Efficient ANN-Guided Distillation: Aligning Rate-based Features of Spiking Neural Networks through Hybrid Block-wise Replacement

Supplementary Material

1. Impact of timesteps

Tab. A shows the ablation study of various timesteps, where a larger T improves performance. Meanwhile, our method could also optimize memory and time costs when T is larger (Fig. 4 in main text), which provides a good alternative to BPTT. Coupled with rate-based propagation, it supports the distillation of larger models with low computational costs for strong scalability potential. While our current results focus on standard benchmarks, our method's memory efficiency and computational benefits suggest that it can be extended to larger models under limited resources.

Table A. **Ablation Study on Timesteps.** "R" denotes ResNet. The teacher column lists the ResNet model along with its accuracy. The values for each timestep indicate both the accuracy and precision differences between the student and teacher models.

Dataset	Teacher	Student	T=2	T=4	T=6	T=8
CIFAR-100	R18 (79.95)	R18	77.06 (-2.89)	78.85 (-1.10)	79.40 (-0.55)	79.56 (-0.39)
CIFAR-100	R34 (81.90)	R19	81.44 (-0.46)	81.53 (-0.37)	81.71 (-0.19)	81.84 (-0.06)
CIFAR-10	R18 (96.92)	R18	95.19 (-1.73)	95.92 (-1.00)	96.14 (-0.78)	96.21 (-0.71)
CIFAR-10	R34 (97.24)	R19	96.56 (-1.68)	96.84 (-0.40))	96.96 (-0.28)	97.04 (-0.20)

2. Experiments with ViT backbones

In addition to ResNet, we have also validated the effectiveness of our method on ViT backbones. As shown in Table B, we conduct experiments on CIFAR-10/100 datasets using Spikingformer [1] as the backbone. It can be observed that our method consistently outperforms Vanilla knowledge distillation, demonstrating the generalizability of our approach across both CNN and Transformer architectures.

Table B. **Performance of ViT on CIFAR-10/100.** Spikingformer-4-256 denotes a model with 4 blocks and 256 feature dimensions. The values in parentheses represent the accuracy gap between the student and teacher models.

Dataset	Method	Teacher	Student	Timesteps	Top1-Acc (%)
CIFAR-100	RateBP	-	Spikingformer-4-256	4	76.15
CIFAR-100	Vanilla KD	ViT-S/16 (91.19)	Spikingformer-4-256	4	76.89 (-14.30)
CIFAR-100	Ours	ViT-S/16 (91.19)	Spikingformer-4-256	4	77.04 (-14.15)
CIFAR-10	RateBP	-	Spikingformer-4-256	4	93.74
CIFAR-10	Vanilla KD	ViT-S/16 (97.83)	Spikingformer-4-256	4	94.16 (-3.67)
CIFAR-10	Ours	ViT-S/16 (97.83)	Spikingformer-4-256	4	94.86 (-2.97)

3. Implement Details

Our method is implemented using PyTorch [2] and SpikingJelly [3]. All experiments are conducted on 8×NVIDIA GeForce RTX 3090 GPUs. Here are the implementation details for our method across various datasets and models, as presented in Tab. C.

For both CIFAR-10 and CIFAR-100 datasets, we utilized ResNet-18 and ResNet-19 as student models, with ResNet-18 and ResNet-34 as teacher models, respectively. The models achieved 96.92% and 97.24% accuracy on CIFAR-10, and 79.95% and 81.90% on CIFAR-100, respectively. All CIFAR-based models were trained for 300 epochs with an initial learning rate of 0.1, a weight decay of 5×10^{-4}, and a cosine learning rate scheduler, using SGD with momentum (SGDM) as the optimizer.

For ImageNet, we experimented with SEW-ResNet-34 and PreAct-ResNet-34 models, using ResNet-34 and ResNet-50 as the teachers. We employ pretrained ResNet models from the timm library for our experiments, achieving an accuracy of 76.32% with ResNet-34 and 80.10% with ResNet-50. The training setup for ImageNet used a higher learning rate of 0.2, with a weight decay of 2×10^{-5}, and followed the same cosine scheduler and SGDM optimizer over 100 epochs.

For the neuromorphic dataset CIFAR10-DVS, we employed ResNet-19 as both the student and teacher model, achieving a teacher accuracy of 83.60%. The training setup mirrored that of CIFAR-10, with a learning rate of 0.1, a weight decay of 5×10^{-4}, 300 epochs, a cosine scheduler, and SGDM.

