A Closer Look at Knowledge Distillation in Spiking Neural Network Training

Xu Liu[1], Na Xia[1*], Jinxing Zhou[1], Jingyuan Xu[1], Dan Guo[1,2*]

[1]School of Computer Science and Information Engineering, Hefei University of Technology
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
xuliu.hfut@gmail.com, {xiananawo,guodan}@hfut.edu.cn

Abstract

Spiking Neural Networks (SNNs) become popular due to excellent energy efficiency, yet facing challenges for effective model training. Recent works improve this by introducing knowledge distillation (KD) techniques, with the pre-trained artificial neural networks (ANNs) used as teachers and the target SNNs as students. This is commonly accomplished through a straightforward element-wise alignment of intermediate features and prediction logits from ANNs and SNNs, often neglecting the intrinsic differences between their architectures. Specifically, ANN's outputs exhibit a continuous distribution, whereas SNN's outputs are characterized by sparsity and discreteness. To mitigate this issue, we introduce two innovative KD strategies. Firstly, we propose the Saliency-scaled Activation Map Distillation (SAMD), which aligns the spike activation map of the student SNN with the class-aware activation map of the teacher ANN. Rather than performing KD directly on the raw features of ANN and SNN, our SAMD directs the student to learn from saliency activation maps that exhibit greater semantic and distribution consistency. Additionally, we propose a Noise-smoothed Logits Distillation (NLD), which utilizes Gaussian noise to smooth the sparse logits of student SNN, facilitating the alignment with continuous logits from teacher ANN. Extensive experiments on multiple datasets demonstrate the effectiveness of our methods.

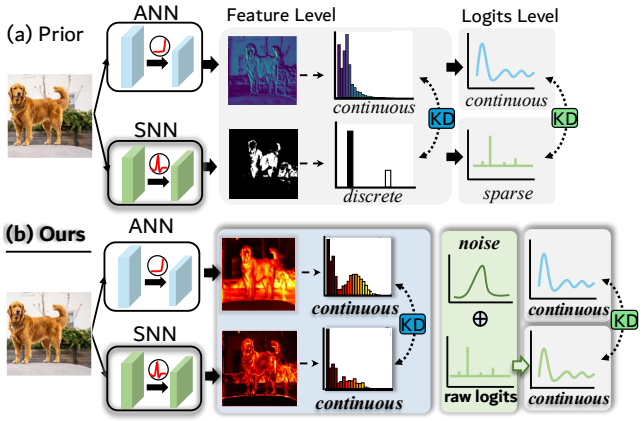

Figure 1: (a) Prior KD methods simply aligns the raw hidden features and output logits between teacher ANN and student SNN, ignoring discrepancies in their distributions. (b) We perform the KD through more precise and semantic-consistent saliency maps, aligning the spiking activation map of SNN with the class activation map with ANN. Besides, we utilize Gaussian noise to smooth the raw logits of SNN, reducing the discrepancy in logits distillation.

Introduction

Spiking Neural Networks (SNNs), inspired by the spiking mechanism of biological neurons, utilize event-driven binary spikes to transmit information, allowing multiplications between activations and weights to be replaced by additions or remain silent, thereby significantly improving energy efficiency (Eshraghian et al. 2023; Davies et al. 2018). Taking advantage of this computational paradigm, SNNs can operate efficiently on neuromorphic hardware and demonstrate autonomous learning capabilities and ultralow power consumption (Mehonic and Kenyon 2022), making them highly promising for intelligent computing tasks (Fang et al. 2023; Zhang et al. 2020). However, training SNNs presents significant challenges due to the inherently discrete and sparse nature of spike-based features, which complicates their optimization process and results in performance and application limitations compared to traditional artificial neural networks (ANNs) (Zhou et al. 2021, 2022; Zhou, Guo, and Wang 2023; Zhou et al. 2024d,a,b,e,c; Li et al. 2025, 2024; Qian et al. 2025; Zhao et al. 2025; Zhou et al. 2025a,b; Jin et al. 2025; Zhou et al. 2025c; Kryklyvets et al. 2025; Shen et al. 2023; Song et al. 2022; Guo et al. 2025).

Specifically, conversion-based methods (Bu et al. 2022) transfer pre-trained ANN parameters to corresponding SNNs but replace ReLU activation with spiking neurons (Huang et al. 2024). This strategy has been experimentally found to require a large number of time steps to achieve satisfactory performance (Bu et al. 2022; Han, Srinivasan, and Roy 2020; Rueckauer et al. 2017). Direct training methods (Wu et al. 2019; Fang et al. 2021; Deng et al. 2022), on the other hand, optimize SNNs using direct backpropagation through the surrogate gradient estimation technique (Zenke and Vogels 2021), leading to significant progress in pattern recognition (Deng et al. 2022; Zhou et al. 2023), natural language processing (Zhang et al. 2024; Xiao et al. 2022), and mul-

*Corresponding authors

timodal tasks (Liu et al. 2025). While this strategy reduces training time steps, the SNN's performance still lags behind that of ANNs.

