FSTA-SNN:Frequency-Based Spatial-Temporal Attention Module for Spiking Neural Networks

Kairong Yu[1*], **Tianqing Zhang**[2*], **Hongwei Wang**[1†], **Qi Xu**[3†]

[1]Zhejiang University-University of Illinois Urbana Champaign Institute, Zhejiang University, Haining, China
[2]The College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[3]School of Computer Science and Technology, Dalian University of Technology, Dalian, China
kairong.22@intl.zju.edu.cn, zhangtianqing@zju.edu.cn, hongweiwang@intl.zju.edu.cn, xuqi@dlut.edu.cn

Abstract

Spiking Neural Networks (SNNs) are emerging as a promising alternative to Artificial Neural Networks (ANNs) due to their inherent energy efficiency. Owing to the inherent sparsity in spike generation within SNNs, the in-depth analysis and optimization of intermediate output spikes are often neglected. This oversight significantly restricts the inherent energy efficiency of SNNs and diminishes their advantages in spatiotemporal feature extraction, resulting in a lack of accuracy and unnecessary energy expenditure. In this work, we analyze the inherent spiking characteristics of SNNs from both temporal and spatial perspectives. In terms of spatial analysis, we find that shallow layers tend to focus on learning vertical variations, while deeper layers gradually learn horizontal variations of features. Regarding temporal analysis, we observe that there is not a significant difference in feature learning across different time steps. This suggests that increasing the time steps has limited effect on feature learning. Based on the insights derived from these analyses, we propose a **F**requency-based **S**patial-**T**emporal **A**ttention (FSTA) module to enhance feature learning in SNNs. This module aims to improve the feature learning capabilities by suppressing redundant spike features. The experimental results indicate that the introduction of the FSTA module significantly reduces the spike firing rate of SNNs, demonstrating superior performance compared to state-of-the-art baselines across multiple datasets.

Code — https://github.com/yukairong/FSTA-SNN

Introduction

Deep learning has achieved significant advancements in tasks like classification (Krizhevsky, Sutskever, and Hinton 2012), object segmentation (Ronneberger, Fischer, and Brox 2015), and natural language processing (Hinton et al. 2012), leading to state-of-the-art performance. However, the high energy consumption of these architectures challenges hardware implementation. Spiking Neural Networks (SNNs) offer a promising alternative, inspired by the mammalian brain's learning mechanisms. SNNs represent a shift in information encoding and feature transmission by simulating the spatio-temporal dynamics of biological neurons (Roy, Jaiswal, and Panda 2019; Schuman et al. 2022). In SNNs, neurons theoretically fire only when the membrane potential surpasses a threshold, enabling event-driven operations on neuromorphic chips (Pei et al. 2019; Ma et al. 2017). This spike-based binary transmission allows SNNs to perform low-cost synaptic accumulations and avoid computations involving zero inputs or activations (Eshraghian et al. 2023; Deng et al. 2020).

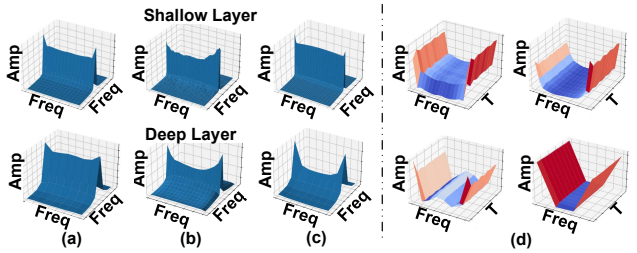


Figure 1: Comparison of the output spike frequency spectrum distribution of SNNs across different model structures, datasets, layer depths, and temporal perspectives. (a) The spectral distribution of spike outputs at different layers of the ResNet architecture at the same time step in static dataset. (b) The spectral distribution of spike outputs at different layers of the ResNet architecture at the same time step in dynamic dataset. (c) The spectral distribution of spike outputs at different layers of the VGG architecture at the same time step in static dataset. (d) The temporal variation of the frequency bands where spectral energy is most concentrated across different layers. Here, Amp, Freq, and T represent amplitude, frequency range and timesteps.

The common belief that computations in SNNs, which are pulse-based neuromorphic computing structures, are inherently sparse is disputable. Due to the lack of reliable mathematical theoretical tools, explorations and analyses in this area are limited. Current investigations predominantly rely on lightweight, spike counting or attention-based strategies. For example, proposed algorithms incorporate penalty functions to exploit spike-aware sparse regularization and compression (Deng et al. 2021; Yin, Corradi, and Bohté 2021), employ techniques like pruning and distillation to re-

*These authors contributed equally.
†Corresponding author.

duce spike occurrences (Shen et al. 2023; Xu et al. 2023b,a) or integrate attention mechanisms like temporal, spatial, or channel attention to suppress redundant spikes (Yao et al. 2023a,b). Despite the potential of these approaches to decrease spikes, they are associated with drawbacks such as decreased accuracy, limited enhancements, or the introduction of additional complexity, without conducting a comprehensive network-level assessment of learning preferences.

