially, and processing feedback on high-dimensional features further increases the model's parameter burden. This raises a natural question: *Why not construct temporal-wise feedback loops directly in the shallow layers of the network?* Such a design could retain the benefits of feedback mechanisms while mitigating the computational overhead caused by high-dimensional feature transformations and minimizing the increase in energy consumption and inference latency caused by repeated forward passes.

Building on this insight, we propose a Shallow-level Temporal Feedback (STF) module that can be seamlessly integrated into the encoding layer of existing models. As illustrated in Figure 1(b), STF comprises Temporal-

* Equal contribution. † Corresponding author.

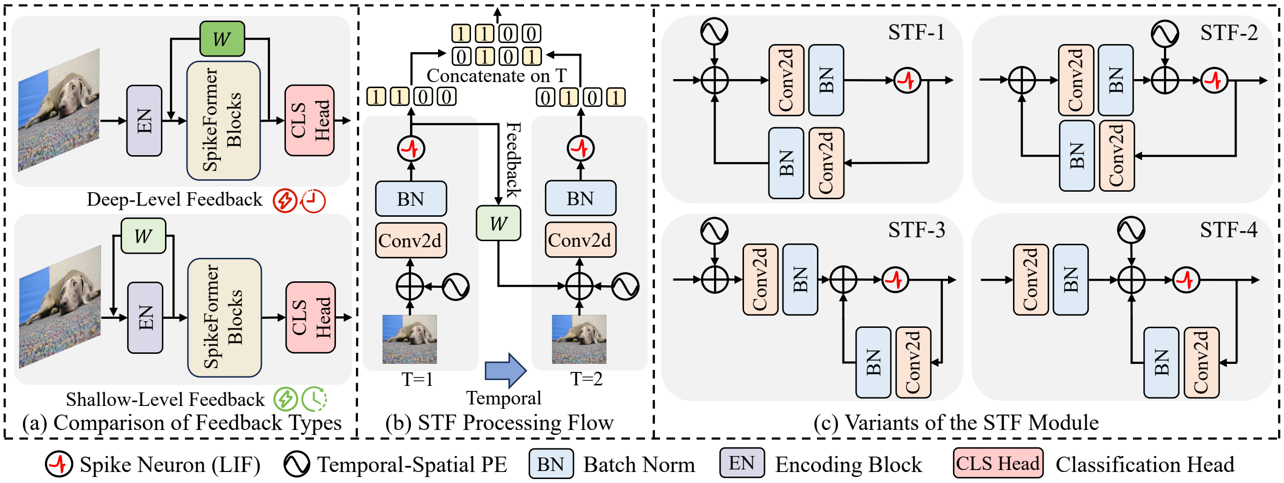

Figure 1: Illustration of the proposed Shallow-level Temporal Feedback (STF) module. (a) Comparison between deep-level and shallow-level feedback. The former incurs higher energy consumption, longer inference latency, and more complex feature transformations. W denotes transformation weights, and darker colors represent higher feature dimensionality. (b) STF processing flow across timesteps. (c) Four architectural variants of the STF module.

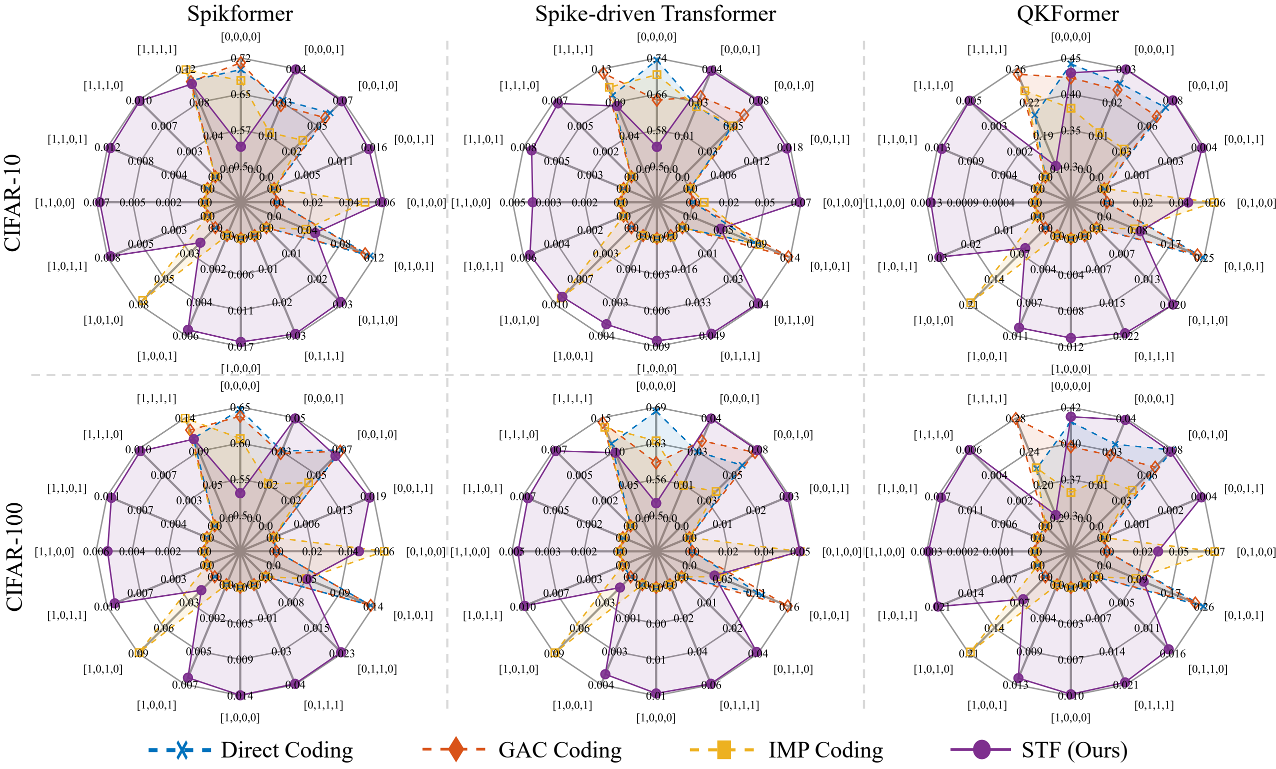

Figure 2: Distribution of spike patterns under different encoding schemes for various Transformer-based SNN backbones at $T = 4$. Both direct coding and GAC coding tend to favor specific spike patterns, resulting in limited diversity. Although IMP coding mitigates this bias to some extent, it still fails to activate the full range of patterns compared to STF.

Spatial Position Embedding (TSPE) and Temporal Feedback (TF), which together introduce temporal feedback efficiently without compromising the spike-driven nature of SNNs. We evaluate the impact of STF on several mainstream Transformer-based SNN architectures, including Spikformer (Zhou et al. 2023), Spike-driven Transformer (SDT) (Yao et al. 2023), and QKFormer (Zhou et al. 2024). Experimental results demonstrate that STF consistently improves accuracy across different spike timesteps on static datasets such as CIFAR-10, CIFAR-100, and ImageNet-1K. Notably, on the large-scale ImageNet-1K dataset, STF improves accuracy by 0.75%, with only a 0.32% increase in parameters, 7.2% more energy consumption, and 6.33% higher inference latency. Further analysis, including visualizations (Figure 2), quantitative comparisons based on spike entropy (Table 1), and theoretical analysis (Section 4.3), reveals that the key factor behind STF's performance gains lies in its ability to enhance the diversity of spike patterns. This enables richer spike-based representations, which are critical for static visual tasks. In addition, we conduct a detailed evaluation of STF against direct coding and its variants, focusing on model performance, robustness, and temporal information preservation. The comparative experiments suggest that STF represents a promising new spike encoding scheme, with the potential to replace the long-standing dominance of direct coding in static data scenarios. Our main contributions can be summarized as follows:

- We propose the STF module, a plug-and-play temporal feedback mechanism applied at the encoding layer. It preserves the spike-driven nature of SNNs by introducing feedback at low-dimensional feature layers and directly modulating spiking neurons. STF consistently improves model performance under varying spike timesteps across CIFAR-10, CIFAR-100, and ImageNet-1K, while incurring only modest increases in model parameters, energy consumption, and inference latency.
- Through detailed visualizations, quantitative comparisons, and theoretical analysis, we identify that the key to STF's performance improvement lies in its ability to enhance the diversity of spike patterns, thereby improving spike-based representation. Besides, we systematically evaluate four architectural variants of STF and determine the optimal configuration.
- We perform a comprehensive evaluation of STF in comparison with direct coding and its variants, with respect to model performance, robustness, and the richness of temporal information. The detailed experimental results indicate that STF has strong potential as a novel spike encoding scheme for static data scenarios, with the capacity to replace the long-standing direct coding.

2 Related Work

2.1 Transformer-based SNNs

Due to their strong performance across a wide range of tasks, Transformer-based SNNs have garnered increasing attention from researchers, leading to continuous architectural innovations. Spikformer (Zhou et al. 2023) was the first to integrate the Transformer architecture with SNNs, replacing the softmax operation with binary spike-based representations to effectively reduce energy consumption. Spike-driven Transformer (Yao et al. 2023, 2024) further advanced the paradigm by substituting the dot-product attention with Hadamard product operations and redesigning residual connections based on membrane potentials to establish a fully spike-driven framework. QKFormer (Zhou et al. 2024) and SpikeGPT (Zhu et al. 2025a) introduced linear attention mechanisms to further reduce computational complexity, while E-SpikeFormer (Yao et al. 2025) adopted integer-based training approaches to enhance both training efficiency and model performance. Despite these advances, most existing Transformer-based SNNs still rely on ANN-inspired designs, neglecting bio-inspired principles in architectural design that may unlock further performance gains.

2.2 Feedback Mechanisms in SNNs

Originally rooted in cybernetics (Wiener 2019) and later adopted in neuroscience (Lamme and Roelfsema 2000), feedback mechanisms provide strong self-adaptive regulation and have been widely integrated into SNNs. BackEISNN (Zhao, Zeng, and Li 2022) and RSNN (Xu et al. 2024) applied recurrent feedback to all neurons and introduced additional hidden states. BRP (Zhang et al. 2021) used cortical reward signal transmission to feed label information back to all hidden layers, enhancing accuracy through local pseudo-gradients. STSF (He et al. 2025) and STNet (Duan et al. 2025) adopted spatio-temporal feedback across cortical hierarchies to mitigate feature degradation and temporal loss. Besides, SpikingVTG (Bal et al. 2024) was the first to utilize high-layer firing rates to modulate attention in intermediate layers, improving performance in video-text temporal grounding. SpiLiFormer (Zheng et al. 2025) and TDFormer (Zhu et al. 2025b) incorporated lateral inhibition and top-down modulation, respectively, to address attention dispersion. However, these approaches primarily rely on structurally deep-level feedback loops, where high-level semantic information is propagated back to multiple or even all hidden layers through high-dimensional feature transformations. In contrast, the shallow-level and temporal feedback mechanism has not yet been thoroughly investigated.

3 Methods

This section introduces the spiking neuron model employed in our study and elaborates on the proposed Shallow-level Temporal Feedback (STF) module.

3.1 Spiking Neuron

Following prior related research (Zhou et al. 2023; Yao et al. 2023; Zhou et al. 2024; Yao et al. 2025), we adopt the leaky integrate-and-fire (LIF) neuron (Maass 1997) as the minimal processing unit. The dynamics of the LIF neuron are described by the following equations:

$$H[t] = U[t-1] + \frac{1}{\tau_{\mathrm{m}}}(I[t] - (U[t-1] - U_{\mathrm{reset}})), \quad (1)$$

$$S[t] = \Theta(H[t] - U_{\mathrm{th}}), \quad (2)$$

$$U[t] = H[t](1 - S[t]) + U_{\mathrm{reset}}S[t], \quad (3)$$

where $H[t]$ denotes the intermediate membrane state at time t, which is influenced by the input current $I[t]$, the previous membrane potential $U[t-1]$, and the reset voltage U_{reset}. The output spike $S[t]$ is generated when $H[t]$ exceeds the threshold U_{th}, determined by the Heaviside function $\Theta(\cdot)$, after which the membrane potential is reset to U_{reset}.

Methods	Architecture	Acc (%) / *Spike Entropy*							
		CIFAR-10				CIFAR-100			
		T=2	T=4	T=6	T=8	T=2	T=4	T=6	T=8
Spikformer	Spikformer-4-384	93.20 *0.964*	95.19 *1.442*	95.14 *1.530*	95.24 *1.664*	75.17 *1.203*	77.86 *1.590*	78.17 *1.774*	78.22 *1.860*
Spikformer w/ GAC	Spikformer-4-384	93.1 *0.966*	94.63 *1.396*	95.03 *1.515*	95.17 *1.627*	75.17 *1.244*	77.16 *1.609*	77.90 *1.760*	78.54 *1.821*
Spikformer w/ IMP	Spikformer-4-384	94.08 *1.406*	94.92 *1.670*	95.42 *1.713*	95.39 *1.739*	75.52 ***1.492***	77.34 *1.883*	78.16 *2.019*	78.73 *2.114*
Spikformer w/ STF	Spikformer-4-384	**94.95** ***1.463***	**95.61** ***2.573***	**95.85** ***4.369***	**95.99** ***4.932***	**76.94** *1.479*	**78.00** ***2.597***	**78.87** ***4.417***	**79.14** ***5.240***
SDT	SDT-2-512	95.01 *0.877*	95.60 *1.318*	96.11 *1.527*	96.24 *1.626*	77.47 *1.104*	78.40 *1.482*	79.43 *1.648*	79.87 *1.733*
SDT w/ GAC	SDT-2-512	95.18 *1.084*	95.69 *1.571*	96.17 *1.630*	95.97 *1.698*	77.65 *1.200*	79.10 *1.710*	79.76 *1.773*	80.04 *1.928*
SDT w/ IMP	SDT-2-512	95.15 *1.073*	95.71 *1.501*	96.09 *1.839*	96.21 *1.903*	77.8 ***1.396***	79.22 *1.797*	79.65 *1.952*	79.98 *2.019*
SDT w/ STF	SDT-2-512	**95.24** ***1.318***	**95.86** ***2.514***	**96.23** ***3.600***	**96.41** ***4.484***	**78.21** *1.381*	**79.44** ***2.609***	**80.34** ***3.720***	**80.61** ***4.439***
QKFormer	HST-4-384	95.79 *1.483*	96.18 *1.911*	96.37 *2.080*	96.35 *2.116*	79.79 *1.490*	81.15 *1.956*	81.35 *2.114*	81.64 *2.134*
QKFormer w/ GAC	HST-4-384	95.76 *1.469*	96.16 *1.906*	96.32 *2.066*	96.37 *2.120*	79.92 *1.490*	80.88 *1.920*	81.40 *2.056*	81.59 *2.139*
QKFormer w/ IMP	HST-4-384	95.28 ***1.810***	96.03 *2.245*	96.35 *2.412*	96.31 *2.398*	80.01 ***1.833***	81.10 *2.325*	81.30 *2.389*	81.69 *2.488*
QKFormer w/ STF	HST-4-384	**96.02** *1.676*	**96.33** ***2.809***	**96.51** ***4.028***	**96.61** ***4.862***	**80.07** *1.722*	**81.26** ***2.819***	**81.51** ***4.054***	**81.89** ***5.076***

Table 1: Performance comparison of our method on 3 backbones: Spikformer, Spike-driven Transformer (SDT), and QKFormer, on the CIFAR datasets across various time steps. Bold indicates the best result, and underlined denotes the second-best.

3.2 Shallow-Level Temporal Feedback Module

As illustrated in Figure 1(b), the STF module consists of two key components, namely Temporal-Spatial Position Embedding (TSPE) and Temporal Feedback (TF).

Temporal-Spatial Position Embedding To effectively capture temporal and spatial dynamics in spike trains, position embedding plays a crucial role in SNNs (Lv et al. 2024). In our formulation, we represent the input image sequence as $I \in \mathbb{R}^{T\times C\times H\times W}$, where T is the number of spike timesteps, and C, H, and W refer to the number of channels, image height, and image width, respectively. In STF-1 and STF-3 configurations, as shown in Figure 1(c), the TSPE mechanism is formulated as follows:

$$I_{\text{embed}}[t] = W_{\text{ConvBN}} \cdot (X_{\text{TPE}}[t] + I[t]), \tag{4}$$

where $X_{\text{TPE}} \in \mathbb{R}^{T\times C\times H\times W}$ represents a spatio-temporal positional embedding, and W_{ConvBN} denotes the weights of a 2D convolution layer followed by batch normalization. Motivated by the inherent similarity between spike trains and sequential video signal processing, a 3D trigonometric positional strategy is adopted to initialize X_{TPE}, following common practice in video transformers (Liu et al. 2022; Liang et al. 2024). To explore the optimal location for incorporating TSPE, two additional variants, STF-2 and STF-4, are introduced. In these configurations, the embedding formulation is modified as follows:

$$I_{\text{embed}}[t] = X_{\text{TPE}}[t] + W_{\text{ConvBN}} \cdot I[t]. \tag{5}$$

Temporal Feedback To maintain the spike-driven nature of the network, TF directly feeds the spike output from the previous time step $S[t-1]$ into the input at the current time step $S[t]$. This forms a feedback loop that captures temporal dependencies. In contrast to conventional recurrent SNNs (Zhao, Zeng, and Li 2022; Xu et al. 2024), TF avoids the use of auxiliary hidden states or non-spiking activations. The feedback operation in STF-1 and STF-2, as illustrated in Figure 1(c), is formulated as:

$$I'[t] = I[t] + W_{\text{TF}} \cdot S[t-1], \tag{6}$$

where W_{TF} denotes a learnable transformation comprising convolution and batch normalization, and $I'[t]$ is the input at time t after integrating the feedback signal. To further explore alternative feedback strategies, STF-3 and STF-4 adopt a more direct approach, in which the previous output is used to update the membrane potential of the LIF neuron. This process is described by the following equation:

$$H[t] = (1 - \frac{1}{\tau_m})U[t-1] + I[t] + W_{\mathrm{TF}} \cdot S[t-1], \quad (7)$$

where all variables strictly adhere to the definitions provided in Equations 1–3.

4 Experiments

In this section, we conduct a series of experiments to investigate the following research questions: **RQ1:** Can the proposed STF module effectively enhance the performance of Transformer-based SNNs on image classification tasks across varying spike timesteps, while maintaining modest increases in model parameters, energy consumption, and inference latency? **RQ2:** What are the underlying factors that contribute to the performance gains achieved by STF? **RQ3:** Can STF serve as a novel spiking encoding scheme that replaces the existing direct coding and its variants?

4.1 Datasets and Baseline Models

To answer the above research questions, we select representative Transformer-based SNNs, including Spikformer (Zhou et al. 2023), Spike-driven Transformer (SDT) (Yao et al. 2023), and QKFormer (Zhou et al. 2024), as baseline models. Following (Yao et al. 2024; Zhou et al. 2024), we use three standard image classification datasets, CIFAR-10 (Krizhevsky et al. 2009), CIFAR-100 (Krizhevsky et al. 2009), and ImageNet-1K (Deng et al. 2009), to evaluate the performance gains introduced by STF. Details of these datasets are provided in Appendix A. In addition, to further investigate **RQ3**, we compare STF with existing spiking encoding schemes, including direct coding (Wu et al. 2019) and its variants such as GAC coding (Qiu et al. 2024) and IMP coding (Shen et al. 2024). The training configuration and hyperparameter settings are detailed in Appendix B.

4.2 Effectiveness of STF on Image Classification Tasks

Before this, we conduct a systematic evaluation of the four STF variants defined in Section 3.2 and illustrated in Figure 1(c) to explore the optimal STF architecture. As shown in Figure 3, STF-4 consistently achieves the best performance across different datasets and spike timesteps. Therefore, unless otherwise specified, we adopt STF-4 as the default configuration throughout this paper.

To answer **RQ1**, we train various models from scratch and evaluate the performance improvements introduced by STF across three datasets and three Transformer-based SNN backbones. As shown in Table 1, STF consistently outperforms the corresponding baselines on both CIFAR-10 and CIFAR-100 across four different timesteps. Specifically, on CIFAR-100, STF improves performance by 1.77% with Spikformer at $T = 2$ and by 1.04% with SDT at $T = 4$.

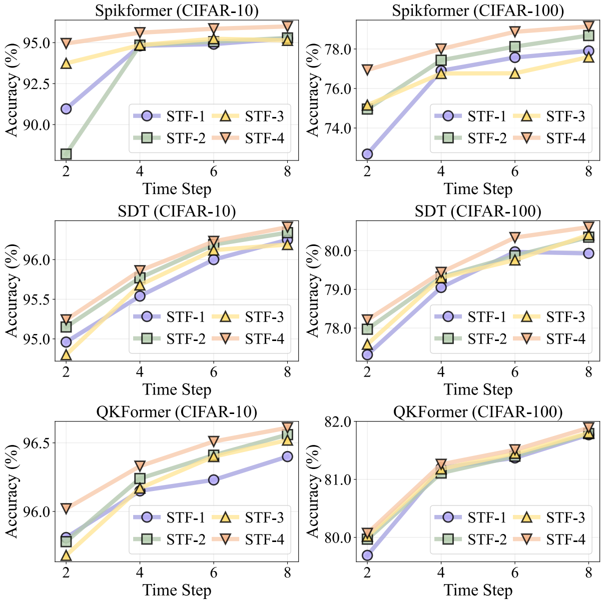

Figure 3: Performance comparison of four STF variants (Figure 1(c)) on CIFAR datasets across various timesteps.

