Appendix: Spiking Wavelet Transformer

Yuetong Fang[1†], Ziqing Wang[1,2†], Lingfeng Zhang[1], Jiahang Cao[1], Honglei Chen[1], and Renjing Xu[1✉] ⋆

The Hong Kong University of Science and Technology (Guangzhou), China
Northwestern University, IL, USA
renjingxu@hkust-gz.edu.cn

1 Implementation Details

This section covers the experimental setups and background information. Spiking Neural Networks (SNNs) are spatio-temporal dynamic networks that naturally handle temporal tasks. For static image classification, it is common practice to repeatedly input the same image at each timestep. Increasing simulation timesteps can improve accuracy but also increase training time, hardware requirements, and inference energy consumption. On the other hand, neuromorphic datasets inherently built with spatio-temporal dynamics can fully exploit the energy-efficient advantages of SNNs. SNN models are implemented in SpikingJelly [4], trained from scratch, and training parameters are provided Table 1. To evaluate the performance, we used a server equipped with 64 × Intel Xeon 8358P CPUs and 8 × NVIDIA A800 GPUs. The code is available at: https://github.com/bic-L/Spiking-Wavelet-Transformer.

1.1 Static Classification Datasets

This section presents the static image datasets used for training and testing our models, chosen for their diverse representation of real-world visual categories, which is essential for comprehensive performance evaluation.

CIFAR-10 [7] consists of 60,000 32×32 color images in 10 classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 testing images.

CIFAR-100 [6] is similar to CIFAR-10 but has 100 classes with 600 images each, divided into 50,000 training and 10,000 testing images.

ImageNet-1k [3] is a large-scale dataset with 1.28 million training, 50,000 validation, and 100,000 test images across 1000 object classes, serving as a benchmark for advanced image classification models.

Training Details. We employ random cropping with a crop size of 256×256 pixels for ImageNet-1k and 32×32 for CIFAR datasets. The Spiking Wavelet Transformer(SWformer) is trained using the AdamW optimizer [10] with a weight decay of 0.01. For CIFAR10/CIFAR100, we train the model for 310 epochs with a peak learning rate of $1e^{-3}$. In the case of ImageNet-1k, we train for 410 epochs

⋆ † Equal Contribution; ✉ Corresponding Author.

with a peak learning rate of $1e^{-4}$, utilizing a warmup period of 30 epochs and a cosine decay learning rate scheduler. To enhance the model's performance and generalization ability, we employ various data augmentation and regularization techniques on the ImageNet-1k classification task, including MixUp MixUp [21], CutMix [20], CutOut [22], RandAugment [1], Label Smoothing [16]. Fig. 1 shows the test accuracy and loss of SWformer on ImageNet.

1.2 Neuromorphic Classification Datasets

SNNs, a key branch of neuromorphic computing, provide an efficient and biologically-inspired approach that is inherently compatible with emerging event-driven processing tasks, offering the potential for high energy efficiency and low latency in real-world applications. Unlike traditional static images, event-driven sensors capture changes in illumination per pixel, encoding spatial-temporal information. We conduct extensive experiments to demonstrate the capability of SNNs in neuromorphic tasks.

CIFAR10-DVS [8], is an event-based version of the CIFAR-10 dataset, created using a Dynamic Vision Sensor (DVS). It contains 10,000 event-based images (128×128 pixels) across 10 classes, split into 9,000 training and 1,000 test samples. This dataset allows for the investigation of neuromorphic approaches in recognizing static image categories within an event-driven paradigm.

N-Caltech101 [13] is an event-based version of the Caltech101 dataset, featuring 8,831 samples (180×240 pixels) across 101 classes. By converting static images into an event-based format, this dataset challenges models to maintain high classification performance in a more complex, temporally rich input space.

N-Cars [15] is a binary classification dataset consisting of 24,029 event-based images (100×120 pixels) categorized into car and background classes. The dataset is divided into 15,422 training samples (7,940 cars, 7,482 background) and 8,607 test samples (4,396 cars, 4,211 background), making it useful for vehicle detection tasks in dynamic environments within the neuromorphic computing domain.

Action Recognition [12] contains 10 action classes at 346×260 resolution, with 30 recordings per class, including arm-crossing, getting up, and waving.