In this work, we analyze the learning preferences of spike discharges in SNNs from a frequency perspective, providing a novel understanding of spike redundancy. As depicted in Fig. 1, to validate the universality of learning in SNNs, we conduct Fourier transforms on spike outputs from intermediate layers of varying depths across different datasets and model architectures. The spectral analysis reveal consistent frequency distributions in both ResNet and VGG structures across static and dynamic datasets. Over time, the frequency distributions across the same layer exhibits significant overlap with only minor variations in amplitude response, suggesting potential parameter sharing to enhance efficiency spatially. Observing network depth, shallow layers are concentrated along a central axis, while deeper layers gradually spread along a vertical axis. This phenomenon suggests that in shallow layers, SNNs preferentially discharge spikes for vertical disparity features, gradually shifting towards horizontal disparity features in deeper layers.

Based on these findings, we propose a novel frequency-based spatial-temporal attention (FSTA) module. This module enhances the learning capabilities of existing networks by amplifying previously overlooked spike features while preserving the network's inherent fitting preferences. Additionally, it effectively suppresses redundant spike features and irrelevant noise. Experimental results show that our module significantly improves SNN performance, reduces spike firing rate, and maintains computational efficiency without increasing energy consumption.

Overall, our main contributions are threefold:

- We present a comprehensive study of SNN learning preferences, introducing a novel frequency-based spike analysis. This framework is essential for optimizing sparsification and energy efficiency and provides a theoretical foundation for enhancing SNN performance.
- We propose a Frequency-Based Spatial-Temporal Attention (FSTA) module, a plug-and-play component that introduces a minimal number of additional parameters. This module effectively reduces spike firing rate while boosting performance.
- We evaluate our method on static datasets CIFAR10, CIFAR100, ImageNet, and dynamic dataset CIFAR10-DVS using widely adopted architectures. The results show that networks integrating our proposed module achieve state-of-the-art performance, with a total spike firing rate reduction of approximately 33.99%.

Related Works

Frequency Domain Learning in ANNs The effectiveness of frequency applications in deep learning tasks is well-established, as real-world datasets often contain rich frequency information. In the ANN domain, approaches that integrate frequency into network architectures can be broadly categorized into two types. One approach involves transforming the model from the spatial domain to the frequency domain for direct learning (Huang et al. 2023; Patro, Namboodiri, and Agneeswaran 2023; Guo et al. 2023a; Kong et al. 2023). These methods typically start with applying Fourier (Brigham 1988) or Wavelet Transforms (Burrus, Gopinath, and Guo 1998) to the original feature maps, followed by linear mappings to convert frequency information, and end with an inverse transform to restore the features. The other approach leverages frequency information for local feature extraction within the network. For example, in (Qin et al. 2021), the introduction of Discrete Cosine Transform (DCT) (Ahmed, Natarajan, and Rao 1974) into the channel attention mechanism SENet (Hu, Shen, and Sun 2018) enhances the channel's ability to perceive more comprehensive frequency information.

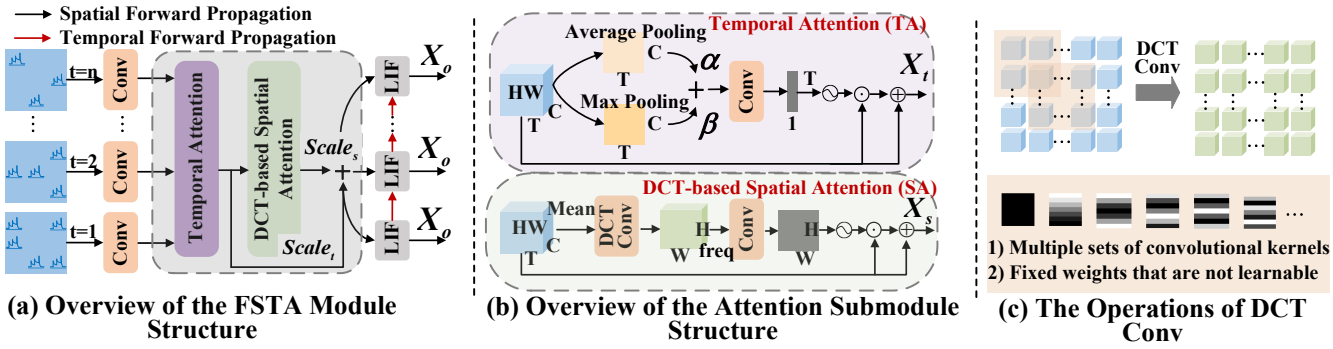
Frequency Applications in SNNs The unique advantages of frequency have led to various application methods in SNNs. Current approaches include using frequency for temporal encoding, such as employing DCT to minimize the number of time steps required for inference (Garg, Chowdhury, and Roy 2021) and introducing spike frequency-adaptive neurons to dynamically adjust neuronal firing thresholds (Chen et al. 2023; Falez et al. 2018). In contrast to these previous methods, our focus is on leveraging frequency-based activation of network output spikes. Furthermore, we propose the integration of a frequency-based attention module between layers to minimize redundancy and reduce the spike firing rate.

Attention in SNNs Attention mechanisms in deep learning have achieved significant success, primarily motivated by the human ability to efficiently focus on salient information within complex scenarios. A common approach is to incorporate attention as an auxiliary module to enhance the representational capacity of ANNs (Li et al. 2022; Guo et al. 2022a). This work (Yao et al. 2021) involves using temporal attention to evaluate the importance of different time steps while bypassing non-essential ones. Furthermore, current research is increasingly exploring multi-dimensional attention modules that encompass time, space, and channels (Zhu et al. 2024; Yao et al. 2023b; Xu et al. 2024). Although these methods effectively reduce spike firing and improve accuracy, they also introduce substantial additional computational burdens to SNNs.