We further evaluate STF on the more challenging ImageNet-1K dataset to assess its performance improvement over SDT and QKFormer, along with the additional parameter and energy consumption overhead. As shown in Table 2, STF improves accuracy by 0.6% to 0.75%. Specifically, for SDT with the timestep of $T = 4$, STF achieves a 0.6% accuracy gain while introducing only a 0.17% increase in parameters and a 12.08% increase in energy consumption. Similarly, for QKFormer with $T = 4$ and a standard input resolution of 224, STF improves accuracy by 0.75% with just a 0.32% increase in parameters and approximately 7.2% more energy consumption. Furthermore, STF allows QKFormer, which uses only 37.5% of the parameters of the best-performing E-SpikeFormer, to reach comparable accuracy (86.2%) with only half the number of timesteps.

It is worth noting that the temporal feedback loop in STF introduces challenges for parallel computation. To quantify the impact of the feedback structure on the inference efficiency, we analyze the inference latency introduced by STF. Although feedback mechanisms inherently add some computational overhead, STF employs a shallow-level feedback design that exhibits significant advantages over deep-level feedback structures such as SpiLiFormer (Zheng et al. 2025). As shown in Table 3, the latency overhead introduced by STF remains below 10% and is often reduced by half or more compared to SpiLiFormer.

4.3 Analyzing the Performance Gains of STF

To answer **RQ2**, we analyze the distribution of spike trains from the same LIF neuron in the encoding layer, both with and without STF, on the CIFAR datasets. Specifically, we visualize the output patterns under STF and the direct coding scheme for comparison. As shown in Figure 2, taking $T = 4$ as an example, STF alleviates the imbalance in the distribution of spike patterns to varying degrees between different main SNN models. It significantly increases the num-

Methods	SNN	Architecture	Input Size	Param(M)	Power(mJ)	Time Step	Top-1 Acc(%)
ViT	✗	ViT-B/16	224	86.59	254.84	1	77.90
Swin	✗	Swin Transformer-B	224	87.77	70.84	1	83.50
	✗	Swin Transformer-B	384	87.77	216.20	1	84.50
SEW ResNet	✓	SEW-ResNet-152	224	60.19	12.89	4	69.26
Spikformer	✓	Spikformer-8-768	224	66.34	21.48	4	74.81
Spikingformer	✓	Spikingformer-8-768	224	66.34	13.68	4	75.85
E-SpikeFormer	✓	E-SpikeFormer-12-768	224	173.0	35.6	$4^\star$	84.70
		E-SpikeFormer-12-768	224	173.0	54.7	$8^\star$	85.10
		E-SpikeFormer-12-768**	384	173.0	-	$8^\star$	86.20
SDT	✓	SDT-6-512	224	23.37	3.56	4	72.41†
SDT w/ STF	✓	SDT-6-512	224	23.41	3.99	4	**73.01**
QKFormer	✓	HST-10-768	224	64.96	38.91	4	84.22
		HST-10-768*	288	64.96	64.27	4	85.20
		HST-10-768**	384	64.96	113.64	4	85.65
QKFormer w/ STF	✓	HST-10-768	224	65.17	41.75	4	**84.97**
		HST-10-768*	288	65.17	68.96	4	**85.89**
		HST-10-768**	384	65.17	121.77	4	**86.26**

Table 2: Performance comparison on ImageNet-1K. Numbers marked with † indicate our implementation based on publicly available code. $*$ and $**$ indicate inference resolutions of 288^2 and 384^2, respectively. $\star$ represents a quantized training method (Yao et al. 2025) using integer-only activations, which are later mapped to spike trains for inference.

ber of activated spike patterns and successfully covers all 16 possible binary combinations. A more diverse spike pattern indicates greater information capacity, facilitating more effective information propagation through the network and ultimately enhancing model representation and performance. Therefore, we consider that the improved diversity of spike patterns is a key contributor to STF performance gains.

To quantitatively measure this property, we adopt spike entropy as a diversity metric, inspired by previous work in neuroscience and information theory (Luczak 2024). The formal definition is provided in Appendix H. As reported in Table 1, STF consistently improves the diversity of spike patterns across all models and datasets, with greater gains at higher spike timesteps.

We further provide a theoretical analysis to explain why STF can improve the diversity of spike patterns. For static datasets, the input image I is duplicated across T spike timesteps to form a sequence $I = [I[1], I[2], ..., I[T]]$, where $I[t] = I$ for all $t \in 1, 2, ..., T$. The spike generation time SG_t of a neuron in the encoding layer without STF can be formulated as follows (the detailed derivation is provided in Appendix G):

$$SG_t = \left\lceil \log_\tau(1 - \frac{U_{\text{th}}(1-\tau)}{I}) \right\rceil . \tag{8}$$

This expression indicates that under fixed values of leakage decay factor τ, U_{th}, and a repeated static input I, baseline models tend to activate only a limited set of specific spike patterns. In contrast, STF incorporates both the current input $I[t]$ and the previous spike output $S[t-1]$ into Equation 7, allowing SG_t to be triggered over a broader temporal range, rather than being constrained by the constant input I. This effectively expands the expressive space of spike patterns, leading to enhanced diversity in spike patterns. In addition, we conduct detailed ablation studies on TSPE and TF, as presented in Appendix J. The results demonstrate that both components significantly contribute to improving the diversity of spike patterns and overall model performance, with TF exhibiting a more pronounced effect.

4.4 Comparison with Existing Spiking Encoding Schemes

To answer **RQ3**, we evaluate two representative spike encoding variants, GAC coding (Qiu et al. 2024) and IMP coding (Shen et al. 2024), in terms of model performance and the diversity of spike patterns on the CIFAR datasets. As illustrated by the spike pattern visualizations in Figure 2 and the results in Table 1, GAC provides performance improvements for certain models. However, it exhibits a strong preference for specific spike patterns, similar to the direct coding scheme, resulting in a concentrated activation distribution. In contrast, IMP demonstrates a more pronounced effect in increasing the diversity of spike patterns, which contributes to performance gains. Nevertheless, it still fails to activate the full range of possible spike combinations, indicating limited diversity. STF, on the other hand, consistently outperforms both GAC and IMP in terms of the diversity of the spike patterns, leading to further improvements in model accuracy. Although STF yields slightly lower spike entropy values than IMP under certain timestep configurations, its advantage becomes increasingly evident as the number of timesteps grows, demonstrating better scalability and stability.

An effective encoding scheme should not only enhance model performance but also improve robustness against adversarial perturbations and capture richer temporal information (Kim et al. 2022; Eshraghian et al. 2023). To evaluate whether STF meets these criteria, we first employ white-box

Methods	Inference Time per Sample (ms)							
	CIFAR-10				CIFAR-100			
	T=2	T=4	T=6	T=8	T=2	T=4	T=6	T=8
Spikformer	0.384	0.499	0.718	0.891	0.389	0.525	0.725	0.932
Spikformer w/ STF	0.404 **(+5.21%)**	0.541 **(+8.42%)**	0.777 **(+8.22%)**	0.962 **(+7.97%)**	0.395 **(+1.54%)**	0.532 **(+1.33%)**	0.749 **(+3.31%)**	0.964 **(+3.34%)**
SDT	0.319	0.521	0.711	0.886	0.304	0.503	0.692	0.884
SDT + STF	0.337 **(+5.64%)**	0.550 **(+5.57%)**	0.745 **(+4.78%)**	0.928 **(4.74%)**	0.317 **(+4.28%)**	0.531 **(+5.57%)**	0.728 **(+5.2%)**	0.919 **(+3.96%)**
QKFormer	0.715	0.726	1.012	1.324	0.547	0.763	0.984	1.270
QKFormer w/ STF	0.769 **(+7.55%)**	0.771 **(+6.20%)**	1.083 **(+7.02%)**	1.414 **(+6.8%)**	0.592 **(+8.23%)**	0.788 **(+3.28%)**	1.035 **(+5.18%)**	1.387 **(+9.21%)**
SpiLiFormer w/o feedback	0.726	0.766	0.995	1.313	0.556	0.697	0.999	1.303
SpiLiFormer	0.919 **(+26.58%)**	0.932 **(+21.67%)**	1.179 **(+18.49%)**	1.521 **(+15.84%)**	0.698 **(+25.54%)**	0.858 **(+23.1%)**	1.228 **(+22.92%)**	1.505 **(+15.50%)**

Table 3: Comparison of inference latency overhead between the shallow-level STF module and traditional deep-level feedback on CIFAR datasets. Corresponding results on ImageNet-1K are provided in Appendix F.

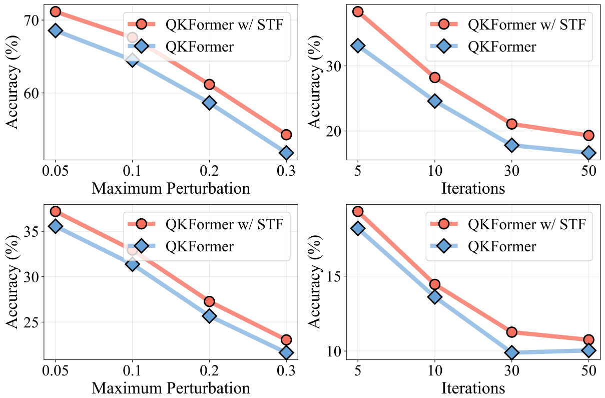

Figure 4: Adversarial robustness of QKFormer (4 timesteps) with and without STF on CIFAR-10 (top) and CIFAR-100 (bottom) under FGSM (left) and PGD (right) attacks.

adversarial attacks, including FGSM (Goodfellow, Shlens, and Szegedy 2014) and PGD (Madry et al. 2017), to comprehensively assess the impact of STF on model robustness under various settings, compared to direct coding. As shown in Figure 4, STF significantly improves robustness without compromising clean accuracy. Additional results of the robustness evaluation are provided in Appendix I. Then, following the previous study (Bu et al. 2023), we employ a random number generator to shuffle the spike trains of neuron outputs, in order to examine whether STF captures richer temporal information. As shown in Table 4, STF-based models are more sensitive to temporal disruptions compared to those using direct coding. Combined with the findings of Bu et al. (Bu et al. 2023), these results confirm that STF effectively encodes richer temporal information.

Besides, we further assess the applicability of STF to neuromorphic datasets. As shown in Table 5 in the appendix, STF leads to performance degradation on CIFAR10-DVS (Li et al. 2017) across different timestep settings, which aligns with our expectations. As analyzed in Section 4.3, the effectiveness of STF in static tasks stems from its ability to break the limitation of Equation 8, thereby enhancing the diversity of the spike patterns. By contrast, event-based data inherently contain temporal structure, and the use of STF may interfere with this built-in temporal information, potentially degrading performance. Therefore, while STF shows promise as a new spike encoding scheme that can replace direct coding and its variants, its advantage appears to be limited to static scenarios.

Methods	T	Clean Acc(%)	Shuffled Acc(%)

QKFormer	2
	4
	6
	8
		79.79
		81.15
		81.35
		81.64
			79.01(-0.78)
			80.38(-0.77)
			80.76(-0.59)
			81.12(-0.52)

QKFormer w/ STF	2
	4
	6
	8
		80.07
		81.26
		81.51
		81.89
			79.10(-0.97)
			80.06(-1.2)
			80.76(-0.75)
			81.05(-0.84)

Table 4: Model performance comparison before and after applying spike shuffling.

5 Conclusion

In this paper, we propose Shallow-level Temporal Feedback (STF), a lightweight module that can be seamlessly integrated into the encoding layer of SNNs. Unlike deep-level feedback loops, STF avoids high-dimensional transformations and redundant forward passes. Experimental results show that STF consistently improves the performance of various Transformer-based SNN backbones under different timestep settings, with only a modest increase in the model size, energy consumption, and inference latency. Further analysis reveals that the performance gain can be primarily attributed to the enhanced diversity of spike patterns. Moreover, we compare STF with direct coding and its variants in terms of adversarial robustness and temporal information, showing that STF is promising as a novel spike encoding scheme for static scenarios. However, additional experiments reveal that STF exhibits suboptimal performance on the neuromorphic dataset. In future work, we will aim to improve its adaptability to event-based scenarios.

References

Bal, M.; Matejek, B.; Jha, S.; and Cobb, A. D. 2024. SpikingVTG: Saliency Feedback Gating Enabled Spiking Video Temporal Grounding. In *Workshop on Machine Learning and Compression, NeurIPS*.

Bu, T.; Ding, J.; Hao, Z.; and Yu, Z. 2023. Rate gradient approximation attack threats deep spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7896–7906.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

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

Duan, D.; Liu, P.; Hui, B.; and Wen, F. 2025. Brain-Inspired Online Adaptation for Remote Sensing with Spiking Neural Network. *IEEE Transactions on Geoscience and Remote Sensing*.

Eshraghian, J. K.; Ward, M.; Neftci, E. O.; Wang, X.; Lenz, G.; Dwivedi, G.; Bennamoun, M.; Jeong, D. S.; and Lu, W. D. 2023. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9): 1016–1054.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.

He, P.; Xiao, R.; Tang, C.; Huang, S.; Lv, J.; and Tang, H. 2025. STSF: Spiking Time Sparse Feedback Learning for Spiking Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Horowitz, M. 2014. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*, 10–14. IEEE.

Hu, Y.; Tang, H.; and Pan, G. 2021. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8): 5200–5205.

Kim, Y.; Park, H.; Moitra, A.; Bhattacharjee, A.; Venkatesha, Y.; and Panda, P. 2022. Rate coding or direct coding: Which one is better for accurate, robust, and energy-efficient spiking neural networks? In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 71–75. IEEE.

Krizhevsky, A.; et al. 2009. Learning multiple layers of features from tiny images.

Lamme, V. A.; and Roelfsema, P. R. 2000. The distinct modes of vision offered by feedforward and recurrent processing. *Trends in neurosciences*, 23(11): 571–579.

Li, H.; Liu, H.; Ji, X.; Li, G.; and Shi, L. 2017. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11: 244131.

Li, W.; Wang, P.; Xiong, R.; and Fan, X. 2024. Spiking tucker fusion transformer for audio-visual zero-shot learning. *IEEE Transactions on Image Processing*.

Li, Z.; Gao, T.; An, Y.; Chen, T.; Zhang, J.; Wen, Y.; Liu, M.; and Zhang, Q. 2025. Brain-Inspired Spiking Neural Networks for Energy-Efficient Object Detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 3552–3562.

Liang, J.; Cao, J.; Fan, Y.; Zhang, K.; Ranjan, R.; Li, Y.; Timofte, R.; and Van Gool, L. 2024. Vrt: A video restoration transformer. *IEEE Transactions on Image Processing*, 33: 2171–2182.

Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; and Hu, H. 2022. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3202–3211.

Luczak, A. 2024. Entropy of neuronal spike patterns. *Entropy*, 26(11): 967.

Luo, X.; Yao, M.; Chou, Y.; Xu, B.; and Li, G. 2024. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *European Conference on Computer Vision*, 253–272. Springer.

Lv, C.; Han, D.; Wang, Y.; Zheng, X.; Huang, X.; and Li, D. 2024. Advancing spiking neural networks for sequential modeling with central pattern generators. *Advances in Neural Information Processing Systems*, 37: 26915–26940.

Maass, W. 1997. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9): 1659–1671.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Qiu, X.; Zhu, R.-J.; Chou, Y.; Wang, Z.; Deng, L.-j.; and Li, G. 2024. Gated attention coding for training high-performance and efficient spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 601–610.

Rathi, N.; Chakraborty, I.; Kosta, A.; Sengupta, A.; Ankit, A.; Panda, P.; and Roy, K. 2023. Exploring neuromorphic computing based on spiking neural networks: Algorithms to hardware. *ACM Computing Surveys*, 55(12): 1–49.

Shen, H.; Zheng, Q.; Wang, H.; and Pan, G. 2024. Rethinking the membrane dynamics and optimization objectives of spiking neural networks. *Advances in Neural Information Processing Systems*, 37: 92697–92720.

Wang, Q.; Zhang, T.; Han, M.; Wang, Y.; Zhang, D.; and Xu, B. 2023. Complex dynamic neurons improved spiking transformer network for efficient automatic speech recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 102–109.

Wang, Z.; Fang, Y.; Cao, J.; Ren, H.; and Xu, R. 2025. Adaptive calibration: A unified conversion framework of spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 1583–1591.

Werbos, P. J. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10): 1550–1560.

Wiener, N. 2019. *Cybernetics or Control and Communication in the Animal and the Machine*. MIT press.

Wu, Y.; Deng, L.; Li, G.; Zhu, J.; and Shi, L. 2018. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12: 331.

Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Xie, Y.; and Shi, L. 2019. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 1311–1318.

Xu, Q.; Fang, X.; Li, Y.; Shen, J.; Ma, D.; Xu, Y.; and Pan, G. 2024. Rsnn: Recurrent spiking neural networks for dynamic spatial-temporal information processing. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 10602–10610.

Yao, M.; Hu, J.; Hu, T.; Xu, Y.; Zhou, Z.; Tian, Y.; Bo, X.; and Li, G. 2024. 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*.

Yao, M.; Hu, J.; Zhou, Z.; Yuan, L.; Tian, Y.; Xu, B.; and Li, G. 2023. Spike-driven transformer. *Advances in neural information processing systems*, 36: 64043–64058.

Yao, M.; Qiu, X.; Hu, T.; Hu, J.; Chou, Y.; Tian, K.; Liao, J.; Leng, L.; Xu, B.; and Li, G. 2025. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, M.; Luo, X.; Wu, J.; Belatreche, A.; Cai, S.; Yang, Y.; and Li, H. 2025. Toward Building Human-Like Sequential Memory Using Brain-Inspired Spiking Neural Models. *IEEE transactions on neural networks and learning systems*.

Zhang, T.; Jia, S.; Cheng, X.; and Xu, B. 2021. Tuning convolutional spiking neural network with biologically plausible reward propagation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12): 7621–7631.

Zhao, D.; Zeng, Y.; and Li, Y. 2022. BackEISNN: A deep spiking neural network with adaptive self-feedback and balanced excitatory–inhibitory neurons. *Neural Networks*, 154: 68–77.

Zheng, Z.; Huang, Y.; Yu, Y.; Zhu, Z.; Tang, J.; Yu, Z.; and Jin, Y. 2025. SpiLiFormer: Enhancing Spiking Transformers with Lateral Inhibition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Zhou, C.; Zhang, H.; Zhou, Z.; Yu, L.; Huang, L.; Fan, X.; Yuan, L.; Ma, Z.; Zhou, H.; and Tian, Y. 2024. QKFormer: Hierarchical Spiking Transformer using QK Attention. *Advances in Neural Information Processing Systems*, 37: 13074–13098.

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

Zhu, R.-J.; Zhao, Q.; Eshraghian, J. K.; and Li, G. 2025a. SpikeGPT: Generative Pre-trained Language Model with Spiking Neural Networks. *The Thirteenth International Conference on Learning Representations*.

Zhu, Z.; Yu, Y.; Zheng, Z.; Yu, Z.; and Jin, Y. 2025b. TD-Former: A Top-Down Attention-Controlled Spiking Transformer. *arXiv preprint arXiv:2505.15840*.

Appendix

A Dataset Description

We conduct our experiments on three widely adopted datasets: ImageNet-1K (Deng et al. 2009), CIFAR-10 (Krizhevsky et al. 2009), and CIFAR-100 (Krizhevsky et al. 2009).

ImageNet-1K: The ImageNet-1K dataset, also known as the ILSVRC benchmark, is a large-scale dataset that has become a cornerstone in visual recognition tasks. It includes approximately 1.28 million training samples drawn from 1,000 object categories, with an additional 50,000 images for validation and 100,000 for testing.

CIFAR-10: As a classic benchmark in image classification, CIFAR-10 comprises 60,000 colored images at a resolution of 32 × 32 pixels. The dataset is evenly split into 10 distinct categories, with 6,000 images per class.

CIFAR-100: Sharing the same structure as CIFAR-10, CIFAR-100 presents a more complex challenge by increasing the number of classes to 100. It also consists of 60,000 color images of 32 × 32 pixels, with 600 samples per class, making it suitable for evaluating fine-grained recognition capabilities.

B Training Configuration and Hyperparameter Settings

To ensure a fair experimental comparison, we follow the experimental settings of previous works (Zhou et al. 2023; Yao et al. 2023; Zhou et al. 2024) for the three Transformer-based SNN backbones: Spikformer, Spike-driven Transformer (SDT), and QKFormer, evaluated on the datasets described in Section A.

B.1 Spikformer

We follow the protocol of Zhou et al. (Zhou et al. 2023) and train all models from scratch on the CIFAR-10 and CIFAR-100 datasets for 300 epochs using the AdamW optimizer, with an initial learning rate of $5e-4$, a weight decay of $6e-2$, and a batch size of 128. A cosine annealing learning rate scheduler is employed, with 20 warm-up epochs and 10 cool-down epochs. To ensure reproducibility, the random seed is set to 42.