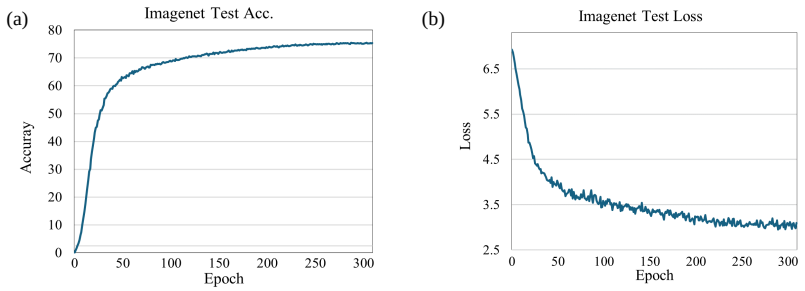

Fig. 1: Test accuracy and loss on ImageNet.

To address the limited size, the Surface of Active Events encoding converts raw data into 4,670 frame images. This dataset enables the study of complex human activities through event-based sensing in neuromorphic vision.

NavGesture [11] contains 1,342 recordings of 6 mid-air hand gestures, totaling about 1 billion events over 47 minutes. The dataset was collected using a wearable device equipped with an event-based dynamic vision sensor and an accelerometer. The gestures include down swipe, up swipe, left swipe, right swipe, select, and home, which are typically used for navigation tasks on smart mobile devices.

Preprocessing Details. Neuromorphic datasets pose unique challenges, such as limited size and overfitting risk. To address these issues, we apply specialized data augmentation techniques tailored for neuromorphic data, as recommended by Li et al. [9]. These methods, including temporal jittering and spatial transformations, increase the diversity and robustness of the training samples. By implementing these targeted preprocessing steps, we aim to enhance the model's generalization capabilities and prevent overfitting, ensuring accurate performance across diverse neuromorphic inputs.

2 Energy Consumption Analysis Details

To estimate the theoretical energy consumption of SWformer, we follow the approach in previous studies [5, 14, 18, 19, 24]. First, we calculate the synaptic operations (SOPs):

$$\mathrm{SOPs}(l) = fr \times \mathrm{T} \times \mathrm{FLOPs}(l) \tag{1}$$

where l represents a block or layer in SWformer, fr denotes the firing rate of the input spike train for that block or layer, and T is the simulation time step of the spiking neuron. The number of floating point operations per layer l is estimated as $\mathrm{FLOPs}(l)$, which refers to the number of multiply-and-accumulate (MAC) operations, while the number of AC operations depends on both the firing rate

Table 1: The details of training hyperparameters of different datasets.

Dataset	Optimizer	Epoch	lr	Batch Size
CIFAR-10 [7]	AdamW [10]	410	1.5e-3	128
CIFAR-100 [6]		410	1.5e-3	128
ImageNet [3]		310	1.0e-4	512
CIFAR10-DVS		105	7.5e-3	16
N-Caltech101		310	5.0e-4	16
N-Cars		310	2.0e-3	16
Action Recognition		210	1.0e-3	8
ASL-DVS		105	5.0e-4	8
NavGesture		105	5.0e-4	8

fr of the input spikes (as shown in Table 2) and the simulation time step T. SOPs represent the total number of spike-based accumulate (AC) operations.

Assuming the MAC and AC operations are implemented on 45nm hardware [2] where each MAC consumes $E_{\text{MAC}} = 4.6\text{pJ}$ and each AC consumes $E_{\text{AC}} = 0.9\text{pJ}$, the total energy consumption of SWformer can be estimated by summing the energy for all MAC and AC operations across layers:

$$\begin{aligned} E_{\text{SWformer}} = & E_{\text{MAC}} \times \text{Convert}^{1}_{\text{SNN CONV}} \\ & + E_{\text{AC}} \times \left(\sum_{n=2}^{N} \text{SOP}^{n}_{\text{SNN Conv}} + \sum_{j=1}^{M} \text{SOP}^{j}_{\text{SNN FC}}\right) \end{aligned} \tag{2}$$

where $\text{Convert}^{1}_{\text{SNN CONV}}$ represents the initial layer that converts static RGB images into spike form, which we omit for neuromorphic tasks. We then aggregate the SOPs over the following layers, encompassing the SOPs from n SNN Conv layers and j SNN Fully Connected (FC) layers. This cumulative SOP count is then multiplied by the energy per spike operation, E_{AC}, yielding the total energy consumption related to the spiking activity.