SNN Frequency Analysis

SNN Fundamentals The fundamental computational unit of SNNs is the spiking neuron, an abstract representation of biological neuronal dynamics. The Leaky Integrate-and-Fire (LIF) model is among the most widely used spiking neuron models, as it effectively balances simplified mathematical representations with the complex dynamics of biological neurons. Mathematically, an LIF neuron is described by the following equation:

$$\tau \frac{du(t)}{dt} = -(u(t) - u_{\text{reset}}) + I(t) \tag{1}$$



(a) Overview of the FSTA Module Structure (b) Overview of the Attention Submodule Structure (c) The Operations of DCT Conv

Figure 2: Overview of the FSTA module and its internal submodules structure

where τ denotes the membrane potential time constant, $u(t)$ represents the membrane potential at time step t, u_{reset} is the neuron's resting potential, and $I(t)$ is the synaptic input at time step t. According to Eq.1, the discrete-time and iterative mathematical representation of LIF-SNNs can be described as follows:

$$V^{t,n} = H^{t-1,n} + \frac{1}{\tau}[I^{t-1,n} - (H^{t-1,n} - V_{reset})] \quad (2)$$

$$S^{t,n} = \Theta(V^{t,n} - v_{th}) \quad (3)$$

$$H^{t,n} = V_{reset} \cdot S^{t,n} + V^{t,n} \odot (1 - S^{t,n}). \quad (4)$$

The Heaviside step function Θ is defined as:

$$\Theta(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0. \end{cases} \quad (5)$$

Among these, $H^{t-1,n}$ denotes the membrane potential after a spike from the previous time step. $I^{t,n}$ and $V^{t,n}$ represent the input and updated membrane potential at time step t for n-th layer, respectively. Additionally, v_{th} is the threshold that determines whether $V^{t,n}$ results in a spike or remains silent, and $S^{t,n}$ indicates the spike sequence at time step t for n-th layer.

Frequency Fundamentals Frequency analysis methods are widely used for feature analysis, with the Discrete Fourier Transform (DFT) being one of the most popular algorithms. The DFT is pivotal in digital signal processing and serves as an essential analytical tool for the preference learning of SNNs discussed in this paper. For clarity, we first consider the one-dimensional DFT. Given a sequence of N complex numbers $x[n]$, where $n = 0, 1, \ldots, N-1$, the one-dimensional DFT converts this sequence into the frequency domain as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn} \quad (6)$$

where j is the imaginary unit. Eq. 6 can be derived by performing the Fourier transform of a discrete signal through sampling in both the time domain and the frequency domain. Since $X[k]$ repeats with a period of length N, it is sufficient to evaluate $X[k]$ at N discrete points at $k = 0, 1, ..., N-1$. Specifically, $X[k]$ represents the frequency spectrum of the sequence $x[n]$ at the frequency $\omega_k = \frac{2\pi k}{N}$.

It is also noteworthy that the DFT is a one-to-one transformation. Given $X[k]$, the original signal $x[n]$ can be reconstructed using the Inverse Discrete Fourier Transform (IDFT).

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j(2\pi/N)kn} \quad (7)$$

The DFT can similarly be extended to 2D signals. Given a 2D signal $X[m, n]$, where $0 \leq m \leq M-1$ and $0 \leq n \leq N-1$, the 2D DFT of $x[m, n]$ is given by the following formula:

$$X[u, v] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m, n] e^{-j2\pi(\frac{um}{M} + \frac{vn}{N})} \quad (8)$$

The 2D DFT can be considered as the sequential application of 1D DFTs in two dimensions.

Frequency Analysis Building on the foundational knowledge outlined above, we analyze the SNN. To accurately represent the network's learning preferences, we perform a statistical analysis of spike outputs from various intermediate layers across different datasets. We use the DFT for frequency domain conversion and center the results, as shown in Fig. 1. Based on these frequency spectra, we investigate the frequency characteristics of the SNNs across spatial, temporal, and other dimensions.

Observation 1. The structural differences and dataset variations do not affect the learning preferences of SNNs As shown in Fig. 1(a), (b), and (c), the frequency distributions of spike average release probabilities are remarkably similar across different architectures and datasets. In shallow layers, the spectral bands of SNNs are highly concentrated along the central horizontal axis, capturing nearly all spectral energy. In deeper layers, network frequency gradually shifts toward the vertical axis, although the central horizontal axis remains significant. According to Eq.7, the spectrum suggests that SNNs initially focus on longitudinal differential features in shallow layers and progressively shift to capturing lateral differential features in deeper layers.

Observation 2: Increasing the time step has limited potential to enhance performance From Fig. 1(d), it is evident that the frequency distribution remains consistent across different time steps, with only minor variations in amplitude. This consistency becomes more pronounced in deeper layers, indicating that features learned within the same layer stabilize over time and suggesting limited acquisition of new feature information beyond a certain time step. Moreover, this observation suggests that spatial enhancement module across different time steps could share parameters, thereby reducing redundancy.

