

Temporal Separation with Entropy Regularization for Knowledge Distillation in Spiking Neural Networks

Kairong Yu¹ Chengting Yu¹ Tianqing Zhang¹ Xiaochen Zhao¹
 Shu Yang¹ Hongwei Wang^{1,†} Qiang Zhang² Qi Xu^{2,†}

¹Zhejiang University ²Dalian University of Technology

{kairong.22, chengting.21, xiaochen.24, shu.23, hongweiwang}@intl.zju.edu.cn
 zhangtianqing@zju.edu.cn
 {xuqi, zhangq}@dlut.edu.cn

Abstract

Spiking Neural Networks (SNNs), inspired by the human brain, offer significant computational efficiency through discrete spike-based information transfer. Despite their potential to reduce inference energy consumption, a performance gap persists between SNNs and Artificial Neural Networks (ANNs), primarily due to current training methods and inherent model limitations. While recent research has aimed to enhance SNN learning by employing knowledge distillation (KD) from ANN teacher networks, traditional distillation techniques often overlook the distinctive spatiotemporal properties of SNNs, thus failing to fully leverage their advantages. To overcome these challenges, we propose a novel logit distillation method characterized by temporal separation and entropy regularization. This approach improves existing SNN distillation techniques by performing distillation learning on logits across different time steps, rather than merely on aggregated output features. Furthermore, the integration of entropy regularization stabilizes model optimization and further boosts the performance. Extensive experimental results indicate that our method surpasses prior SNN distillation strategies, whether based on logit distillation, feature distillation, or a combination of both. Our project is available at <https://github.com/yukairong/TSER>.

1. Introduction

Inspired by the neural firing mechanisms observed in biological systems, Spiking Neural Networks (SNNs) [37] are considered a promising alternative to traditional Artificial Neural Networks (ANNs) due to their superior energy efficiency. Unlike ANNs, which use continuous activation val-

ues for information transmission[31], SNNs transmit and process information through discrete spike events [6, 38], with neurons generating spikes only when their membrane potential exceeds a threshold. This binary, event-driven approach allows SNNs to run efficiently on neuromorphic hardware [6, 42], accumulating synaptic inputs effectively and avoiding unnecessary computations related to zero input or activation [9, 14]. Given their event-driven dynamics and the biomimetic properties of spatiotemporal neuron activity [44, 45], SNNs demonstrate remarkable energy efficiency, robust adaptive learning capabilities [13, 41, 58], and ultra-low power consumption, showing significant potential for computational intelligence applications tasks [44]. Despite the inherent advantages of SNNs, their performance in common tasks such as image classification [27], object segmentation [43], and natural language processing [23] still lags behind that of ANNs. This gap largely stems from the limitations of current training methods and structural constraints in SNNs. In contrast to ANNs, SNNs cannot directly utilize backpropagation (BP) for deep network training. Moreover, attempts to directly adapt ANN methodologies to SNNs encounter compatibility issues, preventing SNNs from fully realizing their theoretical advantages.

Currently, conversion-based and learning-based approaches are two common methods for training SNNs. The former approaches seek to utilize knowledge from ANNs by transferring the parameters of a pre-trained ANN to an equivalent SNN [1, 10, 33, 34]. These methods often necessitate a significant number of time steps to achieve accuracy levels comparable to those of the original ANN [2, 46]. If the number of inference time steps is reduced, the model's capacity for effective information transfer may diminish, resulting in a decline in overall performance. To improve the performance of SNNs and minimize inference time steps, learning-based approaches have been employed

[†]Corresponding authors.

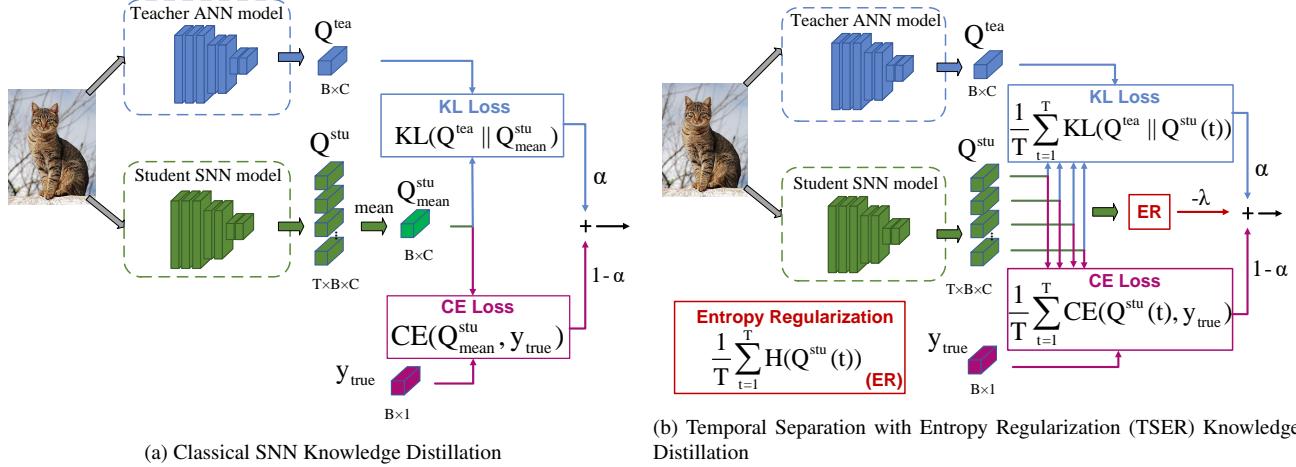


Figure 1. Illustration of the Classical SNN KD and Our Proposed Method. In our approach, we remove the temporal dimension fusion operation present in classical SNN KD and apply a temporal separation strategy to focus the learning process on outputs at individual time steps. The original calculations of KL Loss and CE Loss are adapted to compute the loss for each time step’s output, followed by averaging. Additionally, we incorporate an entropy regularization term to guide the learning direction away from erroneous knowledge. Here, T, B, and C represent the time steps, batch size, and number of classes, respectively.

to facilitate training during inference. These approaches leverage Backpropagation Through Time (BPTT) for effective training [48, 57]. Motivated by the work [25], several studies have utilized surrogate gradient estimation as a method to address the non-differentiability inherent in SNNs. This foundational research has enabled the direct training of CNNs and transformer-based SNNs, achieving strong performance on static datasets with reduced inference time steps. To further reduce the performance gap with ANNs, recent work has explored the inclusion of modules such as attention mechanisms to enhance SNN capabilities [28, 40, 52, 53]. While effective, these additional modules increase computational consumption, counteracting the inherent low-energy benefits of SNNs. Knowledge Distillation (KD) provides a potential solution, where SNNs are guided by a more capable ANN teacher model, addressing the inference burden associated with these modules. Prior works such as KDSNN [50] and LaSNN [24] employ hierarchical knowledge distillation [5] to facilitate the training of SNNs by leveraging the rich semantic information from the ANN teacher model. Recent approaches like BKD-SNN [51] move beyond traditional layer-wise distillation by integrating blurred knowledge distillation techniques with logit distillation, demonstrating significant improvements in performance across both CNNs and Transformer-based models.