B.2 SDT

Following the work of Yao et al. (Yao et al. 2023), we adopt SDT-2-512 and SDT-6-512 to train models from scratch on the CIFAR datasets and ImageNet-1K, respectively. Especially on the large-scale ImageNet-1K dataset, we adopt a data augmentation strategy during training, including RandAugment (with a magnitude of 7 and a standard deviation of 0.5), random resized cropping, horizontal flipping, and color jittering. In addition, Mixup and CutMix are applied with a probability of 0.6 to improve generalization. Furthermore, each image is augmented three times per epoch to enhance data diversity. Detailed hyperparameters and training settings are provided in Table 1.

Hyperparameter	CIFAR Datasets	ImageNet-1K
Head	8	
Layer	2	6
Embed dim	512	512
Min lr	$1e-5$	$2e-5$
Warm-up lr	$1e-6$	$1e-5$
Image size	32	224
Lr	$3e-4$	$1e-3$
Timestep	2/4/6/8	4
Batch size	64	48
Epochs	300	
Weight decay	$6e-2$	$1e-2$
Optimizer	AdamW	LAMB
Warm-up epochs	20	
Cool-down epochs	10	
Random seed	42	
Lr scheduler	Cosine	

Table 1: Training hyperparameters for SDT on the CIFAR datasets and ImageNet-1K

B.3 QKFormer

Following Zhou et al. (Zhou et al. 2024), we train the network models from scratch. Detailed hyperparameters and training settings are provided in Table 2. Additionally, for ImageNet-1K, we follow the data augmentation and training strategy of Zheng et al. (Zheng et al. 2025).

Hyperparameter	CIFAR Datasets	ImageNet-1K
Head	8	12
Layer	4	10
Embed dim	384	768
Min lr	$1e-5$	$1e-6$
Warm-up lr	$1e-5$	$1e-6$
Image size	32	224
Lr	$1e-3$	$6e-4$
Timestep	2/4/6/8	4
Batch size	64	$512^{\ddagger}$
Epochs	400	200
Weight decay	$6e-2$	$5e-2$
Optimizer	AdamW	
Warm-up epochs	20	5
Cool-down epochs	10	5
Random seed	42	0
Lr scheduler	Cosine	

Table 2: Training hyperparameters for QKFormer on the CIFAR datasets and ImageNet-1K. ‡ indicates the use of accumulated gradient iterations (He et al. 2022).

C Computational Environment

C.1 Software Setup

We employ PyTorch version 2.0.1, supported by CUDA 11.8, together with SpikingJelly v0.0.0.0.12 to implement and train our models.

Methods	Architecture	Acc (%)							
		CIFAR-10				CIFAR-100			
		T=2	T=4	T=6	T=8	T=2	T=4	T=6	T=8
Spikformer w/ STF-1	Spikformer-4-384	90.96	94.81	94.91	95.26	72.68	76.90	77.57	77.90
Spikformer w/ STF-2		88.20	94.85	95.06	95.29	74.96	77.43	78.12	78.68
Spikformer w/ STF-3		93.74	94.85	95.24	95.14	75.16	76.76	76.77	77.59
Spikformer w/ STF-4		**94.95**	**95.61**	**95.85**	**95.99**	**76.94**	**78.00**	**78.87**	**79.14**
SDT w/ STF-1	SDT-2-512	94.96	95.54	96.00	96.25	77.31	79.05	79.97	79.93
SDT w/ STF-2		95.15	95.77	96.19	96.34	77.97	79.32	79.87	80.35
SDT w/ STF-3		94.80	95.68	96.12	96.19	77.58	79.30	79.76	80.41
SDT w/ STF-4		**95.24**	**95.86**	**96.23**	**96.41**	**78.21**	**79.44**	**80.34**	**80.61**
QKFormer w/ STF-1	HST-4-384	95.81	96.15	96.23	96.40	79.69	81.20	81.37	81.77
QKFormer w/ STF-2		95.78	96.24	96.41	96.56	79.97	81.11	81.40	81.79
QKFormer w/ STF-3		95.68	96.17	96.40	96.52	80.02	81.18	81.45	81.80
QKFormer w/ STF-4		**96.02**	**96.33**	**96.51**	**96.61**	**80.07**	**81.26**	**81.51**	**81.89**

Table 3: Performance comparison of different STF variants across various Transformer-based SNN backbones and spike timestep settings. Bold indicates the best performance, and underlined denotes the second-best.

C.2 Hardware Setup

To accommodate varying computational demands, we employ different types of GPUs depending on the dataset scale.

For experiments conducted on the smaller datasets, we use a single NVIDIA L40 GPU per run, each equipped with 42 GB of memory. All experiments are performed in a single-GPU setting to ensure consistency across configurations.

For the large-scale ImageNet experiments, we utilize eight NVIDIA H20 GPUs per run, with each GPU offering 96 GB of memory. This multi-GPU setup is adopted to accommodate the computational and memory demands of training on high-resolution data.

D Ablation Study on Variants of the STF Module

Table 3 presents a detailed performance comparison of different STF variants on the CIFAR datasets, corresponding to the visualizations in Figure 3 of the main text. Experiments demonstrate that STF-4 consistently outperforms other STF variants under different timestep settings across various Transformer-based SNN backbones. Unless explicitly specified, STF-4 serves as the default configuration throughout this paper.

E Theoretical Energy Consumption of SNNs and ANNs

As illustrated in Figure 1 of the main text, the temporal feedback in STF follows the LIF neuron and subsequently passes through a convolution and batch normalization (BN) layer, thereby enhancing spike-driven characteristics. Prior studies (Deng et al. 2022; Hu, Tang, and Pan 2021) have shown that, due to the regularity of convolution operations, BN and linear scaling can be fused into the convolutional layer as an effective bias term during deployment. Consequently, the influence of BN can be neglected in theoretical energy estimations.

Based on this observation, the theoretical energy consumption of STF integrated into various Transformer-based SNN backbones is computed as:

$$E_{\text{total}} = E_{\text{MAC}} \times \text{FLOPs}^{1}_{\text{Conv}} + E_{\text{AC}} \times \left(\sum_{t=1}^{T} \text{SOP}'^{\,t}_{\text{Conv}_{\text{STF}}} + \sum_{j=1}^{M} \text{SOP}^{j}_{\text{block}} \right), \quad (9)$$

where E_{MAC} and E_{AC} denote the energy consumption of multiply-and-accumulate (MAC) operations and accumulate (AC) operations, respectively. We assume these operations are executed on 45 nm hardware, with $E_{\text{MAC}} = 4.6\,\text{pJ}$ and $E_{\text{AC}} = 0.9\,\text{pJ}$, as reported in previous studies (Horowitz 2014; Yao et al. 2023; Zhou et al. 2023). $\text{FLOPs}^{1}_{\text{Conv}}$ denotes the number of floating-point MAC operations in the first convolutional layer. $\text{SOP}'^{\,t}_{\text{Conv}_{\text{STF}}}$ represents the number of synaptic operations (SOPs) at time step t under STF feedback, and $\text{SOP}^{j}_{\text{block}}$ indicates the total SOPs over all time steps T in the j-th block of a given Transformer-based SNN backbone. The corresponding formulations for computing $\text{SOP}'^{\,t}_{\text{Conv}_{\text{STF}}}$ and $\text{SOP}^{j}_{\text{block}}$ are given as:

$$\text{SOP}'^{\,t}_{\text{Conv}_{\text{STF}}} = f_r \times \text{FLOPs}^{\text{STF}}, \quad (10)$$

$$\text{SOP}^{j}_{\text{block}} = f_r \times T \times \text{FLOPs}^{j}, \quad (11)$$

where f_r denotes the firing rate of spike trains, and FLOPs represents the number of floating-point operations.

F Inference Latency Statistics on ImageNet-1K

We further analyze the impact of STF on inference latency over the ImageNet-1K dataset and compare it with SpiLiFormer (Zheng et al. 2025) that adopts deep-level feedback.

As shown in Table 4, the relative increase in inference latency caused by STF accounts for roughly half of the proportion introduced by deep-level feedback, which aligns with the observations discussed in Section 4.2 of the main text.

Methods	Inference Time per Sample (ms)
SDT	26.987
SDT w/ STF	28.375 (**+5.14%**)
QKFormer	57.316
QKFormer w/ STF	60.942 (**+6.33%**)
SpiLiFormer w/o feedback	58.990
SpiLiFormer	66.302 (**+12.40%**)

Table 4: Comparison of inference latency overhead between the shallow-level STF module and traditional deep-level feedback on ImageNet-1K.

G SG_t Derivation Under Static Datasets

Building on prior work (Qiu et al. 2024), we provide a step-by-step derivation of the spike generation time (SG_t) formulation under static image input without using STF. To begin with, the membrane dynamics of a standard LIF neuron (Maass 1997) can be described by the following differential equation:

$$\tau_m \frac{\mathrm{d}U[t]}{\mathrm{d}t} = -(U[t] - U_{\text{reset}}) + I[t], \tag{12}$$

where τ_m denotes the membrane time constant, $U[t]$ represents the membrane potential at time step t, and $I[t]$ denotes the input current at time step t. Subsequently, by discretizing the differential equation using the Euler method and assuming $U_{\text{reset}} = 0$ for simplicity in practical implementations, we obtain:

$$\tau_m \frac{U[t] - U[t-1]}{\Delta t} = -U[t-1] + I[t]. \tag{13}$$

After further rearrangement, we obtain the following:

$$U[t] = U[t-1](1 - \frac{\Delta t}{\tau_m}) + \frac{\Delta t}{\tau_m} I[t]. \tag{14}$$

We define $\tau = 1 - \frac{\Delta t}{\tau_m}$ and $\beta = \frac{\Delta t}{\tau_m}$, where τ is the leak decay factor. Following prior work (Zhou et al. 2023; Yao et al. 2023; Zhou et al. 2024), we adopt the default setting $\tau_m = 2.0$, which leads to $\tau = \frac{1}{2}$. Moreover, in practice, β can be further absorbed into the data preprocessing. Thus, we obtain:

$$U[t] = \tau U[t-1] + I[t]. \tag{15}$$

Suppose that a spike is emitted at time step SG_t, meaning no spikes are generated during the interval from $t = 1$ to $t < SG_t$. Based on Equation 15, we can obtain:

$$U[SG_t - 1] = \tau^{SG_t - 1} U[0] + \sum_{t=1}^{SG_t - 1} \tau^{SG_t - 1 - t}\, I[t]. \tag{16}$$

Since $I[t]$ is replicated across T time steps, we have $I[t] = I$ for $t = 1, 2, \ldots, T$. The initial membrane potential is set to $U[0] = 0$ by default. Thus, we obtain:

$$U[SG_t - 1] = \sum_{t=1}^{SG_t - 1} \tau^{SG_t - 1 - t} I < U_{\text{th}}. \tag{17}$$

Since a spike is triggered at time step SG_t, we can derive the following based on Equation 17:

$$\sum_{t=1}^{SG_t - 1} \tau^{SG_t - 1 - t}\, I < U_{\text{th}} \leq \sum_{t=1}^{SG_t} \tau^{SG_t - t}\, I. \tag{18}$$

From Equation 10 and the geometric series properties, we can obtain:

$$\frac{1 - \tau^{SG_t - 1}}{1 - \tau}\, I < U_{\text{th}} \leq \frac{1 - \tau^{SG_t}}{1 - \tau}\, I, \tag{19}$$

$$\tau^{SG_t} \leq 1 - \frac{U_{\text{th}}(1 - \tau)}{I} < \tau^{SG_t - 1}, \tag{20}$$

$$SG_t - 1 < \log_\tau (1 - \frac{U_{\text{th}}(1 - \tau)}{I}) \leq SG_t, \tag{21}$$

$$SG_t = \left\lceil \log_\tau (1 - \frac{U_{\text{th}}(1 - \tau)}{I}) \right\rceil. \tag{22}$$

H Definition of Spike Entropy

Based on the previous work (Luczak 2024), the definition of spike entropy is as follows:

$$\text{Spike Entropy} = -\sum_{i=1}^{2^T} p_i \log_2 p_i, \tag{23}$$

where p_i denotes the proportion of each distinct spike pattern. Assuming the spike timestep is T, the index i ranges from 1 to 2^T, covering all possible patterns. Higher spike entropy reflects greater diversity in neural spike patterns, implying a higher potential information capacity.

I Adversarial Evaluation and Statistical Results

In our adversarial evaluation, we adopt two commonly used methods for generating adversarial examples: the Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2014) and Projected Gradient Descent (PGD) (Madry et al. 2017).

FGSM is a one-step perturbation method that modifies input samples based on the direction of the loss gradient. The adversarial input $\tilde{\mathbf{x}}$ is computed as:

$$\tilde{\mathbf{x}} = \mathbf{x} + \beta \cdot \text{sign}\left(\nabla_{\mathbf{x}} J(\mathbf{x}, \mathbf{y})\right). \tag{24}$$

Here, $\mathbf{x}$ denotes the clean input, $\mathbf{y}$ is the target label, J represents the loss function, and β controls the magnitude of the perturbation.

Methods	Architecture	Acc (%) / *Spike Entropy*							
		CIFAR-10				CIFAR-100			
		T=2	T=4	T=6	T=8	T=2	T=4	T=6	T=8
Spikformer	Spikformer-4-384	93.20 *0.964*	95.19 *1.442*	95.14 *1.530*	95.24 *1.664*	75.17 *1.203*	77.86 *1.590*	78.17 *1.774*	78.22 *1.860*
Spikformer w/ TSPE	Spikformer-4-384	94.72 *1.286*	95.26 *1.866*	95.44 *2.277*	95.69 *2.445*	76.21 *1.263*	77.88 *1.964*	78.43 *2.335*	78.54 *2.593*
Spikformer w/ TF	Spikformer-4-384	94.85 *1.364*	95.52 *2.427*	95.62 *4.245*	95.89 *4.784*	76.40 *1.386*	77.96 *2.566*	78.75 *4.383*	78.72 *5.021*
Spikformer w/ STF	Spikformer-4-384	**94.95** ***1.463***	**95.61** ***2.573***	**95.85** ***4.369***	**95.99** ***4.932***	**76.94** ***1.479***	**78.00** ***2.597***	**78.87** ***4.417***	**79.14** ***5.240***

Table 6: Comparison of ablation studies on the TSPE and TF components.

PGD generalizes FGSM by introducing iterative refinements. Starting from an initial perturbed sample $\tilde{\mathbf{x}}^{(0)}$, PGD updates the adversarial input over multiple steps as follows:

$$\tilde{\mathbf{x}}^{(t)} = \Pi_{\mathcal{B}\infty(\mathbf{x},\epsilon)}\left(\tilde{\mathbf{x}}^{(t-1)} + \eta \cdot \text{sign}\left(\nabla \mathbf{x} J(\tilde{\mathbf{x}}^{(t-1)}, \mathbf{y})\right)\right), \tag{25}$$

where η is the step size, $\Pi_{\mathcal{B}\infty(\mathbf{x},\epsilon)}(\cdot)$ denotes projection onto the $L\infty$ ball of radius ϵ centered at $\mathbf{x}$, ensuring the perturbation remains bounded.

We evaluate three Transformer-based SNN backbones under four different spike timestep settings ($T = 2, 4, 6, 8$) to demonstrate the better robustness of STF over direct coding (baseline models without STF).

As shown in Tables 7, 8, and 9, the detailed and comprehensive experimental results demonstrate that STF achieves better robustness without sacrificing clean performance, compared to direct coding.

J Ablation Study of TSPE and TF Components

We conduct detailed ablation experiments on the TSPE and TF components using Spikformer as the backbone. The experiments are performed under four timestep settings on the CIFAR datasets to evaluate their impact on model performance and the diversity of spike patterns. Following the methodology described in Section H, we use spike entropy as the metric to quantify the diversity of spike patterns. As shown in Table 6, TF leads to more substantial improvements in both the diversity of spike patterns and model performance compared to TSPE, while their combination yields further gains.

K Performance Evaluation on CIFAR10-DVS

We select two Transformer-based SNN backbones, including Spikformer and SDT, to evaluate the effect of STF on model performance under varying spike timestep settings using the neuromorphic dataset CIFAR10-DVS (Li et al. 2017). As presented in Table 5, STF results in performance degradation, which is consistent with our expectations. A detailed analysis can be found in Section 4.4 of the main text.

Architecture	T	Baseline	w/ STF
Spikformer-4-384	10	77.3%	71.4%
	16	79.3%	75.1%
	24	80.4%	80.2%
	32	81.9%	79.6%
SDT-2-512	10	74.3%	64.7%
	16	76.1%	56.3%
	24	77.6%	66.4%
	32	79.0%	73.0%

Table 5: Performance comparison on CIFAR10-DVS

L Detailed Statistics of Spike Pattern Diversity

As shown in Tables 10-19, we compare the spike pattern statistics at the output of LIF neurons in the encoding layer of Spikformer on the CIFAR-10 dataset. The comparison covers direct coding, its variants (GAC coding and IMP coding), and our proposed STF under different time steps ($T = 2, 4, 6, 8$). Additional detailed statistics for other backbone architectures can be found in the submitted supplementary material.

Dataset	Methods	Time Step	Clean	FGSM Maximum Perturbation				PGD Iterations			
				0.05	0.1	0.2	0.3	5	10	30	50
CIFAR-10	Spikformer	2	93.20	46.24	40.33	29.44	23.00	31.44	25.73	23.30	22.38
	Spikformer w/ STF	2	**94.95** **(+1.75)**	**46.69** **(+0.45)**	**42.64** **(+2.31)**	**34.23** **(+4.79)**	**26.22** **(+3.22)**	**33.72** **(+2.28)**	**29.32** **(+3.59)**	**27.14** **(+3.84)**	**26.96** **(+4.58)**
	Spikformer	4	95.19	54.92	51.72	41.20	30.98	27.56	21.08	18.33	17.68
	Spikformer w/ STF	4	**95.61** **(+0.42)**	**58.37** **(+3.45)**	**55.75** **(+4.03)**	**45.40** **(+4.2)**	**35.89** **(+4.91)**	**30.95** **(+3.39)**	**23.81** **(+2.73)**	**20.54** **(+2.21)**	**20.05** **(+2.37)**
	Spikformer	6	95.14	56.13	50.48	35.31	24.93	28.15	22.50	19.88	19.11
	Spikformer w/ STF	6	**95.85** **(+0.71)**	**61.33** **(+5.2)**	**59.72** **(+9.24)**	**50.70** **(+15.39)**	**40.56** **(+15.63)**	**31.37** **(+3.22)**	**25.37** **(+2.87)**	**22.47** **(+2.59)**	**21.97** **(+2.86)**
	Spikformer	8	95.24	53.13	47.21	35.53	26.24	28.61	24.53	22.29	22.04
	Spikformer w/ STF	8	**95.99** **(+0.75)**	**60.59** **(+7.46)**	**58.88** **(+11.67)**	**50.19** **(+14.66)**	**39.58** **(+13.34)**	**32.81** **(+4.20)**	**26.49** **(+1.96)**	**23.44** **(+1.15)**	**22.56** **(+0.52)**
CIFAR-100	Spikformer	2	75.17	21.96	15.82	9.79	6.91	14.03	10.32	9.01	9.06
	Spikformer w/ STF	2	**76.94** **(+1.77)**	**22.10** **(+0.14)**	**16.11** **(+0.29)**	**10.10** **(+0.31)**	**7.07** **(+0.16)**	**14.06** **(+0.03)**	**10.51** **(+0.19)**	**9.11** **(+0.10)**	**9.34** **(+0.28)**
	Spikformer	4	77.86	26.05	21.86	15.19	10.58	15.03	11.52	9.79	9.58
	Spikformer w/ STF	4	**78.00** **(+0.14)**	**26.41** **(+0.36)**	**22.00** **(+0.14)**	**15.75** **(+0.56)**	**11.73** **(+1.15)**	**15.12** **(+0.09)**	**12.36** **(+0.84)**	**10.48** **(+0.69)**	**10.50** **(+0.92)**
	Spikformer	6	78.17	29.94	25.49	16.96	11.86	15.39	12.32	10.55	**10.11**
	Spikformer w/ STF	6	**78.87** **(+0.70)**	**30.68** **(+0.74)**	**27.18** **(+1.69)**	**20.15** **(+3.19)**	**14.59** **(+2.73)**	**15.93** **(+0.54)**	**12.37** **(+0.05)**	**10.64** **(+0.09)**	10.03 (-0.08)
	Spikformer	8	78.22	31.18	26.34	18.22	12.77	15.74	12.55	10.97	10.67
	Spikformer w/ STF	8	**79.14** **(+0.92)**	**31.47** **(+0.29)**	**28.14** **(+1.80)**	**21.17** **(+2.95)**	**15.90** **(+3.13)**	**17.22** **(+1.48)**	**13.54** **(+0.99)**	**11.79** **(+0.82)**	**11.58** **(+0.91)**

Table 7: Adversarial robustness comparison between Spikformer with and without STF on the CIFAR datasets.

