ENHANCING SPIKING TRANSFORMERS WITH BINARY ATTENTION MECHANISMS

Guobin Shen, Dongcheng Zhao, Sicheng Shen & Yi Zeng[†]
Brain-inspired Cognitive Intelligence Lab, Institute of Automation, CAS

ABSTRACT

Spiking Neural Networks (SNNs) are increasingly recognized as an efficient alternative to traditional artificial neural networks. Recent advancements, particularly the integration of SNNs with Transformer structures to create 'SpikFormer', have significantly enhanced the performance of SNNs. However, the current non-spiking form of attention in SpikFormer poses risks of attention value explosion and still results in high computational costs for SNNs. To address this issue, we propose a novel binary attention mechanism. By introducing an attention shift mechanism and adaptive thresholds for neurons, we have successfully binarized the attention matrices in SpikFormer, leading to more efficient and sparser spiking neural networks. Experiments on image and neuromorphic datasets demonstrate that our approach maintains comparable performance to the original SpikFormer while reducing computational costs.

1 INTRODUCTION

In recent years, Spiking Neural Networks (SNNs) (Maass, 1997) have emerged as a biologically-inspired and energy-efficient alternative to traditional artificial neural networks, excelling in time-sensitive and power-constrained applications such as image (Wu et al., 2018; Sengupta et al., 2019; Duan et al., 2022; Shen et al., 2023a;b) and natural language processing Bal & Sengupta (2023); Zhu et al. (2023). The advent of 'SpikFormer' Zhou et al. (2022), which combines Transformer's attention mechanism with SNNs' energy efficiency, has further enhanced their performance Zhou et al. (2023); Che et al. (2023). However, SpikFormer's reliance on non-spiking attention mechanisms leads to potential attention value explosion and increased computational costs, contradicting the low-power ethos of SNNs. Our research addresses this by introducing a novel binary attention mechanism for SpikFormer, integrating an attention shift mechanism and adaptive neuron thresholds to binarize attention matrices, resulting in a more efficient, sparser SNN that maintains the system's energy efficiency.

2 METHOD

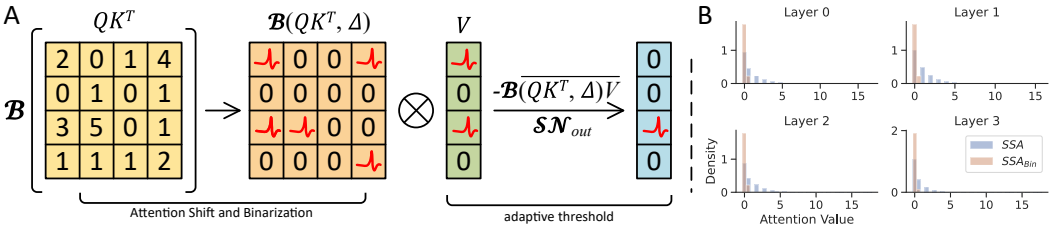

Figure 1: A. Illustration of SSA_{Bin}. B. Attention values for different layers on CIFAR10.

In this section, we present our proposed binary attention mechanism for SpikFormer. The Spiking Self-Attention (SSA) is the central component of the architecture. Given an input feature sequence $X \in \mathbb{R}^{T\times N\times D}$, the SSA involves three key components: query (Q), key (K), and value (V). These components are computed from the input X using learnable linear matrices $W_Q, W_K, W_V \in \mathbb{R}^{D\times D}$ as follows:

$$Q = \mathcal{SN}_Q(XW_Q), \quad K = \mathcal{SN}_K(XW_K), \quad V = \mathcal{SN}_V(XW_V) \tag{1}$$