4. Visualization results of class activation mapping

Class Activation Mapping (CAM) is a technique that highlights the regions in an image most relevant to a specific class prediction, offering insight into which features influence the model's decision.

Fig. A displays the CAM generated using the Grad-CAM method. When the SNN is applied in the ANN-guided distillation framework, the resulting CAMs closely resemble those of the ANN. This similarity indicates that the SNN's feature extraction and gradient backpropagation capabilities align with those of the ANN, validating that the SNN effec-

Table C. **Training settings for our method across various datasets and models.**

Dataset	Model	Tea-Model	Tea-Acc (%)	Learning Rate	Weight Decay	Epochs	Scheduler	Optimizer
CIFAR-10	ResNet-18	ResNet-18	96.92	0.1	5e-4	300	Cosine	SGDM
	ResNet-19	ResNet-34	97.24					
CIFAR-100	ResNet-18	ResNet-18	79.95	0.1	5e-4	300	Cosine	SGDM
	ResNet-19	ResNet-34	81.90					
ImageNet	SEW-ResNet-34	ResNet-34	76.32	0.2	2e-5	100	Cosine	SGDM
	PreAct-ResNet-34	ResNet-34	76.32					
	PreAct-ResNet-34	ResNet-50	80.10					
CIFAR10-DVS	ResNet-19	ResNet-19	83.60	0.1	5e-4	300	Cosine	SGDM

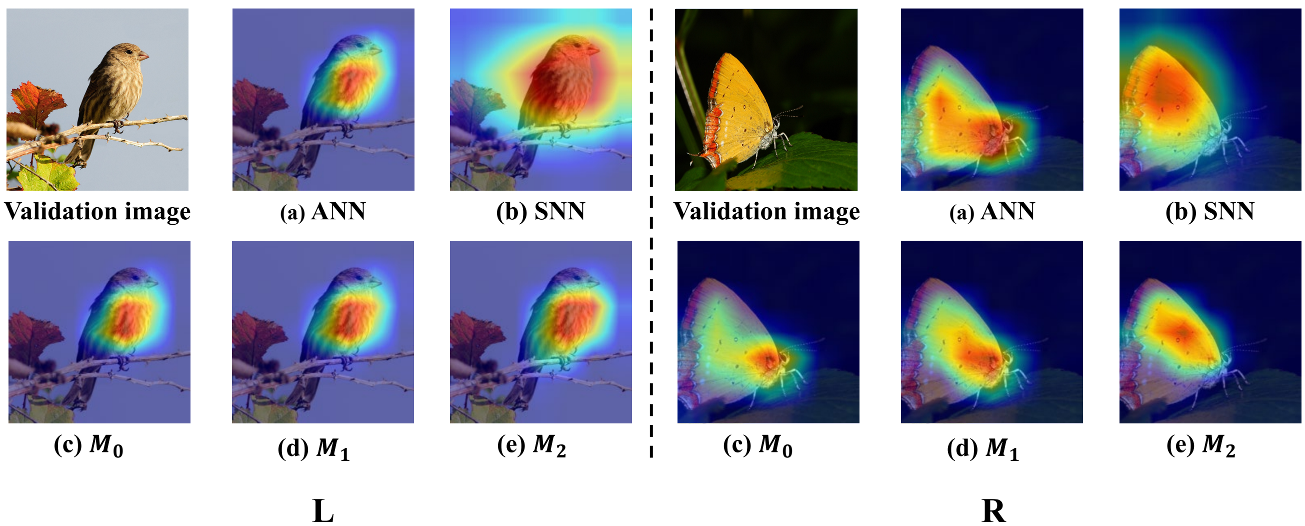

Figure A. **Visualizations of CAMs.** The visualization is based on the output layer of a ResNet-34 model trained on the ImageNet dataset. Panels L and R show the results for two randomly selected samples from the validation set. (a-b) depict the results for the ANN model and the SNN model trained with ANN-guided distillation. (c-e) show the results for the hybrid models M_0, M_1, and M_2.

tively learns from the ANN within this distillation framework. Observing the results from the hybrid model M_k, we can clearly see that the model's focus gradually shifts from ANN-like regions to more SNN-like areas. Overall, the CAM results illustrate the effectiveness of the ANN-guided SNN distillation framework.

References

[1] Zhou C, Yu L, Zhou Z, et al. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv:2304.11954*, 2023. 1

[2] Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 2019, 32. 1

[3] Fang W, Chen Y, Ding J, et al. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 2023, 9(40): eadi1480. 1