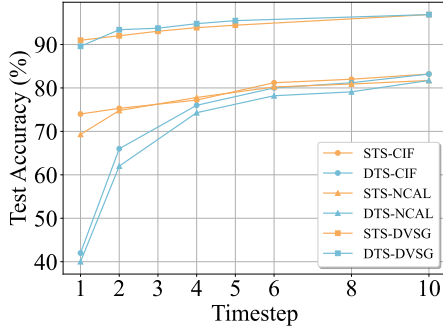


# Supplementary Materials: Towards Low-latency Event-based Visual Recognition with Hybrid Step-wise Distillation Spiking Neural Networks

Anonymous Author(s)

## 1 PERFORMANCE DIFFERENCES BETWEEN STS AND DTS

As illustrated in Figure 1, the orange solid circle line indicates that both training and inference stages use the same time step on CIFAR10-DVS (STS-CIF), while the blue solid circle line signifies that during both stages employ different time steps with a training time step of 10 on CIFAR10-DVS (DTS-CIF). In STS-CIF, accuracy decreases from 83.17% in  $T = 10$  to 78.80% in  $T = 5$ . Similarly, in DTS-CIF, accuracy declines from 83.17% in  $T = 10$  to 77.60% in  $T = 5$ .



**Figure 1: Accuracy of TET [2] on CIFAR10-DVS, N-CALTECH101 and DVS-GESTURE across various time steps with STS and DTS.**

To address the reduction of latency in the inference stage, we believe that it can be divided into STS and DTS, where STS representing ensuring consistency between the time steps during training and inference stages in Spiking Neural Networks (SNNs), thereby reducing the overall time steps during SNN training, DTS does not guarantee the consistency of time steps between the training and inference stages of SNNs, it only decreases the time step during the inference stage. And as shown in Figure 1, it can be observed that as the time steps exceed 5, the performance gap between the STS and DTS methods of the TET [2] gradually narrows across all three datasets. This phenomenon is primarily attributed to the increasing number of event frames input to SNNs during the inference stage with the increment of time steps. Consequently, the SNNs acquires more comprehensive event data information learned during the inference stage, leading to a significant improvement in the performance of the DTS.

## 2 ANALYSIS AND DISCUSSION OF STEP-WISE KNOWLEDGE DISTILLATION

Compared to the absence of Knowledge Distillation (KD) and the application of vanilla KD, employing Step-wise Knowledge Distillation (SKD) to make the output distribution of SNN more stable.

This enables the SNN to achieve higher classification accuracy at lower time steps on neuromorphic datasets.

In our Hybrid Step-wise Distillation (HSD), we assume that SKD is the process of transferring the output distribution from the teacher model Artificial Neural Network (ANN) to the output distribution of SNN at each time step. So the Step-wise knowledge distillation is definitely greater than ordinary knowledge distillation. To verify this hypothesis, we start with the meaning of Kullback-Leibler (KL) divergence itself. KL divergence also known as relative entropy, is a measure of the difference between two probability distributions. The following will verify that the ordinary  $\mathcal{L}_{KD}$  is the lower bound of  $\mathcal{L}_{SKD}$ . Prove as follows:

$$\begin{aligned}
 \mathcal{L}_{SKD} &= \frac{1}{T_2} \sum_{t=1}^{T_2} \sum_{i=1}^N \left( p_{\tau}^a(i) \log \frac{p_{\tau}^a(i)}{p_{\tau}^{s,t}(i)} \right) \\
 &= \frac{1}{T_2} \sum_{t=1}^{T_2} \sum_{i=1}^N \left( p_{\tau}^a(i) \log p_{\tau}^a(i) - p_{\tau}^a(i) \log p_{\tau}^{s,t}(i) \right) \\
 &= \sum_{i=1}^N \left( p_{\tau}^a(i) \log p_{\tau}^a(i) - \sum_{t=1}^N \left( p_{\tau}^a(i) \log \left( \prod_{t=1}^{T_2} p_{\tau}^{s,t}(i) \right)^{\frac{1}{T_2}} \right) \right) \quad (1) \\
 &\geq \sum_{i=1}^N \left( p_{\tau}^a(i) \log p_{\tau}^a(i) - \sum_{t=1}^N \left( p_{\tau}^a(i) \log \left( \frac{1}{T_2} \sum_{t=1}^{T_2} p_{\tau}^{s,t}(i) \right) \right) \right) \\
 &= \sum_{i=1}^N \left( p_{\tau}^a(i) \log p_{\tau}^a(i) - \sum_{i=1}^N \left( p_{\tau}^a(i) \log p_{\tau}^s(i) \right) \right) \\
 &= \mathcal{L}_{KD},
 \end{aligned}$$

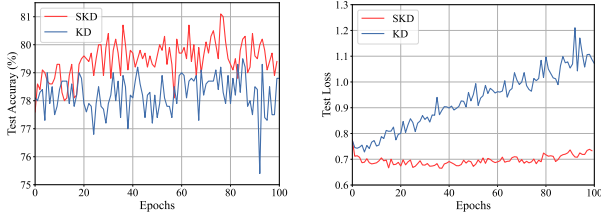
where the inequality is given by the Arithmetic Mean-Geometric Mean Inequality. Because logarithmic functions are concave functions, which means that the KL dispersion of SKD is bigger than that of vanilla KD.