In preliminary experiments, we observed that current SNN distillation architectures are not optimally designed for the unique structure of SNNs. As depicted in Fig. 1a, conventional SNN KD techniques distill the mean of outputs across time steps. However, the inherent nature

of SNN processing implies that significant information is embedded within the temporal dimension. Employing a mean approach for fusion prediction during classification resembles a voting selection process. Therefore, distributing the distillation of outputs across various time steps during the training phase can substantially alleviate the limitations that currently hinder distillation performance. To mitigate the impact of outliers on network performance and to rectify overly confident incorrect semantics produced by the teacher network, we introduce an entropy regularization term as a constraint. We conduct extensive experimental analyses to assess its efficacy. Furthermore, we evaluate our proposed method on several datasets, including CIFAR10 [26], CIFAR100 [26], and ImageNet [8]. The results of our experiments demonstrate that this method achieves state-of-the-art (SOTA) performance. In summary, our contributions are threefold:

- We introduce a temporal separation strategy into the SNN distillation method, effectively addressing the incompatibility issues present in previous SNN distillation frameworks and yielding a more effective universal distillation architecture.
- We integrate entropy regularization to correct the overly confident errors from the teacher network, thereby maximizing the performance of the logit-based distillation method.
- We conduct comprehensive evaluations across multiple datasets, showing that our proposed approach outperforms previous state-of-the-art (SOTA) methods.

2. Related Work

Conversion-based Methods. To fully leverage the knowledge acquired from pre-trained ANNs, conversion methods directly transfer the learned parameters from an ANN to the corresponding SNN, avoiding the difficulties of training SNNs from scratch. Several studies [19, 20] replace the activation functions in ANNs with spiking neurons and introduce optimizations like weight normalization [12] to generate corresponding SNNs. The core idea of these methods is to closely align the outputs of the ANN and SNN, maximizing the utilization of the pre-trained ANN knowledge. While later works [5, 10, 12] achieve nearly lossless accuracy conversion, they often involve long inference times and fail to retain the spatiotemporal characteristics inherent to spike-based processing, significantly limiting the practical applicability of converted SNNs.

Learning-based Methods. The development of surrogate gradient training methods [56] has resolved the challenge of training non-differentiable SNNs directly. Initially, [39] used a surrogate gradient approach to construct deep SNN models, achieving competitive accuracy on neuromorphic datasets. The DCT-SNN [16] applied frequency domain techniques to reduce the number of inference time steps. To support deeper SNN training, [15] introduced SEW ResNet to address gradient vanishing and exploding issues in Spiking ResNet [60], extending SNNs to architectures exceeding 100 layers. TA-SNN [52] introduced a temporal attention mechanism, paving the way for future attention mechanism advancements in the field. MA-SNN [53] went beyond single-dimensional information enhancement by proposing a temporal-channel-spatial attention mechanism. In Transformer architectures, models like SpikeFormer [61], SpikingFormer [62], and Meta-SpikerFormer [63] demonstrated outstanding classification performance through the design and improvement of innovative spike-based self-attention blocks.

Knowledge Distillation for SNN. Knowledge Distillation (KD), a well-established transfer learning technique, has proven effective across various tasks [3, 4, 22]. Recent studies [17, 30, 32, 47, 59] have applied KD to SNN training, using either large isomorphic SNNs to extract knowledge for smaller SNNs or leveraging the logits from pre-trained ANNs to train SNNs. KDSNN [50] utilizes a joint distillation approach based on logits and features, showing effectiveness across multiple datasets. Xu *et al.* [49] explore biologically inspired structure learning using reverse knowledge distillation. Furthermore, LaSNN [24] proposes a hierarchical feature distillation framework that achieves accuracy comparable to ANNs on the Tiny ImageNet dataset. BKDSNN [51] improves SNN performance

on complex datasets by using blurred knowledge to replicate ANN features. Despite the advantages of these distillation methods, as shown by the experiments in Fig. 2, current SNN distillation frameworks have yet to fully account for the unique characteristics of SNNs. Developing tailored distillation paradigms could enhance model accuracy in a simple and effective way.

3. Methodology

In this section, we start from preliminaries, introducing the fundamental concepts of SNNs and SNN distillation. Then we introduce our knowledge distillation approach that incorporates temporal separation with entropy regularization.

3.1. Preliminaries

Spiking Neuron Model. In SNNs, the fundamental computational unit is the spiking neuron, which serves as an abstract model of the dynamics of biological neurons. Currently, the Leaky Integrate-and-Fire (LIF) model is one of the most commonly used spiking neuron models, as it strikes a balance between simplified mathematical formulation and the complex dynamics of biological neurons. Therefore, in this study, we employ the LIF model as the foundational neuron model for the student SNN. Mathematically, a LIF neuron can be represented by Eq. 1.

$$\mu \frac{du(t)}{dt} = -(u(t) - u_{\text{reset}}) + I(t). \quad (1)$$

Where μ denotes the time constant of the membrane potential, $u(t)$ represents the membrane potential at time t , u_{reset} is the resting potential of the neuron, and $I(t)$ is the presynaptic input at time t . Based on this differential equation, the discrete-time and iterative mathematical representation of the LIF-SNN can be described as follows:

$$\begin{aligned} V(t) &= H(t-1) + \frac{1}{\mu}[I(t-1) - (H(t-1) - u_{\text{reset}})] \\ S(t) &= \Theta(V(t) - v_{th}) \\ H(t) &= u_{\text{reset}} \cdot S(t) + V(t) \cdot (1 - S(t)) \end{aligned} \quad (2)$$

The Heaviside step function Θ is defined as:

$$\Theta(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0. \end{cases} \quad (3)$$

Among these, $H(t-1)$ denotes the membrane potential following spike generation at the previous time step, while $I(t)$ and $V(t)$ represent the input and the updated membrane potential at time step t , respectively. Furthermore, v_{th} is the threshold that determines whether $V(t)$ should fire a spike or remain silent, and $S(t)$ indicates the spike sequence generated after triggering the action potential at time step t .