Recent works (Xu et al. 2024, 2023) improve SNN training using knowledge distillation (KD) (Hinton, Vinyals, and Dean 2014) techniques, with pretrained ANNs as the teacher model and SNNs as the student model, showing promising results on multiple datasets. These methods perform KD by element-wise aligning the output features (Xu et al. 2023) or classification logits (Xu et al. 2023) between ANNs and SNNs. Such a KD paradigm distills knowledge from teacher ANNs to student SNNs, however, prior works ignore two critical issues: **(i) Discrepancy between raw features.** As illustrated in Fig. 1(a), features extracted from ANNs via a single forward pass are represented as *continuous floating-point* values. In contrast, features in SNNs, obtained by forward propagation over multiple time steps, are expressed as *discrete binary spikes*. Moreover, whereas ANN features encapsulate patterns spanning the entire image, SNN spikes primarily highlight *salient regions*. **(ii) Discrepancy between raw logits.** The logits (*i.e.*, raw classification scores) are derived from the hidden features. Consequently, a notable disparity emerges between the raw logits of the two models. Specifically, SNN logits display greater sparsity and a more peaked distribution relative to those of ANNs.

In this paper, we propose novel knowledge distillation strategies, having a Closer look at KD for SNN. Specifically, to address the first challenge, we propose the **Saliency-scaled Activation Map Distillation (SAMD)**. As illustrated in Fig. 1, unlike previous methods that directly perform knowledge distillation using raw features, our SAMD leverages the Class Activation Map (CAM) (Zhou et al. 2016) of the teacher ANN, which provides more precise and focused knowledge, clearly describing the salient image regions related to the target class. Notably, unlike traditional activation map-based distillation methods from ANNs (*e.g.*, e^2KD (Parchami-Araghi et al. 2024) and CATKD (Guo et al. 2023c)), we discover that the surrogate gradient estimation in SNNs prevents the use of precise gradient estimation methods like Grad-CAM (Selvaraju et al. 2017) to generate saliency activation maps. Instead, we redesign the activation map distillation by aligning the Spiking Activation Map (SAM) of the student SNN with the CAM of the teacher ANN. Although both SAM and CAM originate from features, they are more consistent than raw features due to the use of saliency maps. Furthermore, we consider the numerical magnitude differences between the activation maps generated from SNN features (*i.e.*, SAM) and ANN features (*i.e.*, CAM). To more accurately assess the contribution of each pixel in the saliency maps, we apply the softmax function to convert both CAM and SAM into probability distributions. In this way, the scaled CAM and SAM remain consistent in both semantics and numerical magnitude, facilitating their alignment. To address the second challenge, we propose the **Noise-smoothed Logits Distillation (NLD)**. As demonstrated in Fig. 1(b), NLD employs Gaussian noise to moderate the prediction logits of student SNNs. Specifically, we sample Gaussian noise with mean and variance parameters derived from the SNN logits, ensuring that the original distribution of these logits remains largely preserved. After the addition of noise, the logits of SNNs transit from a sparse and sharply peaked distribution to one that is denser and broader, resembling the distribution of ANN logits, thereby facilitating knowledge transmission between teacher and student. We evaluate the effectiveness and superiority of the proposed two KD strategies on CIFAR-10, CIFAR-100, and ImageNet-1K da of SNNs.

In summary, our main contributions are as follows:

- We propose a saliency-scaled activation map distillation strategy that directs the student SNN's spike activation map to align with the teacher ANN's class activation map, emphasizing spike generation in salient image regions to improve knowledge transfer.
- We propose a noise-smoothed logits distillation strategy that employs Gaussian noise to moderate the sparse logits of the student SNN, facilitating alignment with the continuous logits of the teacher ANN.
- Our method achieves new state-of-the-art performance on multiple datasets and can be flexibly integrated into existing KD approaches for SNN training, maintaining a good balance between accuracy and energy efficiency.

Related Work

Spiking Neural Networks (SNNs) are brain-inspired models that mimic biological neural systems by transmitting information through discrete spikes, achieving lower energy consumption. SNNs are typically trained using two main approaches: ANN-SNN conversion (Meng et al. 2022; Bu et al. 2022; Deng and Gu 2021; Hu, Tang, and Pan 2023) and direct training (Fang et al. 2021; Guo et al. 2023a,b; Deng et al. 2022; Meng et al. 2023). The conversion methods directly transform pre-trained ANN into SNN by replacing its ReLU activation functions with integrate-and-fire (IF) (Bu et al. 2022) neurons. However, the converted SNNs often require prolonged time steps to collect sufficient spike signals to ensure accuracy. The direct training methods directly train the SNN by backpropagating surrogate gradients (Fang et al. 2021) through multiple time steps, which can alleviate excessive time steps, enabling efficient training and inference. However, a significant accuracy gap persists between the trained SNNs and ANNs due to the approximation errors (Xu et al. 2023) inherent in surrogate gradients. Unlike them, our method utilizes the informative knowledge in pretrained ANNs to better supervise SNN training, achieving balance between accuracy and efficiency.