For ANNs, the theoretical energy consumption of block b can be expressed as:

$$\text{Power(b)} = 4.6pJ \times \text{FLOPs(b)} \tag{3}$$

In contrast, for SNNs, the power consumption of block b is given by:

$$\text{Power(b)} = 0.9pJ \times \text{SOPs(b)} \tag{4}$$

2.1 Method Analysis

Effectiveness of Spiking Frequency Representation SNNs mimic biological vision by continuously sampling input data and independently generating spikes in response to changes in the visual scene. Additionally, neuromorphic data mainly captures brightness changes, like edges and textures, making the ability to learn features in the frequency spectrum essential for SNNs. Fig. 2(c-d) presents the Fourier-transformed feature maps of the last Transformer block of SWformer with and without spiking wavelet transform, while the number of parameters remains constant for the models in Fig. 2 2(a-b). The model that explicitly incorporates spiking frequency representation captures diverse frequency components across different channels, with an accuracy of 83.9% compared to 80.9% on Cifar10-DVS for the model without spiking wavelet transform (see Table 4 in the main paper). This empirical evidence strongly suggests that the explicit incorporation of spiking frequency representation is crucial for enhancing the learning of frequency features and ultimately improving the overall performance of the model. The discussion on static datasets can be found in Sec. 4.4 in the main paper.

Effectiveness of Frequency Learner Existing Spiking Transformers [18, 23, 24] adopt the vanilla Transformer [17] design, which uses a global operation to exchange information among non-overlapping patch tokens, enabling them to effectively capture global information. However, the signal transfer mechanism in SNNs generates output activation maps rich in high-frequency components, as evidenced by Fig. 3(a), which illustrates that data processed by the Spiking Patch Splitting (SPS) module, the initial embedding operation before the visual data is processed by Transformer blocks, contains a significant amount of high-frequency information. In stark contrast, our SWformer's FATM effectively captures specific frequency information on the corresponding channel, enabling more comprehensive feature learning across a wide frequency range. This is clearly illustrated in Fig. 3(b), which shows that SWformer maintains the transmission of high-frequency information even as the layers become deeper. The enhanced frequency learning capability is directly attributed to the Frequency Learner in SWformer, which leads to more comprehensive feature learning. These findings strongly suggest that the incorporation of Frequency Learner in SWformer significantly improves its ability to capture and utilize high-frequency components, ultimately resulting in superior performance compared to existing Spiking Transformers.

Effectiveness of Frequency-Aware Token Mixer The channel mixer (CM), spatial learner (SL), and frequency learner (FL) in the Frequency-Aware Token Mixer (FATM) have distinct but complementary roles: CM employs point-wise convolution to aggregate multi-scale features on the channel dimension, while SL uses 3x3 convolution to capture local spatial information, and FL weights information in the time-frequency domain. Table 3 shows that removing either the SL, CM or FL branches results in decreased accuracy on Cifar10-DVS and Cifar100. To support the effectiveness of FATM in token mixing, Fig. 4 demonstrates that the combination of FATM branches achieves a larger receptive field than global spiking self-attention, enabling a better understanding of long-range dependencies. This empirical evidence demonstrates the effectiveness of FATM as a token mixer.

Table 2: The spike firing rates in SWformer-6-512.