$$SSA(Q, K, V) = \mathcal{SN}_{out}(\frac{QK^T}{\sqrt{D}}V) \tag{2}$$

[†]Corresponding Author. yi.zeng@ia.ac.cn

In Eq. 1 and 2, The $\mathcal{SN}(\cdot)$ represents the sequence of spikes generated by the neuron when the input is $I = [I(0), I(1), \ldots, I(T)]$. $I(t)$ is the input current at step t. In Eq. 2, $\frac{QK^T}{\sqrt{D}}$ first converts spikes into floating-point numbers before multiplying with V. However, due to the lack of a Softmax mechanism, the result of QK^T can be quite large, especially after matrix multiplication with V. This leads to excessively high input currents for $\mathcal{SN}_{out}$, for a detailed definition, please refer to Appendix A.1. To address this issue, we propose a novel binary attention mechanism. By binarizing the attention matrix, we transform the attention mechanism into a spiking form, reducing the risk of high input currents to $\mathcal{SN}_{out}$ and decreasing computational costs.

$$SSA_{Bin}(Q, K, V) = \mathcal{SN}_{out}(\mathcal{B}(QK^T, \Delta)V - \overline{\mathcal{B}(QK^T, \Delta)V}) \tag{3}$$

$$\mathcal{B}(x) = (x > \Delta) \odot \kappa \tag{4}$$

In Eq. 3, $SSA_{Bin}(Q, K, V)$ represents the binary spiking self-attention mechanism, where $\mathcal{B}$ denotes the binarization process applied to the product of QK^T. The threshold Δ is a hyperparameter, determining the binarization cutoff. The term $\mathcal{B}(QK^T, \Delta)V - \overline{\mathcal{B}(QK^T, \Delta)V}$ represents the computation after binarization, where the subtraction of the average value $\overline{\mathcal{B}(QK^T, \Delta)V}$ serves to normalize the binary attention output. Eq. 4 defines the binarization function $\mathcal{B}(x)$, which outputs κ for values of x greater than the threshold Δ, and 0 otherwise. The operator $\odot$ indicates element-wise multiplication, as shown in Fig. 1.

3 EXPERIMENTS

We evaluate the efficacy of our proposed binary attention mechanism in SpikFormer through a series of experiments. These experiments are designed to test the performance of our binary spiking self-attention mechanism (SSA_{Bin}) in various settings, focusing on image recognition tasks. We conduct extensive experiments on both traditional image datasets, CIFAR10/100 Krizhevsky et al. (2009); Xu et al. (2015), as well as on neuromorphic datasets, DVS-CIFAR10 Li et al. (2017) and N-CALTECH101 Orchard et al. (2015), which present more challenging and dynamic visual inputs. We use both the SSA and the plain binary version (SSA_{Bin}) as baselines for comparison. We further examine the impact of introducing the attention shift mechanism, denoted as $\overline{\mathcal{B}(QK^T, \Delta)V}$, and the adaptive threshold $\Delta \neq 0$.

Table 1: Comparison of different attention mechanisms on image and neuromorphic datasets.

Method	CIFAR10			CIFAR100			DVS-CIFAR10			NCALTECH101		
	Acc	fr_{attn}	fr_{out}	Acc	fr_{attn}	fr_{out}	Acc	fr_{attn}	fr_{out}	Acc	fr_{attn}	fr_{out}
SSA	95.51	0.54	0.52	78.21	0.58	0.52	80.9	0.44	0.82	79.42	0.24	0.39
baseline	95.52	0.26	0.30	78.3	0.30	0.22	80.3	0.12	0.34	78.62	0.30	**0.22**
w/ shift	95.77	0.16	0.23	78.93	0.31	0.31	81.1	**0.09**	0.31	80.16	0.27	0.33
w/ mean	95.84	0.23	0.27	79.22	0.10	0.22	79.2	0.21	0.32	79.86	0.14	0.34
SSA_{Bin}	**95.77**	**0.08**	**0.08**	**79.25**	**0.10**	**0.09**	**81.2**	0.17	**0.26**	**80.23**	**0.13**	0.30

Tab. 1 presents the performance metrics of our experiments. Across all datasets, it is evident that the introduction of the attention shift and adaptive threshold significantly enhances the accuracy (Acc) and reduces the firing rates (fr_attn and fr_output), which are indicative of computational cost, in comparison to the SSA and SSA_{Bin} plain baselines. Notably, SSA_{Bin} with the adaptive threshold achieves the best balance between accuracy and efficiency, demonstrating the effectiveness of our binary attention mechanism in SpikFormer.