Knowledge Distillation (KD) (Hinton, Vinyals, and Dean 2014) was initially proposed for compressing artificial neural networks (ANNs) by transferring knowledge from complex teacher models to lightweight student models. Traditional logits-based KD methods typically minimize the difference in classification probabilities (*i.e.*, logits) between the student and teacher models (Sun et al. 2024; Jin, Wang, and Lin 2023). In addition, feature-based KD methods enhance the performance of the student model by learning the feature representations of the teacher model (Wang et al. 2025; Guo et al. 2023c). Recently, a few works start to apply KD technique to facilitate SNN training (Xu et al. 2024, 2023), significantly improving the model's performance. *i.e.*, KDSNN (Xu et al.

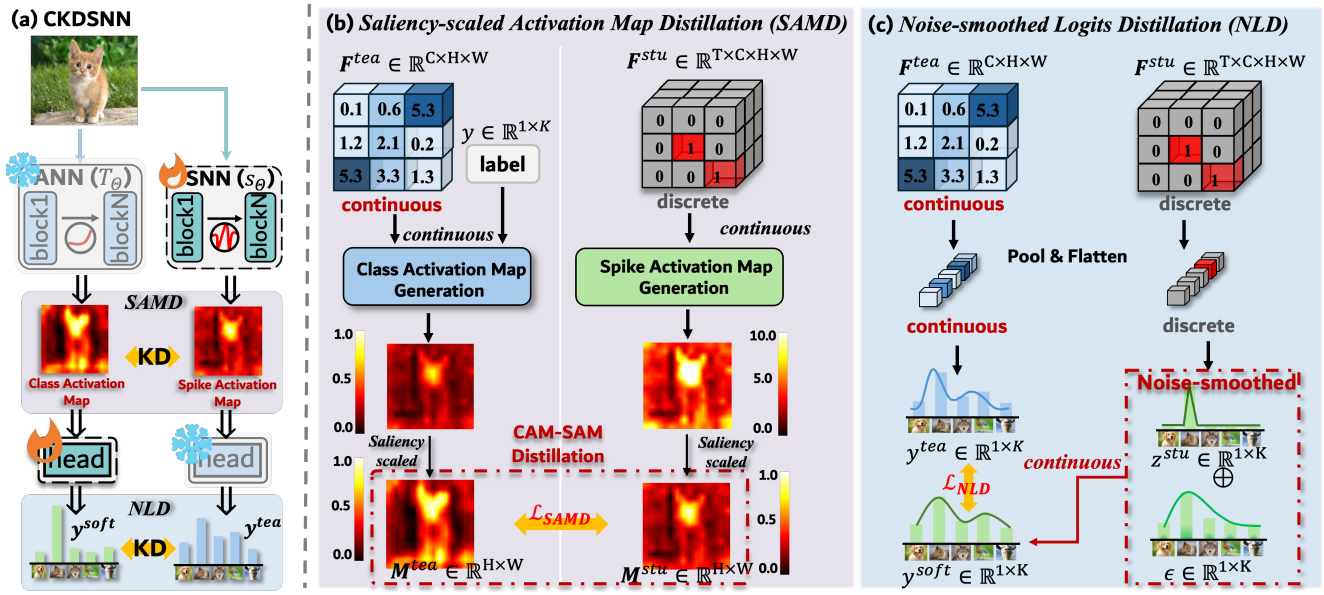

Figure 2: Overview of our CKDSNN. (a) CKDSNN framework aims to improve the student SNN training by distilling knowledge from a pretrained teacher ANN. (b) The Saliency-scaled Activation Map Distillation (SAMD) utilizes the class activation map (CAM) from the ANN to guide the SNN to generate precise spike activations in salient regions. (c) The Noise-smoothed Logits Distillation (NLD) utilizes Gaussian noise to soften the sparse logits of the SNN, better matching with logits of the ANN.

2023) regularizes consistency of the output features and logits between ANNs and SNNs, while BKDSNN (Xu et al. 2024) enhances feature-level matching by further processing the spike features of SNNs with a blurring matrix. However, these methods largely overlook the discrepancies of the raw features and logits from teacher ANNs and student SNNs. Specifically, ANNs generate *continuous floating-point* features, while SNNs generate *discrete spike features*. The logits of SNNs are also more sparse than that of ANNs. Thus, we perform KD from the perspective of semantic saliency activation maps, which are more semantically aligned. But, prior activation map-based KD methods (Parchami-Araghi et al. 2024; Zagoruyko and Komodakis 2017) from ANNs cannot be directly applied to SNNs, due to the surrogate *gradient estimation errors* in SNNs, which affect the generation of semantic saliency activation maps. In contrast, we design a semantic saliency activation map-based KD method for the characteristics of SNNs, which can be applied to various architectures. Moreover, we also design a noise-smoothing distillation method at the logits-level to further enhance the performance of SNNs.