Dataset	Methods	Time Step	Clean	FGSM Maximum Perturbation				PGD Iterations			
				0.05	0.1	0.2	0.3	5	10	30	50
CIFAR-10	SDT	2	95.01	52.65	42.74	33.97	28.23	35.77	27.6	23.17	23.99
	SDT w/ STF	2	**95.24** **(+0.23)**	**52.89** **(+0.24)**	**43.29** **(+0.55)**	**34.22** **(+0.25)**	**28.95** **(+0.72)**	**36.12** **(+0.35)**	**28.32** **(+0.72)**	**24.74** **(+1.57)**	**24.92** **(+0.93)**
	SDT	4	95.60	62.11	56.18	50.54	46.46	38.59	30.81	26.1	**25.62**
	SDT w/ STF	4	**95.86** **(+0.26)**	**63.91** **(+1.80)**	**59.57** **(+3.39)**	**53.78** **(+3.24)**	**50.00** **(+3.54)**	**39.01** **(+0.42)**	**30.85** **(+0.04)**	**26.48** **(+0.38)**	25.53 (-0.09)
	SDT	6	96.11	69.31	66.37	61.13	57.01	42.29	34.81	30.45	28.77
	SDT w/ STF	6	**96.23** **(+0.12)**	**70.22** **(+0.91)**	**67.74** **(+1.37)**	**63.51** **(+2.38)**	**59.74** **(+2.73)**	**43.03** **(+0.74)**	**35.1** **(+0.29)**	**30.57** **(+0.12)**	**29.6** **(+0.83)**
	SDT	8	96.24	70.26	67.94	63.04	58.28	42.85	35.42	30.63	30.16
	SDT w/ STF	8	**96.41** **(+0.17)**	**72.65** **(+2.39)**	**69.06** **(+1.12)**	**63.89** **(+0.85)**	**59.22** **(+0.94)**	**43.99** **(+1.14)**	**36.37** **(+0.95)**	**31.59** **(+0.96)**	**30.70** **(+0.54)**
CIFAR-100	SDT	2	77.47	32.29	26.66	21.63	18.30	20.73	15.88	14.82	14.4
	SDT w/ STF	2	**78.21** **(+0.74)**	**33.23** **(+0.94)**	**27.86** **(+1.20)**	**22.91** **(+1.28)**	**20.02** **(+1.72)**	**21.02** **(+0.29)**	**16.81** **(+0.93)**	**15.17** **(+0.35)**	**14.83** **(+0.43)**
	SDT	4	78.40	37.23	**32.97**	27.69	25.02	23.16	17.91	16.84	16.25
	SDT w/ STF	4	**79.44** **(+1.04)**	**37.46** **(+0.23)**	32.60 (-0.37)	**28.52** **(+0.83)**	**25.08** **(+0.06)**	**23.43** **(+0.27)**	**18.76** **(+0.85)**	**17.30** **(+0.46)**	**16.48** **(+0.23)**
	SDT	6	79.43	38.01	34.75	30.13	26.88	22.15	17.71	15.64	15.49
	SDT w/ STF	6	**80.34** **(+0.91)**	**39.39** **(+1.38)**	**35.53** **(+0.78)**	**31.67** **(+1.54)**	**28.24** **(+1.36)**	**22.80** **(+0.65)**	**18.18** **(+0.47)**	**16.35** **(+0.71)**	**16.10** **(+0.61)**
	SDT	8	79.87	40.22	36.49	31.79	28.49	22.31	18.07	14.81	14.71
	SDT w/ STF	8	**80.61** **(+0.74)**	**40.33** **(+0.11)**	**36.53** **(+0.04)**	**32.42** **(+0.63)**	**29.38** **(+0.89)**	**22.54** **(+0.23)**	**18.41** **(+0.34)**	**15.90** **(+1.09)**	**15.57** **(+0.86)**

Table 8: Adversarial robustness comparison between SDT with and without STF on the CIFAR datasets.

Dataset	Methods	Time Step	Clean	FGSM Maximum Perturbation				PGD Iterations			
				0.05	0.1	0.2	0.3	5	10	30	50
CIFAR-10	QKFormer	2	95.79	59.08	51.57	43.19	37.19	31.37	22.11	16.50	15.45
	QKFormer w/ STF	2	**96.02** **(+0.23)**	**59.52** **(+0.44)**	**51.85** **(+0.28)**	**43.36** **(+0.17)**	**37.31** **(+0.12)**	**31.81** **(+0.44)**	**22.23** **(+0.12)**	**17.09** **(+0.59)**	**15.86** **(+0.41)**
	QKFormer	4	96.18	68.56	64.50	58.64	51.77	33.11	24.59	17.80	16.63
	QKFormer w/ STF	4	**96.33** **(+0.15)**	**71.15** **(+2.59)**	**67.61** **(+3.11)**	**61.16** **(+2.52)**	**54.26** **(+2.49)**	**38.33** **(+5.22)**	**28.21** **(+3.62)**	**21.07** **(+3.27)**	**19.31** **(+2.68)**
	QKFormer	6	96.37	66.07	61.30	53.19	45.80	34.21	24.71	18.28	17.09
	QKFormer w/ STF	6	**96.51** **(+0.14)**	**66.88** **(+0.81)**	**64.06** **(+2.76)**	**58.61** **(+5.42)**	**52.83** **(+7.03)**	**34.43** **(+0.22)**	**25.08** **(+0.37)**	**18.47** **(+0.19)**	**17.13** **(+0.04)**
	QKFormer	8	96.35	68.51	65.68	60.12	54.37	35.43	26.20	20.30	**18.93**
	QKFormer w/ STF	8	**96.61** **(+0.26)**	**68.79** **(+0.28)**	**66.97** **(+1.29)**	**61.61** **(+1.49)**	**54.69** **(+0.32)**	**36.59** **(+1.16)**	**27.49** **(+1.29)**	**20.32** **(+0.02)**	18.86 (-0.07)
CIFAR-100	QKFormer	2	79.79	32.63	28.05	22.98	18.95	17.25	12.52	10.15	9.86
	QKFormer w/ STF	2	**80.07** **(+0.28)**	**36.21** **(+3.58)**	**31.44** **(+3.39)**	**25.42** **(+2.44)**	**21.04** **(+2.09)**	**20.44** **(+3.19)**	**15.29** **(+2.77)**	**12.62** **(+2.47)**	**12.09** **(+2.23)**
	QKFormer	4	81.15	35.56	31.37	25.67	21.63	18.19	13.61	9.89	10.04
	QKFormer w/ STF	4	**81.26** **(+0.11)**	**37.19** **(+1.63)**	**32.94** **(+1.57)**	**27.27** **(+1.60)**	**23.04** **(+1.41)**	**19.32** **(+1.13)**	**14.45** **(+0.84)**	**11.26** **(+1.37)**	**10.75** **(+0.71)**
	QKFormer	6	81.35	36.90	33.39	28.43	23.97	18.40	13.32	9.89	9.10
	QKFormer w/ STF	6	**81.51** **(+0.16)**	**38.33** **(+1.43)**	**35.27** **(+1.88)**	**30.01** **(+1.58)**	**25.42** **(+1.45)**	**20.49** **(+2.09)**	**15.41** **(+2.09)**	**11.84** **(+1.95)**	**11.13** **(+2.03)**
	QKFormer	8	81.64	37.97	34.69	28.44	23.70	35.44	26.20	20.29	**18.93**
	QKFormer w/ STF	8	**81.89** **(+0.25)**	**39.96** **(+1.99)**	**36.84** **(+2.15)**	**30.52** **(+2.08)**	**25.95** **(+2.25)**	**36.59** **(+1.15)**	**27.50** **(+1.30)**	**20.30** **(+0.01)**	18.91 (-0.02)

Table 9: Adversarial robustness comparison between QKFormer with and without STF on the CIFAR datasets.

Spike Pattern	Count	Ratio
$[0, 0]$	386,181,630	0.78569
$[0, 1]$	54,595,182	0.11107
$[1, 0]$	0	0
$[1, 1]$	50,743,188	0.10324
Spike Entropy	0.9637	
Firing Rate	0.1588	

(a) Direct Coding

Spike Pattern	Count	Ratio
$[0, 0]$	385,766,667	0.78484
$[0, 1]$	54,736,027	0.11136
$[1, 0]$	0	0
$[1, 1]$	51,017,306	0.10379
Spike Entropy	0.9662	
Firing Rate	0.1595	

(b) GAC Coding

Spike Pattern	Count	Ratio
$[0, 0]$	332,652,925	0.67678
$[0, 1]$	44,688,373	0.09092
$[1, 0]$	42,131,763	0.08572
$[1, 1]$	72,046,939	0.14658
Spike Entropy	1.4056	
Firing Rate	0.2349	

(c) IMP Coding

Spike Pattern	Count	Ratio
$[0, 0]$	318,063,858	0.64710
$[0, 1]$	70,499,779	0.14343
$[1, 0]$	29,936,350	0.06091
$[1, 1]$	73,020,013	0.14856
Spike Entropy	1.4627	
Firing Rate	0.2507	

(d) STF (Ours)

Table 10: Distribution of the spike patterns in the encoding layer of Spikformer with different schemes on CIFAR-10 at $T = 2$.

Spike Pattern	Count	Ratio
$[0, 0, 0, 0]$	342,242,368	0.69629
$[0, 0, 0, 1]$	13,510,753	0.02749
$[0, 0, 1, 0]$	28,708,328	0.05841
$[0, 0, 1, 1]$	0	0
$[0, 1, 0, 0]$	0	0
$[0, 1, 0, 1]$	56,695,620	0.11535
$[0, 1, 1, 0]$	0	0
$[0, 1, 1, 1]$	0	0
$[1, 0, 0, 0]$	0	0
$[1, 0, 0, 1]$	0	0
$[1, 0, 1, 0]$	0	0
$[1, 0, 1, 1]$	0	0
$[1, 1, 0, 0]$	0	0
$[1, 1, 0, 1]$	0	0
$[1, 1, 1, 0]$	0	0
$[1, 1, 1, 1]$	50,362,931	0.10246
Spike Entropy	1.4417	
Firing Rate	0.1816	

(a) Direct Coding

Spike Pattern	Count	Ratio
$[0, 0, 0, 0]$	349,113,731	0.71027
$[0, 0, 0, 1]$	12,453,653	0.02534
$[0, 0, 1, 0]$	26,487,893	0.05389
$[0, 0, 1, 1]$	0	0
$[0, 1, 0, 0]$	0	0
$[0, 1, 0, 1]$	53,701,975	0.10926
$[0, 1, 1, 0]$	0	0
$[0, 1, 1, 1]$	0	0
$[1, 0, 0, 0]$	0	0
$[1, 0, 0, 1]$	0	0
$[1, 0, 1, 0]$	0	0
$[1, 0, 1, 1]$	0	0
$[1, 1, 0, 0]$	0	0
$[1, 1, 0, 1]$	0	0
$[1, 1, 1, 0]$	0	0
$[1, 1, 1, 1]$	49,762,748	0.10124
Spike Entropy	1.3955	
Firing Rate	0.1757	

(b) GAC Coding

Spike Pattern	Count	Ratio
$[0, 0, 0, 0]$	331,272,166	0.67397
$[0, 0, 0, 1]$	7,127,049	0.01450
$[0, 0, 1, 0]$	16,387,942	0.03334
$[0, 0, 1, 1]$	0	0
$[0, 1, 0, 0]$	24,011,229	0.04885
$[0, 1, 0, 1]$	18,759,892	0.03817
$[0, 1, 1, 0]$	0	0
$[0, 1, 1, 1]$	0	0
$[1, 0, 0, 0]$	0	0
$[1, 0, 0, 1]$	0	0
$[1, 0, 1, 0]$	37,373,113	0.07604
$[1, 0, 1, 1]$	0	0
$[1, 1, 0, 0]$	0	0
$[1, 1, 0, 1]$	0	0
$[1, 1, 1, 0]$	0	0
$[1, 1, 1, 1]$	56,588,609	0.11513
Spike Entropy	1.6701	
Firing Rate	0.1964	

(c) IMP Coding

Spike Pattern	Count	Ratio
$[0, 0, 0, 0]$	265,198,402	0.53955
$[0, 0, 0, 1]$	19,567,140	0.03981
$[0, 0, 1, 0]$	33,923,411	0.06902
$[0, 0, 1, 1]$	7,457,890	0.01517
$[0, 1, 0, 0]$	28,667,675	0.05833
$[0, 1, 0, 1]$	24,183,612	0.04920
$[0, 1, 1, 0]$	11,907,156	0.02423
$[0, 1, 1, 1]$	15,956,240	0.03246
$[1, 0, 0, 0]$	8,005,800	0.01629
$[1, 0, 0, 1]$	2,780,431	0.00566
$[1, 0, 1, 0]$	7,693,307	0.01565
$[1, 0, 1, 1]$	3,815,101	0.00776
$[1, 1, 0, 0]$	3,338,439	0.00679
$[1, 1, 0, 1]$	5,733,335	0.01166
$[1, 1, 1, 0]$	4,863,576	0.00989
$[1, 1, 1, 1]$	48,428,485	0.09853
Spike Entropy	2.5732	
Firing Rate	0.2491	

(d) STF (Ours)

Table 11: Distribution of the spike patterns in the encoding layer of Spikformer with different schemes on CIFAR-10 at $T = 4$.

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
$[0,0,0,0,0,0]$	342,373,731	0.69656	$[1,0,0,0,0,0]$	0	0
$[0,0,0,0,0,1]$	3,111,285	0.00633	$[1,0,0,0,0,1]$	0	0
$[0,0,0,0,1,0]$	6,282,613	0.01278	$[1,0,0,0,1,0]$	0	0
$[0,0,0,1,0,0]$	13,225,515	0.02691	$[1,0,0,1,0,0]$	0	0
$[0,0,0,1,0,1]$	0	0	$[1,0,0,1,0,1]$	0	0
$[0,0,0,0,0,1,1]$	0	0	$[1,0,0,0,1,1]$	0	0
$[0,0,1,1,0]$	0	0	$[1,0,0,1,1,0]$	0	0
$[0,0,0,1,1,1]$	0	0	$[1,0,0,1,1,1]$	0	0
$[0,0,1,0,0,0]$	0	0	$[1,0,1,0,0,0]$	0	0
$[0,0,1,0,0,1]$	27,506,090	0.05596	$[1,0,1,0,0,1]$	0	0
$[0,0,1,0,1,0]$	0	0	$[1,0,1,0,1,0]$	0	0
$[0,0,1,0,1,1]$	0	0	$[1,0,1,0,1,1]$	0	0
$[0,0,1,1,0,0]$	0	0	$[1,0,1,1,0,0]$	0	0
$[0,0,1,1,0,1]$	0	0	$[1,0,1,1,0,1]$	0	0
$[0,0,1,1,1,0]$	0	0	$[1,0,1,1,1,0]$	0	0
$[0,0,1,1,1,1]$	0	0	$[1,0,1,1,1,1]$	0	0
$[0,1,0,0,0,0]$	0	0	$[1,1,0,0,0,0]$	0	0
$[0,1,0,0,0,1]$	0	0	$[1,1,0,0,0,1]$	0	0
$[0,1,0,0,1,0]$	0	0	$[1,1,0,0,1,0]$	0	0
$[0,1,0,0,1,1]$	0	0	$[1,1,0,0,1,1]$	0	0
$[0,1,0,1,0,0]$	0	0	$[1,1,0,1,0,0]$	0	0
$[0,1,0,1,0,1]$	52,721,780	0.10726	$[1,1,0,1,0,1]$	0	0
$[0,1,0,1,1,0]$	0	0	$[1,1,0,1,1,0]$	0	0
$[0,1,0,1,1,1]$	0	0	$[1,1,0,1,1,1]$	0	0
$[0,1,1,0,0,0]$	0	0	$[1,1,1,0,0,0]$	0	0
$[0,1,1,0,0,1]$	0	0	$[1,1,1,0,0,1]$	0	0
$[0,1,1,0,1,0]$	0	0	$[1,1,1,0,1,0]$	0	0
$[0,1,1,0,1,1]$	0	0	$[1,1,1,0,1,1]$	0	0
$[0,1,1,1,0,0]$	0	0	$[1,1,1,1,0,0]$	0	0
$[0,1,1,1,0,1]$	0	0	$[1,1,1,1,0,1]$	0	0
$[0,1,1,1,1,0]$	0	0	$[1,1,1,1,1,0]$	0	0
$[0,1,1,1,1,1]$	0	0	$[1,1,1,1,1,1]$	46,298,986	0.0942
Spike Entropy			1.5296		
Firing Rate			0.1742		

Table 12: Distribution of the spike patterns in the encoding layer of Spikformer with direct coding on CIFAR-10 at $T = 6$.

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
$[0,0,0,0,0,0]$	343,464,329	0.69878	$[1,0,0,0,0,0]$	0	0
$[0,0,0,0,0,1]$	2,946,366	0.00599	$[1,0,0,0,0,1]$	0	0
$[0,0,0,0,1,0]$	6,005,346	0.01222	$[1,0,0,0,1,0]$	0	0
$[0,0,0,0,1,1]$	0	0	$[1,0,0,0,1,1]$	0	0
$[0,0,0,1,0,0]$	12,401,887	0.02523	$[1,0,0,1,0,0]$	0	0
$[0,0,0,1,0,1]$	0	0	$[1,0,0,1,0,1]$	0	0
$[0,0,0,1,1,0]$	0	0	$[1,0,0,1,1,0]$	0	0
$[0,0,0,1,1,1]$	0	0	$[1,0,0,1,1,1]$	0	0
$[0,0,1,0,0,0]$	0	0	$[1,0,1,0,0,0]$	0	0
$[0,0,1,0,0,1]$	26,284,530	0.05348	$[1,0,1,0,0,1]$	0	0
$[0,0,1,0,1,0]$	0	0	$[1,0,1,0,1,0]$	0	0
$[0,0,1,0,1,1]$	0	0	$[1,0,1,0,1,1]$	0	0
$[0,0,1,1,0,0]$	0	0	$[1,0,1,1,0,0]$	0	0
$[0,0,1,1,0,1]$	0	0	$[1,0,1,1,0,1]$	0	0
$[0,0,1,1,1,0]$	0	0	$[1,0,1,1,1,0]$	0	0
$[0,0,1,1,1,1]$	0	0	$[1,0,1,1,1,1]$	0	0
$[0,1,0,0,0,0]$	0	0	$[1,1,0,0,0,0]$	0	0
$[0,1,0,0,0,1]$	0	0	$[1,1,0,0,0,1]$	0	0
$[0,1,0,0,1,0]$	0	0	$[1,1,0,0,1,0]$	0	0
$[0,1,0,0,1,1]$	0	0	$[1,1,0,0,1,1]$	0	0
$[0,1,0,1,0,0]$	0	0	$[1,1,0,1,0,0]$	0	0
$[0,1,0,1,0,1]$	53,059,139	0.10795	$[1,1,0,1,0,1]$	0	0
$[0,1,0,1,1,0]$	0	0	$[1,1,0,1,1,0]$	0	0
$[0,1,0,1,1,1]$	0	0	$[1,1,0,1,1,1]$	0	0
$[0,1,1,0,0,0]$	0	0	$[1,1,1,0,0,0]$	0	0
$[0,1,1,0,0,1]$	0	0	$[1,1,1,0,0,1]$	0	0
$[0,1,1,0,1,0]$	0	0	$[1,1,1,0,1,0]$	0	0
$[0,1,1,0,1,1]$	0	0	$[1,1,1,0,1,1]$	0	0
$[0,1,1,1,0,0]$	0	0	$[1,1,1,1,0,0]$	0	0
$[0,1,1,1,0,1]$	0	0	$[1,1,1,1,0,1]$	0	0
$[0,1,1,1,1,0]$	0	0	$[1,1,1,1,1,0]$	0	0
$[0,1,1,1,1,1]$	0	0	$[1,1,1,1,1,1]$	47,358,403	0.09635
Spike Entropy			1.5150		
Firing Rate			0.1754		

Table 13: Distribution of the spike patterns in the encoding layer of Spikformer with GAC coding on CIFAR-10 at $T = 6$.