SNN Knowledge Distillation. We start from the original SNN Knowledge Distillation (KD) method. To illustrate the process of KD, we consider C-way classification task and denote $Z \in \mathbb{R}^C$ as the output of the network. Then, the class probability is given by:

$$Q_i = \frac{\exp(Z_i/\tau)}{\sum_{j=1}^C \exp(Z_j/\tau)}, \quad (4)$$

where Q_i and Z_i are the probability value on the i -th class and τ is the temperature scaling hyper-parameter. In knowledge distillation, τ is typically larger than 1.0, which helps control the flatness of the feature distribution to mitigate the phenomenon of overconfidence within the network. When τ equals 1.0, the output reverts to the vanilla Softmax output, denoted here as $S(\cdot)$ to represent this specific case. The objective of KD is to transfer knowledge from a high-performing teacher model to a lightweight student model. For the rescaled outputs, the original KD method achieves distillation by minimizing the KL divergence between the outputs of the teacher and student models.

$$KL(Q^{tea} || Q^{stu}) = \sum_{i=1}^C Q_i^{tea} \log\left(\frac{Q_i^{tea}}{Q_i^{stu}}\right), \quad (5)$$

where Q_i^{tea} and Q_i^{stu} indicates the probability value on the i -th category of the teacher and student output, respectively.

In addition, due to the potential errors in teachers' knowledge, KD also necessitates the use of cross-entropy to enable students to learn the true distribution of labels.

$$CE(Q^{stu}, y_{true}) = - \sum_{i=1}^C Q_i^{stu} \log(y_{true}) \quad (6)$$

where y_{true} denote the true labels.

The distinction between SNNs and ANNs lies in the fact that SNNs incorporate an additional temporal dimension T , which denotes the time step. Consequently, traditional SNN distillation attempts to adapt the distillation methods designed for ANNs by employing a simplistic approach that averages the outputs across the temporal dimension. By introducing a parameter α to control the weight ratio between true labels and soft labels. The loss calculation formula can be expressed as follows:

$$\begin{aligned} \mathcal{L}_{SKD} = & (1 - \alpha) \cdot CE\left(\frac{1}{T} \sum_{t=1}^T Q^{stu}(t), y_{true}\right) \\ & + \alpha \cdot KL(Q^{tea} || \frac{1}{T} \sum_{t=1}^T Q^{stu}(t)) \end{aligned} \quad (7)$$

3.2. Temporal Separation Knowledge Distillation with Entropy Regularization

In this section, we provide an in-depth explanation of the temporal separation knowledge distillation method with entropy regularization. Our approach consists of two core

components: temporal separation strategy and entropy regularization.

Temporal Separation Strategy. As illustrated in Fig. 1a, the original SNN distillation merely transferred the conventional KD methods to SNNs in a naive manner. The use of mean values to integrate outputs across the temporal dimension fails to adequately account for the unique spatiotemporal characteristics of SNNs. Due to the incremental temporal processing in SNNs, the output features at each time step exhibit variability and contain rich information. However, this also implies the potential presence of local outliers in the outputs at different time steps. While the averaging operation mitigates the impact of these anomalies, it poses risks for the subsequent model learning and probabilistic confidence. Consequently, implementing a temporal separation strategy for learning time step's output is crucial.

As shown in Fig. 1b, we apply temporal separation strategy to CE and KL in Eq. 7, moving the mean operation outside. We directly use the outputs at each time step to distill the learning of the true labels and the teacher distribution. The loss expression for temporal separation knowledge distillation is given by

$$\begin{aligned} \mathcal{L}_{TS} = & (1 - \alpha) \cdot \frac{1}{T} \sum_{t=1}^T CE(Q^{stu}(t), y_{true}) \\ & + \alpha \cdot \frac{1}{T} \sum_{t=1}^T \cdot KL(Q^{tea} || Q^{stu}(t)). \end{aligned} \quad (8)$$

Entropy Regularization. After introducing the temporal separation strategy, the soft labels from the teacher network directly impact the output distribution at each time step. This approach, however, can inadvertently reinforce erroneous information from the teacher network, increasing the likelihood that the student model will adopt incorrect knowledge. To counter this, we introduce an entropy regularization term, which constrains and adjusts the teacher's outputs to reduce excessive confidence in potentially erroneous knowledge. The formula for entropy is as follows:

$$H(Q^{stu}) = - \sum_{i=1}^C Q_i^{stu} \log(Q_i^{stu}). \quad (9)$$

Then, we obtain the loss formula for the entropy regularization term:

$$\mathcal{L}_{ER} = -\lambda \cdot \frac{1}{T} \sum_{t=1}^T H(Q^{stu}(t)), \quad (10)$$

where λ is the penalty factor.

Finally, we integrate the temporal separation strategy, as shown in Eq. 8, with the entropy regularization term from Eq. 10 to derive a new loss function as follows:

$$\mathcal{L}_{TSER} = \mathcal{L}_{TS} + \mathcal{L}_{ER}. \quad (11)$$

And Algorithm. 1 outlines the overall training process of our proposed method.

4. Experiment

In this section, we conduct extensive experiments to validate the effectiveness of our method. Initially, we compare our approach with existing SNN distillation methods. Furthermore, we conduct ablation studies, sensitivity analyses, energy consumption assessments, and visualizations to provide a comprehensive evaluation of our method.

4.1. Experimental Settings

Our proposed method is evaluated on three datasets. For teacher ANN networks, we select several advanced models, including ResNet-19 and ResNet-34 [21] as well as VGG-16. In contrast, for the student SNN models, we employ neural network architectures with equivalent or fewer layers, specifically ResNet-18/19, and VGG-11/16. Additionally, we conduct multiple experiments by varying the time step.

