Efficient Spiking Transformer Enabled By Partial Information

Ziqing Wang$^{1,2,*}$, Yuetong Fang$^1*$, Jiahang Cao$^1$, Zhongrui Wang$^3*$, Renjing Xu$^1*$

$^1$The Hong Kong University of Science and Technology (Guangzhou)
$^2$Sun Yat-sen University
$^3$The University of Hong Kong

*Equal contribution
*Email: renjingxu@ust.hk, zrwang@eee.hku.hk
Abstract

Spiking neural networks (SNNs) have received substantial attention in recent years due to their sparse and asynchronous communication nature, and thus can be deployed in neuromorphic hardware and achieve extremely high energy efficiency. However, SNNs currently can hardly realize a comparable performance to that of artificial neural networks (ANNs) because their limited scalability does not allow for large-scale networks. Especially for Transformer, as a model of ANNs that has accomplished remarkable performance in various machine learning tasks, its implementation in SNNs by conventional methods requires a large number of neurons, notably in the self-attention module. Inspired by the mechanisms in the nervous system, we propose an efficient spiking Transformer (EST) framework enabled by partial information to address the above problem. In this model, we not only implemented the self-attention module with a reasonable number of neurons, but also introduced partial-information self-attention (PSA), which utilizes only partial input signals, further reducing computational resources compared to conventional methods. The experimental results show that our EST can outperform the state-of-the-art SNN model in terms of accuracy and the number of time steps on both Cifar-10/100 and ImageNet datasets. In particular, the proposed EST model achieves 78.48% top-1 accuracy on the ImageNet dataset with only 16 time steps. In addition, our proposed PSA reduces flops by 49.8% with negligible performance loss compared to a self-attention module with full information.

KEYWORDS: spiking Transformer, partial-information self-attention, spiking neural networks
Introduction

Spiking neural networks (SNNs), considered as the next generation neural networks (1), are brain-inspired neural networks based on the dynamic characteristics of biological neurons (2, 3). SNNs have attracted great attention due to their unique properties in handling sparse data, which can yield great energy efficiency benefits on neuromorphic hardware. Due to their specialties, they have been widely utilized in various fields, such as classification tasks on dynamic vision sensor (DVS) datasets (4) and static datasets (5), object detection (6) and tracking (7), etc. Meanwhile, popular deep learning models, including recurrent neural networks (RNN) (8), multilayer perceptron (MLP) (9), and convolutional neural networks (CNN) (10), have been implemented in the form of SNNs. In the case of Transformer (11), it has not been deployed as an SNN to date, although it has achieved excellent performance in both natural language processing (NLP) and computer vision (CV) (12). Therefore, taking advantage of the features and benefits of both models would be promising to achieve both high performance and high energy efficiency.

Recently, several efforts have been made to combine SNNs and the Transformer to take advantage of the biological nature of SNNs and the high performance of the Transformer. However, these methods either used SNNs and the Transformer as two separate branches to extract features (13) or replaced part of the Transformer with SNNs; in either case, the non-SNN part means the entire model will not be able to run independently on neuromorphic hardware, which in turn hindering the fully exploitation of high energy efficiency benefits of the SNNs. The biggest challenge for the complete implementation of the SNN version of Transformers mainly lies on the self-attention (SA) module, which is considered to be one of the key reasons why Transformers are able to achieve superior performance. However, this also incurs the massive computing operations and huge energy consumption. The most expensive part of the self-attention module is the multiplication of two large matrices, i.e., the query and key matrices. While in SNNs, each element of a given matrix is often considered as a neuron; therefore, the
multiplication of two large matrices require a large amount of computational resource to correspondingly construct the neuron counterparts, which limits the scalability of Transformer based on SNNs.