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
$[0, 0, 0, 0, 0, 0]$	338,159,121	0.68799	$[1, 0, 0, 0, 0, 0]$	0	0
$[0, 0, 0, 0, 0, 1]$	1,639,282	0.00333	$[1, 0, 0, 0, 0, 1]$	0	0
$[0, 0, 0, 0, 1, 0]$	3,392,152	0.00690	$[1, 0, 0, 0, 1, 0]$	0	0
$[0, 0, 0, 0, 1, 1]$	0	0	$[1, 0, 0, 0, 1, 1]$	0	0
$[0, 0, 0, 1, 0, 0]$	7,176,057	0.01460	$[1, 0, 0, 1, 0, 0]$	0	0
$[0, 0, 0, 1, 0, 1]$	0	0	$[1, 0, 0, 1, 0, 1]$	0	0
$[0, 0, 0, 1, 1, 0]$	0	0	$[1, 0, 0, 1, 1, 0]$	0	0
$[0, 0, 0, 1, 1, 1]$	0	0	$[1, 0, 0, 1, 1, 1]$	0	0
$[0, 0, 1, 0, 0, 0]$	12,294,224	0.02501	$[1, 0, 1, 0, 0, 0]$	0	0
$[0, 0, 1, 0, 0, 1]$	4,026,400	0.00819	$[1, 0, 1, 0, 0, 1]$	0	0
$[0, 0, 1, 0, 1, 0]$	0	0	$[1, 0, 1, 0, 1, 0]$	34,806,496	0.07081
$[0, 0, 1, 0, 1, 1]$	0	0	$[1, 0, 1, 0, 1, 1]$	0	0
$[0, 0, 1, 1, 0, 0]$	0	0	$[1, 0, 1, 1, 0, 0]$	0	0
$[0, 0, 1, 1, 0, 1]$	0	0	$[1, 0, 1, 1, 0, 1]$	0	0
$[0, 0, 1, 1, 1, 0]$	0	0	$[1, 0, 1, 1, 1, 0]$	0	0
$[0, 0, 1, 1, 1, 1]$	0	0	$[1, 0, 1, 1, 1, 1]$	0	0
$[0, 1, 0, 0, 0, 0]$	0	0	$[1, 1, 0, 0, 0, 0]$	0	0
$[0, 1, 0, 0, 0, 1]$	105,634	0.00021	$[1, 1, 0, 0, 0, 1]$	0	0
$[0, 1, 0, 0, 1, 0]$	23,232,901	0.04727	$[1, 1, 0, 0, 1, 0]$	0	0
$[0, 1, 0, 0, 1, 1]$	0	0	$[1, 1, 0, 0, 1, 1]$	0	0
$[0, 1, 0, 1, 0, 0]$	0	0	$[1, 1, 0, 1, 0, 0]$	0	0
$[0, 1, 0, 1, 0, 1]$	17,930,876	0.03648	$[1, 1, 0, 1, 0, 1]$	0	0
$[0, 1, 0, 1, 1, 0]$	0	0	$[1, 1, 0, 1, 1, 0]$	0	0
$[0, 1, 0, 1, 1, 1]$	0	0	$[1, 1, 0, 1, 1, 1]$	0	0
$[0, 1, 1, 0, 0, 0]$	0	0	$[1, 1, 1, 0, 0, 0]$	0	0
$[0, 1, 1, 0, 0, 1]$	0	0	$[1, 1, 1, 0, 0, 1]$	0	0
$[0, 1, 1, 0, 1, 0]$	0	0	$[1, 1, 1, 0, 1, 0]$	0	0
$[0, 1, 1, 0, 1, 1]$	0	0	$[1, 1, 1, 0, 1, 1]$	0	0
$[0, 1, 1, 1, 0, 0]$	0	0	$[1, 1, 1, 1, 0, 0]$	0	0
$[0, 1, 1, 1, 0, 1]$	0	0	$[1, 1, 1, 1, 0, 1]$	0	0
$[0, 1, 1, 1, 1, 0]$	0	0	$[1, 1, 1, 1, 1, 0]$	0	0
$[0, 1, 1, 1, 1, 1]$	0	0	$[1, 1, 1, 1, 1, 1]$	48,756,857	0.09920
Spike Entropy			1.7133		
Firing Rate			0.1797		

Table 14: Distribution of the spike patterns in the encoding layer of Spikformer with IMP coding on CIFAR-10 at $T = 6$.

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
$[0, 0, 0, 0, 0, 0]$	153,390,794	0.31207	$[1, 0, 0, 0, 0, 0]$	13,437,249	0.02734
$[0, 0, 0, 0, 0, 1]$	17,672,955	0.03596	$[1, 0, 0, 0, 0, 1]$	1,539,807	0.00313
$[0, 0, 0, 0, 1, 0]$	21,481,933	0.04371	$[1, 0, 0, 0, 1, 0]$	1,668,172	0.00339
$[0, 0, 0, 0, 1, 1]$	4,090,165	0.00832	$[1, 0, 0, 0, 1, 1]$	367,067	0.00075
$[0, 0, 0, 1, 0, 0]$	25,066,475	0.05100	$[1, 0, 0, 1, 0, 0]$	21,684,054	0.04412
$[0, 0, 0, 1, 0, 1]$	4,208,713	0.00856	$[1, 0, 0, 1, 0, 1]$	3,148,724	0.00641
$[0, 0, 0, 1, 1, 0]$	5,314,403	0.01081	$[1, 0, 0, 1, 1, 0]$	3,829,962	0.00779
$[0, 0, 0, 1, 1, 1]$	1,395,765	0.00284	$[1, 0, 0, 1, 1, 1]$	879,695	0.00179
$[0, 0, 1, 0, 0, 0]$	24,662,353	0.05018	$[1, 0, 1, 0, 0, 0]$	2,519,839	0.00513
$[0, 0, 1, 0, 0, 1]$	28,702,361	0.05840	$[1, 0, 1, 0, 0, 1]$	2,557,085	0.00520
$[0, 0, 1, 0, 1, 0]$	3,846,168	0.00783	$[1, 0, 1, 0, 1, 0]$	388,368	0.00079
$[0, 0, 1, 0, 1, 1]$	5,584,658	0.01136	$[1, 0, 1, 0, 1, 1]$	502,201	0.00102
$[0, 0, 1, 1, 0, 0]$	4,763,066	0.00969	$[1, 0, 1, 1, 0, 0]$	4,483,259	0.00912
$[0, 0, 1, 1, 0, 1]$	6,549,139	0.01332	$[1, 0, 1, 1, 0, 1]$	7,232,276	0.01471
$[0, 0, 1, 1, 1, 0]$	1,180,092	0.00240	$[1, 0, 1, 1, 1, 0]$	919,458	0.00187
$[0, 0, 1, 1, 1, 1]$	1,815,570	0.00369	$[1, 0, 1, 1, 1, 1]$	1,597,980	0.00325
$[0, 1, 0, 0, 0, 0]$	18,269,651	0.03717	$[1, 1, 0, 0, 0, 0]$	2,522,659	0.00513
$[0, 1, 0, 0, 0, 1]$	2,068,725	0.00421	$[1, 1, 0, 0, 0, 1]$	314,479	0.00064
$[0, 1, 0, 0, 1, 0]$	24,728,089	0.05031	$[1, 1, 0, 0, 1, 0]$	2,570,239	0.00523
$[0, 1, 0, 0, 1, 1]$	3,734,288	0.00760	$[1, 1, 0, 0, 1, 1]$	443,169	0.00090
$[0, 1, 0, 1, 0, 0]$	3,142,395	0.00639	$[1, 1, 0, 1, 0, 0]$	3,455,308	0.00703
$[0, 1, 0, 1, 0, 1]$	560,545	0.00114	$[1, 1, 0, 1, 0, 1]$	550,655	0.00112
$[0, 1, 0, 1, 1, 0]$	5,397,830	0.01098	$[1, 1, 0, 1, 1, 0]$	6,574,824	0.01338
$[0, 1, 0, 1, 1, 1]$	1,139,751	0.00232	$[1, 1, 0, 1, 1, 1]$	1,149,593	0.00234
$[0, 1, 1, 0, 0, 0]$	4,462,547	0.00908	$[1, 1, 1, 0, 0, 0]$	589,074	0.00120
$[0, 1, 1, 0, 0, 1]$	4,297,173	0.00874	$[1, 1, 1, 0, 0, 1]$	565,157	0.00115
$[0, 1, 1, 0, 1, 0]$	5,386,879	0.01096	$[1, 1, 1, 0, 1, 0]$	624,925	0.00127
$[0, 1, 1, 0, 1, 1]$	8,595,064	0.01749	$[1, 1, 1, 0, 1, 1]$	884,239	0.00180
$[0, 1, 1, 1, 0, 0]$	972,876	0.00198	$[1, 1, 1, 1, 0, 0]$	1,000,862	0.00204
$[0, 1, 1, 1, 0, 1]$	1,077,323	0.00219	$[1, 1, 1, 1, 0, 1]$	1,295,744	0.00264
$[0, 1, 1, 1, 1, 0]$	1,391,814	0.00283	$[1, 1, 1, 1, 1, 0]$	1,625,378	0.00331
$[0, 1, 1, 1, 1, 1]$	2,334,636	0.00475	$[1, 1, 1, 1, 1, 1]$	3,314,303	0.00674
Spike Entropy			4.3685		
Firing Rate			0.2438		

Table 15: Distribution of the spike patterns in the encoding layer of Spikformer with STF on CIFAR-10 at $T = 6$.

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
[0, 0, 0, 0, 0, 0, 0, 0]	325,647,136	0.66253	[1, 0, 0, 0, 0, 0, 0, 0]	0	0
[0, 0, 0, 0, 0, 0, 0, 1]	780,771	0.00159	[1, 0, 0, 0, 0, 0, 0, 1]	0	0
[0, 0, 0, 0, 0, 0, 1, 0]	1,549,957	0.00315	[1, 0, 0, 0, 0, 0, 1, 0]	0	0
[0, 0, 0, 0, 0, 0, 1, 1]	0	0	[1, 0, 0, 0, 0, 0, 1, 1]	0	0
[0, 0, 0, 0, 0, 1, 0, 0]	3,343,140	0.0068	[1, 0, 0, 0, 0, 1, 0, 0]	0	0
[0, 0, 0, 0, 0, 1, 0, 1]	0	0	[1, 0, 0, 0, 0, 1, 0, 1]	0	0
[0, 0, 0, 0, 0, 1, 1, 0]	0	0	[1, 0, 0, 0, 0, 1, 1, 0]	0	0
[0, 0, 0, 0, 0, 1, 1, 1]	0	0	[1, 0, 0, 0, 0, 1, 1, 1]	0	0
[0, 0, 0, 0, 1, 0, 0, 0]	6,669,556	0.01357	[1, 0, 0, 0, 1, 0, 0, 0]	0	0
[0, 0, 0, 0, 1, 0, 0, 1]	0	0	[1, 0, 0, 0, 1, 0, 0, 1]	0	0
[0, 0, 0, 0, 1, 0, 1, 0]	0	0	[1, 0, 0, 0, 1, 0, 1, 0]	0	0
[0, 0, 0, 0, 1, 0, 1, 1]	0	0	[1, 0, 0, 0, 1, 0, 1, 1]	0	0
[0, 0, 0, 0, 1, 1, 0, 0]	0	0	[1, 0, 0, 0, 1, 1, 0, 0]	0	0
[0, 0, 0, 0, 1, 1, 0, 1]	0	0	[1, 0, 0, 0, 1, 1, 0, 1]	0	0
[0, 0, 0, 0, 1, 1, 1, 0]	0	0	[1, 0, 0, 0, 1, 1, 1, 0]	0	0
[0, 0, 0, 0, 1, 1, 1, 1]	0	0	[1, 0, 0, 0, 1, 1, 1, 1]	0	0
[0, 0, 0, 1, 0, 0, 0, 0]	0	0	[1, 0, 0, 1, 0, 0, 0, 0]	0	0
[0, 0, 0, 1, 0, 0, 0, 1]	13,888,765	0.02826	[1, 0, 0, 1, 0, 0, 0, 1]	0	0
[0, 0, 0, 1, 0, 0, 1, 0]	0	0	[1, 0, 0, 1, 0, 0, 1, 0]	0	0
[0, 0, 0, 1, 0, 0, 1, 1]	0	0	[1, 0, 0, 1, 0, 0, 1, 1]	0	0
[0, 0, 0, 1, 0, 1, 0, 0]	0	0	[1, 0, 0, 1, 0, 1, 0, 0]	0	0
[0, 0, 0, 1, 0, 1, 0, 1]	0	0	[1, 0, 0, 1, 0, 1, 0, 1]	0	0
[0, 0, 0, 1, 0, 1, 1, 0]	0	0	[1, 0, 0, 1, 0, 1, 1, 0]	0	0
[0, 0, 0, 1, 0, 1, 1, 1]	0	0	[1, 0, 0, 1, 0, 1, 1, 1]	0	0
[0, 0, 0, 1, 1, 0, 0, 0]	0	0	[1, 0, 0, 1, 1, 0, 0, 0]	0	0
[0, 0, 0, 1, 1, 0, 0, 1]	0	0	[1, 0, 0, 1, 1, 0, 0, 1]	0	0
[0, 0, 0, 1, 1, 0, 1, 0]	0	0	[1, 0, 0, 1, 1, 0, 1, 0]	0	0
[0, 0, 0, 1, 1, 0, 1, 1]	0	0	[1, 0, 0, 1, 1, 0, 1, 1]	0	0
[0, 0, 0, 1, 1, 1, 0, 0]	0	0	[1, 0, 0, 1, 1, 1, 0, 0]	0	0
[0, 0, 0, 1, 1, 1, 0, 1]	0	0	[1, 0, 0, 1, 1, 1, 0, 1]	0	0
[0, 0, 0, 1, 1, 1, 1, 0]	0	0	[1, 0, 0, 1, 1, 1, 1, 0]	0	0
[0, 0, 0, 1, 1, 1, 1, 1]	0	0	[1, 0, 0, 1, 1, 1, 1, 1]	0	0
[0, 0, 1, 0, 0, 0, 0, 0]	0	0	[1, 0, 1, 0, 0, 0, 0, 0]	0	0
[0, 0, 1, 0, 0, 0, 0, 1]	0	0	[1, 0, 1, 0, 0, 0, 0, 1]	0	0
[0, 0, 1, 0, 0, 0, 1, 0]	0	0	[1, 0, 1, 0, 0, 0, 1, 0]	0	0
[0, 0, 1, 0, 0, 0, 1, 1]	0	0	[1, 0, 1, 0, 0, 0, 1, 1]	0	0
[0, 0, 1, 0, 0, 1, 0, 0]	29,723,360	0.06047	[1, 0, 1, 0, 0, 1, 0, 0]	0	0
[0, 0, 1, 0, 0, 1, 0, 1]	0	0	[1, 0, 1, 0, 0, 1, 0, 1]	0	0
[0, 0, 1, 0, 0, 1, 1, 0]	0	0	[1, 0, 1, 0, 0, 1, 1, 0]	0	0
[0, 0, 1, 0, 0, 1, 1, 1]	0	0	[1, 0, 1, 0, 0, 1, 1, 1]	0	0
[0, 0, 1, 0, 1, 0, 0, 0]	0	0	[1, 0, 1, 0, 1, 0, 0, 0]	0	0
[0, 0, 1, 0, 1, 0, 0, 1]	0	0	[1, 0, 1, 0, 1, 0, 0, 1]	0	0
[0, 0, 1, 0, 1, 0, 1, 0]	0	0	[1, 0, 1, 0, 1, 0, 1, 0]	0	0
[0, 0, 1, 0, 1, 0, 1, 1]	0	0	[1, 0, 1, 0, 1, 0, 1, 1]	0	0
[0, 0, 1, 0, 1, 1, 0, 0]	0	0	[1, 0, 1, 0, 1, 1, 0, 0]	0	0
[0, 0, 1, 0, 1, 1, 0, 1]	0	0	[1, 0, 1, 0, 1, 1, 0, 1]	0	0
[0, 0, 1, 0, 1, 1, 1, 0]	0	0	[1, 0, 1, 0, 1, 1, 1, 0]	0	0
[0, 0, 1, 0, 1, 1, 1, 1]	0	0	[1, 0, 1, 0, 1, 1, 1, 1]	0	0
[0, 0, 1, 1, 0, 0, 0, 0]	0	0	[1, 0, 1, 1, 0, 0, 0, 0]	0	0
[0, 0, 1, 1, 0, 0, 0, 1]	0	0	[1, 0, 1, 1, 0, 0, 0, 1]	0	0
[0, 0, 1, 1, 0, 0, 1, 0]	0	0	[1, 0, 1, 1, 0, 0, 1, 0]	0	0
[0, 0, 1, 1, 0, 0, 1, 1]	0	0	[1, 0, 1, 1, 0, 0, 1, 1]	0	0
[0, 0, 1, 1, 0, 1, 0, 0]	0	0	[1, 0, 1, 1, 0, 1, 0, 0]	0	0
[0, 0, 1, 1, 0, 1, 0, 1]	0	0	[1, 0, 1, 1, 0, 1, 0, 1]	0	0

Continued on next page

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
[0, 0, 1, 1, 0, 1, 1, 0]	0	0	[1, 0, 1, 1, 0, 1, 1, 0]	0	0
[0, 0, 1, 1, 0, 1, 1, 1]	0	0	[1, 0, 1, 1, 0, 1, 1, 1]	0	0
[0, 0, 1, 1, 1, 0, 0, 0]	0	0	[1, 0, 1, 1, 1, 0, 0, 0]	0	0
[0, 0, 1, 1, 1, 0, 0, 1]	0	0	[1, 0, 1, 1, 1, 0, 0, 1]	0	0
[0, 0, 1, 1, 1, 0, 1, 0]	0	0	[1, 0, 1, 1, 1, 0, 1, 0]	0	0
[0, 0, 1, 1, 1, 0, 1, 1]	0	0	[1, 0, 1, 1, 1, 0, 1, 1]	0	0
[0, 0, 1, 1, 1, 1, 0, 0]	0	0	[1, 0, 1, 1, 1, 1, 0, 0]	0	0
[0, 0, 1, 1, 1, 1, 0, 1]	0	0	[1, 0, 1, 1, 1, 1, 0, 1]	0	0
[0, 0, 1, 1, 1, 1, 1, 0]	0	0	[1, 0, 1, 1, 1, 1, 1, 0]	0	0
[0, 0, 1, 1, 1, 1, 1, 1]	0	0	[1, 0, 1, 1, 1, 1, 1, 1]	0	0
[0, 1, 0, 0, 0, 0, 0, 0]	0	0	[1, 1, 0, 0, 0, 0, 0, 0]	0	0
[0, 1, 0, 0, 0, 0, 0, 1]	0	0	[1, 1, 0, 0, 0, 0, 0, 1]	0	0
[0, 1, 0, 0, 0, 0, 1, 0]	0	0	[1, 1, 0, 0, 0, 0, 1, 0]	0	0
[0, 1, 0, 0, 0, 0, 1, 1]	0	0	[1, 1, 0, 0, 0, 0, 1, 1]	0	0
[0, 1, 0, 0, 0, 1, 0, 0]	0	0	[1, 1, 0, 0, 0, 1, 0, 0]	0	0
[0, 1, 0, 0, 0, 1, 0, 1]	0	0	[1, 1, 0, 0, 0, 1, 0, 1]	0	0
[0, 1, 0, 0, 0, 1, 1, 0]	0	0	[1, 1, 0, 0, 0, 1, 1, 0]	0	0
[0, 1, 0, 0, 0, 1, 1, 1]	0	0	[1, 1, 0, 0, 0, 1, 1, 1]	0	0
[0, 1, 0, 0, 1, 0, 0, 0]	0	0	[1, 1, 0, 0, 1, 0, 0, 0]	0	0
[0, 1, 0, 0, 1, 0, 0, 1]	0	0	[1, 1, 0, 0, 1, 0, 0, 1]	0	0
[0, 1, 0, 0, 1, 0, 1, 0]	0	0	[1, 1, 0, 0, 1, 0, 1, 0]	0	0
[0, 1, 0, 0, 1, 0, 1, 1]	0	0	[1, 1, 0, 0, 1, 0, 1, 1]	0	0
[0, 1, 0, 0, 1, 1, 0, 0]	0	0	[1, 1, 0, 0, 1, 1, 0, 0]	0	0
[0, 1, 0, 0, 1, 1, 0, 1]	0	0	[1, 1, 0, 0, 1, 1, 0, 1]	0	0
[0, 1, 0, 0, 1, 1, 1, 0]	0	0	[1, 1, 0, 0, 1, 1, 1, 0]	0	0
[0, 1, 0, 0, 1, 1, 1, 1]	0	0	[1, 1, 0, 0, 1, 1, 1, 1]	0	0
[0, 1, 0, 1, 0, 0, 0, 0]	0	0	[1, 1, 0, 1, 0, 0, 0, 0]	0	0
[0, 1, 0, 1, 0, 0, 0, 1]	0	0	[1, 1, 0, 1, 0, 0, 0, 1]	0	0
[0, 1, 0, 1, 0, 0, 1, 0]	0	0	[1, 1, 0, 1, 0, 0, 1, 0]	0	0
[0, 1, 0, 1, 0, 0, 1, 1]	0	0	[1, 1, 0, 1, 0, 0, 1, 1]	0	0
[0, 1, 0, 1, 0, 1, 0, 0]	0	0	[1, 1, 0, 1, 0, 1, 0, 0]	0	0
[0, 1, 0, 1, 0, 1, 0, 1]	57,422,345	0.11683	[1, 1, 0, 1, 0, 1, 0, 1]	0	0
[0, 1, 0, 1, 0, 1, 1, 0]	0	0	[1, 1, 0, 1, 0, 1, 1, 0]	0	0
[0, 1, 0, 1, 0, 1, 1, 1]	0	0	[1, 1, 0, 1, 0, 1, 1, 1]	0	0
[0, 1, 0, 1, 1, 0, 0, 0]	0	0	[1, 1, 0, 1, 1, 0, 0, 0]	0	0
[0, 1, 0, 1, 1, 0, 0, 1]	0	0	[1, 1, 0, 1, 1, 0, 0, 1]	0	0
[0, 1, 0, 1, 1, 0, 1, 0]	0	0	[1, 1, 0, 1, 1, 0, 1, 0]	0	0
[0, 1, 0, 1, 1, 0, 1, 1]	0	0	[1, 1, 0, 1, 1, 0, 1, 1]	0	0
[0, 1, 0, 1, 1, 1, 0, 0]	0	0	[1, 1, 0, 1, 1, 1, 0, 0]	0	0
[0, 1, 0, 1, 1, 1, 0, 1]	0	0	[1, 1, 0, 1, 1, 1, 0, 1]	0	0
[0, 1, 0, 1, 1, 1, 1, 0]	0	0	[1, 1, 0, 1, 1, 1, 1, 0]	0	0
[0, 1, 0, 1, 1, 1, 1, 1]	0	0	[1, 1, 0, 1, 1, 1, 1, 1]	0	0
[0, 1, 1, 0, 0, 0, 0, 0]	0	0	[1, 1, 1, 0, 0, 0, 0, 0]	0	0
[0, 1, 1, 0, 0, 0, 0, 1]	0	0	[1, 1, 1, 0, 0, 0, 0, 1]	0	0
[0, 1, 1, 0, 0, 0, 1, 0]	0	0	[1, 1, 1, 0, 0, 0, 1, 0]	0	0
[0, 1, 1, 0, 0, 0, 1, 1]	0	0	[1, 1, 1, 0, 0, 0, 1, 1]	0	0
[0, 1, 1, 0, 0, 1, 0, 0]	0	0	[1, 1, 1, 0, 0, 1, 0, 0]	0	0
[0, 1, 1, 0, 0, 1, 0, 1]	0	0	[1, 1, 1, 0, 0, 1, 0, 1]	0	0
[0, 1, 1, 0, 0, 1, 1, 0]	0	0	[1, 1, 1, 0, 0, 1, 1, 0]	0	0
[0, 1, 1, 0, 0, 1, 1, 1]	0	0	[1, 1, 1, 0, 0, 1, 1, 1]	0	0
[0, 1, 1, 0, 1, 0, 0, 0]	0	0	[1, 1, 1, 0, 1, 0, 0, 0]	0	0
[0, 1, 1, 0, 1, 0, 0, 1]	0	0	[1, 1, 1, 0, 1, 0, 0, 1]	0	0
[0, 1, 1, 0, 1, 0, 1, 0]	0	0	[1, 1, 1, 0, 1, 0, 1, 0]	0	0
[0, 1, 1, 0, 1, 0, 1, 1]	0	0	[1, 1, 1, 0, 1, 0, 1, 1]	0	0
[0, 1, 1, 0, 1, 1, 0, 0]	0	0	[1, 1, 1, 0, 1, 1, 0, 0]	0	0