Method

The overall pipeline of our CKDSNN framework is illustrated in Fig. 2. Given a pretrained ANN teacher model and a learnable SNN student model, the proposed CKDSNN aims to train the student model by effectively distilling knowledge from the teacher model from two aspects: **1) Saliency-scaled Activation Map Distillation (SAMD)**. The class activation map (Selvaraju et al. 2017) obtained from the teacher ANN is used to guide the student SNN to fire spikes in salient image regions. During the distillation process, we address the magnitude discrepancy between the two types of activation maps by scaling them into the same range. **2) Noise-smoothed Logits Distillation (NLD).** The classification output logits of the student model is smoothed using additional Gaussian noise, making the vanilla sparse logits distribution of student SNN close to the continuous logits distribution of teacher ANN. This facilitates more precise logit-level knowledge distillation. We first have a brief introduction to the spiking neuron used in SNN, which leads to the discrepancy on the features and logits of ANN and SNN. Then, we elaborate on the proposed SAMD and NLD strategies, respectively.

Discrepancy Caused by SNN Spiking Neuron

Typical SNNs (Fang et al. 2021; Deng et al. 2022; Meng et al. 2023; Jiang et al. 2024; Deng et al. 2024) adopt the integrate-and-fire (IF) neuron as the fundamental unit. Specifically, the IF neuron first integrates input currents by updating its membrane potential, and then compares it with a pre-set threshold to generate a spike signal, followed by a reset mechanism of the membrane potential. This process can be formulated as:

$$\begin{aligned} H[t] &= V[t-1] + I[t], \\ S[t] &= \Theta(H[t] - V_{\text{th}}) = \begin{cases} 1, & H[t] \geq V_{\text{th}}, \\ 0, & H[t] < V_{\text{th}}, \end{cases} \\ V[t] &= H[t](1 - S[t]) + V_{\text{reset}} S[t], \end{aligned} \tag{1}$$

where $I[t]$ is the input current at time step t and $V[t-1]$ is the membrane potential at previous $t-1$ time step. γ denotes the membrane time constant and $\Theta(\cdot)$ represents the Heaviside function (Huang et al. 2024). The IF accumulates input

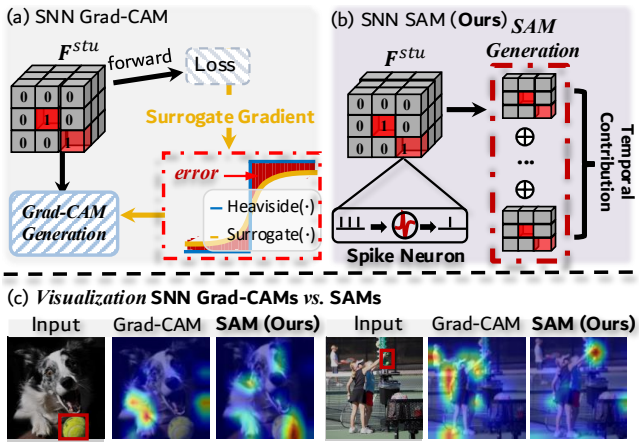

Figure 3: Illustration of (a) the main challenge of applying Grad-CAM-like strategies in SNNs is the error caused by surrogate gradients. (b) Our SAM directly computes the spike activation rate of SNNs via SAM Generation. (c) Visualization of the generated CAMs and SAMs for SNN.

currents to update the membrane potential $H[t]$. When $H[t]$ exceeds the threshold V_{th}, a spike $S[t]$ is generated, and the membrane potential is reset to V_{reset} (Meng et al. 2023).

In image classification, the above is attached to each feature encoding block of SNN. Specifically, each encoding block contains multiple linear transformation layers, batch normalization layers, and IF spiking neurons, as shown in Fig. 2(a). This generates *discrete* spike features $\boldsymbol{F}^{st}\in\mathbb{R}^{T\times C\times H\times W}$ over T time steps, where C, H, and W denote the number of channels, height, and width of the feature, respectively. The discrete spike features $\boldsymbol{F}^{st}$ generated by the IF activation mechanism differ significantly from the *continuous* features $\boldsymbol{F}^{te}\in\mathbb{R}^{C\times H\times W}$ produced by ANN. This leads to a significant discrepancy in the feature distribution between $\boldsymbol{F}^{st}$ and $\boldsymbol{F}^{te}$, resulting in different prediction logits. Prior KD methods (Xu et al. 2023) directly match these two types of features or logits element-wise, ignoring the essential differences in feature representation and distribution, leading to suboptimal distillation results. We address these issues through carefully designed KD strategies.