4.2. Comparison on Static Datasets

Our method is compared with previous approaches on CIFAR10 and CIFAR100, as shown in the Table. 1. The re-

Algorithm 1 Training student SNN model with TSER knowledge distillation for one epoch.

Require: pre-trained teacher ANN model M_t ; an initialized student SNN model M_s ; input dataset sample X ; and the true labels y_{true} ; total training iteration in one epoch I_{train} ; total validation iteration in one epoch I_{val}

Ensure: SNN model with TSER KD

```

for  $i = 1$  to  $I_{train}$  do
    Get mini-batch training data, and true label:  $X^i, y_{true}^i$ ;
    Compute the  $M_s$  output  $Z^{stu,i}(t)$  of each time step;
    Compute the  $M_t$  output  $Z^{tea,i}$ ;
    Calculate loss function:  $\mathcal{L}_{TSER}$ ;
    Backpropagation and update model parameters;
end for
for  $i = 1$  to  $I_{val}$  do
    Get mini-batch validation data, and true label:  $X^i, y_{true}^i$ ;
    Compute the SNN average output  $Z_{mean}^{stu,i} = \frac{1}{T} \sum_{t=1}^T Z^{stu,i}(t)$ ;
    Compare the  $Z_{mean}^{stu,i}$  and  $y_{true}^i$  for classification;
end for

```

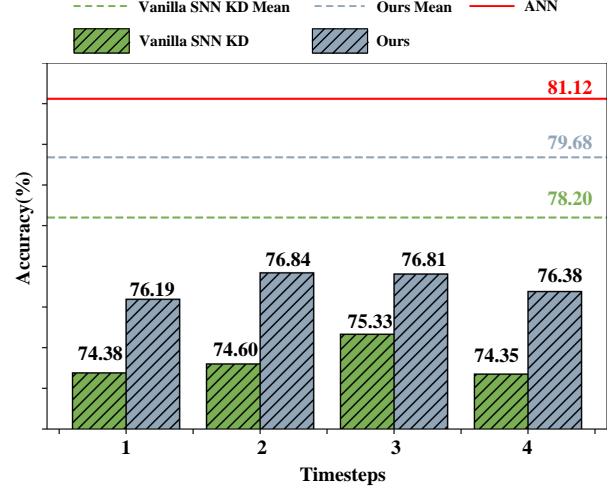


Figure 2. Prediction Accuracy Distribution at Different Time Steps for Vanilla SNN KD and Our Proposed Method. The solid red line indicates the teacher model’s accuracy, while the dashed lines represent the prediction accuracies of different distillation methods after averaging outputs over time steps. The bars show the prediction accuracies of each distillation method at individual time steps.

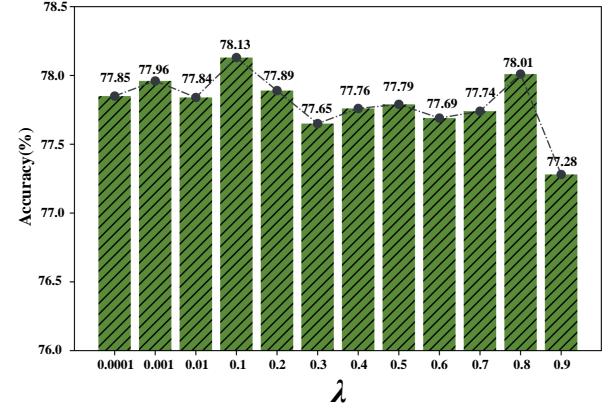


Figure 3. Accuracy Distribution for Different λ Values. Experiments conducted on the CIFAR100 dataset with a fixed time step of 2, testing various λ values.

sults demonstrate our method surpasses the SOTA performance achieved by prior method across both ResNet and VGG architectures. Specifically, ResNet19 outperform the previous SOTA by 0.57% and 0.41% on CIFAR10 and CIFAR100, respectively, achieving accuracies of 96.72% and 81.29%. This indicates that our approach, although based on logit distillation, performs exceptionally well compared to previous, more complex distillation methods. By addressing the limitations of simple transfer in existing distillation, our framework effectively learns from the teacher’s knowledge, minimizing the gap between the student and

Method	Archি.	Acc.(%)		Acc.(%)		Archি.	Acc.(%)		Acc.(%)				
		CF-100	CF-10	CF-100	CF-10		CF-100	CF-10	CF-100	CF-10			
		Timestep=2								Timestep=4			
without KD	GLIF[55]	R-19	75.48	94.44	R-18	74.60	94.15	R-19	77.05	94.85	R-18	76.42	94.67
	TET[11]	R-19	72.87	94.16	-	-	-	R-19	74.47	94.44	-	-	-
	LSG[35]	R-19	76.32	94.41	-	-	-	R-19	76.85	95.17	-	-	-
	PFA[7]	R-19	76.70	95.60	-	-	-	R-19	78.10	95.71	-	-	-
	MPBN[18]	R-19	79.51	96.47	V-16	73.88	93.96	R-19	80.10	96.52	V-16	74.74	94.44
	IM-LIF[36]	R-19	77.21	95.29	-	-	-	R-19	77.42	95.66	-	-	-
	Spikformer[64]	-	-	-	-	-	-	S-4-256	75.96	93.94	S-4-384	77.86	95.19
	Spikingformer[61]	-	-	-	-	-	-	Sg-4-256	77.43	94.77	Sg-4-384	79.09	95.61
with KD	SD Transformer [54]	-	-	-	-	-	-	S-2-512	78.40	95.60	-	-	-
	Teacher ANN Timestep=1												
	Teacher ANN	R-34	81.12	97.10	V-16	78.08	96.06	R-34	81.12	97.10	V-16	78.08	96.06
		R-19	81.97	97.08	V-16	78.08	96.06	-	-	-	V-16	78.08	96.06
	Timestep=2						Timestep=4						
	LaSNN [24]	R-18	76.17	93.64	V-16	73.80	93.90	R-18	78.12	95.09	V-16	74.99	94.49
		R-19	80.30	95.26	V-11	69.89	90.23	-	-	-	V-11	70.74	90.42
	KDSNN [50]	R-18	77.16	95.25	V-16	74.65	94.26	R-18	78.46	95.72	V-16	75.98	94.85
		R-19	80.88	96.15	V-11	72.17	92.81	-	-	-	V-11	73.05	92.91
	BKDSNN [51]	R-18	73.30	93.64	V-16	73.80	94.10	R-18	75.57	95.09	V-16	74.99	94.55
		R-19	75.74	95.26	V-11	69.89	92.73	-	-	-	V-11	70.74	92.88
	Ours	R-18	78.30 ± 0.08	95.58 ± 0.08	V-16	75.81 ± 0.12	94.55 ± 0.09	R-18	79.69 ± 0.02	96.18 ± 0.05	V-16	77.06 ± 0.04	95.01 ± 0.10
		R-19	81.29 ± 0.14	96.72 ± 0.02	V-11	73.67 ± 0.04	93.31 ± 0.02	-	-	-	V-11	74.65 ± 0.07	93.68 ± 0.02