As illustrated in Figure 2, compared to KD, SKD incorporates the temporal dimension information of SNN, namely, the output distribution of SNN at each time step. This diminishes the impact of individual time step outliers on the average output distribution, thereby enhancing the generalization performance of SNN and consequently achieving higher accuracy. Additionally, we visualize the loss during the inference stage and observe that the loss of SKD is lower than that of KD, further validating the efficacy of the SKD module.

However, during the inference stage, we notice that both KD and SKD exhibit a certain degree of upward trend in test loss, with KD showing a more pronounced increase. This phenomenon can mainly be attributed to the presence of some redundant information in the

**Table 1: Comparisons of Top-1 accuracy (%) performances with state-of-the-art methods on CIFAR10,  $T$  represents the time steps of the SNN during the inference stage.**

Method Type	Method	Venue	Model	$T \downarrow$	Acc $\uparrow$
Hybrid Training ANN-SNN Conversion	Hybrid training [7]	ICLR '20	ResNet-20	250	92.22
	QCFS [1]	ICLR '22	ResNet-20	128	93.48
Direct Training	STBP-tdBN [9]	AAAI '21	ResNet-19	4	92.92
	Dspike [4]	NeurIPS '21	ResNet-18	4	93.66
	TET [2]	ICLR '22	ResNet-19	4	94.44
	SLTT [6]	ICCV '23	ResNet-18	6	94.59
Hybrid Training	HSD (Ours)	-	ResNet-19	4	<b>94.71</b>
	HSD (Ours)	-	ResNet-19	1	93.24

**Figure 2: Comparisons of test accuracy and test loss performances at each epoch of KD and SKD in  $T = 5$  on CIFAR10-DVS.**

dataset, making the fine-tuning phase prone to overfitting. Nevertheless, our proposed SKD module greatly alleviates the impact of this phenomenon, effectively improving the model's performance.

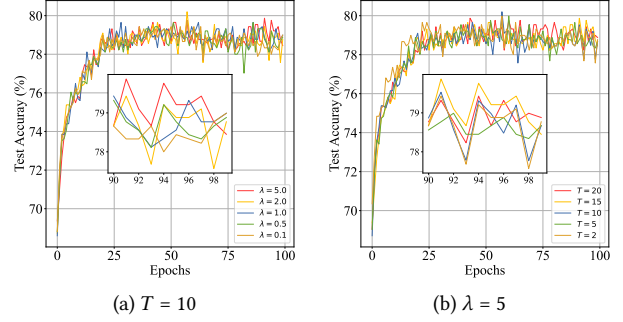
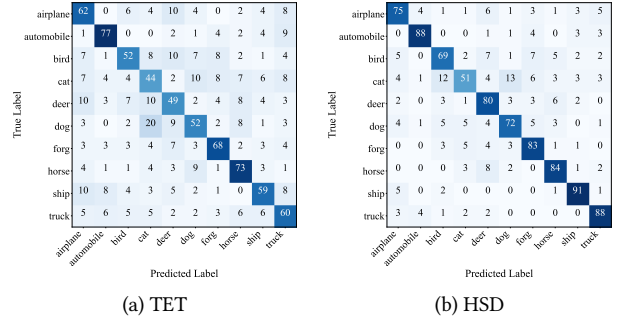
### 3 TEMPERATURE COEFFICIENT $T$ AND $\lambda$

In this section, we examine the impact of hyper-parameters in SKD and the principles guiding their selection. We show the effect of hyper-parameters in SKD and the principles of their selection. As depicted in Figure 3, the results underscore that the temperature coefficient  $T$ , when either too large or too small, fails to accurately capture the differences in probabilities between categories. Therefore, selecting an appropriate temperature coefficient  $T$  is crucial. In our method, we set the temperature  $T$  to 10 for optimal results. Additionally, the parameter  $\lambda$  governs the extent to which SKD module influences the final loss. If  $\lambda$  is excessively large or too small, it may not effectively guide the loss function in updating the weights. Hence, in our method, we set  $\lambda$  to 5 for optimal performance.

It is noteworthy that in the initial epochs, the accuracy on testing set reaches 70.00%. This can be attributed to the utilization of weight and threshold information during the pre-training phase using HSD.

### 4 VISUALIZATION OF CONFUSION MATRIX

Figure 4 shows the confusion matrix of our HSD and TET [2] on CIFAR10-DVS. In the right plot, the diagonal elements appear darker than in the left plot, indicating that HSD outperforms TET. Enriched event frame information and teacher model ANN assist SNN in better distinguishing differences between different categories at lower time steps.