Methodology

Based on the previous analysis and conclusions, we majorly focus on four issues as follows: (1) Sharing enhancement modules across identical frequency distributions at different time steps within the same layer. (2) Reducing redundancy in feature learning within shallow networks. (3) Expanding feature learning capacity in deep networks. (4) Adjusting amplitude variations across time steps using temporal attention. To address these issues, we employ a Frequency-based Spatial-Temporal Attention (FSTA) module.

DCT-based Spatial Attention Submodule

We begin with a detailed explanation of the Discrete Cosine Transform (DCT) (Ahmed, Natarajan, and Rao 1974). The basis function of the two-dimensional (2D) DCT is defined as:

$$B_{u,v}^{i,j} = \cos\left(\frac{\pi u}{H}(i+\frac{1}{2})\right)\cos\left(\frac{\pi v}{W}(j+\frac{1}{2})\right) \tag{9}$$

The 2D DCT can then be expressed as:

$$f_{u,v} = \sum_{i=0}^{H-1}\sum_{j=0}^{W-1} x_{i,j} B_{u,v}^{i,j} \tag{10}$$

where $u \in \{0,1,...,H-1\}, v \in \{0,1,...,W-1\}$. Here, $f \in \mathbb{R}^{H,W}$ represents the 2D DCT frequency spectrum, while $x \in \mathbb{R}^{H,W}$ denotes the input, with H and W being the height and width of x, respectively.

In typical attention modules, global average pooling (GAP) is widely used for its computational simplicity and effective compression. Other methods such as global max pooling and global standard pooling are also common. However, as shown in Eq. 10, the addition of x and B closely resembles convolution operations. This establishes a link between traditional feature extraction methods and frequency domain analysis. We will demonstrate that the compression method used in GAP is, in fact, a special case of 2D DCT, with results proportional to the lowest frequency component of the 2D DCT.

Proof. Assume u and v in Eq. 10 are both set to 0. Then:

$$\begin{aligned} f_{0,0} &= \sum_{i=0}^{H-1}\sum_{j=0}^{W-1} x_{i,j}\cos\left(\frac{0}{H}(i+\frac{1}{2})\right)\cos\left(\frac{0}{W}(j+\frac{1}{2})\right) \\ &= \sum_{i=0}^{H-1}\sum_{j=0}^{W-1} x_{i,j} = GAP(x)HW \end{aligned} \tag{11}$$

Since H and W are constants, linear operations in neural network can be transformed through mapping layers without altering the feature distribution. Consequently, from Eq. 11, it is evident that GAP corresponds to the lowest frequency component of the 2D DCT. This suggests that the spatial attention features previously used are confined to specific frequency bands and fail to capture comprehensive information. Based on these insights, we propose a DCT-based full-spectrum spatial attention module that better aligns with learning preferences of SNNs.

As illustrated in Fig. 2(b), the first step involves averaging the input X along the temporal dimension. This is because the frequency feature distributions in Obs. 2 are highly similar across different time steps. Consequently, the shape of X changes from $\mathbb{R}^{T,C,H,W}$ to $\mathbb{R}^{C,H,W}$. Next, the feature map X_{mean} is analyzed using the full-band frequency basis of the DCT to extract the complete frequency feature $Freq$, where $Freq \in \mathbb{R}^{freq,H,W}$ and $freq$ represents the number of frequency components. This entire extract process can be achieved through concatenation:

$$X_{mean} = mean(X) \tag{12}$$

$$Freq = Conv_{dct}(X_{mean}) \tag{13}$$

Here, $Conv_{dct}$ represents a non-trainable kernel that performs a convolution operation using fixed weights, as derived from Eq. 9. In the experimental section, we evaluate how different kernel size, which correspond to various frequency extraction ranges, affect overall network performance.

After extracting the frequency features $Freq$, the linear layer compresses these features and applies the Sigmoid function to obtain the spatial attention weight matrix $freq_w \in \mathbb{R}^{H,W}$, which encodes where to emphasize or suppress. Finally, $freq_w$ is dot-multiplied with input X and added, thereby enhancing the feature matrix. The complete process of compression and enhancement is as follows:

$$freq_w = Sigmoid(Linear(Freq)) \tag{14}$$

$$X_s = X + X \cdot freq_w \tag{15}$$

To further reduce computational complexity and avoid complex floating-point operations, the weights of $Conv_{dct}$ are precomputed and stored as fixed constants, eliminating the need for repetitive cosine calculations. This module design adds only a minimal number of parameters while avoiding excessive computational overhead. Additionally, it broadens the frequency range of the extracted information, reducing spike redundancy and enhancing feature representation.

Temporal Attention Submodule for Amplitude Regulation

In Obs. 2, it is noted that SNN layers do not exhibit frequency range fitting biases temporally, but rather amplitude differences. Therefore, attention must be focused on learning spike features over time. As depicted in Fig. 2(b), for the input spike feature map X, temporal channel features are initially aggregated using average and max pooling operations. To effectively integrate these temporal features, we

