STAA-SNN: Spatial-Temporal Attention Aggregator for Spiking Neural Networks

Supplementary Material

7. Datasets Details and Augmentation

CIFAR-10. The CIFAR-10 dataset serves as a widely recognized benchmark for images, all with dimensions of 32×32. This dataset contains 10 distinct classes, encompassing various common objects such as airplanes, cars, birds, and cats. It is frequently used as a standard for assessing the effectiveness of image classification algorithms, presenting a diverse array of visual challenges. In our approach, we implement data augmentation techniques, including cropping, horizontal flipping, and cutout, throughout the entire training dataset. In addition, during the training phase, we introduce random augmentation by selecting two strategies from the contrast enhancement, rotation, and translation. These adjustments add robustness to the model and enhance its ability to generalize to diverse visual scenarios.

CIFAR-100 CIFAR-100 [35] represents an extension of the CIFAR-10 dataset and is designed to address more complex classification tasks. It comprises 50,000 training images and 10,000 test images, all standardized at 32×32 dimensions. The dataset comprises 100 classes, each of which belongs to one of the 20 superclasses. CIFAR-100 offers a more demanding challenge compared to CIFAR-10, making it a more suitable benchmark for evaluating the effectiveness of models in classification tasks. The data augmentation strategy employed for CIFAR-100 aligns with that of CIFAR-10.

ImageNet. ImageNet contains a training dataset of 1.3 million images across 1,000 categories, along with an additional 50,000 images for validation. Compared to the CIFAR-10/100 datasets, ImageNet presents a larger and more complex collection of images, providing a more robust benchmark for evaluating model generalization and learning capabilities. In our experiments, we employ the data augmentation techniques outlined in [29]. Images are randomly cropped from either their original version or a horizontally flipped version to a size of 224×224 pixels, followed by data normalization. For testing samples, images are resized to 224×224 pixels and subject to center cropping, after which data normalization is also applied.

CIFAR10-DVS. CIFAR10-DVS, introduced in [40], represents one of the largest visual neuromorphic datasets currently available. It comprises 10,000 event streams, each with a size of 128×128, derived from the frame-based CIFAR-10 images using a dynamic vision sensor (DVS). The dataset comprises 10 categories, each containing 1,000 images. During the training phase, the dataset is divided into training and testing datasets at a ratio of 9:1. In the preprocessing stage, the training dataset undergoes random horizontal flipping, followed by the random selection of an augmentation technique such as cropping, translation, rotation, cutout, or erasing. These techniques are used to enhance the diversity of the training dataset and strengthen the model's generalization capabilities[20, 21].

DVS128 Gesture. The DVS128 Gesture dataset, as presented in [4], is specifically curated for gesture recognition tasks. It comprises 1,176 training images and 288 testing images, each with dimensions of 128×128. The dataset features 11 different gestures performed by 29 subjects under 3 illumination conditions, adding complexity to the recognition task. This dataset serves as a valuable resource for evaluating models designed for gesture recognition in dynamic and varying conditions. To enhance the dataset, each frame undergoes cutout and mixup operations, and random augmentations such as rotation, shear, and translation are utilized. These techniques aim to enrich the dataset and improve the model's capacity to generalize across various gesture recognition scenarios.

8. Experimental Settings

All code implementations are based on the PyTorch framework. Experiments were conducted on a single RTX 3090 GPU for all datasets except ImageNet, which was trained using a configuration of eight RTX 4090 GPUs. In the experimental setup, we utilized the SGD optimizer with a momentum of 0.9 across all datasets and employed the CosineAnnealing learning rate adjustment strategy.

CIFAR-10/100. For CIFAR-10/100, we configured the initial learning rate to 0.1, batch size to 128, number of training epochs to 500, α to 2, and β to 0. Additionally, the dimension scaling factor of the hidden representations of W_q in the GC is set to 4, while the dimension scaling factor for the hidden representations in the SA is set to 16.

CIFAR10-DVS. In the case of CIFAR10-DVS, the initial learning rate is set to 0.01, batch size to 8, and the number of training epochs to 200. Furthermore, α is set to 2, and

β to 0.1. The scaling factor of the dimension size of the intermediate representations in the GC is set to 4, and in the SA it is set to 16.

ImageNet. For the ImageNet dataset, the initial learning rate is set to 0.1, with a batch size of 64 and a total of 350 training epochs. Furthermore, the parameters α and β are configured to 2 and 0.1, respectively. The scaling factors for the intermediate dimension in GC and SA are set to 4 and 16, respectively.

DVS128 Gesture. For the DVS128 Gesture dataset, the learning rate is initialized at 0.01, and the batch size is set to 8. The model undergoes 500 training epochs, with α and β values set to 2 and 0.1, respectively. The scaling coefficients for the two intermediate dimensions in GC and SA are set to 4 and 16.