Table 1. Comparison results with training-based SNN SOTA methods, including CNN- and transformer-based approaches, on CIFAR-10/100, with and without Knowledge Distillation (KD). **Acc.** denotes accuracy, **CF** denotes CIFAR, and **Archি.** denotes architecture. The abbreviations R, V, S, and Sg represent ResNet, VGG, Spikeformer, and Spikingformer architectures, respectively, while SD Transformer refers to the Spike-Driven Transformer. For KD methods, the student model aligns with the architecture of the teacher ANN.

teacher models. This reduction in disparity validates the superiority of our method.

Further comparisons on ImageNet dataset are detailed in Table. 2 . The SEW ResNet-18 and SEW ResNet-34 architectures achieve top-1 accuracies of 69.24% and 73.16%, respectively. These results indicate that our method is capable of delivering competitive performance even within large-scale datasets, effectively harnessing the knowledge from ANNs to enhance the learning capabilities of SNNs.

4.3. Ablation Study

Temporal Separation Strategies and Entropy Regularization.

We investigate different configurations of $\mathcal{L}_{TSE,R}$, focusing on the temporal separation applied within the KL and CE losses, as well as the integration of an entropy regularization term. The results are summarized in Table. 3. Our experiments reveal that incorporating tempo-

ral separation strategies notably enhances model accuracy, with improvements of 0.49% and 0.26% for the standard KL and CE loss function, respectively. When temporal separation is applied concurrently to both KL and CE losses, the overall accuracy increase reaches 0.84%. Furthermore, we find that optimally adjust the parameter λ in the entropy regularization term, alongside temporal separation, leads to an enhancement of up to 0.98%. These findings underscore the effectiveness of combining temporal separation and entropy regularization in maximizing the performance of SNN distillation.

Comparison of Output Accuracy at Different Time Steps.

To illustrate the adaptability of our proposed distillation method for SNNs, we perform a comparative analysis of accuracy across different time steps between vanilla SNN KD and our method. As shown in Fig. 2, the accu-

	Method	Architecture	T	Accuracy
without KD	TET [11]	R-34	6	64.79%
	GLIF [55]	R-34	4	67.52%
	MPBN [18]	R-18	4	63.14%
		R-34	4	64.71%
	SEW ResNet [15]	R-18	4	63.18%
		R-34	4	67.04%
	Spikformer [64]	S-8-384	4	70.24%
		S-6-512	4	72.46%
	Spikingformer [61]	Sg-8-384	4	72.45%
	Spike-driven Transformer [54]	Sg-8-384	4	72.28%
with KD	Teacher ANN	SEW R-18	1	71.69%
		SEW R-34	1	75.38%
	LaSNN [24]	SEW R-18	4	63.33%
		SEW R-34	4	66.98%
	KDSNN [50]	SEW R-18	4	63.61%
		SEW R-34	4	67.28%
	BKDSNN [51]	SEW R-18	4	63.43%
		SEW R-34	4	67.21%
	Ours	SEW R-18	4	69.24% ± 0.19
		SEW R-34	4	73.16% ± 0.15

Table 2. Comparison results with training-based SNN SOTA methods, including CNN- and Transformer-based approaches, on ImageNet, with and without Knowledge Distillation (KD). **T** denotes Timestep. R, S, Sg denotes ResNet, Spikeformer and Spikingformer architectures respectively.

Temporal Separation		Entropy Regularization	Accuracy
KL Loss	CE Loss		
✗	✗	✗	77.15%
✗	✓	✗	77.41%
✓	✗	✗	77.64%
✓	✓	✗	77.99%
✓	✓	✓	78.13%

Table 3. Ablation of Temporal Separation Strategy and Entropy Regularization.

racy discrepancy among time steps for the vanilla SNN KD is 0.98%, notably higher than the 0.65% difference with our approach. Furthermore, the vanilla SNN KD yields an average output accuracy of 74.67%, whereas our method achieves 76.56%. And as the accuracy of individual time step outputs improved, we observe a significant enhancement in performance following the mean operation on these outputs, aligning with our hypothesis. This clearly demonstrates the effectiveness of our method and highlights the significance of temporal separation in the SNN distillation process.

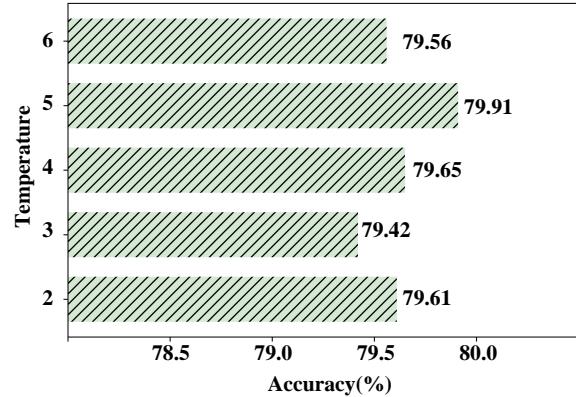


Figure 4. Temperature Coefficient Sensitivity. Our method demonstrates stability across various τ hyperparameters. This experiment is conducted on CIFAR100 with ResNet-34 as the teacher model and ResNet-18 as the student model.