Continued on next page

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
$[0,1,1,0,1,1,0,1]$	0	0	$[1,1,1,0,1,1,0,1]$	0	0
$[0,1,1,0,1,1,1,0]$	0	0	$[1,1,1,0,1,1,1,0]$	0	0
$[0,1,1,0,1,1,1,1]$	0	0	$[1,1,1,0,1,1,1,1]$	0	0
$[0,1,1,1,0,0,0,0]$	0	0	$[1,1,1,1,0,0,0,0]$	0	0
$[0,1,1,1,0,0,0,1]$	0	0	$[1,1,1,1,0,0,0,1]$	0	0
$[0,1,1,1,0,0,1,0]$	0	0	$[1,1,1,1,0,0,1,0]$	0	0
$[0,1,1,1,0,0,1,1]$	0	0	$[1,1,1,1,0,0,1,1]$	0	0
$[0,1,1,1,0,1,0,0]$	0	0	$[1,1,1,1,0,1,0,0]$	0	0
$[0,1,1,1,0,1,0,1]$	0	0	$[1,1,1,1,0,1,0,1]$	0	0
$[0,1,1,1,0,1,1,0]$	0	0	$[1,1,1,1,0,1,1,0]$	0	0
$[0,1,1,1,0,1,1,1]$	0	0	$[1,1,1,1,0,1,1,1]$	0	0
$[0,1,1,1,1,0,0,0]$	0	0	$[1,1,1,1,1,0,0,0]$	0	0
$[0,1,1,1,1,0,0,1]$	0	0	$[1,1,1,1,1,0,0,1]$	0	0
$[0,1,1,1,1,0,1,0]$	0	0	$[1,1,1,1,1,0,1,0]$	0	0
$[0,1,1,1,1,0,1,1]$	0	0	$[1,1,1,1,1,0,1,1]$	0	0
$[0,1,1,1,1,1,0,0]$	0	0	$[1,1,1,1,1,1,0,0]$	0	0
$[0,1,1,1,1,1,0,1]$	0	0	$[1,1,1,1,1,1,0,1]$	0	0
$[0,1,1,1,1,1,1,0]$	0	0	$[1,1,1,1,1,1,1,0]$	0	0
$[0,1,1,1,1,1,1,1]$	0	0	$[1,1,1,1,1,1,1,1]$	52,494,970	0.1068
Spike Entropy			1.6643		
Firing Rate			0.1905		

Table 16: Distribution of the spike patterns in the encoding layer of Spikformer with direct coding on CIFAR-10 at $T = 8$.

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
[0, 0, 0, 0, 0, 0, 0, 0]	330,957,691	0.67334	[1, 0, 0, 0, 0, 0, 0, 0]	0	0
[0, 0, 0, 0, 0, 0, 0, 1]	771,824	0.00157	[1, 0, 0, 0, 0, 0, 0, 1]	0	0
[0, 0, 0, 0, 0, 0, 1, 0]	1,528,431	0.00311	[1, 0, 0, 0, 0, 0, 1, 0]	0	0
[0, 0, 0, 0, 0, 0, 1, 1]	0	0	[1, 0, 0, 0, 0, 0, 1, 1]	0	0
[0, 0, 0, 0, 0, 1, 0, 0]	3,246,666	0.00660	[1, 0, 0, 0, 0, 1, 0, 0]	0	0
[0, 0, 0, 0, 0, 1, 0, 1]	0	0	[1, 0, 0, 0, 0, 1, 0, 1]	0	0
[0, 0, 0, 0, 0, 1, 1, 0]	0	0	[1, 0, 0, 0, 0, 1, 1, 0]	0	0
[0, 0, 0, 0, 0, 1, 1, 1]	0	0	[1, 0, 0, 0, 0, 1, 1, 1]	0	0
[0, 0, 0, 0, 1, 0, 0, 0]	6,402,526	0.01303	[1, 0, 0, 0, 1, 0, 0, 0]	0	0
[0, 0, 0, 0, 1, 0, 0, 1]	0	0	[1, 0, 0, 0, 1, 0, 0, 1]	0	0
[0, 0, 0, 0, 1, 0, 1, 0]	0	0	[1, 0, 0, 0, 1, 0, 1, 0]	0	0
[0, 0, 0, 0, 1, 0, 1, 1]	0	0	[1, 0, 0, 0, 1, 0, 1, 1]	0	0
[0, 0, 0, 0, 1, 1, 0, 0]	0	0	[1, 0, 0, 0, 1, 1, 0, 0]	0	0
[0, 0, 0, 0, 1, 1, 0, 1]	0	0	[1, 0, 0, 0, 1, 1, 0, 1]	0	0
[0, 0, 0, 0, 1, 1, 1, 0]	0	0	[1, 0, 0, 0, 1, 1, 1, 0]	0	0
[0, 0, 0, 0, 1, 1, 1, 1]	0	0	[1, 0, 0, 0, 1, 1, 1, 1]	0	0
[0, 0, 0, 1, 0, 0, 0, 0]	0	0	[1, 0, 0, 1, 0, 0, 0, 0]	0	0
[0, 0, 0, 1, 0, 0, 0, 1]	13,208,866	0.02687	[1, 0, 0, 1, 0, 0, 0, 1]	0	0
[0, 0, 0, 1, 0, 0, 1, 0]	0	0	[1, 0, 0, 1, 0, 0, 1, 0]	0	0
[0, 0, 0, 1, 0, 0, 1, 1]	0	0	[1, 0, 0, 1, 0, 0, 1, 1]	0	0
[0, 0, 0, 1, 0, 1, 0, 0]	0	0	[1, 0, 0, 1, 0, 1, 0, 0]	0	0
[0, 0, 0, 1, 0, 1, 0, 1]	0	0	[1, 0, 0, 1, 0, 1, 0, 1]	0	0
[0, 0, 0, 1, 0, 1, 1, 0]	0	0	[1, 0, 0, 1, 0, 1, 1, 0]	0	0
[0, 0, 0, 1, 0, 1, 1, 1]	0	0	[1, 0, 0, 1, 0, 1, 1, 1]	0	0
[0, 0, 0, 1, 1, 0, 0, 0]	0	0	[1, 0, 0, 1, 1, 0, 0, 0]	0	0
[0, 0, 0, 1, 1, 0, 0, 1]	0	0	[1, 0, 0, 1, 1, 0, 0, 1]	0	0
[0, 0, 0, 1, 1, 0, 1, 0]	0	0	[1, 0, 0, 1, 1, 0, 1, 0]	0	0
[0, 0, 0, 1, 1, 0, 1, 1]	0	0	[1, 0, 0, 1, 1, 0, 1, 1]	0	0
[0, 0, 0, 1, 1, 1, 0, 0]	0	0	[1, 0, 0, 1, 1, 1, 0, 0]	0	0
[0, 0, 0, 1, 1, 1, 0, 1]	0	0	[1, 0, 0, 1, 1, 1, 0, 1]	0	0
[0, 0, 0, 1, 1, 1, 1, 0]	0	0	[1, 0, 0, 1, 1, 1, 1, 0]	0	0
[0, 0, 0, 1, 1, 1, 1, 1]	0	0	[1, 0, 0, 1, 1, 1, 1, 1]	0	0
[0, 0, 1, 0, 0, 0, 0, 0]	0	0	[1, 0, 1, 0, 0, 0, 0, 0]	0	0
[0, 0, 1, 0, 0, 0, 0, 1]	0	0	[1, 0, 1, 0, 0, 0, 0, 1]	0	0
[0, 0, 1, 0, 0, 0, 1, 0]	0	0	[1, 0, 1, 0, 0, 0, 1, 0]	0	0
[0, 0, 1, 0, 0, 0, 1, 1]	0	0	[1, 0, 1, 0, 0, 0, 1, 1]	0	0
[0, 0, 1, 0, 0, 1, 0, 0]	27,747,968	0.05645	[1, 0, 1, 0, 0, 1, 0, 0]	0	0
[0, 0, 1, 0, 0, 1, 0, 1]	0	0	[1, 0, 1, 0, 0, 1, 0, 1]	0	0
[0, 0, 1, 0, 0, 1, 1, 0]	0	0	[1, 0, 1, 0, 0, 1, 1, 0]	0	0
[0, 0, 1, 0, 0, 1, 1, 1]	0	0	[1, 0, 1, 0, 0, 1, 1, 1]	0	0
[0, 0, 1, 0, 1, 0, 0, 0]	0	0	[1, 0, 1, 0, 1, 0, 0, 0]	0	0
[0, 0, 1, 0, 1, 0, 0, 1]	0	0	[1, 0, 1, 0, 1, 0, 0, 1]	0	0
[0, 0, 1, 0, 1, 0, 1, 0]	0	0	[1, 0, 1, 0, 1, 0, 1, 0]	0	0
[0, 0, 1, 0, 1, 0, 1, 1]	0	0	[1, 0, 1, 0, 1, 0, 1, 1]	0	0
[0, 0, 1, 0, 1, 1, 0, 0]	0	0	[1, 0, 1, 0, 1, 1, 0, 0]	0	0
[0, 0, 1, 0, 1, 1, 0, 1]	0	0	[1, 0, 1, 0, 1, 1, 0, 1]	0	0
[0, 0, 1, 0, 1, 1, 1, 0]	0	0	[1, 0, 1, 0, 1, 1, 1, 0]	0	0
[0, 0, 1, 0, 1, 1, 1, 1]	0	0	[1, 0, 1, 0, 1, 1, 1, 1]	0	0
[0, 0, 1, 1, 0, 0, 0, 0]	0	0	[1, 0, 1, 1, 0, 0, 0, 0]	0	0
[0, 0, 1, 1, 0, 0, 0, 1]	0	0	[1, 0, 1, 1, 0, 0, 0, 1]	0	0
[0, 0, 1, 1, 0, 0, 1, 0]	0	0	[1, 0, 1, 1, 0, 0, 1, 0]	0	0
[0, 0, 1, 1, 0, 0, 1, 1]	0	0	[1, 0, 1, 1, 0, 0, 1, 1]	0	0
[0, 0, 1, 1, 0, 1, 0, 0]	0	0	[1, 0, 1, 1, 0, 1, 0, 0]	0	0
[0, 0, 1, 1, 0, 1, 0, 1]	0	0	[1, 0, 1, 1, 0, 1, 0, 1]	0	0

Continued on next page

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
[0, 0, 1, 1, 0, 1, 1, 0]	0	0	[1, 0, 1, 1, 0, 1, 1, 0]	0	0
[0, 0, 1, 1, 0, 1, 1, 1]	0	0	[1, 0, 1, 1, 0, 1, 1, 1]	0	0
[0, 0, 1, 1, 1, 0, 0, 0]	0	0	[1, 0, 1, 1, 1, 0, 0, 0]	0	0
[0, 0, 1, 1, 1, 0, 0, 1]	0	0	[1, 0, 1, 1, 1, 0, 0, 1]	0	0
[0, 0, 1, 1, 1, 0, 1, 0]	0	0	[1, 0, 1, 1, 1, 0, 1, 0]	0	0
[0, 0, 1, 1, 1, 0, 1, 1]	0	0	[1, 0, 1, 1, 1, 0, 1, 1]	0	0
[0, 0, 1, 1, 1, 1, 0, 0]	0	0	[1, 0, 1, 1, 1, 1, 0, 0]	0	0
[0, 0, 1, 1, 1, 1, 0, 1]	0	0	[1, 0, 1, 1, 1, 1, 0, 1]	0	0
[0, 0, 1, 1, 1, 1, 1, 0]	0	0	[1, 0, 1, 1, 1, 1, 1, 0]	0	0
[0, 0, 1, 1, 1, 1, 1, 1]	0	0	[1, 0, 1, 1, 1, 1, 1, 1]	0	0
[0, 1, 0, 0, 0, 0, 0, 0]	0	0	[1, 1, 0, 0, 0, 0, 0, 0]	0	0
[0, 1, 0, 0, 0, 0, 0, 1]	0	0	[1, 1, 0, 0, 0, 0, 0, 1]	0	0
[0, 1, 0, 0, 0, 0, 1, 0]	0	0	[1, 1, 0, 0, 0, 0, 1, 0]	0	0
[0, 1, 0, 0, 0, 0, 1, 1]	0	0	[1, 1, 0, 0, 0, 0, 1, 1]	0	0
[0, 1, 0, 0, 0, 1, 0, 0]	0	0	[1, 1, 0, 0, 0, 1, 0, 0]	0	0
[0, 1, 0, 0, 0, 1, 0, 1]	0	0	[1, 1, 0, 0, 0, 1, 0, 1]	0	0
[0, 1, 0, 0, 0, 1, 1, 0]	0	0	[1, 1, 0, 0, 0, 1, 1, 0]	0	0
[0, 1, 0, 0, 0, 1, 1, 1]	0	0	[1, 1, 0, 0, 0, 1, 1, 1]	0	0
[0, 1, 0, 0, 1, 0, 0, 0]	0	0	[1, 1, 0, 0, 1, 0, 0, 0]	0	0
[0, 1, 0, 0, 1, 0, 0, 1]	0	0	[1, 1, 0, 0, 1, 0, 0, 1]	0	0
[0, 1, 0, 0, 1, 0, 1, 0]	0	0	[1, 1, 0, 0, 1, 0, 1, 0]	0	0
[0, 1, 0, 0, 1, 0, 1, 1]	0	0	[1, 1, 0, 0, 1, 0, 1, 1]	0	0
[0, 1, 0, 0, 1, 1, 0, 0]	0	0	[1, 1, 0, 0, 1, 1, 0, 0]	0	0
[0, 1, 0, 0, 1, 1, 0, 1]	0	0	[1, 1, 0, 0, 1, 1, 0, 1]	0	0
[0, 1, 0, 0, 1, 1, 1, 0]	0	0	[1, 1, 0, 0, 1, 1, 1, 0]	0	0
[0, 1, 0, 0, 1, 1, 1, 1]	0	0	[1, 1, 0, 0, 1, 1, 1, 1]	0	0
[0, 1, 0, 1, 0, 0, 0, 0]	0	0	[1, 1, 0, 1, 0, 0, 0, 0]	0	0
[0, 1, 0, 1, 0, 0, 0, 1]	0	0	[1, 1, 0, 1, 0, 0, 0, 1]	0	0
[0, 1, 0, 1, 0, 0, 1, 0]	0	0	[1, 1, 0, 1, 0, 0, 1, 0]	0	0
[0, 1, 0, 1, 0, 0, 1, 1]	0	0	[1, 1, 0, 1, 0, 0, 1, 1]	0	0
[0, 1, 0, 1, 0, 1, 0, 0]	0	0	[1, 1, 0, 1, 0, 1, 0, 0]	0	0
[0, 1, 0, 1, 0, 1, 0, 1]	55,169,205	0.11224	[1, 1, 0, 1, 0, 1, 0, 1]	0	0
[0, 1, 0, 1, 0, 1, 1, 0]	0	0	[1, 1, 0, 1, 0, 1, 1, 0]	0	0
[0, 1, 0, 1, 0, 1, 1, 1]	0	0	[1, 1, 0, 1, 0, 1, 1, 1]	0	0
[0, 1, 0, 1, 1, 0, 0, 0]	0	0	[1, 1, 0, 1, 1, 0, 0, 0]	0	0
[0, 1, 0, 1, 1, 0, 0, 1]	0	0	[1, 1, 0, 1, 1, 0, 0, 1]	0	0
[0, 1, 0, 1, 1, 0, 1, 0]	0	0	[1, 1, 0, 1, 1, 0, 1, 0]	0	0
[0, 1, 0, 1, 1, 0, 1, 1]	0	0	[1, 1, 0, 1, 1, 0, 1, 1]	0	0
[0, 1, 0, 1, 1, 1, 0, 0]	0	0	[1, 1, 0, 1, 1, 1, 0, 0]	0	0
[0, 1, 0, 1, 1, 1, 0, 1]	0	0	[1, 1, 0, 1, 1, 1, 0, 1]	0	0
[0, 1, 0, 1, 1, 1, 1, 0]	0	0	[1, 1, 0, 1, 1, 1, 1, 0]	0	0
[0, 1, 0, 1, 1, 1, 1, 1]	0	0	[1, 1, 0, 1, 1, 1, 1, 1]	0	0
[0, 1, 1, 0, 0, 0, 0, 0]	0	0	[1, 1, 1, 0, 0, 0, 0, 0]	0	0
[0, 1, 1, 0, 0, 0, 0, 1]	0	0	[1, 1, 1, 0, 0, 0, 0, 1]	0	0
[0, 1, 1, 0, 0, 0, 1, 0]	0	0	[1, 1, 1, 0, 0, 0, 1, 0]	0	0
[0, 1, 1, 0, 0, 0, 1, 1]	0	0	[1, 1, 1, 0, 0, 0, 1, 1]	0	0
[0, 1, 1, 0, 0, 1, 0, 0]	0	0	[1, 1, 1, 0, 0, 1, 0, 0]	0	0
[0, 1, 1, 0, 0, 1, 0, 1]	0	0	[1, 1, 1, 0, 0, 1, 0, 1]	0	0
[0, 1, 1, 0, 0, 1, 1, 0]	0	0	[1, 1, 1, 0, 0, 1, 1, 0]	0	0
[0, 1, 1, 0, 0, 1, 1, 1]	0	0	[1, 1, 1, 0, 0, 1, 1, 1]	0	0
[0, 1, 1, 0, 1, 0, 0, 0]	0	0	[1, 1, 1, 0, 1, 0, 0, 0]	0	0
[0, 1, 1, 0, 1, 0, 0, 1]	0	0	[1, 1, 1, 0, 1, 0, 0, 1]	0	0
[0, 1, 1, 0, 1, 0, 1, 0]	0	0	[1, 1, 1, 0, 1, 0, 1, 0]	0	0
[0, 1, 1, 0, 1, 0, 1, 1]	0	0	[1, 1, 1, 0, 1, 0, 1, 1]	0	0
[0, 1, 1, 0, 1, 1, 0, 0]	0	0	[1, 1, 1, 0, 1, 1, 0, 0]	0	0

Continued on next page

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
$[0, 1, 1, 0, 1, 1, 0, 1]$	0	0	$[1, 1, 1, 0, 1, 1, 0, 1]$	0	0
$[0, 1, 1, 0, 1, 1, 1, 0]$	0	0	$[1, 1, 1, 0, 1, 1, 1, 0]$	0	0
$[0, 1, 1, 0, 1, 1, 1, 1]$	0	0	$[1, 1, 1, 0, 1, 1, 1, 1]$	0	0
$[0, 1, 1, 1, 0, 0, 0, 0]$	0	0	$[1, 1, 1, 1, 0, 0, 0, 0]$	0	0
$[0, 1, 1, 1, 0, 0, 0, 1]$	0	0	$[1, 1, 1, 1, 0, 0, 0, 1]$	0	0
$[0, 1, 1, 1, 0, 0, 1, 0]$	0	0	$[1, 1, 1, 1, 0, 0, 1, 0]$	0	0
$[0, 1, 1, 1, 0, 0, 1, 1]$	0	0	$[1, 1, 1, 1, 0, 0, 1, 1]$	0	0
$[0, 1, 1, 1, 0, 1, 0, 0]$	0	0	$[1, 1, 1, 1, 0, 1, 0, 0]$	0	0
$[0, 1, 1, 1, 0, 1, 0, 1]$	0	0	$[1, 1, 1, 1, 0, 1, 0, 1]$	0	0
$[0, 1, 1, 1, 0, 1, 1, 0]$	0	0	$[1, 1, 1, 1, 0, 1, 1, 0]$	0	0
$[0, 1, 1, 1, 0, 1, 1, 1]$	0	0	$[1, 1, 1, 1, 0, 1, 1, 1]$	0	0
$[0, 1, 1, 1, 1, 0, 0, 0]$	0	0	$[1, 1, 1, 1, 1, 0, 0, 0]$	0	0
$[0, 1, 1, 1, 1, 0, 0, 1]$	0	0	$[1, 1, 1, 1, 1, 0, 0, 1]$	0	0
$[0, 1, 1, 1, 1, 0, 1, 0]$	0	0	$[1, 1, 1, 1, 1, 0, 1, 0]$	0	0
$[0, 1, 1, 1, 1, 0, 1, 1]$	0	0	$[1, 1, 1, 1, 1, 0, 1, 1]$	0	0
$[0, 1, 1, 1, 1, 1, 0, 0]$	0	0	$[1, 1, 1, 1, 1, 1, 0, 0]$	0	0
$[0, 1, 1, 1, 1, 1, 0, 1]$	0	0	$[1, 1, 1, 1, 1, 1, 0, 1]$	0	0
$[0, 1, 1, 1, 1, 1, 1, 0]$	0	0	$[1, 1, 1, 1, 1, 1, 1, 0]$	0	0
$[0, 1, 1, 1, 1, 1, 1, 1]$	0	0	$[1, 1, 1, 1, 1, 1, 1, 1]$	52,486,823	0.10679
Spike Entropy			1.6272		
Firing Rate			0.1868		

Table 17: Distribution of the spike patterns in the encoding layer of Spikformer with GAC coding on CIFAR-10 at $T = 8$.

