Supplementary Material for AAAI 2025 NO.3832

Yimeng Shan
yimengshan2001@gmail.com

Abstract

Although our article has been accepted for publication, the limited length of the main text prevents a comprehensive presentation of certain experimental designs and additional explorations. Therefore, we have prepared this appendix to supplement our article.

APPENDIX

A Training details

A.1 Network structure

In the experiment detailed in Chap. 4 of the primary text, we employed MS-ResNet networks with varying numbers of layers on the ImageNet-1K dataset. Additionally, we utilized an 8-layer VGG architecture on the mainstream neural morphology dataset, as depicted in Tab. 1. It is worth noting that in both the SMA-ResNet and SMA-VGG architectures, our SMA module closely follows the convolutional layer, and an SMA module is added after each residual connection. This suggests that the actual configuration of the second module in SMA-ResNet18 consists of Conv-SMA-Conv-Conv-SMA-Conv, while the actual structure of the second module of SMA-VGG is Conv-SMA-MaxPool. For clarity, we omitted LIF neurons and BatchNorm in Tab. 1.

A.2 Training strategy

The experiments conducted on the neuromorphic datasets in this article were all carried out using $4 \times$ 3090GPUs, while experiments on Imagenet-1K were performed using 6/8/10 $\times$ 3090D GPUs depending on the ResNet layers. Tab. 2 presents the hyperparameters necessary for training each dataset, while Tab. 3 delineates the membrane constants of LIF neurons for both the neuromorphic dataset and Imagenet-1K dataset.

A.3 Equipment details

This research utilized three sets of experimental apparatus, the configurations of which are detailed in Tab. 4. Both Equipment 2 and Equipment 3 were leased from the AutoDL cloud computing platform.

B Hyperparameter selection

For general attention modules, particularly those based on SE modules, the hyperparameters within the modules are crucial, as they frequently have the potential to influence the performance of attention mechanisms to a certain extent.

B.1 SMA hyperparameter selection

The SMA module features only two hyperparameters: the channel reduction ratio and the time reduction channel. As in Chap. 4 of the primary text, we undertook experiments using the DVS128 Gesture dataset to ascertain the optimal reduction ratios in both the channel and time dimensions. Initially, we integrated SMA solely along the channel dimension into the network for experimentation,

Main paper is: Shan Y, Zhang M, Zhu R, et al. Advancing Spiking Neural Networks towards Multiscale Spatiotemporal Interaction Learning[J]. arXiv preprint arXiv:2405.13672, 2024.

Table 1: Structures for SMA-VGG. "CR" and "TR" denote the compression ratios of the C-MSE and T-MSE modules in SMA, respectively.

Block	SMA-VGG	Output Size		
		DVS128 Gesture	CIFAR10-DVS	N-Caltech101
1	3×3,64	128×128	128×128	180×240
	MaxPool(2,2,0,1)	64×64	64×64	90×120
2	3×3,128	32×32	32×32	45×60
	SMA(CR=4,TR=4)			
	MaxPool(2,2,0,1)			
3	3×3,256	16×16	16×16	22×30
	SMA(CR=4,TR=4)			
	MaxPool(2,2,0,1)			
4	3×3,512	8×8	8×8	11×15
	SMA(CR=4,TR=4)			
	MaxPool(2,2,0,1)			
5	3×3,512	4×4	4×4	5×7
	SMA(CR=4,TR=4)			
	MaxPool(2,2,0,1)			
FC-1	AveragePool	1×1	1×1	1×1
	FC(2048)			
	Dropout(0.5)			
FC-2	FC(1024)	1×1	1×1	1×1
	Dropout(0.5)			
FC-3	FC(11/10/101)	1×1	1×1	1×1

with the results depicted in Fig. 1(a). Notably, setting the reduction ratio in the channel dimension to 4 yielded the most favorable outcomes. Building upon this finding, we further investigated the optimal reduction ratio in the time dimension, as illustrated in Fig. 1(b), where the optimal setting remained at 4.

The insights from Fig. 1 highlight the critical role of hyperparameter selection within the attention module, as an incorrect choice can detrimentally impact model performance. Upon close examination of the findings in Fig. 1, it becomes evident that excessive compression may blur the importance of information within the attention module, while insufficient compression may prevent the model from achieving the correct balance of importance.

B.2 AZO hyperparameter selection

The AZO regularization method utilizes two hyperparameters: the Replacement Time Ratio(RTR) and the Replacement Channel Ratio(RCR). Given the short simulation timesteps in our experiment, the influence of the timestep on the AZO effect appears to be considerably less significant than the number of channels, a notion supported by Fig. 3. Given the diverse nature of the datasets involved—CIFAR10-DVS and N-Caltech101, which are converted datasets, and DVS128 Gesture, which represents natural neural morphology—we investigated the optimal RCR for the DVS128 Gesture and CIFAR10-DVS datasets, setting the RTR at 4, as illustrated in Fig. 2. Following this, we further explored the optimal RTR for the DVS128 Gesture dataset, as depicted in Fig. 3.