Dataset	Method	Type	Architecture	Timestep	Accuracy
CIFAR-10	SpikeNorm (Sengupta et al. 2019)	ANN2SNN	VGG16	2500	91.55%
	Hybrid-Train (Rathi et al. 2020)	Hybrid training	VGG16	200	92.02%
	PTL (Wu et al. 2021)	Tandem learning	VGG11	16	91.24%
	DSR (Meng et al. 2022)	SNN training	ResNet18	20	95.40%
	KDSNN (Xu et al. 2023b)	SNN training	ResNet18	4	93.41%
	Joint A-SNN (Guo et al. 2023b)	SNN training	ResNet18	4	95.45%
	RMP-Loss (Guo et al. 2023b)	SNN training	ResNet19	2	95.31%
				4	95.51%
			ResNet20	4	91.89%
	RecDis-SNN (Guo et al. 2022c)	SNN training	ResNet19	2	93.64%
				4	95.53%
	TET (Deng et al. 2022)	SNN training	ResNet19	2	94.16%
				4	94.44%
	Real Spike (Guo et al. 2022d)	SNN training	ResNet19	2	95.31%
				4	95.51%
				6	96.10%
	MPBN (Guo et al. 2023c)	SNN training	ResNet19	1	96.06%
				2	96.47%
			ResNet20	1	92.22%
				2	93.54%
				4	94.28%
	Ours	**SNN training**	**ResNet19**	**1**	**96.21%** $\pm$ 0.10%
				2	**96.52%** $\pm$ 0.09%
			ResNet20	**1**	**93.01%** $\pm$ 0.12%
				2	**94.18%** $\pm$ 0.10%
				4	**94.72%** $\pm$ 0.09%
CIFAR-100	RMP (Han, Srinivasan, and Roy 2020)	ANN2SNN	ResNet20	2048	67.82%
	LTL (Yang et al. 2022)	Tandem learning	ResNet20	31	76.08%
	Real Spike (Guo et al. 2022d)	SNN training	ResNet20	5	70.62%
	Dspike (Li et al. 2021)	SNN training	ResNet20	2	71.68%
				4	73.35%
	TET (Deng et al. 2022)	SNN training	ResNet19	2	72.87%
				4	74.47%
	GLIF (Yao et al. 2022)	SNN training	ResNet19	2	75.48%
				4	77.05%
	TEBN (Duan et al. 2022)	SNN training	ResNet19	2	75.86%
				4	76.13%
				6	76.41%
	MPBN (Guo et al. 2023c)	SNN training	ResNet19	1	78.71%
				2	79.51%
			ResNet20	2	70.79%
				4	72.30%
	Ours	**SNN training**	**ResNet19**	**1**	**78.87%** $\pm$ 0.11%
				2	**80.42%** $\pm$ 0.09%
			ResNet20	**1**	**69.64%** $\pm$ 0.10%
				2	**72.15%** $\pm$ 0.12%
				4	**73.44%** $\pm$ 0.09%

Table 1: Comparison with SOTA methods on CIFAR-10/100

introduce two learnable parameters α and β, which balance global (max pooling) and local (average pooling) information. Specifically, $f_{avg}, f_{max} \in \mathbb{R}^{T,1,1}$. Feature extraction M is performed using the following equation:

$$M = \alpha \cdot f_{avg}(X) + \beta \cdot f_{max}(X) \tag{16}$$

where $M \in \mathbb{R}^{T,C}$.

After extraction, the mean is computed across the temporal dimension of M. Subsequently, linear layers followed by a Sigmoid function are used to obtain weights $T_w \in \mathbb{R}^T$ for different time steps. Finally, as with the spatial module, temporal enhancement is applied to the input.

$$M_{mean} = mean(M) \tag{17}$$

$$T_w = Sigmoid(Linear(M_{mean})) \tag{18}$$

$$X_t = X + X \cdot T_w \tag{19}$$

Frequency-based Spatial-Temporal Attention Module

After detailing the temporal attention (TA) submodule and the DCT-based spatial attention (SA) submodule, we proceed to integrate the entire module. Our experiments (see

Method	Architecture	T	Accuracy
STBP-tdBN (2021)	ResNet34	6	63.72%
TET (2022)	ResNet34	6	64.79%
RecDis-SNN (2022c)	ResNet34	6	67.33%
GLIF (2022)	ResNet34	4	67.52%
IM-Loss (2022b)	ResNet18	6	67.43%
Real Spike (2022d)	ResNet18	4	63.68%
	ResNet34	4	67.69%
RMP-Loss (2023b)	ResNet18	4	63.03%
	ResNet34	4	65.17%
MPBN (2023c)	ResNet18	4	63.14%
	ResNet34	4	64.71%
SEW ResNet (2021)	ResNet18	4	63.18%
	ResNet34	4	67.04%
Ours	**ResNet18**	**4**	**68.21%** $\pm 0.20\%$
	ResNet34	**4**	**70.23%** $\pm 0.12\%$

Table 2: Comparison with SNN training based SOTA methods on ImageNet

Method	Architecture	T	Accuracy
DSR (2022)	VGG11	20	77.27%
GLIF (2022)	7B-wideNet	16	78.10%
STBP-tdBN (2021)	ResNet19	10	67.80%
RecDis-SNN (2022c)	ResNet19	10	72.42%
Real Spike (2022d)	ResNet19	10	72.85%
TET (2022)	VGGSNN	10	77.30%
MPBN (2023c)	ResNet19	10	74.40%
	ResNet20	10	78.70%
Ours	**ResNet20**	**10**	**81.50%** $\pm 0.25\%$
		16	**82.70%** $\pm 0.10\%$

Table 3: Comparison with SNN training based SOTA methods on CIFAR10-DVS

the experimental section for details) show that concatenating the TA and SA achieves the optimal combination, balancing computational complexity and performance effectively. For the intermediate output X in the SNN, we first apply TA to enhance temporal dynamics, followed by spatial frequency enhancement using SA. To address significant loss of intermediate feature information, we introduce essential temporal and spatial scaling factors $Scale_t$ and $Scale_s$ to control this process and obtain the fused output. The complete computational process is outlined as follows:

$$X_t = TA(X) \tag{20}$$

$$X_s = SA(X_t) \tag{21}$$

$$X_o = Scale_t \cdot X_t + Scale_s \cdot X_s \tag{22}$$

in which, X_t, X_s, X_o maintain the same shape as X.

