Supplementary Materials for: Masked Spiking Transformer

Ziqing Wang[1,2*] Yuetong Fang[1*] Jiahang Cao[1] Qiang Zhang[1] Zhongrui Wang[3,4†] Renjing Xu[1†]

[1]The Hong Kong University of Science and Technology (Guangzhou) [2]North Carolina State University
[3]The University of Hong Kong [4]ACCESS - AI Chip Center for Emerging Smart Systems

zwang247@ncsu.edu, {yfang870, jcao248, qzhang749}@connect.hkust-gz.edu.cn
zrwang@eee.hku.hk, renjingxu@ust.hk

1. Implementation Details

1.1. Dataset Description and Preprocessing

To validate the superiority of the proposed Masked Spiking Transformer (MST) on both static and neuromorphic datasets, we evaluated our model on the following benchmarks:

CIFAR-10 The CIFAR-10 [5] dataset encompasses 60,000 (50,000 training samples and 10,000 testing samples) 32×32 color images, with 10 distinct classes.

CIFAR-100 The CIFAR-100 dataset [4] comprises 60,000 (50,000 training samples and 10,000 testing samples) 32×32 color images, distributed across 100 object classes.

ImageNet-1k The ImageNet-1k dataset [3] consists of 1,281,167 training images, 50,000 validation images, and 100,000 test images, categorized into 1000 object classes.

The above datasets are static image datasets. In the preprocessing stage, the images are resized to a resolution of 224×224 to match the input dimensions of the model. Data augmentation techniques, such as random cropping and horizontal flipping, are applied to increase the diversity of the training data and improve the generalization ability of the model. In addition, data normalization is applied to the input images to ensure they have zero mean and unit variance. These are common preprocessing steps in machine learning that improve the performance of the model by enhancing the quality of the inputs.

*Equal contribution.
†Corresponding author

CIFAR10-DVS The CIFAR10-DVS dataset [6] encompasses 10,000 event-based images, with a 128×128 resolution and 10 classes, derived from the CIFAR-10 dataset via an event-based sensor. This dataset is partitioned into 9,000 training samples and 1,000 test samples.

N-Caltech101 The N-Caltech101 dataset [11] incorporates 8,831 event-based images, with a 180×240 resolution and 101 classes, generated from the original Caltech101 dataset through an event-based sensor.

N-Cars The N-Cars dataset [12] contains 24,029 event-based images, with a 100×120 resolution and two classes (car and background). The training set comprises 7,940 cars and 7,482 background samples, whereas the test set includes 4,396 cars and 4,211 background samples.

Action Recognition The Action Recognition dataset [9] encompasses 10 classes at a 346×260 resolution, including arm-crossing, getting up, jumping, kicking, picking up, sitting down, throwing, turning around, walking, and waving. The dataset comprises 30 recordings per class, averaging 5 seconds of actual action. Because of the dataset's limited size, the raw data is converted into 4,670 frame images utilizing the Surface of Active Events (SAE) [10] encoding technique.

ASL-DVS The ASL-DVS [1] dataset constitutes a comprehensive 24-class collection of handshape recordings captured under authentic conditions. Its 24 classes correspond to 24 letters (A-Y, excluding J) from American Sign Language (ASL). Five subjects were instructed to exhibit various static handshapes about the camera, introducing natural variability into the dataset. Each letter contains 4,200 samples, culminating in a total of 100,800 samples,

with each sample enduring approximately 100 milliseconds.

The above datasets are specifically designed for neuromorphic computing. In the pre-processing stage, the images are resized to a resolution of 224×224 in order to match the input dimensions of the model. Neuromorphic datasets are generally small in size and prone to overfitting. Herein, we applied the data augmentation techniques that are specifically designed for neuromorphic datasets[7] before loading the data. This helps to improve the generalization ability of the model by preventing it from overfitting to the specific characteristics of the training data.

1.2. Training Hyperparameters

The quantization clip-floor-shift (QCFS) [2] method is used during the ANN-to-SNN conversion. The QCFS function can be expressed by:

$$\boldsymbol{a}^l = f\left(\boldsymbol{a}^{l-1}\right) = \frac{\lambda^l}{L}\,\mathrm{clip}\left(\left\lfloor \frac{\boldsymbol{W}^l \boldsymbol{a}^{l-1} L}{\lambda^l} + \frac{1}{2} \right\rfloor, 0, L\right) \tag{1}$$

where L denotes the ANN quantization step and λ^l is the trainable threshold of the outputs in ANN layer l, which is mapped to the threshold θ^l in SNN layer l. In our work, the quantization steps L is set to 16 and the initial threshold θ is set to 3.