9. Supplemental Experiments

9.1. Impact of Intermediate Dimension Scaling Coefficients in GC.

The GC module incorporates multiple 1×1 convolutions. Keeping the number of channels C consistent for the last two convolutions within the module would notably escalate computational costs. To ensure the lightweightness of the module, a scaling coefficient r is incorporated in the GC to compress features. This efficient approach reduces the module's parameter count from $C \cdot C$ to $2 \cdot C \cdot C/r$.

Evaluation involving different values of r is conducted on the ResNet-20 architecture using a time step of 4 on the CIFAR-10 dataset. The corresponding test accuracies for varied r values are detailed in Table 5, revealing optimal performance when r is set to 4.

Dataset	Architecture	r	Accuracy
CIFAR-10	ResNet-20	1	94.93%
		2	94.85%
		4	95.03%
		8	94.81%
		16	94.72%

Table 5. Impact of different scaling coefficients r for intermediate feature dimensions in the GC module on the CIFAR-10 dataset.

9.2. Pooling Method Selection in SA.

Evaluation of various pooling methods for image classification was conducted, and the outcomes are detailed in Table 6. The analysis revealed that the overall performance was relatively better, and the network could converge effectively when utilizing average pooling. In contrast, the incorporation of max pooling posed significant challenges, including difficulties in training and the potential failure to converge. Moreover, the utilization of max pooling substantially extended the training time.

Pooling Method	Timestep	Accuracy
+1 MaxPool	1	-
	2	93.86%
+1 AvgPool	1	91.27%
	2	93.96%
+1 AvgPool, +1 MaxPool	1	89.60%
	2	94.44%
+2 AvgPool	1	92.91%
	2	94.27%

Table 6. Ablation Study of Pooling Method Combinations in SA with ResNet-20 on CIFAR-10.

Further experimentation involving different combinations of average pooling demonstrated that the use of two average pooling layers yielded superior performance compared to a single layer. This observation validates the integration of the α parameter in SA, set to 2. Consequently, to balance both accuracy and training speed, our final selection involved a combination scheme utilizing multiple average pooling layers.

10. Analysis of Computation Efficiency

In ANNs, each operation involves a multiplication and accumulation (MAC) process. The total number of MAC operations (#MAC) in an ANN can be calculated directly and remains constant for a given network structure. In contrast, spiking neural networks (SNNs) perform only an accumulation computation (AC) per operation, which occurs when an incoming spike is received. The number of AC operations can be estimated by taking the layer-wise product and sum of the average spike activities, in relation to the number of synaptic connections.

$$\begin{cases} \#\text{MAC} = \sum_{l=1}^{L} (\#\text{MAC}_l) \\ \#\text{AC} = \sum_{l=2}^{L} (\#\text{MAC}_l \times a_l) \times T \end{cases} \tag{15}$$

Here, a_l represents the average spiking activity for layer l. The first, rate-encoding layer of an SNN does not benefit from multiplication-free operations and therefore involves MACs, while the subsequent layers rely on ACs for computation.

The energy consumption E for both ANN and SNN, accounting for MACs and ACs across all network layers, is given by:

Arch.	Res.	T	ACs(G)	MACs(G)	FLOPs(G)	Params(M)	Energy(mJ)
ResNet20	224x224	4	5.38	2.82	42.62	13.20	17.814
VGG13	224x224	4	3.45	13.56	45.07	11.17	65.481
ResNet20	32x32	4	0.10	0.06	0.87	12.69	0.366
VGG13	32x32	4	0.05	0.28	0.92	10.67	1.333

Table 7. Table of computational consumption for different models on CIFAR-100 and ImageNet.

$$\begin{cases} E_{SNN} = \#\text{MAC}_1 \times E_{MAC} + \#\text{AC} \times E_{AC} \\ E_{ANN} = \#\text{MAC} \cdot E_{MAC} \end{cases} \tag{16}$$

Based on previous studies in SNN research [10, 68], we assume that the operations are implemented using 32-bit floating-point (FL) on a 45 nm 0.9V chip [31], where a MAC operation consumes 4.6 pJ and an AC operation consumes 0.9 pJ. This comparison suggests that one synaptic operation in an ANN is roughly equivalent to five synaptic operations in an SNN. It is important to note that this estimation is conservative, and the energy consumption of SNNs on specialized hardware designs can be significantly lower, potentially reduced by up to 12× to 77 fJ per synaptic operation (SOP) [49]. We conduct energy consumption tests on different models using CIFAR-100 and ImageNet dataset, and the specific results are recorded in Table 7.