Saliency-scaled Activation Map Distillation

We propose the Saliency-scaled Activation Map Distillation (SAMD) to train SNN rather than distilling the element-wise discrepant features used in existing works (Xu et al. 2023, 2024). Specifically, SAMD consists of three steps:

1) Class Activation Map (CAM) Generation. The class activation map is initially used as a visualization tool to enhance the explainability of convolutional neural network (Zhou et al. 2016), which can highlight the related image region of a specific class. In our work, we use it as the teacher's knowledge for distillation. We follow the typical method of Grad-CAM (Selvaraju et al. 2017) to generate the class activation map. Specifically, given an input image x, we first extract the intermediate features using a pretrained teacher ANN model, denoted as $\boldsymbol{F}^{te}\in\mathbb{R}^{C\times H\times W}$. Then, the gradients of $\boldsymbol{F}^{te}$ are calculated according to the forward loss between prediction and the ground truth label $y\in\mathbb{R}^{K}$ (K is the total number of classes). The class activation map $\boldsymbol{M}^{te}\in\mathbb{R}^{H\times W}$ can be obtained by associating $\boldsymbol{F}^{te}$ with the class label y. This process can be formulated as follows:

$$\begin{aligned}\alpha &= \frac{1}{W\cdot H}\sum_{i=1}^{W}\sum_{j=1}^{H}\frac{\partial y}{\partial \boldsymbol{F}^{te}_{i,j}},\\ \boldsymbol{M}^{te} &= \text{ReLU}\left(\sum_{k=1}^{K}\alpha_k \boldsymbol{F}^{te}_{k}\right),\end{aligned} \tag{2}$$

where α, the gradient-based weights from label y, measures each channel's contribution to the target class. Using α for a weighted sum of the feature map $\boldsymbol{F}^{te}$ across the channel dimension C yields the activation map for input image x.

2) Spike Activation Map (SAM) Generation. Unlike the gradient-based CAM generation in ANNs, the gradient estimation error in SNNs leads to inaccurate activation maps. As shown in Fig. 3(a), the gradient-based activation map is not accurate in SNNs, mainly due to the *gradient estimation error*. Therefore, as show in Fig. 3(b), we abandon the gradient-based CAM generation method and design a SAM generation method that directly computes the spike activation map based on the spikes in features. As shown in Fig. 3(c), we leverage the characteristics of SNNs, where spikes are generated only in salient regions, and consider the contribution of spikes at different time steps, *i.e.*, spikes at each time step t are accumulated into the SAM. Specifically, the intermediate features $\boldsymbol{F}^{st}\in\mathbb{R}^{T\times C\times H\times W}$ can be obtained from a student SNN. $\boldsymbol{F}^{st}$ indicates the activated spikes over T time steps. For each time step, the spatial regions are fired with spikes in different channels, where each channel captures key saliency information related to the target class. Therefore, $\boldsymbol{F}^{st}$ is able to reveal the informative and salient regions. So we generate the spike activation map $\boldsymbol{M}^{st}\in\mathbb{R}^{H\times W}$ by directly averaging $\boldsymbol{F}^{st}$ in the channel and time dimensions, computed as,

$$\boldsymbol{M}^{st}=\sum_{t=1}^{T}\sum_{c=1}^{C}\boldsymbol{F}^{st}_{t,c}. \tag{3}$$

$\boldsymbol{M}^{st}$ leverages the characteristics of SNN and is able to integrate semantic information across multiple time steps. Unlike the class activation map generation, this process is computationally efficient and is performed online, allowing later dynamical distillation learning through forward and backward propagation.

3) Saliency-scaled CAM-SAM Distillation. Although both the CAM and SAM highlight the salient regions (semantic aligned), there is a discrepancy between $\boldsymbol{M}^{te}$ and $\boldsymbol{M}^{st}$ in the feature magnitude. This is because that $\boldsymbol{M}^{st}$ is the summarization of discrete SNN binary 0/1 spikes, while $\boldsymbol{M}^{te}$ is derived by weighting float-point ANN features that range between 0 and 1. To address this, as shown in Fig. 2(b), we propose saliency scaling that normalizes $\boldsymbol{M}^{st}$ and $\boldsymbol{M}^{te}$ to probability distributions on the same scale using a softmax function: $f(\boldsymbol{M})=\exp\left(\frac{\boldsymbol{M}}{\mathcal{T}}\right)/\sum_{i=1}^{WH}\exp\left(\frac{\boldsymbol{M}}{\mathcal{T}}\right)$, where $\mathcal{T}$ is a constant to control the distribution smoothness.