Experiments

We perform comprehensive experiments to assess the proposed method and compare it with other recent SOTA methods on several widely used architectures. These experiments utilize static datasets CIFAR-10, CIFAR-100 (Krizhevsky, Nair, and Hinton 2010), and ImageNet (Deng et al. 2009), as well as the dynamic dataset CIFAR10-DVS (Li et al. 2017) dataset.

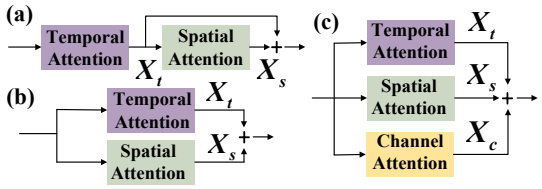


Figure 3: The different combinations of modules

Mode	ACs	MACs	Accuracy
a	109.91M	38.93M	72.15%
b	107.31M	38.93M	70.83%
c	89.17M	41.74M	72.68%

Table 4: Comparative analysis of the computational performance of different combination methods

Ablation Study

Combination Mode of Submodules To enhance module integration performance with the original SNN, we examine various submodule combination strategies. In these experiments, we use ResNet20 as the baseline network and evaluate its performance on the CIFAR-100 dataset with a time step of 2. As illustrated in Fig. 3, we explore both parallel and serial combinations, and include a channel attention enhancement submodule for comparison. Results in Tab. 4 show that the serial combination mode slightly improves accuracy by 1.32% over the parallel mode. Although channel attention enhances performance, it increases the number of MAC operations and parameters, thereby raising energy consumption. Ultimately, the serial combination of spatiotemporal attention provides the best balance between performance and energy efficiency.

Selection of Frequency Range Eq. 10 demonstrates that frequency feature extraction can be performed using convolutional operations. The size of the convolutional kernel directly influences the frequency range of u and v. To enhance module versatility and reduce manual computation, $Conv_{dct}$ uses a block-wise approach with fixed kernel sizes for feature extraction. Thus, exploring various kernel shapes, corresponding to different frequency ranges, is justified for improving network performance. In Tab. 6, we compare different kernel sizes and strides to maintain input-output consistency. With experimental conditions kept identical except for kernel sizes of $3\times3, 5\times5$, and 7×7, the performance results are $92.36\%, 92.58\%$, and 93.01%, respectively. These results suggest that finer frequency ranges lead to improved performance within a certain range.

Comparion with SOTA methods

In this section, we compare our approach with previous SOTA methods. We report the average top-1 accuracy from these experiments. Evaluations are first conducted on the static datasets CIFAR-10 and CIFAR-100, with results shown in Tab. 1. On CIFAR-10, our method improves upon the previous best results by 0.44% and 0.05% using

Dataset	Architecture	Resolution	Timestep	ACs	MACs	FLOPs	Param(M)	Energy(mJ)
CIFAR100	ResNet20	32x32	4	260.05M	67.26M	880.36M	11.3	0.54
CIFAR10-DVS	ResNet20	128x128	10	2.14G	582.87M	8.71G	11.2	4.60
ImageNet	SEW ResNet18	224x224	4	2.45G	1.05G	7.37G	11.7	7.03

Table 5: Energy costs and model structure across different datasets

Kernel Size	Frequency Range	Padding	Accuracy
3x3	9	1	92.36%
5x5	25	2	92.58%
7x7	49	3	93.01%

Table 6: Performance comparison of different frequency ranges using ResNet20 structure on CIFAR-10

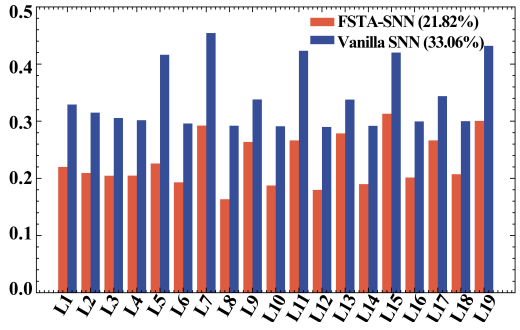


Figure 4: Comparison of spike firing rates at various layers between FSTA-SNN and vanilla SNN. The legend represents the average firing rate of the entire network.

ResNet20 and ResNet19, achieving 94.72% and 96.52%, respectively. This enhancement is observed even under different timestep conditions. On CIFAR-100, our method with ResNet19 and ResNet20 at time steps 2 and 4 yields 80.42% and 73.44%, respectively, surpassing previous best methods. Results in Tab. 1 clearly demonstrate the superior and efficient performance of our approach. Additionally, we test our method on the more complex ImageNet dataset, with comparative results shown in Tab. 2. Our approach achieves 68.21% and 70.23% using the same models, reflecting significant performance improvements over existing methods. Finally, on the dynamic dataset CIFAR10-DVS, evaluations using ResNet20 achieve 81.50% and 82.70% at time steps 10 and 16, respectively. This represents a substantial improvement over previous approaches.