Selection of Parameter λ . The entropy regularization term is essential for \mathcal{L}_{TSEER} , as it partially alleviates the erroneous knowledge transferred by the teacher network. Thus, selecting an optimal value for λ is crucial. We conduct experiments with various values of λ , and the results are shown in Fig. 3. Increasing λ enhances the correction of the teacher’s inaccuracies but also dilutes the distribution of the correct knowledge. Our findings indicate that maintaining λ within the range of 0.0001 to 0.2 enhances the network’s overall training stability. However, excessively high values of λ can result in negative loss values, as described in Eq. 11, which must be avoided.

Temperature Coefficient Sensitivity. In our experiments, we configure the temperature to $\tau = 5$. Utilizing the prior experimental setup, we select five temperature points: $\tau = [2, 3, 4, 5, 6]$. We evaluate performance with ResNet-34 as the teacher model and ResNet-18 as the student model on the CIFAR100 dataset. The results depicted in Fig. 4, indicate that our method exhibits stable performance across different temperature coefficients.

4.4. Performance Analysis and Visualization

Firing Rate Analysis. In Table. 4, the firing rates of our proposed method show a slight increase compared to vanilla SNN KD across both ResNet and VGG architectures. Fig. 2 illustrates that an increase in firing rate of approximately 0.1% corresponds to a 1.48% improvement in overall performance. Notably, when comparing our method to other distillation techniques within the ResNet architecture, it maintains performance gains while simultaneously reducing firing rate and increasing spike sparsity, thus minimizing redundant spikes. Although LaSNN exhibits lower firing rates, this comes at the expense of some model per-

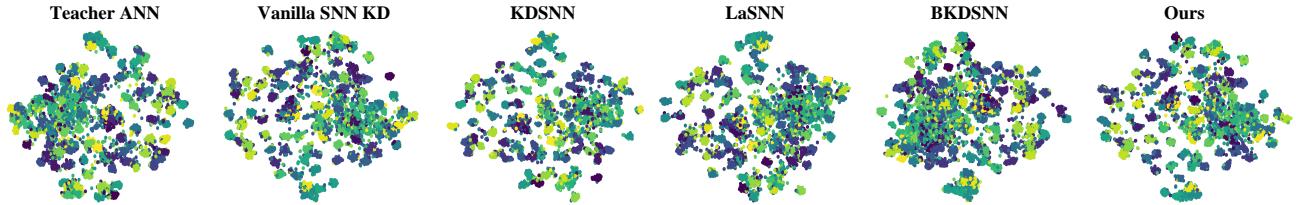


Figure 5. t-SNE Visualization of features learned by teacher ANN and different distillation methods.

Architecture	Methods	OPs	ACs	MACs	Energy	Training Time	Firing Rate
ResNet18	Vanilla SNN KD	2.23G	405.33M	12.2M	0.42mJ	2.78s	22.85%
	KDSNN	2.23G	411.49M	12.2M	0.43mJ	2.67s	23.23%
	LaSNN	2.23G	300.94M	12.2M	0.33mJ	2.79s	16.76%
	BKDSNN	2.23G	511.57M	12.2M	0.52mJ	2.84s	34.93%
	Ours	2.23G	405.98M	12.2M	0.42mJ	2.70s	22.95%
VGG16	Vanilla SNN KD	1.26G	159.96M	274.2M	1.40mJ	2.43s	18.91%
	KDSNN	1.26G	170.79M	274.2M	1.39mJ	2.39s	18.59%
	LaSNN	1.26G	150.96M	274.2M	1.41mJ	2.44s	17.25%
	BKDSNN	1.26G	171.94M	274.2M	1.40mJ	2.58s	18.66%
	Ours	1.26G	160.92M	274.2M	1.41mJ	2.40s	20.05%

Table 4. Energy consumption, training time, and spike firing rate under various distillation methods and architectures on the CIFAR100 dataset. Training Time refers to the time required per batch during training.

formance. Furthermore, unlike our logit-based approach, LaSNN employs a feature-based method that optimizes the feature map directly, making it more advantageous for feature sparsification.

Energy Consumption Analysis. Our energy consumption analysis, based on the energy model in [29], is summarized in Table. 4. The results reveal that the most significant variations in energy consumption occur in Accumulation Computation (AC), which is related to the firing rate. Notably, a lower firing rate correlates with reduced AC. Consequently, our approach does not increase the computational load of SNNs and is competitive regarding energy efficiency.

Training Time. We evaluate the training times of various competitive KD methods on the CIFAR100 dataset, focusing on the training time per batch. As shown in Table. 4, our method achieves the second-fastest training time, surpassed only by KDSNN. This advantage is largely due to KDSNN simplifying the original KL divergence into cross-entropy in the logit-based framework, while lowering computational cost, also affect accuracy. Our approach outperforms other techniques by utilizing only the logits output for knowledge distillation, thus eliminating the need for additional auxiliary training modules or feature map comparisons. In contrast, previous methods demand more time and resources to extract knowledge from intermediate layers.

Visualization. We employ t-SNE to visualize features learned from various distillation methods on the CIFAR100 dataset, utilizing ResNet-34 as the teacher and ResNet-18 as the student model. As shown in Fig. 5, our approach significantly enhances the distinguishability of deeper features compared to existing SNN distillation techniques.

5. Conclusion

This work addresses the issue of existing SNN distillation methods that are simply derived from ANN. We propose a new framework specifically designed for SNN distillation, which better exploits the temporal information contained in the logit output of SNNs for the process of knowledge distillation. Specifically, we incorporate a temporal separation strategy and introduce an entropy regularization term into the original distillation method. We aim to uncover the potential of SNN distillation by fully leveraging temporal information and rectifying the erroneous teacher knowledge. Extensive experimental results demonstrate the efficacy of this approach.

6. Acknowledgements

This work was supported in part by National Natural Science Foundation of China (NSFC) (62476035, 62206037, 62276230), and Natural Science Foundation of Zhejiang Province (LDT23F02023F02), and State Key Laboratory (SKL) of Biobased Transportation Fuel Technology.