During the pre-training of the ANN, the AdamW optimizer was utilized and the model was trained for 300 epochs. The CosineLRScheduler [8] algorithm was employed to schedule the learning rate according to a cosine function, decreasing it as training progressed. A weight decay of 0.05 was also applied as a regularization technique. The CosineLRScheduler algorithm modulates the learning rate according to a cosine function, decreasing it as training progresses. These training parameters were chosen in order to optimize the performance of the ANN during the pre-training phase.

In the following experiments, the weights of the model without masking were fine-tuned for an additional 100 epochs with various masking ratios, using the same training parameters as in the pre-training phase. To reduce the influence of randomness on the accuracy of the model during inference, we conducted 10 inferences with different seeds (0, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000) and calculated the mean and variance of the accuracy across these 10 runs. This was done for all cases in order to provide a more robust comparison and analysis. Averaging the results of multiple inferences helps to mitigate the influence of random fluctuations on the accuracy of the model and allows for a more reliable assessment of its performance. The details of hyperparameters can be found in Tab. 1.

Dataset	Optimizer	Epoch	lr	Batch Size
CIFAR-10	Adamw	300	1e-4	64
CIFAR-100				64
ImageNet				64
CIFAR10-DVS				16
N-Caltech101				16
N-Cars				64
Action Recognition				16
ASL-DVS				16

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

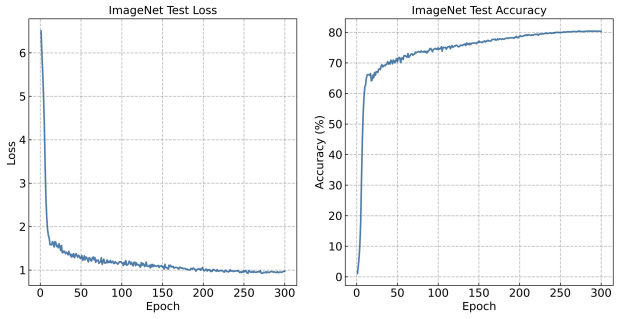

Figure 1: Test loss and test accuracy on the ImageNet dataset.

1.3. Training Curves

In Fig. 1, We present the test loss and accuracy of the proposed MST in the ANN pre-training process on the Imagenet dataset. In order to achieve optimal performance, the ANN is trained for 300 epochs. As shown in Fig. 1, the curve of test loss demonstrates a clear convergence, which indicates that the ANN has reached a steady state.

2. Effect of Masking Ratios and Time Steps on Accuracy

In Fig. 2, we analyze the effect of different time steps on the accuracy of the Masked Spiking Transformer (MST) model, which employs the Random Spike Masking (RSM) method with different masking ratios, on the CIFAR-10, CIFAR-100, and ImageNet datasets. To mitigate the effect of random seed on the accuracy, each masking ratio scenario was executed 10 times. The dark line in the figures represents the mean accuracy of these 10 trials, while the lighter-shaded area signifies the accuracy variance.

To clearly illustrate the difference in accuracy among different masking ratios, the time step in the figures commences at 64 for the CIFAR-10 and CIFAR-100 datasets and at 128 for the ImageNet dataset. As the time step increases, the accuracy initially shows a rapid increase fol-

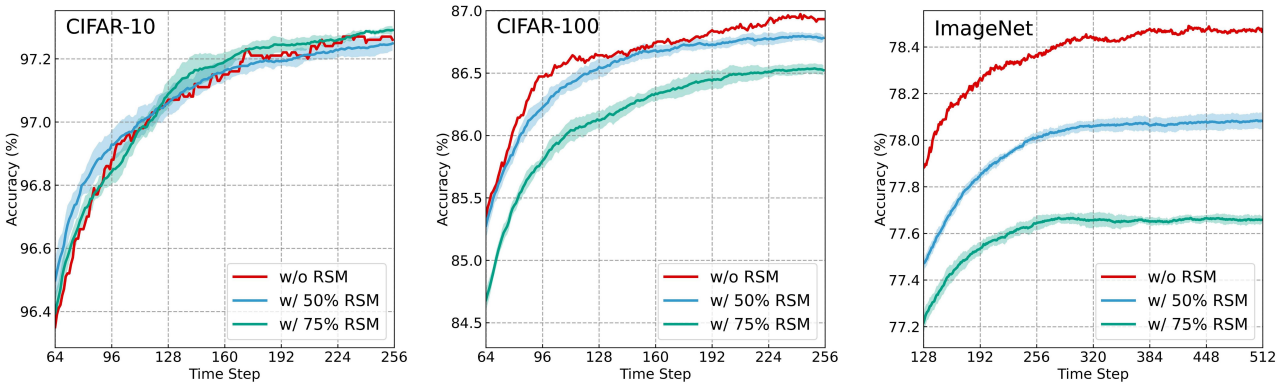

Figure 2: Comparision of the accuracy with increasing time steps at different masking ratios.

lowed by a period of plateauing.