Inspired by the cross-correlation and partial information observed in the nervous system, we propose an efficient spiking Transformer (EST) with partial-information self-attention (PSA), as shown in Figure 1. In our PSA module, spike trains generated by individual neurons are used to represent each row of the matrix used in the self-attention module instead of representing each element, which greatly reduces the number of neurons required. Moreover, since matrix multiplication between the query and the key in the self-attention module consumes a massive amount of computational power, we convert only part of the input information into these matrices, which greatly reduces the number of spikes in the computation process thus reducing energy consumption. We evaluate our EST model on the Cifar-10/100 and ImageNet datasets, and the experimental results show that our proposed model outperforms the state-of-art SNNs and requires only a small number of time steps. In addition, it has been shown that partial information operation in PSA reduced flops by 49.8% compared to SA, while accuracy was reduced by only 0.4%.

RESULTS

Integrate-and-fire (IF) neuron model is one of the most widely used models for analyzing the behavior of the nervous system, which can be represented as:

\[ u_i^t = u_i^{t-1} + \sum_j w_{ij} o_j^t - v o_i^{t-1} \]

\[ o_i^{t-1} = \begin{cases} 1, & \text{if } u_i^{t-1} > v \\ 0, & \text{otherwise} \end{cases} \]  

(1)

where \( u \) is the membrane potential, subscript \( i \) and \( j \) represent the post-neuron and pre-neuron, respectively, superscript \( t \) is the time step, \( w \) is the learnable weight, \( o \) is the binary
output spike, and $v$ is the threshold. When the membrane potential exceeds the threshold, the neuron will fire.

The training of SNNs with a large amount of IF neurons is a computationally expensive process, which limits the scalability of SNNs. On the other hand, self-attention is also a computationally expensive module in the Transformer that requires a huge amount of operations, therefore it is often a focused part of the efforts people make to optimize the Transformer for energy efficiency. Inspired by the nervous system, we not only implemented the self-attention module with SNNs with much fewer required neurons, but also reduced the number of operations substantially with a partial information approach, as described below.

In the Transformer, the self-attention process is given as:

$$A_{ij} = Q_i K_j^T$$  \hspace{1cm}(2)

where $A_{ij}$ represents the $(i, j)^{th}$ element in the attention score calculated by the dot product of the $i^{th}$ row of the query matrix and the $j^{th}$ row of the key matrix. The attention score is a $n$ by $n$ matrix, where $n$ is the number of rows in the query and key, usually a large number. Therefore, this process contains a large number of dot product operations, which need substantial computational resources. In the conventional approach, the implementation of the self-attention module in SNNs requires a spiking neuron for each element in the query and value matrices. Therefore, a large number of neurons will be deployed to perform the dot product of these two matrices, which makes the network difficult to train. Inspired by the auditory system, each dot product can be accomplished with much fewer neurons.

Figure 1a illustrates the calculation of cross-correlation for the two ear-input signals in the auditory system: binaural cross-correlation first involves shifting the signal of one ear by a delay and performing a point-by-point multiplication of the signals of the left and right ears (14). The point-by-point multiplication illustrated in the figure, exactly the dot-product operation required in the self-attention module, involves only two input neurons and one output neuron. Therefore, the implementation of the self-attention module in SNNs is redesigned in a similar manner. As
shown in Figure 1b, the query and the key matrix will be converted to spike trains generated by two individual neurons. Afterward, the connected output neuron will perform the dot product for two spike trains. By the above process, only a small number of neurons are required to perform the matrix operations in self-attention, which largely reduces the computational resources.

The entire process can be expressed as:

\[
Q_{\text{spk}}[t] = IF(X[t]W_q) K_{\text{spk}}[t] = IF(X[t]W_k) V_{\text{spk}}[t] = IF(X[t]W_v)
\]  

where \(Q_{\text{spk}}, K_{\text{spk}}, V_{\text{spk}}\) are the spike trains of the query, key and value, \(IF(\cdot)\) is the IF neuron function, \(W_q, W_k, W_v\) are the weights.

Attention score is expressed as:

\[
A_{\text{spk,ij}}[t] = IF\left(\frac{Q_{\text{spk,i}}[t] \ast K^{T}_{\text{spk,j}}[t]}{d}\right)
\]

where \(A_{\text{spk,ij}}\) represents the spike trains of attention score calculated by the dot product of the \(i^{th}\) row of the query spike trains and the \(j^{th}\) row of the key spike trains, and \(d\) is a scale factor that is equal to the feature dimension of one attention head.