Energy Estimation

In this section, we evaluate the energy performance of the proposed method. Fig. 4 presents a statistical analysis of the spike firing rate in the vanilla SNN compared to our proposed method across various layers. The results show a substantial reduction in spike firing rate at each layer, with an overall decrease of 33.99% across the network. This reduction indicates that our approach effectively eliminates redundant spikes, thereby enhancing the feature representation capability of network. In Tab. 5, we provide detailed data on

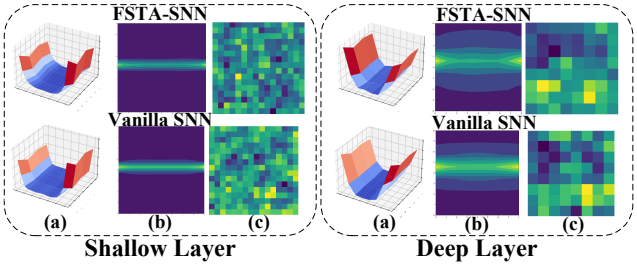


Figure 5: Visualization of the FSTA-SNN and vanilla SNN analysis. (a) Distribution of energy accumulation across frequency bands in the time dimension; (b) Contour plots of the averaged temporal frequency spectrum energy; (c) Grad-CAM visualization for the same sample.

energy consumption for the model and dataset. The table indicates that our model introduces only a minimal number of additional parameters while effectively reducing MAC operations and increasing AC computations. Notably, this performance improvement is achieved without a significant increase in energy consumption.

Result Analysis

We also analyze the spike spectral distribution of the model using DFT with the proposed method. As shown in Fig. 5, the frequency distribution remains largely consistent with vanilla SNN after incorporating the FSTA module. However, the magnitudes have significantly decreased. This reduction indicates that the proposed method effectively suppresses unnecessary spikes, consistent with the reduced spike firing rate reported in Fig. 4. Fig. 5(b) further illustrates that the introduction of the module narrows the region of shallow spectral energy concentration, reducing redundant spike generation. In contrast, deeper layers expand the frequency distribution, revealing new spike features that the vanilla SNN could not capture. The spectral diagrams correspond well with the heat map shown in Fig. 5(c).

Conclusion

In this paper, we analyze the spike output of SNN from a novel perspective of frequency, revealing the learning preferences of the network. Based on these findings, we propose a plug-and-play Frequency-Based Spatial-Temporal Attention (FSTA) module. This module enhances the inherent characteristics of SNNs to improve feature learning capability and reduce redundant spikes. Experimental results demonstrate that our method significantly enhances model performance across multiple datasets while maintaining low energy consumption.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62476035, and 62206037.

References

Ahmed, N.; Natarajan, T.; and Rao, K. R. 1974. Discrete cosine transform. *IEEE transactions on Computers*, 100(1): 90–93.

Brigham, E. O. 1988. *The fast Fourier transform and its applications*. Prentice-Hall, Inc.

Burrus, C. S.; Gopinath, R. A.; and Guo, H. 1998. Wavelets and wavelet transforms. *rice university, houston edition*, 98.

Chen, T.; Wang, L.; Li, J.; Duan, S.; and Huang, T. 2023. Improving spiking neural network with frequency adaptation for image classification. *IEEE Transactions on Cognitive and Developmental Systems*.

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. ISSN: 1063-6919.

Deng, L.; Wu, Y.; Hu, X.; Liang, L.; Ding, Y.; Li, G.; Zhao, G.; Li, P.; and Xie, Y. 2020. Rethinking the performance comparison between SNNS and ANNS. *Neural networks*, 121: 294–307.

Deng, L.; Wu, Y.; Hu, Y.; Liang, L.; Li, G.; Hu, X.; Ding, Y.; Li, P.; and Xie, Y. 2021. Comprehensive SNN Compression Using ADMM Optimization and Activity Regularization. *Institute of Electrical and Electronics Engineers (IEEE)*, (99).

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, C.; Ding, J.; Chen, S.; Yu, Z.; and Huang, T. 2022. Temporal effective batch normalization in spiking neural networks. *Advances in Neural Information Processing Systems*, 35: 34377–34390.

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*.

Falez, P.; Tirilly, P.; Bilasco, I. M.; Devienne, P.; and Boulet, P. 2018. Mastering the output frequency in spiking neural networks. In *2018 international joint conference on neural networks (IJCNN)*, 1–8. IEEE.

Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34: 21056–21069.

Garg, I.; Chowdhury, S. S.; and Roy, K. 2021. Dct-snn: Using dct to distribute spatial information over time for low-latency spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4671–4680.

Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.-H.; Martin, R. R.; Cheng, M.-M.; and Hu, S.-M. 2022a. Attention mechanisms in computer vision: A survey. *Computational visual media*, 8(3): 331–368.

Guo, S.; Yong, H.; Zhang, X.; Ma, J.; and Zhang, L. 2023a. Spatial-frequency attention for image denoising. *arXiv preprint arXiv:2302.13598*.