**Figure 3: Comparisons of test accuracy performances at each epoch of HSD in  $T = 5$  on N-CALTECH101 with different hyper-parameter settings.****Figure 4: Comparisons performances with confusion matrix of TET and HSD in  $T = 5$  on CIFAR10-DVS.**

### 5 PERFORMANCE OF CIFAR10

To assess the generalization capabilities of our HSD method, we conduct experiments on the static CIFAR10 dataset. We employ the direct training SNN using the commonly used ResNet-19 [2] architecture, following the experimental setup inspired by the TET method. Our ANN-SNN conversion is independently trained and not initialized with pre-existing weights. Our method is regarded as a fusion of ANN-SNN conversion and direct training SNN, introducing knowledge distillation. In this context, the ANN serves not only as a pre-training model but also as a teacher model, aiming to leverage the entirety of knowledge acquired during ANN training. Given that current methods achieve respectable classification accuracy

**Table 2: Comparison of model parameter and types of neurons with state-of-the-art methods on CIFAR10-DVS.**

Method	Model	Time Steps	Neuron	Param (M)	Acc (%)
PLIF [3]	SNN-4	20	PLIF	17.4	74.8
Dspike [4]	ResNet-18	10	LIF	11.7	75.4
EventMixer [8]	ResNet-18	10	PLIF	11.7	81.5
NDA [5]	VGG-11	10	LIF	132.9	81.7
TET [2]	VGG-SNN	10	LIF	9.3	83.2
SLTT [6]	VGG-SNN	10	LIF	9.3	83.1
TET [2]	VGG-SNN	5	LIF	9.3	78.8
SLTT [6]	VGG-SNN	5	LIF	9.3	76.1
HSD (Ours)	VGG-SNN	5	IF	9.3	<b>81.1</b>

at relatively lower time steps through direct training SNN on the static CIFAR10 dataset, such as at 2 and 4, we conduct experiments with consistent training and inference time steps.

In Table 1, our HSD method achieves a performance of 93.24% in  $T = 1$ , surpassing even the results of some methods in  $T = 4$ , and our HSD outperforms TET with an accuracy of 94.44%, showing an improvement of 0.27% in  $T = 4$ . At the same time, our HSD method, built upon the ANN-SNN conversion, integrates direct training SNN. Therefore, in terms of latency, compared to ANN-SNN conversion methods, it achieves a lower time step without a significant decrease in accuracy. Compared to the direct training SNN, we utilize an ANN as a teacher model, enabling HSD to achieve higher classification accuracy at lower time steps.

## 6 COMPARISON OF MODEL PARAMETERS

To validate the efficiency of our model, we select the model parameters for evaluation. Due to the integration of ANN in our method, the model's parameters increases during the training stage. However, during the inference stage, only the SNN is utilized. Therefore, we evaluate the model parameters during the inference stage. As shown in Table 2, the selected VGG-SNN model [2] has a parameter count of 9.3 M, lower than the ResNet-18 model used by Dspike [4]. Furthermore, our model achieves higher accuracy. Under the same model parameters, the accuracy of our HSD model also surpasses that of TET. Additionally, since SNNs employ IF neurons, which only involve additive operations due to the absence of a leakage factor compared to commonly used LIF and PLIF neurons, they exhibit better computational efficiency when deployed on hardware.

## REFERENCES

- [1] Tong Bu, Wei Fang, Jianhao Ding, Penglin Dai, Zhaofer Yu, and Tiejun Huang. 2022. Optimal ANN-SNN Conversion for High-accuracy and Ultra-low-latency Spiking Neural Networks. (2022).
- [2] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. 2022. Temporal Efficient Training of Spiking Neural Network via Gradient Re-weighting. In *Proc. Int. Conf. Learn. Represent.*
- [3] Wei Fang, Zhaofer Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. 2021. Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*
- [4] Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. 2021. Differentiable Spike: Rethinking Gradient-Descent for Training Spiking Neural Networks. In *Adv. Neural Inf. Process. Syst.* 23426–23439.
- [5] Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. 2022. Neuromorphic Data Augmentation for Training Spiking Neural Networks. In *Proc. Eur. Conf. Comput. Vis.* 631–649.
- [6] Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. 2023. Towards Memory- and Time-Efficient Backpropagation for

Training Spiking Neural Networks. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.* 6143–6153.

- [7] Nitin Rath, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. 2020. Enabling Deep Spiking Neural Networks with Hybrid Conversion and Spike Timing Dependent Backpropagation. In *Proc. Int. Conf. Learn. Represent.*
- [8] Guobin Shen, Dongcheng Zhao, and Yi Zeng. 2023. EventMix: An Efficient Data Augmentation Strategy for Event-Based Learning. *Inf. Sci.* 644 (2023), 119170.
- [9] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. 2021. Going Deeper with Directly-Trained Larger Spiking Neural Networks. In *Proc. AAAI Conf. Artif. Intell.* 11062–11070.