References

- [1] 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. *arXiv preprint arXiv:2303.04347*, 2023. 1
- [2] Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113:54–66, 2015. 1
- [3] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020. 3
- [4] Li Chen, Chunyan Yu, and Lvcai Chen. A new knowledge distillation for incremental object detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019. 3
- [5] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019. 2, 3
- [6] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99, 2018. 1
- [7] Haoyu Deng, Ruijie Zhu, Xuerui Qiu, Yule Duan, Malu Zhang, and Liang-Jian Deng. Tensor decomposition based attention module for spiking neural networks. *Knowledge-Based Systems*, 295:111780, 2024. 6
- [8] 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. 2
- [9] Lei Deng, Yujie Wu, Xing Hu, Ling Liang, Yufei Ding, Guoqi Li, Guangshe Zhao, Peng Li, and Yuan Xie. Rethinking the performance comparison between snns and anns. *Neural networks*, 121:294–307, 2020. 1
- [10] Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. *arXiv preprint arXiv:2103.00476*, 2021. 1, 3
- [11] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. In *International Conference on Learning Representations*. 6, 7
- [12] Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks (IJCNN)*, pages 1–8. ieee, 2015. 3
- [13] Jianhao Ding, Tong Bu, Zhaofei Yu, Tiejun Huang, and Jian Liu. Snn-rat: Robustness-enhanced spiking neural network through regularized adversarial training. *Advances in Neural Information Processing Systems*, 35:24780–24793, 2022. 1
- [14] Jason K Eshraghian, Max Ward, Emre O Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Ben-
- namoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 2023. 1
- [15] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021. 3, 7
- [16] Isha Garg, Sayeed Shafayet Chowdhury, and Kaushik Roy. Dct-snn: Using dct to distribute spatial information over time for low-latency spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4671–4680, 2021. 3
- [17] Yufei Guo, Weihang Peng, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Xuhui Huang, and Zhe Ma. Joint a-snn: Joint training of artificial and spiking neural networks via self-distillation and weight factorization. *Pattern Recognition*, 142:109639, 2023. 3
- [18] Yufei Guo, Yuhang Zhang, Yuanpei Chen, Weihang Peng, Xiaode Liu, Liwen Zhang, Xuhui Huang, and Zhe Ma. Membrane potential batch normalization for spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19420–19430, 2023. 6, 7
- [19] Bing Han and Kaushik Roy. Deep spiking neural network: Energy efficiency through time based coding. In *European conference on computer vision*, pages 388–404. Springer, 2020. 3
- [20] Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13558–13567, 2020. 3
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5
- [22] Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3
- [23] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdelrahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012. 1
- [24] Di Hong, Jiangrong Shen, Yu Qi, and Yueming Wang. Lasnn: Layer-wise ann-to-snn distillation for effective and efficient training in deep spiking neural networks. *arXiv preprint arXiv:2304.09101*, 2023. 2, 3, 6, 7
- [25] Saeed Reza Kheradpisheh, Mohammad Ganjtabesh, Simon J Thorpe, and Timothée Masquelier. Stdः-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99:56–67, 2018. 2
- [26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2
- [27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural net-

- works. *Advances in neural information processing systems*, 25, 2012. 1
- [28] Souvik Kundu, Gourav Datta, Massoud Pedram, and Peter A Beerel. Spike-thrift: Towards energy-efficient deep spiking neural networks by limiting spiking activity via attention-guided compression. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3953–3962, 2021. 2
- [29] Souvik Kundu, Massoud Pedram, and Peter A Beerel. Hir-snn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5209–5218, 2021. 8
- [30] Ravi Kumar Kushawaha, Saurabh Kumar, Biplab Banerjee, and Rajbabu Velmurugan. Distilling spikes: Knowledge distillation in spiking neural networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4536–4543. IEEE, 2021. 3
- [31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1
- [32] Dongjin Lee, Seongsik Park, Jongwan Kim, Wuhyeong Doh, and Sungroh Yoon. Energy-efficient knowledge distillation for spiking neural networks. *arXiv preprint arXiv:2106.07172*, 2021. 3
- [33] Chen Li, Lei Ma, and Steve Furber. Quantization framework for fast spiking neural networks. *Frontiers in Neuroscience*, 16:918793, 2022. 1
- [34] Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In *International conference on machine learning*, pages 6316–6325. PMLR, 2021. 1
- [35] Shuang Lian, Jiangrong Shen, Qianhui Liu, Ziming Wang, Rui Yan, and Huajin Tang. Learnable surrogate gradient for direct training spiking neural networks. In *IJCAI*, pages 3002–3010, 2023. 6
- [36] Shuang Lian, Jiangrong Shen, Ziming Wang, and Huajin Tang. Im-lif: Improved neuronal dynamics with attention mechanism for direct training deep spiking neural network. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024. 6
- [37] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997. 1
- [38] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014. 1
- [39] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. 3
- [40] Kleanthis C Neokleous, Marios N Avraamides, Costas K Neokleous, and Christos N Schizas. Selective attention and consciousness: investigating their relation through computational modelling. *Cognitive Computation*, 3:321–331, 2011. 2
- [41] Srdjan Ostojic. Two types of asynchronous activity in networks of excitatory and inhibitory spiking neurons. *Nature neuroscience*, 17(4):594–600, 2014. 1
- [42] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019. 1
- [43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18, pages 234–241. Springer, 2015. 1
- [44] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019. 1
- [45] Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Bill Kay, et al. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022. 1
- [46] Abhroneil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13:95, 2019. 1
- [47] Sugahara Takuya, Renyuan Zhang, and Yasuhiko Nakashima. Training low-latency spiking neural network through knowledge distillation. In *2021 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)*, pages 1–3. IEEE, 2021. 3
- [48] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence*, pages 1311–1318, 2019. 2
- [49] Qi Xu, Yaxin Li, Xuanye Fang, Jiangrong Shen, Jian K. Liu, Huajin Tang, and Gang Pan. Biologically inspired structure learning with reverse knowledge distillation for spiking neural networks, 2023. *arXiv:2304.09500 [cs]*. 3
- [50] Qi Xu, Yaxin Li, Jiangrong Shen, Jian K Liu, Huajin Tang, and Gang Pan. Constructing deep spiking neural networks from artificial neural networks with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7886–7895, 2023. 2, 3, 6, 7
- [51] Zekai Xu, Kang You, Qinghai Guo, Xiang Wang, and Zhezhi He. Bkdsnn: Enhancing the performance of learning-based spiking neural networks training with blurred knowledge distillation. *arXiv preprint arXiv:2407.09083*, 2024. 2, 3, 6, 7
- [52] Man Yao, Huanhuan Gao, Guangsue Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10221–10230, 2021. 2, 3