Guo, Y.; Chen, Y.; Zhang, L.; Liu, X.; Wang, Y.; Huang, X.; and Ma, Z. 2022b. IM-loss: information maximization loss for spiking neural networks. *Advances in Neural Information Processing Systems*, 35: 156–166.

Guo, Y.; Peng, W.; Chen, Y.; Zhang, L.; Liu, X.; Huang, X.; and Ma, Z. 2023b. Joint a-snn: Joint training of artificial and spiking neural networks via self-distillation and weight factorization. *Pattern Recognition*, 142: 109639.

Guo, Y.; Tong, X.; Chen, Y.; Zhang, L.; Liu, X.; Ma, Z.; and Huang, X. 2022c. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 326–335.

Guo, Y.; Zhang, L.; Chen, Y.; Tong, X.; Liu, X.; Wang, Y.; Huang, X.; and Ma, Z. 2022d. Real spike: Learning real-valued spikes for spiking neural networks. In *European Conference on Computer Vision*, 52–68. Springer.

Guo, Y.; Zhang, Y.; Chen, Y.; Peng, W.; Liu, X.; Zhang, L.; Huang, X.; and Ma, Z. 2023c. Membrane potential batch normalization for spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19420–19430.

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

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6): 82–97.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huang, Z.; Zhang, Z.; Lan, C.; Zha, Z.-J.; Lu, Y.; and Guo, B. 2023. Adaptive frequency filters as efficient global token mixers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6049–6059.

Kong, L.; Dong, J.; Ge, J.; Li, M.; and Pan, J. 2023. Efficient frequency domain-based transformers for high-quality image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5886–5895.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2010. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 5(4): 1.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Li, G.; Fang, Q.; Zha, L.; Gao, X.; and Zheng, N. 2022. HAM: Hybrid attention module in deep convolutional neural networks for image classification. *Pattern Recognition*, 129: 108785.

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.

Li, Y.; Guo, Y.; Zhang, S.; Deng, S.; Hai, Y.; and Gu, S. 2021. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in Neural Information Processing Systems*, 34: 23426–23439.

Ma, D.; Shen, J.; Gu, Z.; Zhang, M.; Zhu, X.; Xu, X.; Xu, Q.; Shen, Y.; and Pan, G. 2017. Darwin: A neuromorphic hardware co-processor based on spiking neural networks. *Journal of systems architecture*, 77: 43–51.

Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; and Luo, Z.-Q. 2022. Training high-performance low-latency spiking neural networks by differentiation on spike representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12444–12453.

Patro, B. N.; Namboodiri, V. P.; and Agneeswaran, V. S. 2023. SpectFormer: Frequency and Attention is what you need in a Vision Transformer. *arXiv preprint arXiv:2304.06446*.

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

Qin, Z.; Zhang, P.; Wu, F.; and Li, X. 2021. Fcanet: Frequency channel attention networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 783–792.

Rathi, N.; Srinivasan, G.; Panda, P.; and Roy, K. 2020. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. *arXiv preprint arXiv:2005.01807*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer.

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

Schuman, C. D.; Kulkarni, S. R.; Parsa, M.; Mitchell, J. P.; Kay, B.; et al. 2022. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1): 10–19.

Sengupta, A.; Ye, Y.; Wang, R.; Liu, C.; and Roy, K. 2019. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in neuroscience*, 13: 95.

Shen, J.; Xu, Q.; Liu, J. K.; Wang, Y.; Pan, G.; and Tang, H. 2023. Esl-snns: An evolutionary structure learning strategy for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 86–93.

Wu, J.; Xu, C.; Han, X.; Zhou, D.; Zhang, M.; Li, H.; and Tan, K. C. 2021. Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7824–7840.

Xu, Q.; Gao, Y.; Shen, J.; Li, Y.; Ran, X.; Tang, H.; and Pan, G. 2024. Enhancing adaptive history reserving by spiking convolutional block attention module in recurrent neural networks. *Advances in Neural Information Processing Systems*, 36.

Xu, Q.; Li, Y.; Fang, X.; Shen, J.; Liu, J. K.; Tang, H.; and Pan, G. 2023a. Biologically inspired structure learning with reverse knowledge distillation for spiking neural networks. *arXiv preprint arXiv:2304.09500*.

Xu, Q.; Li, Y.; Shen, J.; Liu, J. K.; Tang, H.; and Pan, G. 2023b. Constructing deep spiking neural networks from artificial neural networks with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7886–7895.

Yang, Q.; Wu, J.; Zhang, M.; Chua, Y.; Wang, X.; and Li, H. 2022. Training spiking neural networks with local tandem learning. *Advances in Neural Information Processing Systems*, 35: 12662–12676.

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

Yao, M.; Hu, J.; Zhao, G.; Wang, Y.; Zhang, Z.; Xu, B.; and Li, G. 2023a. Inherent redundancy in spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16924–16934.

Yao, M.; Zhao, G.; Zhang, H.; Hu, Y.; Deng, L.; Tian, Y.; Xu, B.; and Li, G. 2023b. Attention spiking neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 45(8): 9393–9410.

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

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

Zheng, H.; Wu, Y.; Deng, L.; Hu, Y.; and Li, G. 2021. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11062–11070.

Zhu, R.-J.; Zhang, M.; Zhao, Q.; Deng, H.; Duan, Y.; and Deng, L.-J. 2024. Tcja-snn: Temporal-channel joint attention for spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.