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
[0, 0, 0, 0, 0, 0, 0, 0]	338,033,724	0.68773	[1, 0, 0, 0, 0, 0, 0, 0]	0	0
[0, 0, 0, 0, 0, 0, 0, 1]	420,120	0.00085	[1, 0, 0, 0, 0, 0, 0, 1]	0	0
[0, 0, 0, 0, 0, 0, 1, 0]	791,838	0.00161	[1, 0, 0, 0, 0, 0, 1, 0]	0	0
[0, 0, 0, 0, 0, 0, 1, 1]	0	0	[1, 0, 0, 0, 0, 0, 1, 1]	0	0
[0, 0, 0, 0, 0, 1, 0, 0]	1,570,624	0.00319	[1, 0, 0, 0, 0, 1, 0, 0]	0	0
[0, 0, 0, 0, 0, 1, 0, 1]	0	0	[1, 0, 0, 0, 0, 1, 0, 1]	0	0
[0, 0, 0, 0, 0, 1, 1, 0]	0	0	[1, 0, 0, 0, 0, 1, 1, 0]	0	0
[0, 0, 0, 0, 0, 1, 1, 1]	0	0	[1, 0, 0, 0, 0, 1, 1, 1]	0	0
[0, 0, 0, 0, 1, 0, 0, 0]	3,244,095	0.00660	[1, 0, 0, 0, 1, 0, 0, 0]	0	0
[0, 0, 0, 0, 1, 0, 0, 1]	0	0	[1, 0, 0, 0, 1, 0, 0, 1]	0	0
[0, 0, 0, 0, 1, 0, 1, 0]	0	0	[1, 0, 0, 0, 1, 0, 1, 0]	0	0
[0, 0, 0, 0, 1, 0, 1, 1]	0	0	[1, 0, 0, 0, 1, 0, 1, 1]	0	0
[0, 0, 0, 0, 1, 1, 0, 0]	0	0	[1, 0, 0, 0, 1, 1, 0, 0]	0	0
[0, 0, 0, 0, 1, 1, 0, 1]	0	0	[1, 0, 0, 0, 1, 1, 0, 1]	0	0
[0, 0, 0, 0, 1, 1, 1, 0]	0	0	[1, 0, 0, 0, 1, 1, 1, 0]	0	0
[0, 0, 0, 0, 1, 1, 1, 1]	0	0	[1, 0, 0, 0, 1, 1, 1, 1]	0	0
[0, 0, 0, 1, 0, 0, 0, 0]	5,826,008	0.01185	[1, 0, 0, 1, 0, 0, 0, 0]	0	0
[0, 0, 0, 1, 0, 0, 0, 1]	1,168,235	0.00238	[1, 0, 0, 1, 0, 0, 0, 1]	0	0
[0, 0, 0, 1, 0, 0, 1, 0]	0	0	[1, 0, 0, 1, 0, 0, 1, 0]	0	0
[0, 0, 0, 1, 0, 0, 1, 1]	0	0	[1, 0, 0, 1, 0, 0, 1, 1]	0	0
[0, 0, 0, 1, 0, 1, 0, 0]	0	0	[1, 0, 0, 1, 0, 1, 0, 0]	0	0
[0, 0, 0, 1, 0, 1, 0, 1]	0	0	[1, 0, 0, 1, 0, 1, 0, 1]	0	0
[0, 0, 0, 1, 0, 1, 1, 0]	0	0	[1, 0, 0, 1, 0, 1, 1, 0]	0	0
[0, 0, 0, 1, 0, 1, 1, 1]	0	0	[1, 0, 0, 1, 0, 1, 1, 1]	0	0
[0, 0, 0, 1, 1, 0, 0, 0]	0	0	[1, 0, 0, 1, 1, 0, 0, 0]	0	0
[0, 0, 0, 1, 1, 0, 0, 1]	0	0	[1, 0, 0, 1, 1, 0, 0, 1]	0	0
[0, 0, 0, 1, 1, 0, 1, 0]	0	0	[1, 0, 0, 1, 1, 0, 1, 0]	0	0
[0, 0, 0, 1, 1, 0, 1, 1]	0	0	[1, 0, 0, 1, 1, 0, 1, 1]	0	0
[0, 0, 0, 1, 1, 1, 0, 0]	0	0	[1, 0, 0, 1, 1, 1, 0, 0]	0	0
[0, 0, 0, 1, 1, 1, 0, 1]	0	0	[1, 0, 0, 1, 1, 1, 0, 1]	0	0
[0, 0, 0, 1, 1, 1, 1, 0]	0	0	[1, 0, 0, 1, 1, 1, 1, 0]	0	0
[0, 0, 0, 1, 1, 1, 1, 1]	0	0	[1, 0, 0, 1, 1, 1, 1, 1]	0	0
[0, 0, 1, 0, 0, 0, 0, 0]	0	0	[1, 0, 1, 0, 0, 0, 0, 0]	0	0
[0, 0, 1, 0, 0, 0, 0, 1]	273,521	0.00056	[1, 0, 1, 0, 0, 0, 0, 1]	0	0
[0, 0, 1, 0, 0, 0, 1, 0]	11,686,870	0.02378	[1, 0, 1, 0, 0, 0, 1, 0]	0	0
[0, 0, 1, 0, 0, 0, 1, 1]	0	0	[1, 0, 1, 0, 0, 0, 1, 1]	0	0
[0, 0, 1, 0, 0, 1, 0, 0]	3,856,410	0.00785	[1, 0, 1, 0, 0, 1, 0, 0]	0	0
[0, 0, 1, 0, 0, 1, 0, 1]	0	0	[1, 0, 1, 0, 0, 1, 0, 1]	0	0
[0, 0, 1, 0, 0, 1, 1, 0]	0	0	[1, 0, 1, 0, 0, 1, 1, 0]	0	0
[0, 0, 1, 0, 0, 1, 1, 1]	0	0	[1, 0, 1, 0, 0, 1, 1, 1]	0	0
[0, 0, 1, 0, 1, 0, 0, 0]	0	0	[1, 0, 1, 0, 1, 0, 0, 0]	0	0
[0, 0, 1, 0, 1, 0, 0, 1]	0	0	[1, 0, 1, 0, 1, 0, 0, 1]	0	0
[0, 0, 1, 0, 1, 0, 1, 0]	0	0	[1, 0, 1, 0, 1, 0, 1, 0]	35,350,352	0.07192
[0, 0, 1, 0, 1, 0, 1, 1]	0	0	[1, 0, 1, 0, 1, 0, 1, 1]	0	0
[0, 0, 1, 0, 1, 1, 0, 0]	0	0	[1, 0, 1, 0, 1, 1, 0, 0]	0	0
[0, 0, 1, 0, 1, 1, 0, 1]	0	0	[1, 0, 1, 0, 1, 1, 0, 1]	0	0
[0, 0, 1, 0, 1, 1, 1, 0]	0	0	[1, 0, 1, 0, 1, 1, 1, 0]	0	0
[0, 0, 1, 0, 1, 1, 1, 1]	0	0	[1, 0, 1, 0, 1, 1, 1, 1]	0	0
[0, 0, 1, 1, 0, 0, 0, 0]	0	0	[1, 0, 1, 1, 0, 0, 0, 0]	0	0
[0, 0, 1, 1, 0, 0, 0, 1]	0	0	[1, 0, 1, 1, 0, 0, 0, 1]	0	0
[0, 0, 1, 1, 0, 0, 1, 0]	0	0	[1, 0, 1, 1, 0, 0, 1, 0]	0	0
[0, 0, 1, 1, 0, 0, 1, 1]	0	0	[1, 0, 1, 1, 0, 0, 1, 1]	0	0
[0, 0, 1, 1, 0, 1, 0, 0]	0	0	[1, 0, 1, 1, 0, 1, 0, 0]	0	0
[0, 0, 1, 1, 0, 1, 0, 1]	0	0	[1, 0, 1, 1, 0, 1, 0, 1]	0	0

Continued on next page

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
[0, 0, 1, 1, 0, 1, 1, 0]	0	0	[1, 0, 1, 1, 0, 1, 1, 0]	0	0
[0, 0, 1, 1, 0, 1, 1, 1]	0	0	[1, 0, 1, 1, 0, 1, 1, 1]	0	0
[0, 0, 1, 1, 1, 0, 0, 0]	0	0	[1, 0, 1, 1, 1, 0, 0, 0]	0	0
[0, 0, 1, 1, 1, 0, 0, 1]	0	0	[1, 0, 1, 1, 1, 0, 0, 1]	0	0
[0, 0, 1, 1, 1, 0, 1, 0]	0	0	[1, 0, 1, 1, 1, 0, 1, 0]	0	0
[0, 0, 1, 1, 1, 0, 1, 1]	0	0	[1, 0, 1, 1, 1, 0, 1, 1]	0	0
[0, 0, 1, 1, 1, 1, 0, 0]	0	0	[1, 0, 1, 1, 1, 1, 0, 0]	0	0
[0, 0, 1, 1, 1, 1, 0, 1]	0	0	[1, 0, 1, 1, 1, 1, 0, 1]	0	0
[0, 0, 1, 1, 1, 1, 1, 0]	0	0	[1, 0, 1, 1, 1, 1, 1, 0]	0	0
[0, 0, 1, 1, 1, 1, 1, 1]	0	0	[1, 0, 1, 1, 1, 1, 1, 1]	0	0
[0, 1, 0, 0, 0, 0, 0, 0]	0	0	[1, 1, 0, 0, 0, 0, 0, 0]	0	0
[0, 1, 0, 0, 0, 0, 0, 1]	0	0	[1, 1, 0, 0, 0, 0, 0, 1]	0	0
[0, 1, 0, 0, 0, 0, 1, 0]	0	0	[1, 1, 0, 0, 0, 0, 1, 0]	0	0
[0, 1, 0, 0, 0, 0, 1, 1]	0	0	[1, 1, 0, 0, 0, 0, 1, 1]	0	0
[0, 1, 0, 0, 0, 1, 0, 0]	103,065	0.00021	[1, 1, 0, 0, 0, 1, 0, 0]	0	0
[0, 1, 0, 0, 0, 1, 0, 1]	0	0	[1, 1, 0, 0, 0, 1, 0, 1]	0	0
[0, 1, 0, 0, 0, 1, 1, 0]	0	0	[1, 1, 0, 0, 0, 1, 1, 0]	0	0
[0, 1, 0, 0, 0, 1, 1, 1]	0	0	[1, 1, 0, 0, 0, 1, 1, 1]	0	0
[0, 1, 0, 0, 1, 0, 0, 0]	0	0	[1, 1, 0, 0, 1, 0, 0, 0]	0	0
[0, 1, 0, 0, 1, 0, 0, 1]	22,682,460	0.04615	[1, 1, 0, 0, 1, 0, 0, 1]	0	0
[0, 1, 0, 0, 1, 0, 1, 0]	0	0	[1, 1, 0, 0, 1, 0, 1, 0]	0	0
[0, 1, 0, 0, 1, 0, 1, 1]	0	0	[1, 1, 0, 0, 1, 0, 1, 1]	0	0
[0, 1, 0, 0, 1, 1, 0, 0]	0	0	[1, 1, 0, 0, 1, 1, 0, 0]	0	0
[0, 1, 0, 0, 1, 1, 0, 1]	0	0	[1, 1, 0, 0, 1, 1, 0, 1]	0	0
[0, 1, 0, 0, 1, 1, 1, 0]	0	0	[1, 1, 0, 0, 1, 1, 1, 0]	0	0
[0, 1, 0, 0, 1, 1, 1, 1]	0	0	[1, 1, 0, 0, 1, 1, 1, 1]	0	0
[0, 1, 0, 1, 0, 0, 0, 0]	0	0	[1, 1, 0, 1, 0, 0, 0, 0]	0	0
[0, 1, 0, 1, 0, 0, 0, 1]	0	0	[1, 1, 0, 1, 0, 0, 0, 1]	0	0
[0, 1, 0, 1, 0, 0, 1, 0]	0	0	[1, 1, 0, 1, 0, 0, 1, 0]	0	0
[0, 1, 0, 1, 0, 0, 1, 1]	0	0	[1, 1, 0, 1, 0, 0, 1, 1]	0	0
[0, 1, 0, 1, 0, 1, 0, 0]	0	0	[1, 1, 0, 1, 0, 1, 0, 0]	0	0
[0, 1, 0, 1, 0, 1, 0, 1]	17,535,536	0.03568	[1, 1, 0, 1, 0, 1, 0, 1]	0	0
[0, 1, 0, 1, 0, 1, 1, 0]	0	0	[1, 1, 0, 1, 0, 1, 1, 0]	0	0
[0, 1, 0, 1, 0, 1, 1, 1]	0	0	[1, 1, 0, 1, 0, 1, 1, 1]	0	0
[0, 1, 0, 1, 1, 0, 0, 0]	0	0	[1, 1, 0, 1, 1, 0, 0, 0]	0	0
[0, 1, 0, 1, 1, 0, 0, 1]	0	0	[1, 1, 0, 1, 1, 0, 0, 1]	0	0
[0, 1, 0, 1, 1, 0, 1, 0]	0	0	[1, 1, 0, 1, 1, 0, 1, 0]	0	0
[0, 1, 0, 1, 1, 0, 1, 1]	0	0	[1, 1, 0, 1, 1, 0, 1, 1]	0	0
[0, 1, 0, 1, 1, 1, 0, 0]	0	0	[1, 1, 0, 1, 1, 1, 0, 0]	0	0
[0, 1, 0, 1, 1, 1, 0, 1]	0	0	[1, 1, 0, 1, 1, 1, 0, 1]	0	0
[0, 1, 0, 1, 1, 1, 1, 0]	0	0	[1, 1, 0, 1, 1, 1, 1, 0]	0	0
[0, 1, 0, 1, 1, 1, 1, 1]	0	0	[1, 1, 0, 1, 1, 1, 1, 1]	0	0
[0, 1, 1, 0, 0, 0, 0, 0]	0	0	[1, 1, 1, 0, 0, 0, 0, 0]	0	0
[0, 1, 1, 0, 0, 0, 0, 1]	0	0	[1, 1, 1, 0, 0, 0, 0, 1]	0	0
[0, 1, 1, 0, 0, 0, 1, 0]	0	0	[1, 1, 1, 0, 0, 0, 1, 0]	0	0
[0, 1, 1, 0, 0, 0, 1, 1]	0	0	[1, 1, 1, 0, 0, 0, 1, 1]	0	0
[0, 1, 1, 0, 0, 1, 0, 0]	0	0	[1, 1, 1, 0, 0, 1, 0, 0]	0	0
[0, 1, 1, 0, 0, 1, 0, 1]	0	0	[1, 1, 1, 0, 0, 1, 0, 1]	0	0
[0, 1, 1, 0, 0, 1, 1, 0]	0	0	[1, 1, 1, 0, 0, 1, 1, 0]	0	0
[0, 1, 1, 0, 0, 1, 1, 1]	0	0	[1, 1, 1, 0, 0, 1, 1, 1]	0	0
[0, 1, 1, 0, 1, 0, 0, 0]	0	0	[1, 1, 1, 0, 1, 0, 0, 0]	0	0
[0, 1, 1, 0, 1, 0, 0, 1]	0	0	[1, 1, 1, 0, 1, 0, 0, 1]	0	0
[0, 1, 1, 0, 1, 0, 1, 0]	0	0	[1, 1, 1, 0, 1, 0, 1, 0]	0	0
[0, 1, 1, 0, 1, 0, 1, 1]	0	0	[1, 1, 1, 0, 1, 0, 1, 1]	0	0
[0, 1, 1, 0, 1, 1, 0, 0]	0	0	[1, 1, 1, 0, 1, 1, 0, 0]	0	0

Continued on next page

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
$[0, 1, 1, 0, 1, 1, 0, 1]$	0	0	$[1, 1, 1, 0, 1, 1, 0, 1]$	0	0
$[0, 1, 1, 0, 1, 1, 1, 0]$	0	0	$[1, 1, 1, 0, 1, 1, 1, 0]$	0	0
$[0, 1, 1, 0, 1, 1, 1, 1]$	0	0	$[1, 1, 1, 0, 1, 1, 1, 1]$	0	0
$[0, 1, 1, 1, 0, 0, 0, 0]$	0	0	$[1, 1, 1, 1, 0, 0, 0, 0]$	0	0
$[0, 1, 1, 1, 0, 0, 0, 1]$	0	0	$[1, 1, 1, 1, 0, 0, 0, 1]$	0	0
$[0, 1, 1, 1, 0, 0, 1, 0]$	0	0	$[1, 1, 1, 1, 0, 0, 1, 0]$	0	0
$[0, 1, 1, 1, 0, 0, 1, 1]$	0	0	$[1, 1, 1, 1, 0, 0, 1, 1]$	0	0
$[0, 1, 1, 1, 0, 1, 0, 0]$	0	0	$[1, 1, 1, 1, 0, 1, 0, 0]$	0	0
$[0, 1, 1, 1, 0, 1, 0, 1]$	0	0	$[1, 1, 1, 1, 0, 1, 0, 1]$	0	0
$[0, 1, 1, 1, 0, 1, 1, 0]$	0	0	$[1, 1, 1, 1, 0, 1, 1, 0]$	0	0
$[0, 1, 1, 1, 0, 1, 1, 1]$	0	0	$[1, 1, 1, 1, 0, 1, 1, 1]$	0	0
$[0, 1, 1, 1, 1, 0, 0, 0]$	0	0	$[1, 1, 1, 1, 1, 0, 0, 0]$	0	0
$[0, 1, 1, 1, 1, 0, 0, 1]$	0	0	$[1, 1, 1, 1, 1, 0, 0, 1]$	0	0
$[0, 1, 1, 1, 1, 0, 1, 0]$	0	0	$[1, 1, 1, 1, 1, 0, 1, 0]$	0	0
$[0, 1, 1, 1, 1, 0, 1, 1]$	0	0	$[1, 1, 1, 1, 1, 0, 1, 1]$	0	0
$[0, 1, 1, 1, 1, 1, 0, 0]$	0	0	$[1, 1, 1, 1, 1, 1, 0, 0]$	0	0
$[0, 1, 1, 1, 1, 1, 0, 1]$	0	0	$[1, 1, 1, 1, 1, 1, 0, 1]$	0	0
$[0, 1, 1, 1, 1, 1, 1, 0]$	0	0	$[1, 1, 1, 1, 1, 1, 1, 0]$	0	0
$[0, 1, 1, 1, 1, 1, 1, 1]$	0	0	$[1, 1, 1, 1, 1, 1, 1, 1]$	48,977,142	0.09964
Spike Entropy			1.7386		
Firing Rate			0.1825		

Table 18: Distribution of the spike patterns in the encoding layer of Spikformer with IMP coding on CIFAR-10 at $T = 8$.