4 CONCLUSION

We have proposed a binary attention mechanism for SpikFormer, enhancing its efficiency and sparsity. Our evaluations on CIFAR10/100 and neuromorphic datasets confirm that our method maintains accuracy while reducing computational costs. The binary attention mechanism, with its attention shift and adaptive threshold, addresses the limitations of traditional SNNs and demonstrates the viability of energy-efficient, spike-based processing. This advancement marks a significant step toward practical SNN applications and opens new avenues for optimizing neural computation.

ACKNOWLEDGEMENTS

This research is financially supported by a funding from Institute of Automation, Chinese Academy of Sciences (Grant No. E411230101).

URM STATEMENT

Authors G.S. and S.S meet the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

Malyaban Bal and Abhronil Sengupta. Spikingbert: Distilling bert to train spiking language models using implicit differentiation. *arXiv preprint arXiv:2308.10873*, 2023.

Kaiwei Che, Zhaokun Zhou, Zhengyu Ma, Wei Fang, Yanqi Chen, Shuaijie Shen, Li Yuan, and Yonghong Tian. Auto-spikformer: Spikformer architecture search. *arXiv preprint arXiv:2306.00807*, 2023.

Chaoteng Duan, Jianhao Ding, Shiyan Chen, Zhaofei Yu, and Tiejun Huang. Temporal effective batch normalization in spiking neural networks. *Advances in Neural Information Processing Systems*, 35:34377–34390, 2022.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178. IEEE, 2004.

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

Giacomo Indiveri, Federico Corradi, and Ning Qiao. Neuromorphic architectures for spiking deep neural networks. In *2015 IEEE International Electron Devices Meeting (IEDM)*, pp. 4–2. IEEE, 2015.

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

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

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

Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13:95, 2019.

Guobin Shen, Dongcheng Zhao, and Yi Zeng. Backpropagation with biologically plausible spatiotemporal adjustment for training deep spiking neural networks. *Patterns*, 3(6), 2022.

Guobin Shen, Dongcheng Zhao, Yiting Dong, and Yi Zeng. Brain-inspired neural circuit evolution for spiking neural networks. *Proceedings of the National Academy of Sciences*, 120(39): e2218173120, 2023a.

Guobin Shen, Dongcheng Zhao, and Yi Zeng. Exploiting high performance spiking neural networks with efficient spiking patterns. *arXiv preprint arXiv:2301.12356*, 2023b.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

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

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

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

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

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

A Appendix

A.1 Neuron Model

The LIF neuron model is characterized by its membrane potential dynamics, which are governed by a differential equation accounting for the leaky integration of synaptic currents. In our SpikFormer architecture, the LIF model is discretized and implemented as follows:

$$\begin{aligned} u(t) &= v(t-1) + \frac{1}{\tau}(I(t) - v(t-1)) \\ s(t) &= g(u(t) - v_{th}) \\ v(t) &= u(t)(1 - s(t)) + v_{reset}s(t) \end{aligned} \tag{5}$$

In this model, $u(t)$ represents the membrane potential at time t, $v(t-1)$ is the membrane potential from the previous time step, and $I(t)$ is the input current at time t. The term τ is the membrane time constant that modulates the potential's decay towards the resting potential in the absence of input. The function $g(\cdot)$ is a Heaviside step function that outputs 1 if the membrane potential exceeds the threshold v_{th}, causing the neuron to spike, and 0 otherwise. After a spike is emitted ($s(t) = 1$), the membrane potential $v(t)$ is reset to v_{reset}, a lower value representing the hyperpolarized state of the neuron post-firing. This reset mechanism introduces a refractory period during which the neuron is less likely to fire. The LIF neuron thus captures the essential spiking behavior of biological neurons, allowing our SpikFormer to process information in a temporally dynamic and energy-efficient manner.