We posit that the distinct data formats and acquisition methods necessitate uniquely suitable hyperparameter settings for each dataset when using the AZO regularization method. Due to the extensive data volume, we refrained from conducting an exhaustive exploration of optimal hyperparameters on the N-Caltech101 dataset, opting instead to provide a set of parameters that demonstrated improved performance, as detailed in Tab. 5.

B.3 Comparison of accuracy between LIF based and ReLU based SMA

In the Discussion section of the primary text, we conducted extensive experiments utilizing the LIF version of the SMA module. The discrete nature of signals in SNNs facilitated easy visualization and analysis of attention modules based on spike count or firing rate—a method widely adopted in numerous studies. As illustrated in Tab.6, our experiments showcased that while the ReLU version of the SMA module exhibits slightly superior performance, the disparity in performance between it

Table 2: The hyperparameters on each dataset. In the 125-layer structure, we incorporate AZO regularization methods following each SMA, fine-tuning them with hyperparameters as indicated in parentheses.

Model	Dataset	Name	Value
SMA-VGG	DVS128 Gesture	lr	1e-4
		T	16
		batch_size	9
		train_epoch	200
		loss_function	MSE_loss
		optimizer	AdamW(momentum_decay=1e-3)
	CIFAR10-DVS	lr	1e-3
		T	10
		batch_size	24
		train_epoch	200
		loss_function	MSE_loss
		optimizer	Adam
	N-Caltech101	lr	1e-3
		T	14
		batch_size	6
		train_epoch	300
		loss_function	TET_loss
		optimizer	NAdam(momentum_decay=1e-3)
SMA-ResNet18/34	Imagenet-1K	lr	0.1
		T	6
		batch_size	384/256
		train_epoch	125
		loss_function	Label_smoothing(0.1)
		optimizer	SGD(Momentum=0.9, weight_decay=1e-4)
SMA-ResNet104	Imagenet-1K	lr	0.1(1e-3)
		T	5
		batch_size	224
		train_epoch	125(50)
		loss_function	Label_smoothing(0.1)
		optimizer	SGD(Momentum=0.9, weight_decay=1e-4(1e-5)

Table 3: Neuronal configuration parameters of LIF.

Datasets	Parameter	Value
Neuromorphic Datasets	$\boldsymbol{U}_{threshold}$	1.0
	$\boldsymbol{U}_{reset}$	0.0
	τ	2.0
	Surrogate Gradient Function	ATan()
Imagenet-1K	$\boldsymbol{U}_{threshold}$	0.5
	$\boldsymbol{U}_{reset}$	0.0
	τ	4.0
	Surrogate Gradient Function	$sign(\lvert \boldsymbol{I}_t^n - \boldsymbol{U}_{threshold} \rvert \leq \frac{1}{2})$

Table 4: Equipment configuration used in the experiment.

No	CPU	GPU	Memory	CUDA	Pytorch	OS
1	Gold 6133 × 2	RTX 3090 × 4	128GB	12.1	2.0.1	Ubuntu 18.04
2	Gold 6430	RTX 4090 × 8	960GB	12.1	2.1.0	Ubuntu 22.04
3	AMD EPYC 9754	RTX 4090D × 10	600GB	12.1	2.1.0	Ubuntu 22.04