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
[0, 0, 0, 0, 0, 0, 0, 0]	166,978,088	0.33972	[1, 0, 0, 0, 0, 0, 0, 0]	9,513,446	0.01936
[0, 0, 0, 0, 0, 0, 0, 1]	8,830,253	0.01796	[1, 0, 0, 0, 0, 0, 0, 1]	392,394	0.00080
[0, 0, 0, 0, 0, 0, 1, 0]	8,802,154	0.01791	[1, 0, 0, 0, 0, 0, 1, 0]	2,181,415	0.00444
[0, 0, 0, 0, 0, 0, 1, 1]	924,376	0.00188	[1, 0, 0, 0, 0, 0, 1, 1]	135,224	0.00028
[0, 0, 0, 0, 0, 1, 0, 0]	10,439,976	0.02124	[1, 0, 0, 0, 0, 1, 0, 0]	418,115	0.00085
[0, 0, 0, 0, 0, 1, 0, 1]	6,276,028	0.01277	[1, 0, 0, 0, 0, 1, 0, 1]	230,261	0.00047
[0, 0, 0, 0, 0, 1, 1, 0]	707,642	0.00144	[1, 0, 0, 0, 0, 1, 1, 0]	109,016	0.00022
[0, 0, 0, 0, 0, 1, 1, 1]	654,040	0.00133	[1, 0, 0, 0, 0, 1, 1, 1]	80,885	0.00016
[0, 0, 0, 0, 1, 0, 0, 0]	10,414,246	0.02119	[1, 0, 0, 0, 1, 0, 0, 0]	1,152,106	0.00234
[0, 0, 0, 0, 1, 0, 0, 1]	709,126	0.00144	[1, 0, 0, 0, 1, 0, 0, 1]	60,478	0.00012
[0, 0, 0, 0, 1, 0, 1, 0]	6,254,257	0.01272	[1, 0, 0, 0, 1, 0, 1, 0]	1,728,587	0.00352
[0, 0, 0, 0, 1, 0, 1, 1]	653,259	0.00133	[1, 0, 0, 0, 1, 0, 1, 1]	133,119	0.00027
[0, 0, 0, 0, 1, 1, 0, 0]	1,055,761	0.00215	[1, 0, 0, 0, 1, 1, 0, 0]	74,482	0.00015
[0, 0, 0, 0, 1, 1, 0, 1]	611,561	0.00124	[1, 0, 0, 0, 1, 1, 0, 1]	42,115	0.00009
[0, 0, 0, 0, 1, 1, 1, 0]	608,929	0.00124	[1, 0, 0, 0, 1, 1, 1, 0]	112,564	0.00023
[0, 0, 0, 0, 1, 1, 1, 1]	569,225	0.00116	[1, 0, 0, 0, 1, 1, 1, 1]	96,101	0.00020
[0, 0, 0, 1, 0, 0, 0, 0]	13,875,066	0.02823	[1, 0, 0, 1, 0, 0, 0, 0]	801,618	0.00163
[0, 0, 0, 1, 0, 0, 0, 1]	2,203,497	0.00448	[1, 0, 0, 1, 0, 0, 0, 1]	96,118	0.00020
[0, 0, 0, 1, 0, 0, 1, 0]	586,408	0.00119	[1, 0, 0, 1, 0, 0, 1, 0]	133,181	0.00027
[0, 0, 0, 1, 0, 0, 1, 1]	155,802	0.00032	[1, 0, 0, 1, 0, 0, 1, 1]	26,500	0.00005
[0, 0, 0, 1, 0, 1, 0, 0]	8,315,241	0.01692	[1, 0, 0, 1, 0, 1, 0, 0]	357,330	0.00073
[0, 0, 0, 1, 0, 1, 0, 1]	11,461,260	0.02332	[1, 0, 0, 1, 0, 1, 0, 1]	378,856	0.00077
[0, 0, 0, 1, 0, 1, 1, 0]	433,513	0.00088	[1, 0, 0, 1, 0, 1, 1, 0]	71,315	0.00015
[0, 0, 0, 1, 0, 1, 1, 1]	860,104	0.00175	[1, 0, 0, 1, 0, 1, 1, 1]	105,466	0.00021
[0, 0, 0, 1, 1, 0, 0, 0]	883,826	0.00180	[1, 0, 0, 1, 1, 0, 0, 0]	87,782	0.00018
[0, 0, 0, 1, 1, 0, 0, 1]	167,854	0.00034	[1, 0, 0, 1, 1, 0, 0, 1]	13,783	0.00003
[0, 0, 0, 1, 1, 0, 1, 0]	405,627	0.00082	[1, 0, 0, 1, 1, 0, 1, 0]	99,569	0.00020
[0, 0, 0, 1, 1, 0, 1, 1]	108,798	0.00022	[1, 0, 0, 1, 1, 0, 1, 1]	22,741	0.00005
[0, 0, 0, 1, 1, 1, 0, 0]	755,319	0.00154	[1, 0, 0, 1, 1, 1, 0, 0]	55,661	0.00011
[0, 0, 0, 1, 1, 1, 0, 1]	1,068,954	0.00217	[1, 0, 0, 1, 1, 1, 0, 1]	67,964	0.00014
[0, 0, 0, 1, 1, 1, 1, 0]	359,435	0.00073	[1, 0, 0, 1, 1, 1, 1, 0]	69,454	0.00014
[0, 0, 0, 1, 1, 1, 1, 1]	744,798	0.00152	[1, 0, 0, 1, 1, 1, 1, 1]	115,930	0.00024
[0, 0, 1, 0, 0, 0, 0, 0]	13,875,540	0.02823	[1, 0, 1, 0, 0, 0, 0, 0]	5,314,878	0.01081
[0, 0, 1, 0, 0, 0, 0, 1]	586,328	0.00119	[1, 0, 1, 0, 0, 0, 0, 1]	165,056	0.00034
[0, 0, 1, 0, 0, 0, 1, 0]	2,202,679	0.00448	[1, 0, 1, 0, 0, 0, 1, 0]	1,868,819	0.00380
[0, 0, 1, 0, 0, 0, 1, 1]	155,493	0.00032	[1, 0, 1, 0, 0, 0, 1, 1]	86,177	0.00017
[0, 0, 1, 0, 0, 1, 0, 0]	885,163	0.00180	[1, 0, 1, 0, 0, 1, 0, 0]	230,771	0.00047
[0, 0, 1, 0, 0, 1, 0, 1]	404,652	0.00082	[1, 0, 1, 0, 0, 1, 0, 1]	100,306	0.00020
[0, 0, 1, 0, 0, 1, 1, 0]	167,824	0.00034	[1, 0, 1, 0, 0, 1, 1, 0]	90,448	0.00018
[0, 0, 1, 0, 0, 1, 1, 1]	108,576	0.00022	[1, 0, 1, 0, 0, 1, 1, 1]	50,985	0.00010
[0, 0, 1, 0, 1, 0, 0, 0]	8,283,191	0.01685	[1, 0, 1, 0, 1, 0, 0, 0]	3,759,113	0.00765
[0, 0, 1, 0, 1, 0, 0, 1]	434,009	0.00088	[1, 0, 1, 0, 1, 0, 0, 1]	150,195	0.00031
[0, 0, 1, 0, 1, 0, 1, 0]	11,461,617	0.02332	[1, 0, 1, 0, 1, 0, 1, 0]	25,557,536	0.05200
[0, 0, 1, 0, 1, 0, 1, 1]	863,192	0.00176	[1, 0, 1, 0, 1, 0, 1, 1]	1,423,576	0.00290
[0, 0, 1, 0, 1, 1, 0, 0]	754,439	0.00153	[1, 0, 1, 0, 1, 1, 0, 0]	236,823	0.00048
[0, 0, 1, 0, 1, 1, 0, 1]	358,953	0.00073	[1, 0, 1, 0, 1, 1, 0, 1]	105,855	0.00021
[0, 0, 1, 0, 1, 1, 1, 0]	1,070,523	0.00218	[1, 0, 1, 0, 1, 1, 1, 0]	1,705,014	0.00347
[0, 0, 1, 0, 1, 1, 1, 1]	749,485	0.00153	[1, 0, 1, 0, 1, 1, 1, 1]	1,076,246	0.00219
[0, 0, 1, 1, 0, 0, 0, 0]	2,069,223	0.00421	[1, 0, 1, 1, 0, 0, 0, 0]	619,136	0.00126
[0, 0, 1, 1, 0, 0, 0, 1]	246,054	0.00050	[1, 0, 1, 1, 0, 0, 0, 1]	59,275	0.00012
[0, 0, 1, 1, 0, 0, 1, 0]	246,872	0.00050	[1, 0, 1, 1, 0, 0, 1, 0]	174,810	0.00036
[0, 0, 1, 1, 0, 0, 1, 1]	51,835	0.00011	[1, 0, 1, 1, 0, 0, 1, 1]	26,124	0.00005
[0, 0, 1, 1, 0, 1, 0, 0]	1,162,307	0.00236	[1, 0, 1, 1, 0, 1, 0, 0]	264,551	0.00054
[0, 0, 1, 1, 0, 1, 0, 1]	1,138,064	0.00231	[1, 0, 1, 1, 0, 1, 0, 1]	228,051	0.00046

Continued on next page

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
[0, 0, 1, 1, 0, 1, 1, 0]	173,135	0.00035	[1, 0, 1, 1, 0, 1, 1, 0]	86,136	0.00017
[0, 0, 1, 1, 0, 1, 1, 1]	258,765	0.00053	[1, 0, 1, 1, 0, 1, 1, 1]	99,880	0.00020
[0, 0, 1, 1, 1, 0, 0, 0]	1,156,988	0.00235	[1, 0, 1, 1, 1, 0, 0, 0]	403,385	0.00082
[0, 0, 1, 1, 1, 0, 0, 1]	172,341	0.00035	[1, 0, 1, 1, 1, 0, 0, 1]	50,620	0.00010
[0, 0, 1, 1, 1, 0, 1, 0]	1,139,849	0.00232	[1, 0, 1, 1, 1, 0, 1, 0]	2,158,541	0.00439
[0, 0, 1, 1, 1, 0, 1, 1]	260,298	0.00053	[1, 0, 1, 1, 1, 0, 1, 1]	413,268	0.00084
[0, 0, 1, 1, 1, 1, 0, 0]	938,898	0.00191	[1, 0, 1, 1, 1, 1, 0, 0]	256,378	0.00052
[0, 0, 1, 1, 1, 1, 0, 1]	1,051,950	0.00214	[1, 0, 1, 1, 1, 1, 0, 1]	249,442	0.00051
[0, 0, 1, 1, 1, 1, 1, 0]	1,053,394	0.00214	[1, 0, 1, 1, 1, 1, 1, 0]	1,597,605	0.00325
[0, 0, 1, 1, 1, 1, 1, 1]	1,832,734	0.00373	[1, 0, 1, 1, 1, 1, 1, 1]	2,463,570	0.00501
[0, 1, 0, 0, 0, 0, 0, 0]	9,590,581	0.01951	[1, 1, 0, 0, 0, 0, 0, 0]	1,647,164	0.00335
[0, 1, 0, 0, 0, 0, 0, 1]	2,226,551	0.00453	[1, 1, 0, 0, 0, 0, 0, 1]	340,639	0.00069
[0, 1, 0, 0, 0, 0, 1, 0]	397,399	0.00081	[1, 1, 0, 0, 0, 0, 1, 0]	342,632	0.00070
[0, 1, 0, 0, 0, 0, 1, 1]	137,468	0.00028	[1, 1, 0, 0, 0, 0, 1, 1]	101,762	0.00021
[0, 1, 0, 0, 0, 1, 0, 0]	1,172,184	0.00238	[1, 1, 0, 0, 0, 1, 0, 0]	157,183	0.00032
[0, 1, 0, 0, 0, 1, 0, 1]	1,748,051	0.00356	[1, 1, 0, 0, 0, 1, 0, 1]	196,883	0.00040
[0, 1, 0, 0, 0, 1, 1, 0]	61,588	0.00013	[1, 1, 0, 0, 0, 1, 1, 0]	36,292	0.00007
[0, 1, 0, 0, 0, 1, 1, 1]	133,859	0.00027	[1, 1, 0, 0, 0, 1, 1, 1]	60,848	0.00012
[0, 1, 0, 0, 1, 0, 0, 0]	425,154	0.00086	[1, 1, 0, 0, 1, 0, 0, 0]	156,747	0.00032
[0, 1, 0, 0, 1, 0, 0, 1]	112,041	0.00023	[1, 1, 0, 0, 1, 0, 0, 1]	36,216	0.00007
[0, 1, 0, 0, 1, 0, 1, 0]	233,155	0.00047	[1, 1, 0, 0, 1, 0, 1, 0]	196,911	0.00040
[0, 1, 0, 0, 1, 0, 1, 1]	81,906	0.00017	[1, 1, 0, 0, 1, 0, 1, 1]	61,451	0.00013
[0, 1, 0, 0, 1, 1, 0, 0]	75,695	0.00015	[1, 1, 0, 0, 1, 1, 0, 0]	22,643	0.00005
[0, 1, 0, 0, 1, 1, 0, 1]	114,457	0.00023	[1, 1, 0, 0, 1, 1, 0, 1]	28,090	0.00006
[0, 1, 0, 0, 1, 1, 1, 0]	43,172	0.00009	[1, 1, 0, 0, 1, 1, 1, 0]	27,892	0.00006
[0, 1, 0, 0, 1, 1, 1, 1]	96,608	0.00020	[1, 1, 0, 0, 1, 1, 1, 1]	51,863	0.00011
[0, 1, 0, 1, 0, 0, 0, 0]	5,357,109	0.01090	[1, 1, 0, 1, 0, 0, 0, 0]	736,347	0.00150
[0, 1, 0, 1, 0, 0, 0, 1]	1,897,437	0.00386	[1, 1, 0, 1, 0, 0, 0, 1]	247,622	0.00050
[0, 1, 0, 1, 0, 0, 1, 0]	168,450	0.00034	[1, 1, 0, 1, 0, 0, 1, 0]	113,743	0.00023
[0, 1, 0, 1, 0, 0, 1, 1]	86,691	0.00018	[1, 1, 0, 1, 0, 0, 1, 1]	55,728	0.00011
[0, 1, 0, 1, 0, 1, 0, 0]	3,782,721	0.00770	[1, 1, 0, 1, 0, 1, 0, 0]	399,293	0.00081
[0, 1, 0, 1, 0, 1, 0, 1]	25,384,806	0.05165	[1, 1, 0, 1, 0, 1, 0, 1]	1,931,294	0.00393
[0, 1, 0, 1, 0, 1, 1, 0]	151,732	0.00031	[1, 1, 0, 1, 0, 1, 1, 0]	72,079	0.00015
[0, 1, 0, 1, 0, 1, 1, 1]	1,418,875	0.00289	[1, 1, 0, 1, 0, 1, 1, 1]	460,687	0.00094
[0, 1, 0, 1, 1, 0, 0, 0]	235,228	0.00048	[1, 1, 0, 1, 1, 0, 0, 0]	68,648	0.00014
[0, 1, 0, 1, 1, 0, 0, 1]	92,351	0.00019	[1, 1, 0, 1, 1, 0, 0, 1]	25,366	0.00005
[0, 1, 0, 1, 1, 0, 1, 0]	102,671	0.00021	[1, 1, 0, 1, 1, 0, 1, 0]	67,715	0.00014
[0, 1, 0, 1, 1, 0, 1, 1]	51,490	0.00011	[1, 1, 0, 1, 1, 0, 1, 1]	33,684	0.00007
[0, 1, 0, 1, 1, 1, 0, 0]	239,042	0.00049	[1, 1, 0, 1, 1, 1, 0, 0]	53,849	0.00011
[0, 1, 0, 1, 1, 1, 0, 1]	1,699,280	0.00346	[1, 1, 0, 1, 1, 1, 0, 1]	284,509	0.00058
[0, 1, 0, 1, 1, 1, 1, 0]	107,626	0.00022	[1, 1, 0, 1, 1, 1, 1, 0]	54,865	0.00011
[0, 1, 0, 1, 1, 1, 1, 1]	1,071,170	0.00218	[1, 1, 0, 1, 1, 1, 1, 1]	405,608	0.00082
[0, 1, 1, 0, 0, 0, 0, 0]	805,300	0.00164	[1, 1, 1, 0, 0, 0, 0, 0]	735,791	0.00150
[0, 1, 1, 0, 0, 0, 0, 1]	137,191	0.00028	[1, 1, 1, 0, 0, 0, 0, 1]	114,502	0.00023
[0, 1, 1, 0, 0, 0, 1, 0]	98,351	0.00020	[1, 1, 1, 0, 0, 0, 1, 0]	245,772	0.00050
[0, 1, 1, 0, 0, 0, 1, 1]	27,141	0.00006	[1, 1, 1, 0, 0, 0, 1, 1]	55,104	0.00011
[0, 1, 1, 0, 0, 1, 0, 0]	89,126	0.00018	[1, 1, 1, 0, 0, 1, 0, 0]	68,869	0.00014
[0, 1, 1, 0, 0, 1, 0, 1]	100,748	0.00020	[1, 1, 1, 0, 0, 1, 0, 1]	67,376	0.00014
[0, 1, 1, 0, 0, 1, 1, 0]	14,038	0.00003	[1, 1, 1, 0, 0, 1, 1, 0]	25,216	0.00005
[0, 1, 1, 0, 0, 1, 1, 1]	22,889	0.00005	[1, 1, 1, 0, 0, 1, 1, 1]	33,126	0.00007
[0, 1, 1, 0, 1, 0, 0, 0]	359,617	0.00073	[1, 1, 1, 0, 1, 0, 0, 0]	396,169	0.00081
[0, 1, 1, 0, 1, 0, 0, 1]	72,902	0.00015	[1, 1, 1, 0, 1, 0, 0, 1]	71,954	0.00015
[0, 1, 1, 0, 1, 0, 1, 0]	383,809	0.00078	[1, 1, 1, 0, 1, 0, 1, 0]	1,940,450	0.00395
[0, 1, 1, 0, 1, 0, 1, 1]	107,570	0.00022	[1, 1, 1, 0, 1, 0, 1, 1]	463,704	0.00094
[0, 1, 1, 0, 1, 1, 0, 0]	56,599	0.00012	[1, 1, 1, 0, 1, 1, 0, 0]	53,706	0.00011

Continued on next page

Spike Pattern	Count	Ratio	Spike Pattern	Count	Ratio
$[0, 1, 1, 0, 1, 1, 0, 1]$	69,347	0.00014	$[1, 1, 1, 0, 1, 1, 0, 1]$	55,115	0.00011
$[0, 1, 1, 0, 1, 1, 1, 0]$	68,691	0.00014	$[1, 1, 1, 0, 1, 1, 1, 0]$	288,965	0.00059
$[0, 1, 1, 0, 1, 1, 1, 1]$	117,212	0.00024	$[1, 1, 1, 0, 1, 1, 1, 1]$	410,063	0.00083
$[0, 1, 1, 1, 0, 0, 0, 0]$	626,320	0.00127	$[1, 1, 1, 1, 0, 0, 0, 0]$	524,667	0.00107
$[0, 1, 1, 1, 0, 0, 0, 1]$	176,202	0.00036	$[1, 1, 1, 1, 0, 0, 0, 1]$	141,241	0.00029
$[0, 1, 1, 1, 0, 0, 1, 0]$	60,238	0.00012	$[1, 1, 1, 1, 0, 0, 1, 0]$	140,653	0.00029
$[0, 1, 1, 1, 0, 0, 1, 1]$	26,423	0.00005	$[1, 1, 1, 1, 0, 0, 1, 1]$	53,288	0.00011
$[0, 1, 1, 1, 0, 1, 0, 0]$	407,982	0.00083	$[1, 1, 1, 1, 0, 1, 0, 0]$	282,270	0.00057
$[0, 1, 1, 1, 0, 1, 0, 1]$	2,151,062	0.00438	$[1, 1, 1, 1, 0, 1, 0, 1]$	1,174,816	0.00239
$[0, 1, 1, 1, 0, 1, 1, 0]$	51,179	0.00010	$[1, 1, 1, 1, 0, 1, 1, 0]$	83,582	0.00017
$[0, 1, 1, 1, 0, 1, 1, 1]$	411,145	0.00084	$[1, 1, 1, 1, 0, 1, 1, 1]$	456,045	0.00093
$[0, 1, 1, 1, 1, 0, 0, 0]$	270,206	0.00055	$[1, 1, 1, 1, 1, 0, 0, 0]$	282,787	0.00057
$[0, 1, 1, 1, 1, 0, 0, 1]$	87,441	0.00018	$[1, 1, 1, 1, 1, 0, 0, 1]$	83,519	0.00017
$[0, 1, 1, 1, 1, 0, 1, 0]$	231,147	0.00047	$[1, 1, 1, 1, 1, 0, 1, 0]$	1,183,180	0.00241
$[0, 1, 1, 1, 1, 0, 1, 1]$	101,422	0.00021	$[1, 1, 1, 1, 1, 0, 1, 1]$	458,876	0.00093
$[0, 1, 1, 1, 1, 1, 0, 0]$	258,769	0.00053	$[1, 1, 1, 1, 1, 1, 0, 0]$	228,059	0.00046
$[0, 1, 1, 1, 1, 1, 0, 1]$	1,586,548	0.00323	$[1, 1, 1, 1, 1, 1, 0, 1]$	1,057,587	0.00215
$[0, 1, 1, 1, 1, 1, 1, 0]$	252,294	0.00051	$[1, 1, 1, 1, 1, 1, 1, 0]$	1,065,182	0.00217
$[0, 1, 1, 1, 1, 1, 1, 1]$	2,453,351	0.00499	$[1, 1, 1, 1, 1, 1, 1, 1]$	8,398,365	0.01709
Spike Entropy	4.9319				
Firing Rate	0.2464				

Table 19: Distribution of the spike patterns in the encoding layer of Spikformer with STF on CIFAR-10 at $T = 8$.