During the training of our SpikFormer model, we encounter non-differentiable components in the form of the spiking function g and the binary threshold function $\mathcal{B}$. To facilitate gradient-based optimization, we use surrogate gradient functions for both. Specifically, we employ the Sigmoid function as the surrogate for the spiking function's gradient. The Sigmoid function, defined as $\sigma(x) = \frac{1}{1+e^{-x}}$, is smooth and differentiable, with its gradient given by:

$$g'(x) = \sigma(\alpha x)(1 - \sigma(\alpha x)) = \frac{1}{1 + e^{-\alpha x}}\left(1 - \frac{1}{1 + e^{-\alpha x}}\right). \tag{6}$$

As in Eq. 6, α is used to control the width of the approximation function, and in the experimental section, we use the same hyperparameters as in Zhou et al. (2022), $\alpha = 4$.

For the binary attention mechanism, where the threshold function $\mathcal{B}(x)$ is applied, we approximate its derivative using the gradient of the Softmax function. The Softmax function for a vector $\mathbf{x}$ and its i-th component is defined as $\text{Softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$, and its gradient with respect to an input x_i can be written as:

$$\frac{d\mathcal{B}(x)}{dx_i} = \text{Softmax}(\mathbf{x})_i(1 - \text{Softmax}(\mathbf{x})_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}\left(1 - \frac{e^{x_i}}{\sum_j e^{x_j}}\right). \tag{7}$$

These surrogate gradients are continuous and differentiable, allowing us to apply standard backpropagation techniques for training the SpikFormer network. By using these approximations, we can effectively compute gradients across the otherwise non-differentiable spiking and threshold functions, facilitating the network's ability to learn from data while retaining the computational benefits of spiking models.

A.2 HYPERPARAMETER SETTINGS

In our experiments we use the commonly used SpikFormer model Spikformer-4-384 (Zhou et al., 2022). For training, we set the batch size to 128 to balance computational efficiency and convergence stability. The membrane time constant (τ) was determined to be 2, dictating the leakage rate of the LIF neuron model. The spike intensity κ is set to 1. Following a spike, the membrane potential resets to v_{reset}, which was set to 0. The attention shift scaling factor Δ was computed using a running average during training, and was held constant during testing. The total number of training epochs was set to 400 to ensure ample learning without overfitting. For the static image dataset, the simulation step was set to $T = 4$, while for the neuromorphic dataset, the simulation step was set to $T = 16$.

Let $\overline{V}$ be the running average of $\mathcal{B}(QK^T, \Delta)V$. After each batch, we update $\overline{V}$ using the momentum μ as follows:

$$\overline{V}_{new} = \mu \cdot \overline{V}_{old} + (1 - \mu) \cdot \mathcal{B}(QK^T, \Delta)V \tag{8}$$

where μ is the momentum term, set to 0.99 in our experiments. This running average is computed during the training process and is fixed during the testing phase.

We employ a direct encoding strategy (Wu et al., 2019) where the first layer of the network functions as the encoding layer. This method has been deliberately chosen for its ability to preserve accuracy while significantly reducing the simulation length, a crucial factor in enhancing the computational efficiency of our spiking neural network. The encoding layer receives external stimuli and converts them into a steady stream of input currents. By doing so, it not only maintains a consistent input format but also ensures compatibility with various data formats. This approach facilitates seamless integration of the same network architecture with different data representations, allowing our model to be versatile and applicable to a wide range of tasks without the need for structural adjustments. This encoding method is widely used in SNNs. This encoding method is widely used in SNNs (Zhou et al., 2022; Shen et al., 2022; 2023a; Wu et al., 2018; Duan et al., 2022; Zhu et al., 2023; Zhou et al., 2023; Che et al., 2023).