Let $\boldsymbol{P}^{te}$ and $\boldsymbol{P}^{st}$ be the normalized activation map. The CAM-SAM distillation regularizes the consistency of $\boldsymbol{P}^{te}$ and $\boldsymbol{P}^{st}$. This is achieved by computing the Kullback–Leibler (KL) divergence loss $\mathcal{L}_{\text{SAMD}}$:

$$\begin{aligned}\mathcal{L}_{\text{SAMD}} &= \mathcal{T}^2 \cdot \text{KL}\left(\boldsymbol{P}^{te} \| \boldsymbol{P}^{st}\right) \\ &= \mathcal{T}^2 \cdot \sum_{i=1}^{H}\sum_{j=1}^{W} \boldsymbol{P}_{ij}^{te} \log\left(\frac{\boldsymbol{P}_{ij}^{te}}{\boldsymbol{P}_{ij}^{st}}\right). \end{aligned} \tag{4}$$

Noise-smoothed Logits Distillation

In addition to the feature-driven activation map distillation, we consider the target-driven logits distillation, which aims to align the output probability logits of teacher ANN and student SNN. Although prior work (Xu et al. 2023) considered this, they largely overlook the discrepancy between the logits. This issue still originates from the utilization of different features in logits production where SNNs employ binary spike features that are either 0 or 1, whereas ANNs utilize floating-point features. Consequently, the logits of SNN are very sparse, while those of ANN are more dense and *continuous*. We propose the Noise-Smoothed Logits Distillation (NLD) to address this problem. Thus, we try to soften the logits of student SNN, $z^{st}{\in}\mathbb{R}^{1\times K}$, reducing the discrepancy from $z^{te}{\in}\mathbb{R}^{1\times K}$. Specifically, we add some continuous noise $\epsilon{\in}\mathbb{R}^{1\times K}$ onto z^{st}. To avoid destroying the original distribution of z^{st} while maintaining the characteristic of classification logits, we opt to Gaussian noise with the mean and standard deviation of z^{st}:

$$\epsilon \sim \mathcal{N}(\bar{z}^{st}, \sigma(z^{st})^2), \tag{5}$$

where $\mathcal{N}$ denotes the Gaussian distribution, $\bar{z}$ and $\sigma(z)$ are the mean and standard variance.

Then, the Gaussian noise ϵ is fused with z^{st} using a balance hyper-parameter λ, computed as,

$$z^{soft} = z^{st} + \lambda\epsilon, \tag{6}$$

where z^{soft} is the noise-softened logits of student SNN. This process introduces randomness through noise, promoting the exploration of a broader decision boundary for classification.

The softened logits z^{soft} of student SNN and the raw logits z^{te} from the teacher ANN are processed by a softmax function to generate classification probability y^{soft} and y^{te}, formulated as $y{=}f(z){=}\exp(z_i/\tau)/\sum_{k=1}^{K}\exp(z_k/\tau)$. Then, logits distillation can be performed by aligning y^{soft} with y^{te} through a KL loss:

$$\mathcal{L}_{\text{NLD}} = \tau^2 \cdot \text{KL}\left(y^{te} \| y^{soft}\right). \tag{7}$$

Overall Training Loss

Given the student SNN's prediction y^{st} and the ground truth label y, we can obtain the standard cross-entropy loss $\mathcal{L}_{\text{CE}}$. Then, the total objective for model optimization $\mathcal{L}_{\text{total}}$ is calculated by summarizing the above three losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \beta\mathcal{L}_{\text{SAMD}} + \gamma\mathcal{L}_{\text{NLD}}, \tag{8}$$

where β and γ are hyper-parameters to balance the two distillation losses.

Experiments

Experimental Setups

Datasets. We evaluate our method on three widely used image classification datasets and a neuromorphic dataset in this research area: CIFAR-10/100 (Krizhevsky and Hinton 2009), ImageNet-1K (Deng et al. 2009) and CIFAR10-DVS (Li et al. 2017). The details of these datasets are provided in the supplementary materials.

Model Configuration. To ensure fair comparison, we determine the configurations of teacher ANN and student SNN models based on prior studies (Guo et al. 2024; Xu et al. 2024). Specifically, for the CIFAR-10/100 datasets, we use the ResNet-19/20 versions provided by TET (Deng et al. 2022) and QCFS (Li et al. 2023), with their corresponding ANN versions as the teacher models; and we also test a ViT-based architecture, using ViT-S as the teacher ANN and Spikformer-4-384 (Zhou et al. 2023) as the student model. For the ImageNet-1K dataset, we use the ResNet-18 and ResNet-34 pre-trained on ImageNet-1K as teacher models, while SEW-ResNet (Fang et al. 2021) serves as the student model. For the CIFAR10-DVS dataset, we use ResNet-19 as the student, with its ANN architecture serving as the teacher, trained in the same way as EnOF (Guo et al. 2024).