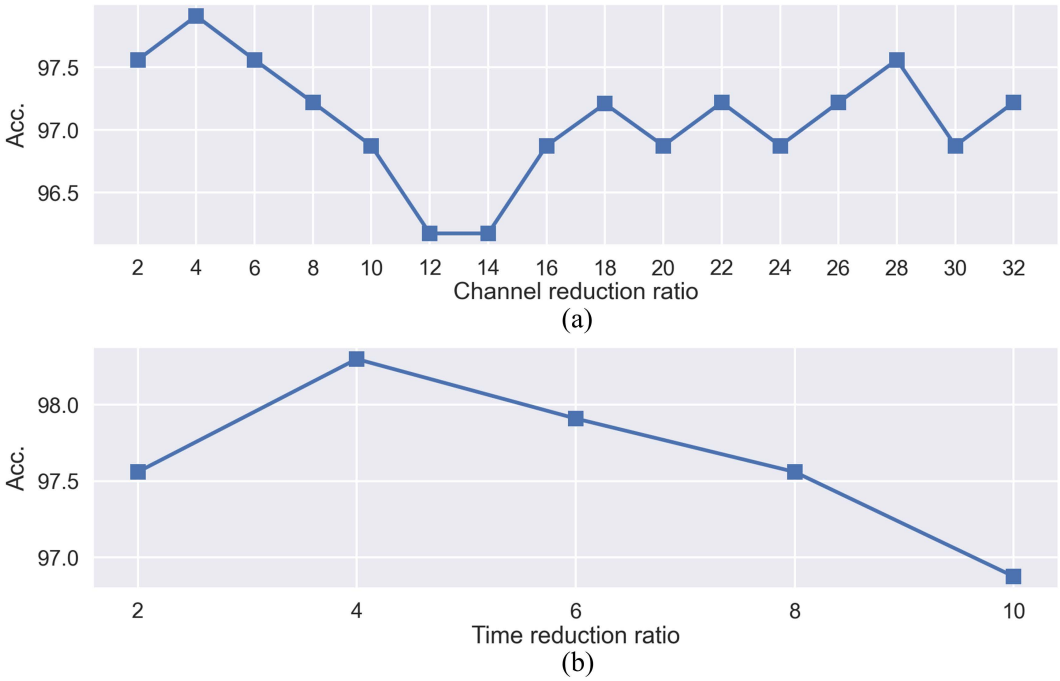

Figure 1: Explore the best reduction ratio. The accuracy of the VGG network without any attention added is 97.92% . (a) Only channel dimension attention has been added in. (b) Explored the time dimension reduction ratio based on a channel reduction ratio of 4.

and the LIF version is minimal. Notably, even on the DVS128 Gesture dataset, both versions yielded identical effects. Hence, it is reasonable to posit that the LIF version and the ReLU version of the SMA module share the same attention mechanism, affirming our discussion in the primary text.

C Further discussion

C.1 Optimizing the implementation of AZO

Clearly, the efficiency of implementing AZO regularization in Python using nested display loops is quite low. Thus, we propose a method for AZO implementation that leverages the tensor parallelism of Python and Numpy. As illustrated in Tab. 7, we compared the time required for a single training iteration between the parallel and the loop-based versions of AZO. We found that the loop version offers a significant efficiency advantage, establishing a solid foundation for the widespread adoption of AZO-like regularization algorithms.

Table 5: The best settings for AZO.

AZO hyperparameters	**DVS128 Gesture**	**CIFAR10-DVS**	**N-Caltech101**
Best RCR	10/12	4	24
Best RTR	4	4	4

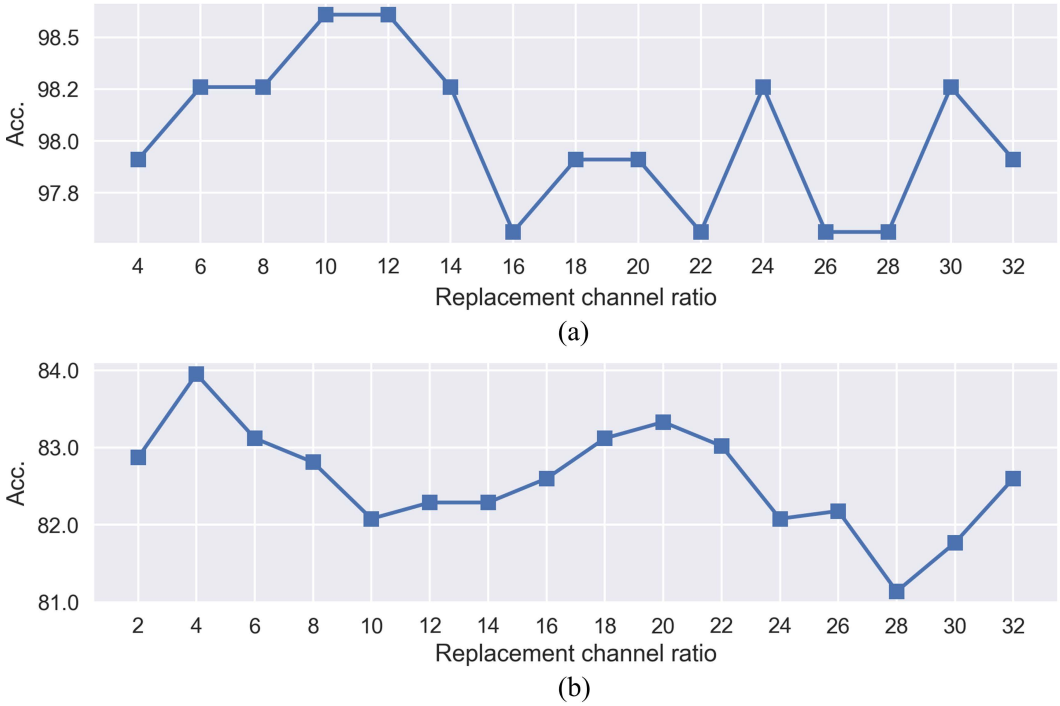

Figure 2: Explore the best RCR. The RTR for all experiments is 4. (a) is an experiment on the DVS128 Gesture dataset, and (b) is an experiment on the CIFAR10-DVS dataset.