			T = 1	T = 2	T = 3	T = 4	Average
SPS	**Conv1**		0.0802	0.1550	0.1226	0.1778	0.1339
	Conv2		0.0434	0.0637	0.0564	0.0504	0.0535
	Conv3		0.0357	0.0499	0.0369	0.0404	0.0407
	Conv4		0.0682	0.1741	0.1754	0.1921	0.1525
Block 1	**FATM**	**Input**	0.2280	0.2674	0.2725	0.2612	0.2573
		H_f	0.2629	0.3523	0.2949	0.3362	0.3116
		H_I	0.1229	0.1461	0.1870	0.1808	0.1592
	MLP	**Layer 1**	0.3661	0.3944	0.4176	0.4082	0.3966
		Layer 2	0.0466	0.0570	0.0554	0.0564	0.0539
Block 2	**FATM**	**Input**	0.3446	0.3678	0.3916	0.4000	0.3760
		H_f	0.2539	0.3484	0.3287	0.3412	0.3181
		H_I	0.1431	0.1836	0.2123	0.2172	0.1891
	MLP	**Layer 1**	0.3585	0.3842	0.4058	0.4132	0.3904
		Layer 2	0.0371	0.0473	0.0493	0.0498	0.0459
Block 3	**FATM**	**Input**	0.3349	0.3579	0.3708	0.3734	0.3593
		H_f	0.2570	0.3772	0.3468	0.3662	0.3368
		H_I	0.1132	0.1609	0.1769	0.1849	0.1590
	MLP	**Layer 1**	0.3566	0.3889	0.4047	0.4102	0.3901
		Layer 2	0.0316	0.0407	0.0427	0.0434	0.0396
Block 4	**FATM**	**Input**	0.3164	0.3490	0.3593	0.3612	0.3465
		H_f	0.2899	0.4477	0.3965	0.4271	0.3903
		H_I	0.1037	0.1633	0.1819	0.1963	0.1613
	MLP	**Layer 1**	0.3432	0.3824	0.3974	0.4009	0.3810
		Layer 2	0.0208	0.0294	0.0310	0.0317	0.0282
Block 5	**FATM**	**Input**	0.2861	0.3267	0.3424	0.3416	0.3242
		H_f	0.2938	0.4594	0.3998	0.4326	0.3964
		H_I	0.1074	0.1760	0.2019	0.2154	0.1752
	MLP	**Layer 1**	0.2935	0.3619	0.3785	0.3828	0.3542
		Layer 2	0.0167	0.0250	0.0266	0.0272	0.0239
Block 6	**FATM**	**Input**	0.2530	0.3333	0.3480	0.3547	0.3223
		H_f	0.3223	0.4946	0.4306	04601	0.4269
		H_I	0.0896	0.1718	0.1779	0.1936	0.1582
	MLP	**Layer 1**	0.2067	0.3512	0.3716	0.3846	0.3285
		Layer 2	0.0466	0.0110	0.0114	0.0107	0.0199
Head	**FC**		0.0004	0.4762	0.4370	0.5682	0.3705

Table 3: Effectiveness of FATM components

Datasets	**Baseline**	**w/o Channel**	**w/o Spatial**	**w/o Frequency**
Cifar10-DVS	83.9	81.7 (-2.2)	82.2 (-1.7)	80.9 (-3.0)
Cifar100	79.6	79.1 (-0.5)	78.3 (-1.3)	78.8 (-0.8)

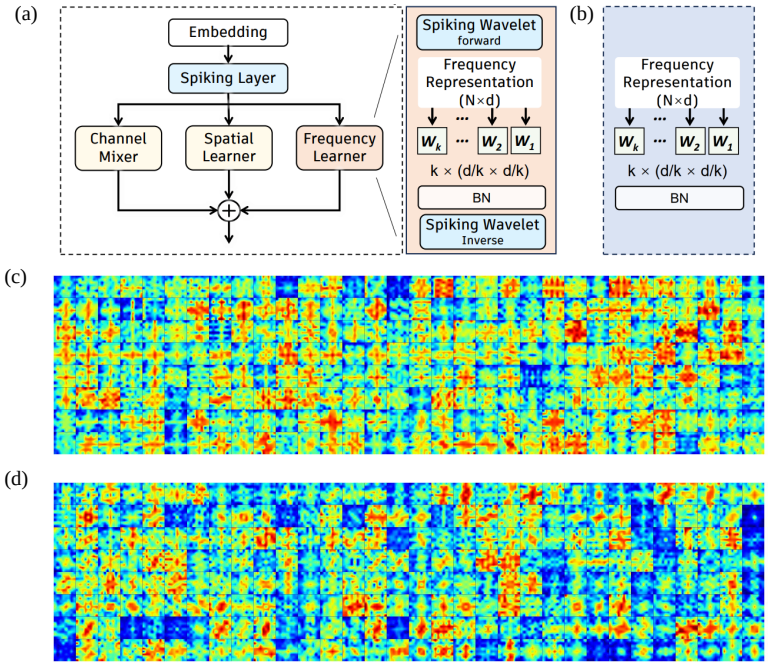

Fig. 2: (a-b) SWformer with and without spiking wavelet transform. (c-d) Fourier-transformed feature map across all channels on CIFAR10-DVS for the model (a) and (b), respectively. Explicitly incorporating spiking frequency representation has been shown to enhance the learning of frequency features, resulting in improved performance.