Implementation Details. To ensure consistency with prior studies, our implementation on CIFAR-10/100 and ImageNet strictly follows the established distillation architecture (Guo et al. 2024; Xu et al. 2024). Specifically, we set the hyperparameters as follows: $\mathcal{T}$ in Eq. 4 and τ in Eq. 7 to 2.0. λ in Eq. 6 to 0.1, β and γ in Eq. 8 to 1.0. The Integrate-and-Fire (IF) neuron settings align with prior works (Fang et al. 2021), and all other training configurations, including batch size, learning rate, and optimizer, remain consistent with those in (Xu et al. 2024). In addition, we conduct sensitivity analysis of all hyper-parameters in the supplementary materials.

Comparison with State-of-the-Arts

We compare our CKDSNN with three types of SNN training approaches to evaluate its effectiveness: 1) *ANN-to-SNN*: Conversion of a pre-trained ANN into SNN. 2) *Direct Training*: Direct training of SNN from scratch. 3) *SNN-KD*: Training SNN using knowledge distillation with the aid of ANN.

Results on CIFAR-10/100. The comparison results between our CKDSNN and previous methods are shown in Tab. 1. We significantly outperform existing methods across different architectures. For example, in the ResNet-19, when the time step is set to 1, CKDSNN achieves an accuracy improvement of 0.74% on the CIFAR-10 dataset and 1.03% on the CIFAR-100 dataset compared to the current most competitive knowledge distillation method, EnOF (Guo et al. 2024), reaching new SOTA accuracies of 96.11% and 78.15%, respectively. Moreover, notably, as the time step increases, CKDSNN continues to significantly outperform prior works.

Results on ImageNet-1K (IN-1K). The comparison results are presented in Tab. 2. The proposed CKDSNN continues to outperform previous SOTA methods using various teacher models, including ResNet-18, ResNet-34, and ResNet-50. Specifically, compared to the prior SOTA methods BKD-SNN (Xu et al. 2024), our CKDSNN improves the top-1

Methods	Venue	Time step	ResNet20 CIFAR10	ResNet20 CIFAR100	ResNet19 CIFAR10	ResNet19 CIFAR100	Spikformer-4-384 CIFAR10	Spikformer-4-384 CIFAR100
ANN-to-SNN								
QCFS (Bu et al. 2022)	ICLR'22	64	92.35	55.37	-	-	-	-
Direct Training								
SEW-R (Fang et al. 2021)	NIPS'21	4	89.07	60.16	93.24	70.84	-	-
STBP (Wu et al. 2019)	AAAI'21	4	-	-	92.92	-	-	-
TET (Deng et al. 2022)	ICLR'22	4	-	-	94.44	74.47	-	-
SLTT (Meng et al. 2023)	ICCV'23	4	-	-	94.56	74.67	-	-
Spikformer (Zhou et al. 2023)	ICLR'22	4	-	-	-	-	95.93	79.65
SNN-KD								
KDSNN (Xu et al. 2023)	CVPR'23	4	89.03	60.18	94.36	74.08	95.88	80.33
BKDSNN (Xu et al. 2024)	ECCV'24	4	89.29	60.92	94.64	74.95	96.06	81.26
EnOFSNN (Guo et al. 2024)	NIPS'24	1	92.66	70.38	95.37	77.08	-	-
		2	93.86	71.55	96.19	82.43	-	-
CKDSNN (Ours)	-	1	**92.85**	**72.45**	**96.11**	**79.11**	**96.93**	**83.07**
CKDSNN (Ours)	-	2	**93.53**	**73.67**	**97.13**	**83.21**	**96.98**	**84.53**
CKDSNN (Ours)	-	4	**94.78**	**73.88**	**97.81**	**83.88**	**97.54**	**84.88**

Table 1: The comparison of Acc↑ (%) with previous works on CIFAR-10/100 datasets. The best results are **bolded**.

Methods	Venue	Time step	ResNet18	ResNet34
ANN-to-SNN				
QCFS (Bu et al. 2022)	ICLR'22	64	-	72.35
Direct Training				
SEW-R (Fang et al. 2021)	NIPS'21	4	63.18	67.04
RMP-Loss (Guo et al. 2023a)	ICCV'23	4	63.14	64.71
MBPN (Guo et al. 2023b)	ICCV'23	4	63.03	65.17
SNN-KD				
KDSNN (Xu et al. 2023)	CVPR'23	4	63.42	67.18
EnOFSNN (Guo et al. 2024)	NIPS'24	4	65.31	67.40
BKDSNN (Xu et al. 2024)	ECCV'24	4	65.60	71.24
CKDSNN (Ours)	-	4	**66.92**	**73.05**

Table 2: Comparison of Top-1 Acc↑ (%) on IN-1K dataset.

	w/o Scaled	L2-norm	Z-score	**Softmax**
Acc↑ (%)	75.56	76.48	74.78	**79.11**

Table 3: Ablation study on the saliency-scaling strategies.

accuracy by 1.32%, 1.81%, and 1.52% using the three types of network architectures, respectively. These results indicate that CKDSNN is effective on the large-scale dataset.