A.3 COMPUTATIONAL COST ANALYSIS

We present an analysis of the synaptic operations (SOPs) and energy consumption for our proposed binary attention mechanism in the SpikFormer model. Synaptic operations (SOPs) (Zhou et al., 2022; Indiveri et al., 2015) are indicative of the computational load placed on the network during inference and are directly related to the energy efficiency of the model.

The Synaptic Operations (SOPs) are calculated by counting the number of operations required during the forward pass of the network, as defined in Equation 9, where fr represents the average firing rate of the network and T represents the number of simulation time steps. FLOPs are the network's floating-point operations. This metric serves as a proxy for the computational complexity and potential energy usage of the network. Table 2 displays the SOPs for different datasets and configurations of our SpikFormer model. As shown, our proposed SSA_Bin mechanism consistently requires fewer SOPs across all datasets compared to the baseline and other variants, illustrating the computational savings achieved by our approach.

$$SOPs = fr \times T \times FLOPs \tag{9}$$

Assuming a linear relationship between SOPs and energy usage, energy consumption is estimated based on the number of SOPs. Following the methodology of (Zhou et al., 2022; Indiveri et al., 2015; Hu et al., 2021), the energy cost per SOP is approximated to be 77 fJ. The energy values, expressed in millijoules (mJ), are summarized in Table 2. These values confirm the efficiency of the SA_{Bin} model, which exhibits lower energy consumption across all datasets, thereby validating

the effectiveness of our binary attention mechanism in reducing the overall energy footprint of the SpikFormer architecture.

Table 2: Synaptic operations (SOPs) and energy consumption for different configurations of the SpikFormer model across various datasets.

SOP (G)				
Dataset	CIFAR10	CIFAR100	DVS-CIFAR10	NCALTECH101
SSA	1.29	1.32	5.73	5.91
baseline	0.78	0.83	3.76	3.94
w / shift	0.69	0.82	3.17	3.76
w / mean	0.64	0.71	3.64	3.27
SSA_{Bin}	**0.51**	**0.54**	**3.22**	**3.16**
Energy (mJ)				
SSA	0.099334	0.101	0.441	0.455
baseline	0.060	0.064	0.289	0.303
w / shift	0.053	0.063	0.244	0.289
w / mean	0.049	0.055	0.280	0.252
SSA_{Bin}	**0.039**	**0.041**	**0.248**	**0.243**

The reduction in SOPs and energy consumption demonstrates the practical benefits of our binary attention mechanism, addressing the reviewer's concern and highlighting our contribution towards creating more efficient Spiking Neural Networks.

A.4 DATASET

Here, we outline the datasets utilized in our experiments to evaluate the performance of the proposed binary attention mechanism in SpikFormer. The datasets include:

CIFAR10 and CIFAR100 (Krizhevsky et al., 2009; Xu et al., 2015) CIFAR10 and CIFAR100 are widely-used image datasets consisting of 60,000 color images categorized into 10 and 100 classes, respectively. Each image is 32×32 pixels in size. These datasets are benchmarks for image classification tasks and allow for the assessment of the model's performance in terms of accuracy and efficiency.

DVS-CIFAR10 (Li et al., 2017) The DVS-CIFAR10 dataset is a neuromorphic dataset derived from the CIFAR10 dataset. It is generated using a Dynamic Vision Sensor (DVS), which captures pixel changes in the form of events or 'spikes'. This dataset is particularly suited for temporally-sensitive models and evaluates the SpikFormer's ability to process spatio-temporal data.

N-CALTECH101 (Orchard et al., 2015) N-CALTECH101 is another neuromorphic dataset, which is a spiking version of the well-known CALTECH101 dataset (Fei-Fei et al., 2004). It contains spike events captured by a DVS camera from the original CALTECH101 images. This dataset challenges the model's capability to handle more complex and varied visual patterns in a spike-based processing framework.