In terms of the impact of masking ratios on accuracy, the general trend shows that accuracy decreases as the masking ratio increases. However, this effect also depends on the complexity of the task. For simpler tasks such as CIFAR-10, the influence of the masking ratio on accuracy is relatively small, whereas for more complex tasks like CIFAR-100 and ImageNet, a substantial accuracy gap occurs between different masking ratios. This observation suggests the presence of more redundant spikes on the CIFAR-10 dataset, which can be pruned without compromising performance.

3. The Impact of the RSM Method on Firing Rate in the Self-Attention Module

We investigate the firing rates of Query, Key, Attention, and Value components within the self-attention module, examining their difference across distinct blocks. Our RSM method is applied to the Query, Key, and Attention components, incorporating various masking ratios. As demonstrated in Fig. 4, the utilization of our RSM method effectively decreases the firing rate of the respective module. Consequently, this reduction leads to a diminished number of spikes within the network, ultimately contributing to a decrease in energy consumption. By incorporating the RSM method, we demonstrate the potential to optimize energy efficiency in the self-attention module without compromising network performance.

4. The Impact of the RSM Method on the Number of Spikes in the Whole Model

Fig. 3 depicts the effect of different masking ratios on the total number of spikes for each block in the overall model, evaluated using the CIFAR-100 dataset. The results show that there is a significant correlation between the increase

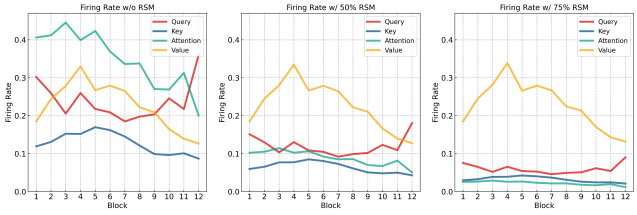

Figure 3: Firing rate variation for different components within Self-Attention module across blocks.

in the masking ratio and the decrease in the total number of spikes. For example, at a masking rate of 50%, the total number of spikes in the model decreased by 23.8% compared to the unmasked model. Furthermore, when the masking rate reached 75%, the total number of spikes in the model was further reduced by 31.4% relative to the unmasked counterpart. Importantly, this reduction resulted in a minimal loss of precision of only 0.37%. These findings suggest that our RSM method is an effective way to reduce the number of spikes in the overall model, which has the potential to improve efficiency and performance.

5. Exploration of the Utility of the RSM Method in Other Modules

In the main text, the RSM approach is applied to the query (Q), key (K), and value (V) components of the self-concern (SA) module, as well as the multilayer perceptron (MLP) module of the MST model. To further evaluate the effectiveness and generality of this approach, we extended the RSM method to other modules of the MST model. This extension allows us to explore the potential benefits offered by RSM in various aspects of the MST model and to evaluate its ability to improve efficiency and performance.

Module	SC1	Q	K	A	V	soft(A)*V	Proj	SC2	MLP1	MLP2	Accuracy (%)
Masking Ratio (%)	0	0.5	0.5	0.5	0	0	0	0	0	0	86.7
	0	0.5	0.5	0.5	0.5	0	0	0	0	0	83.1
	0	0.5	0.5	0.5	0.5	0.5	0	0	0	0	80.5
	0	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	76.7
	0	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0.1	75.2
	0	0.5	0.5	0.5	0.5	0.5	0.5	0	0.1	0	75.7
	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	74.46
	0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	62.36
	0	0.5	0.5	0.5	0.5	0.5	0.5	0	0.5	0	68.67
	0	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0.5	67.91

Table 2: Comparative analysis of RSM application on different components in the MST.

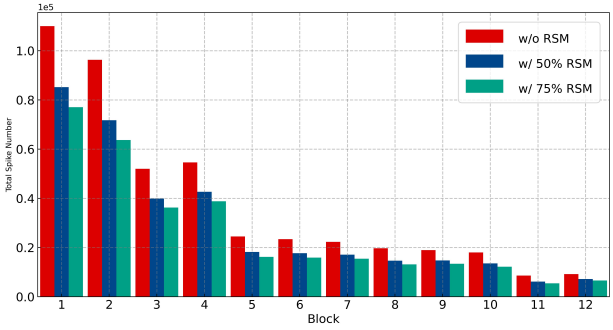

Figure 4: The number of spikes variation of the overall model with different masking ratios.

Tab. 2 presents the results of applying the RSM method to several components of the MST model, including Query (Q), Key (K) and Value (V), Attention (A), Softmax (Attention)∗ Value (soft(A)∗ V), projection layer (Proj), and multilayer perceptron (MLP) modules (MLP1 and MLP2). In addition, the RSM method is used for the first and second layer shortcut (SC1 and SC2) modules within the Swin Transformer block, which serve as inputs to the self-attention and MLP modules, respectively. These modules are followed by a layer of IF neurons whose output spike matrices are subjected to random mask pruning.