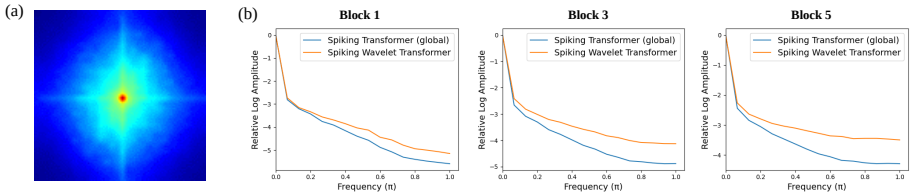

Fig. 3: (a) Fourier-transformed feature map of SPS output. (b) Relative log amplitudes of Fourier-transformed feature maps from model blocks 1, 3, and 5.

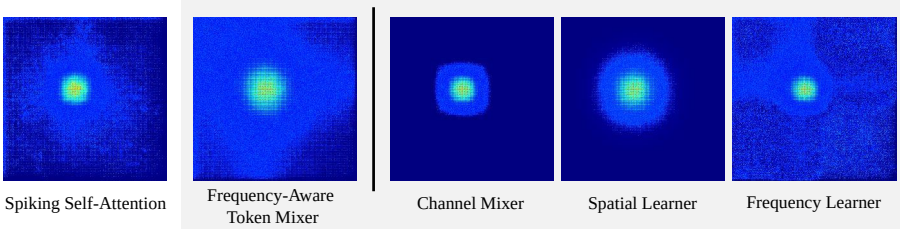

Fig. 4: Visualization of effective receptive field.

References

1. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 702–703 (2020)
2. Davies, M., Srinivasa, N., Lin, T.H., Chinya, G., Cao, Y., Choday, S.H., Dimou, G., Joshi, P., Imam, N., Jain, S., et al.: Loihi: A neuromorphic manycore processor with on-chip learning. Ieee Micro **38**(1), 82–99 (2018)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. Ieee (2009)
4. Fang, W., Chen, Y., Ding, J., Chen, D., Yu, Z., Zhou, H., Masquelier, T., Tian, Y., other contributors: Spikingjelly. https://github.com/fangwei123456/spikingjelly (2020), accessed: 2023-02-21
5. Hu, Y., Tang, H., Pan, G.: Spiking deep residual networks. IEEE Transactions on Neural Networks and Learning Systems (2021)
6. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
7. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (Nov 1998). https://doi.org/10.1109/5.726791
8. Li, H., Liu, H., Ji, X., Li, G., Shi, L.: CIFAR10-DVS: An Event-Stream Dataset for Object Classification. Frontiers in Neuroscience **11** (2017)
9. Li, Y., Kim, Y., Park, H., Geller, T., Panda, P.: Neuromorphic Data Augmentation for Training Spiking Neural Networks. arXiv preprint arXiv:2203.06145 (2022)
10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
11. Maro, J.M., Ieng, S.H., Benosman, R.: Event-based gesture recognition with dynamic background suppression using smartphone computational capabilities. Frontiers in neuroscience **14**, 275 (2020)
12. Miao, S., Chen, G., Ning, X., Zi, Y., Ren, K., Bing, Z., Knoll, A.: Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection. Frontiers in neurorobotics **13**, 38 (2019)
13. Orchard, G., Jayawant, A., Cohen, G.K., Thakor, N.: Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in neuroscience **9**, 437 (2015)
14. Panda, P., Aketi, S.A., Roy, K.: Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization. Frontiers in Neuroscience **14**, 535502 (2020)
15. Sironi, A., Brambilla, M., Bourdis, N., Lagorce, X., Benosman, R.: HATS: Histograms of averaged time surfaces for robust event-based object classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1731–1740 (2018)
16. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, \., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

18. Yao, M., Hu, J., Zhou, Z., Yuan, L., Tian, Y., Xu, B., Li, G.: Spike-driven transformer. arXiv preprint arXiv:2307.01694 (2023)
19. Yao, X., Li, F., Mo, Z., Cheng, J.: Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks. Advances in Neural Information Processing Systems **35**, 32160–32171 (2022)
20. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6023–6032 (2019)
21. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018)
22. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 13001–13008 (2020)
23. Zhou, C., Yu, L., Zhou, Z., Zhang, H., Ma, Z., Zhou, H., Tian, Y.: Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. arXiv preprint arXiv:2304.11954 (2023)
24. Zhou, Z., Zhu, Y., He, C., Wang, Y., Yan, S., Tian, Y., Yuan, L.: Spikformer: When spiking neural network meets transformer. arXiv preprint arXiv:2209.15425 (2022)