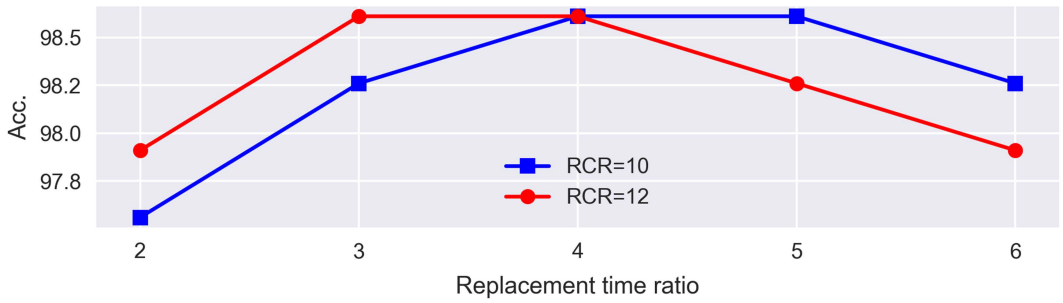

Figure 3: Explore the best RTR while the RCR has been determined.

C.2 AZO effect

As elucidated in Section 3.3 of the primary text, the incorporation of the AZO algorithm is posited to enhance the generalizability of the model. This postulate is grounded in the algorithm's methodology, which perturbs the hidden unit activations with noise during the training of pseudo-sets. To empirically substantiate this hypothesis, we employed the N-Caltech101 dataset—a dataset of considerable complexity, meticulously selected from among a consortium of smaller datasets. Fig. 4 provides a graphical representation of the empirical results, contrasting the model's performance on training and

Table 6: Comparing the effects of LIF neurons and ReLU in SMA

Type	**DVS128 Gesture**		**CIFAR10-DVS**		**N-Caltech101**	
	SMA-C	SMA	SMA-C	SMA	SMA-C	SMA
ReLU	97.9	98.3	82.3	83.1	82.7	83.7
LIF Neuron	97.9	98.3	81.8	82.6	82.5	82.9

Table 7: Comparison of Efficiency between Tensor Parallel and Explicit Loop Implemented AZO Regularization Methods.

Version	Training time(s)		
	Dvs128 Gesture	CIFAR10-DVS	N-Caltech101
Explicit Loop	55.29	802.54	755.07
Tensor Parallelism	4.35	68.61	44.04

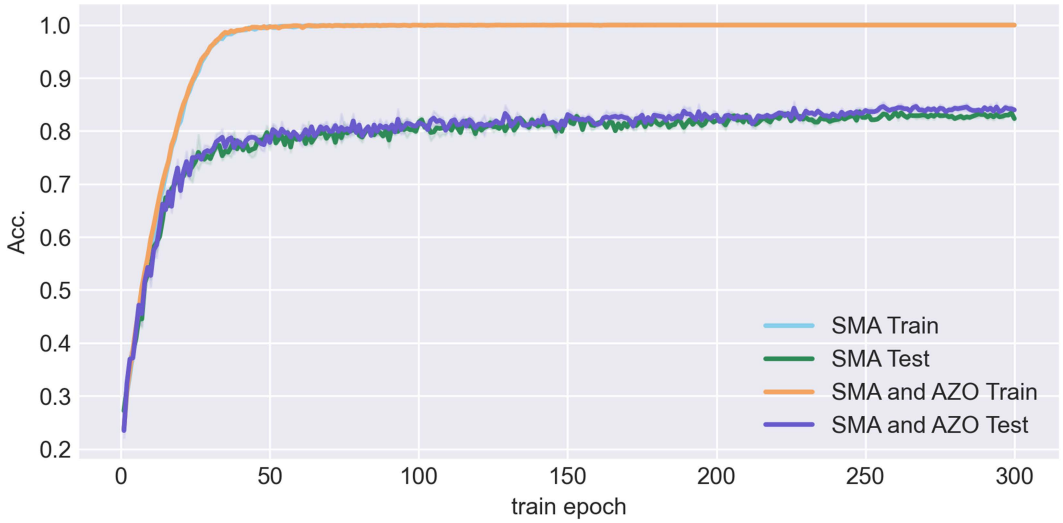

Figure 4: Comparison of generalization errors for five runs.

testing accuracies both pre and post the integration of the AZO algorithm. The results unambiguously indicate that the AZO algorithm materially amplifies the generalization capabilities of the model.