Although the number of neurons required has been greatly reduced, SNN-based self-attention still requires substantial operations because of the large number of spikes involved in the computation. In order to reduce the number of spikes, we notice that there are some inspirations that we can refer to in our nervous system. Figure 1a shows that missing information may happen in neuronal signal transmission. There are at least three potential reasons to explain this behavior: the quantal failure theory (15) indicates that the probability of evoking the release of the quantal package is reported to range from 0.25 to 0.5 with 0.5 being less common and 0.25 being quite common (16), which means there is at least 50% information lost in quantal synaptic transmission; refractory period allows the excited neuron to take time to be ready for a second stimulus, resulting in some input information being ignored; the threshold limit of the neurons prevents the information from being transmitted to the postsynaptic neuron if the threshold is not reached. Inspired by this property in the nervous system, our proposed PSA
module emulates the partial information approach, which implies information loss during spike transmission between spiking neurons.

Based on the above findings, half of the input spike information in our PSA module will be lost during the transmission. As an example illustrated in Figure 2a, if the query, the key, and the value neurons receive a spike train with 4 time steps, only the first 2 time steps of the query and the key spike trains are involved in the membrane potential accumulation, while the value neurons need to utilize all the 4 time steps. In this way, the number of spikes can be further reduced, thus also cutting down the overall amount of operations in the PSA module and making the whole model more energy-efficient.

Our EST applies the mentioned mechanism to the PSA module. Specifically, Swin Transformer (17) is used as the baseline and the ANN-to-SNN conversion algorithm is implemented as the training method. Next, we validate the effectiveness of our model and compare our model with other state-of-the-art methods, including RMP (18), RNL (19), QCFS (20), SNNAP (21), Hybrid (22), tdBN (23), TET (24) and DSR (5), for image classification tasks on Cifar-10 (25), Cifar-100 (26), and ImageNet datasets (27). Figure 2 shows the comparison of our model and the state-of-art model in terms of top-1 accuracy and time steps in Cifar-10/100 and ImageNet datasets. In all three datasets, the top-1 accuracy of our model outperforms all other state-of-the-art models.

As for low latency inference (\( T \leq 20 \)), these models usually use direct training methods, which are supposed to require fewer time steps because they do not need extra time steps to reduce conversion loss like ANN-to-SNN algorithm. However, without pre-trained weights and thresholds, these methods generally take a considerable time to train and have relatively low accuracy. Some state-of-the-art direct training methods, such as DSR, tdBN, TET, are selected for comparison. As for tdBN and TET, they only use 4 time steps for training in Cifar-10/100 datasets. However, their top-1 accuracy is much lower than our EST model. In particular, for tdBN, its accuracy in Cifar100 and ImageNet is more than 10% lower than our EST model.
Compared to DSR with 20 time steps, our EST model also performs better on all datasets with only 16 time steps, which means that our model can achieve the highest accuracy in a reasonable amount of time steps.

In terms of high latency inference (\(T \geq 100\)), these models use ANN-to-SNN conversion algorithms. Compared to the direct training method, ANN-to-SNN conversion algorithms can generally achieve higher accuracy because the weights and threshold are derived from ANN training, allowing the models to converge faster during training. During the conversion process, these methods should map the floating-point activation values of ANN onto the firing rates of SNN. However, there are some quantization errors between ANN and SNN, which requires more time steps to maintain a satisfactory performance. For example, the time step of RMP reaches 4096. In order to have achieve an attracting performance, our EST model take the advantage of the ANN-to-SNN conversion algorithm, but only requires 16 time steps, far less than the state-of-art ANN-to-SNN models. Moreover, the top-1 accuracy of our model is 1.17\%, 4.44\%, 3.01\% higher than the highest accuracy state-of-the-art model on Cifar-10, Cifar-100 and ImageNet datasets, respectively.