- [53] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *arXiv preprint arXiv:2209.13929*, 2022. 2, 3
- [54] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *Advances in neural information processing systems*, 36, 2024. 6, 7
- [55] Xingting Yao, Fanrong Li, Zitao Mo, and Jian Cheng. Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks. *Advances in Neural Information Processing Systems*, 35:32160–32171, 2022. 6, 7
- [56] Qiang Yu, Jialu Gao, Jianguo Wei, Jing Li, Kay Chen Tan, and Tiejun Huang. Improving multispike learning with plastic synaptic delays. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10254–10265, 2022. 3
- [57] Friedemann Zenke and Tim P Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural computation*, 33(4):899–925, 2021. 2
- [58] Friedemann Zenke, Everton J Agnes, and Wulfram Gerstner. Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nature communications*, 6(1):6922, 2015. 1
- [59] Fengzhao Zhang, Chengting Yu, Hanzhi Ma, Zheming Gu, and Er-ping Li. Knowledge Distillation For Spiking Neural Network. In *2023 5th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI)*, pages 1015–1020, 2023. 3
- [60] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11062–11070, 2021. 3
- [61] Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Zhengyu Ma, Han Zhang, Huihui Zhou, and Yonghong Tian. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv preprint arXiv:2304.11954*, 2023. 3, 6, 7
- [62] Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Zhengyu Ma, Huihui Zhou, Xiaopeng Fan, and Yonghong Tian. Enhancing the performance of transformer-based spiking neural networks by snn-optimized downsampling with precise gradient backpropagation. *arXiv preprint arXiv:2305.05954*, 2023. 3
- [63] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021. 3
- [64] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, YAN Shuicheng, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*. 6, 7

Temporal Separation with Entropy Regularization for Knowledge Distillation in Spiking Neural Networks

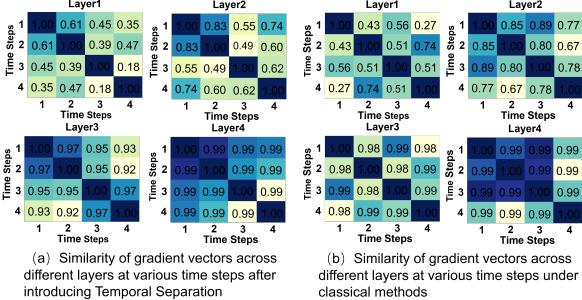


Figure 1. Comparison of gradients between classical methods and methods incorporating the Temporal Separation strategy. The cosine similarity of gradients across different time steps is computed for various layers at different depths using both approaches.

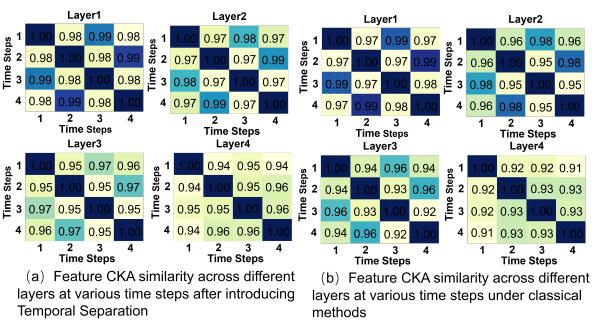


Figure 2. Comparison of similarity of features at different time steps across layers between classical methods and methods incorporating the Temporal Separation strategy.

R1:The Reasons for the Effectiveness of Temporal Separation. As shown in Fig. 1, the introduction of Temporal Separation reduces the gradient similarity between time steps compared to classical methods. This allows each time step to have a distinct optimization direction rather than being broadly optimized towards a potentially incorrect direction. Consequently, this leads to more cohesive and accurate time step features. The strategy introduced in Fig. 2 results in an improvement of 1%-5% in CKA similarity [?], which is consistent with the Fig. 2 in main text. This effectively enhances the accuracy of features across different time steps, thereby improving the overall performance after aggregation.

R1:Design Motivation and Experiments of Entropy Regularization. The introduction of entropy regularization aims to mitigate the impact of high-confidence incorrect predictions from the teacher model. When the value of α is set inappropriately, the learning process of the network may be adversely affected by the erroneous high-confidence

knowledge from the teacher network, leading to optimization bias. Incorporating regularization can alleviate the influence of erroneous information to some extent, thereby enhancing the network performance. Experimental results indicate that adding entropy regularization to the vanilla KD method improves performance by 0.30% to 0.51%.

R1:Firing rate analysis. This was a misstatement. Eq. 12 represents the computation formula for a single layer. The energy consumption analysis in Table. 4 is comprehensive, including all layers except for the final one.

R1:Performance Table is not clear. The term "baseline ANN performance" refers to the accuracy of the teacher network ANN. We aim to compare the learning outcomes of different distillation methods while keeping the teacher network weights fixed. And we will consider your suggestions for future revisions.

R2:Performance Comparison of Transformer-based SNNs. We conducted experiments on the Transformer model using the CIFAR-10/100 datasets. The detailed performance comparisons are presented in Table. 1. Testing results on the ImageNet dataset will be provided in future.

	S-4-256		S-2-384		S-4-384		Time Step
	CF-100	CF-10	CF-100	CF-10	CF-100	CF-10	
Teacher ANN	82.22	96.75	82.22	96.75	82.22	96.75	1
KDSNN	78.38	95.00	79.25	95.59	80.33	95.88	4
LASNN	78.02	94.97	78.91	95.55	79.99	95.79	4
BKDSNN	79.41	95.29	80.63	95.90	81.26	96.06	4
Ours	79.92	95.64	81.07	95.97	81.86	96.72	4

Table 1. Comparison of different distillation methods applied to Transformer-based SNNs on CIFAR10/100 performance.

R2:Inconsistency Between Figures and Equations. We appreciate your careful feedback. We will revise the final paper in accordance with your suggestions.

R3:Analysis of Entropy Regularization. In the main text, Fig. 3 presents a simple experiment on the impact of λ on entropy regularization. A more in-depth theoretical analysis of this aspect will be conducted in future work.

R3:Training Time. In our subsequent experiments, we found that our method is generally faster than vanilla SNN KD training in most cases. This observation contradicts common intuition. I think this discrepancy may be related to the underlying computational logic of PyTorch and CUDA.

020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040

041
042
043
044
045
046
047
048
049
050
051
052
053