Tab. 2 shows the different effects of RSM on the accuracy of each module. Specifically, modules Q, K, and A show relatively small accuracy degradation after masking, while the other modules, especially the MLP module, show greater sensitivity to changes in the masking ratio. This finding suggests that reducing energy consumption while maintaining accuracy may require adjusting the masking ratio according to each module's sensitivity to this process. Therefore, this emphasizes the importance of considering the unique characteristics of each module when applying RSM to effectively achieve a balance of energy efficiency and performance.

6. The Application of the RSM Method on Neuromorphic Datasets

Fig. 5 illustrates the fluctuations of the accuracy corresponding to different masking ratios on the N-Caltech101 and CIFAR10-DVS datasets. The experimental results show similarities to those observed in the static dataset. First, in the SA and MLP modules, the accuracy decreases as the masking ratio increases. However, the sensitivity of these two modules to changes in masking rate is different. In particular, the accuracy of the SA module remains relatively stable over a specific range of masking ratios, while the accuracy of the MLP module decreases more rapidly and shows a high sensitivity to changes in masking ratios. These findings provide valuable insights into the behavior of the different modules when applied to the RSM method, thus indicating the development of strategies to optimize energy efficiency and performance while minimizing the impact on accuracy. Furthermore, these results show that the RAM method is effective for both static and neuromorphic datasets.

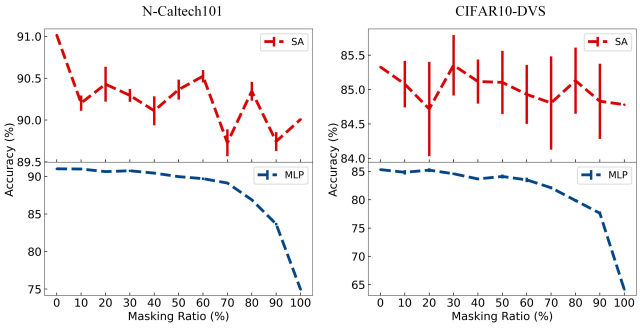

Figure 5: The impact of the RSM method on the performance when applied to the SA and MLP modules on N-Caltech101 and CIFAR10-DVS datasets, respectively, under different masking ratios.

7. Robustness of the RSM Method

In addition to accuracy and power efficiency, robustness is also a key aspect to evaluate the proposed RSM method. To investigate the robustness of the MST model under different masking rates, different levels of pepper noise are introduced in the input images and the ability of the model to resist this noise under different masking rates is compared. From Fig. 6, we can see that the accuracy of the model decreases as the pepper noise density increases. However, we also observe that the model exhibits stronger noise robustness at higher masking ratios.

More specifically, the accuracy gap between the model with and without noise becomes smaller as the masking ratio increases, indicating that the model with a larger masking ratio exhibits higher noise robustness. These results indicate that the proposed RSM method not only reduces the power consumption of the model, but also enhances its noise immunity. As a result, this enhanced robustness makes the model more suitable for deployment in noisy real-world environments.

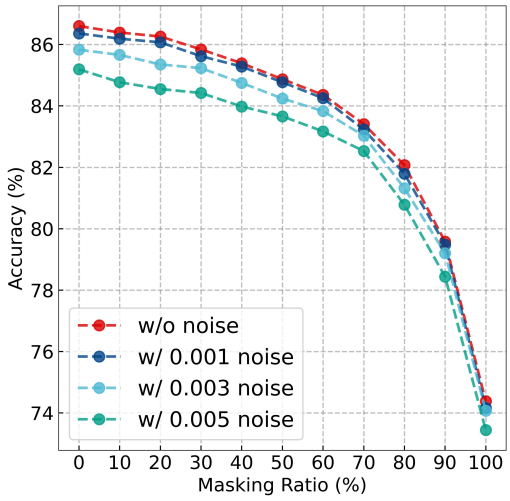

Figure 6: The impact of various pepper noise levels and masking ratios on model robustness. The gap in accuracy between the model with and without noise becomes smaller as the masking ratio increases.

References

[1] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based object classification for neuromorphic vision sensing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 491–501, 2019. 1

[2] Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN Conversion for High-accuracy and Ultra-low-latency Spiking Neural Networks. In *International Conference on Learning Representations*, 2021. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 1

[4] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. 1

[5] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[6] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Frontiers in Neuroscience*, 11, 2017. 1

[7] Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic Data Augmentation for Training Spiking Neural Networks. *arXiv preprint arXiv:2203.06145*, 2022. 2

[8] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 2

[9] Shu Miao, Guang Chen, Xiangyu Ning, Yang Zi, Kejia Ren, Zhenshan Bing, and Alois Knoll. Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection. *Frontiers in neurorobotics*, 13:38, 2019. 1

[10] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 1

[11] 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. 1

[12] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018. 1