Further experiments have been conducted to check the performance of our model with ultra-low time steps. As shown in Figure 3a, our model achieve an accuracy of 97.2\% with only 4 time steps in Cifar-10 dataset, while achieving 83.7\% in Cifar-100 in Figure 3b, which outperforms all other methods shown earlier.

**DISCUSSION**

The proposed PSA module can obtain the query and key matrices, as well as the attention score with only half of the input spike train while maintaining high performance. As illustrated in Figure 4a, the performance of SA is just slightly better than PSA by 0.4\% accuracy in Cifar-100 dataset, but requires a greater amount of neurons compared with the latter one. The mean spike count per layer is calculated to estimate the neuronal activities, and as shown in Figure
the mean spike count of PSA is 49.8% less than that of SA, which means fewer operations are needed to conduct dot product operations owing to the partial information method, thus can also reduce the computational cost.

In order to visualize the differences between PSA and SA, we randomly choose a rabbit, tiger, and crocodile as classification objects from Cifar-100 dataset, and use the GradCam algorithm (28) to draw their corresponding attention maps. As shown in Figure 5, the blue part of the image represents the part that the self-attention module focuses on, while the red part represents the part that is ignored. The results show that the self-attention areas of both modules are similar in terms of the same image. For example, both PSA and SA focus on the rabbit’s head and ears, which refer to the key features used by the models to distinguish rabbits from other animals. As for tigers and crocodiles, both of them focus on their whole bodies to get more information. Although PSA only utilizes partial information, it focuses on similar key information as SA, which promises a comparable accuracy between SA and PSA.

CONCLUSION

In this paper, we proposed the efficient spiking transformer enabled by partial information. In our PSA, the query, key, and value matrices can be processed in a fully spiking manner, and our model only requires partial information to obtain the self-attention score, which reduces computational resources. In Cifar-10/100 and ImageNet datasets, experimental results show that our proposed EST outperforms the state-of-art SNNs with only 16 time steps. This may enable real-time and high-accuracy applications, such as event-based camera tracking in the future.

METHODS

In our EST model, tiny Swin Transformer (17) with 12 blocks is used as baseline. The total parameter of the model is 28.8 millions, and IFs are utilized as spiking neurons. For the datasets,
we use Cifar-10 (25), Cifar-100 (26) and ImageNet-1k (27) datasets. The Cifar-10 dataset contains 60000 (50000 training samples and 10000 testing samples) 32x32 color images in 10 different classes. The Cifar-100 dataset contains 60000 32×32 color images with 100 classes of objects. We choose ImageNet-1k dataset for further verification, which contains 1281167 training images, 50000 validation images and 100000 test images with 1000 object classes. In the pre-processing procedure for the above datasets, the images are resized to 224x224 resolution to make the input dimension align to the designed model, then apply random cropping and horizontal flipping for data augmentation. Also, data normalization is applied to ensure the input images have zero mean and unit variance.

For the training process, the rate norm layer (19) method is used as our ANN-to-SNN algorithm. The key idea of ANN-to-SNN conversion is to map the floating-point activation value in ANN to the firing rate in SNN. Refer to the previous works (6, 29), ReLU activation function is converted to the IF neuron. Therefore, our whole ANN-to-SNN training process is that we first pre-train the corresponding ANN model, where the IF neurons in the PSA module are replaced by the ReLU activation function, based on the official pre-trained weight of Swin Transformer. Then the pre-trained model is converted into SNN by replacing the ReLU activation function in ANN with IF neuron. The weights and thresholds obtained from ANN training are utilized to directly conduct inference on the SNN model. We trained 100 epochs using AdamW as the optimizer in the ANN pre-training process. The learning rate is 0.0001 and the CosineLRScheduler (30) algorithm is used as the scheduler for the learning rate.
Figure 1: Demonstration of PSA inspired by the nervous system. (a) The auditory system utilizes cross-correlation, where the output neuron will perform dot-product for the two input spike trains to accomplish sound localization. (b) Our PSA follows a similar manner, where each row in the query and key is converted into a spike train and fed to another neuron to perform a dot-product to obtain the self-attention score. Moreover, the self-attention score can be done with partial information: we utilize only partial inputs in this process.
Figure 2: (a) The overview of the computing flow in our PSA. (b-d) Performance of our EST model comparing with SOTA approaches: (b) Cifar-10, (c) Cifar-100, and (d) ImageNet.
Figure 3: (a-c) Experiment results of PSA: (a) Cifar-10, (b) Cifar-100, and (c) ImageNet.

Figure 4: The (a) test accuracy and (b) mean spike count of PSA vs SA on Cifar-100 dataset.
Figure 5: The attention maps of PSA vs SA on Cifar-100 dataset.
References and Notes

1. W. Maass, *Neural networks* **10**, 1659 (1997). Publisher: Elsevier.

2. W. S. McCulloch, W. Pitts, *The bulletin of mathematical biophysics* **5**, 115 (1943). Publisher: Springer.

3. E. M. Izhikevich, *IEEE Transactions on neural networks* **14**, 1569 (2003). Publisher: IEEE.

4. Y. Kim, P. Panda, *Neural Networks* **144**, 686 (2021). Publisher: Elsevier.

5. Q. Meng, *et al.*, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 12444–12453.

6. Y. Cao, Y. Chen, D. Khosla, *International Journal of Computer Vision* **113**, 54 (2015). Publisher: Springer.

7. Z. Yang, *et al.*, *arXiv preprint arXiv:1909.12942* (2019).

8. P. U. Diehl, G. Zarrella, A. Cassidy, B. U. Pedroni, E. Neftci, *2016 IEEE International Conference on Rebooting Computing (ICRC)* (IEEE, 2016), pp. 1–8.

9. W. Li, H. Chen, J. Guo, Z. Zhang, Y. Wang, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 783–793.

10. W. Fang, *et al.*, *Advances in Neural Information Processing Systems* **34**, 21056 (2021).

11. A. Vaswani, *et al.*, *Advances in neural information processing systems* **30** (2017).

12. A. Dosovitskiy, *et al.*, *arXiv preprint arXiv:2010.11929* (2020).

13. J. Zhang, *et al.*, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 8801–8810.
14. C. H. Keller, T. T. Takahashi, *Journal of Neuroscience* **16**, 4300 (1996). Publisher: Society for Neuroscience Section: Articles.

15. W. B. Levy, R. A. Baxter, *Journal of Neuroscience* **22**, 4746 (2002). Publisher: Society for Neuroscience Section: ARTICLE.

16. A. M. Thomson, *Trends in neurosciences* **23**, 305 (2000). Publisher: Elsevier.

17. Z. Liu, et al., *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 10012–10022.

18. B. Han, G. Srinivasan, K. Roy, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 13558–13567.

19. J. Ding, Z. Yu, Y. Tian, T. Huang, *arXiv preprint arXiv:2105.11654* (2021).

20. T. Bu, et al., *International Conference on Learning Representations* (2021).

21. Y. Li, S. Deng, X. Dong, R. Gong, S. Gu, *International Conference on Machine Learning* (PMLR, 2021), pp. 6316–6325.

22. N. Rathi, G. Srinivasan, P. Panda, K. Roy, *arXiv preprint arXiv:2005.01807* (2020).

23. H. Zheng, Y. Wu, L. Deng, Y. Hu, G. Li, *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 11062–11070. Issue: 12.

24. S. Deng, Y. Li, S. Zhang, S. Gu, *arXiv preprint arXiv:2202.11946* (2022).

25. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *Proceedings of the IEEE* **86**, 2278 (1998). Publisher: Ieee.

26. A. Krizhevsky, G. Hinton (2009). Publisher: Citeseer.
27. J. Deng, et al., 2009 IEEE conference on computer vision and pattern recognition (Ieee, 2009), pp. 248–255.

28. R. R. Selvaraju, et al., Proceedings of the IEEE international conference on computer vision (2017), pp. 618–626.

29. P. U. Diehl, et al., 2015 International joint conference on neural networks (IJCNN) (ieee, 2015), pp. 1–8.

30. I. Loshchilov, F. Hutter, arXiv preprint arXiv:1608.03983 (2016).