Results on CIFAR10-DVS. The neuromorphic dataset comparison results are shown in Tab. 4. Our CKDSNN also outperforms existing methods on the neuromorphic dataset. For example, under the same architecture and time step settings, CKDSNN achieves an accuracy improvement of 1.05% compared to the most competitive EnOFSNN (Guo et al. 2024).

Ablation Study

We conduct additional ablation experiments to analyze the effectiveness of our proposed strategies. Unless otherwise specified, all experiments are validated based on the **ResNet-19** architecture and the **CIFAR-100** dataset.

Effectiveness of our core KD strategies. Our method primarily consists of the saliency-scaled activation map distillation (SAMD) and noise-smoothed logits distillation (NLD). As presented in Fig. 4 (a), without using either SAMD and NLD,

Methods	Venue	Arch.	Time step	Acc↑(%)
STBP (Wu et al. 2019)	NIPS'21	ResNet19	4	67.80
SEW-R (Fang et al. 2021)	NIPS'21	WideNet	16	74.40
KDSNN (Xu et al. 2023)	CVPR'23	ResNet20	10	78.31
BKDSNN (Xu et al. 2024)	ECCV'24	ResNet20	10	79.53
EnOFSNN (Guo et al. 2024)	NIPS'24	ResNet20	10	80.50
CKDSNN (Ours)	-	ResNet20	10	**81.55**

Table 4: Comparison of Acc↑ (%) on CIFAR10-DVS dataset.

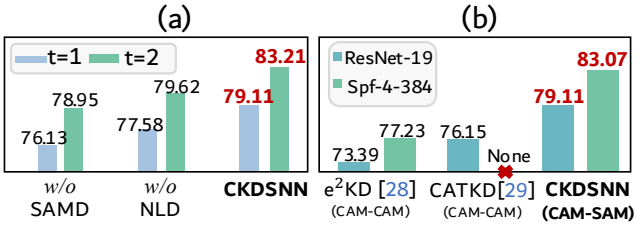

Figure 4: (a) Effectiveness of our core KD strategies. (b) Importance of the CAM-SAM distillation in SAMD.

the model's performance significantly decreases across various experimental setups. This indicates the effectiveness and necessity of each proposed KD strategy.

Different saliency-scaling manners in SAMD. In SAMD, we use the softmax function to re-scale the class activation map $\boldsymbol{M}^{te}$ and spike activation map $\boldsymbol{M}^{st}$ into the same range. We also try other scaling methods, including *w/o* Scaled, Z-score, and L2-norm. As reported in Tab. 3, the softmax scaling strategy exceeds these potential choices by around 2 to 3 points. When no scaling is applied, although the original value characteristics are preserved, the significant difference in magnitude between the class activation map and spike activation map leads to a significant decrease in saliency alignment effectiveness. Further analysis in Supp. material shows that the softmax scaling strategy effectively normalizes and identifies the most salient regions.

Importance of the CAM-SAM distillation in SAMD. As

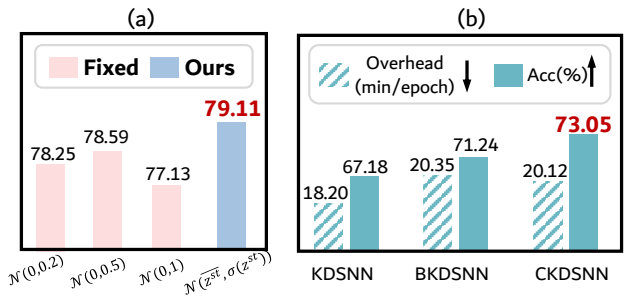

Figure 5: Comparison of (a) different noise strategies on CIFAR100 and (b) training overhead on IN-1K.

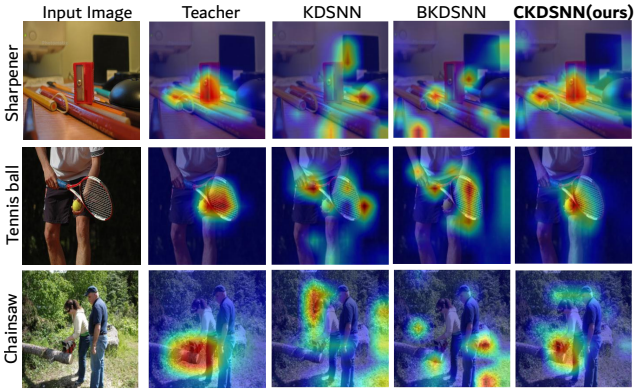

Figure 6: Visualization of the spike activation maps from different kd strategies on IN-1K.

show in Fig. 4 (b), when using ANN-based strategies (*e.g.*, e^{2}KD (Parchami-Araghi et al. 2024) or CATKD (Guo et al. 2023c)) to distill SNNs, the performance is significantly lower than our proposed CAM-SAM distillation. Specifically, They use gradient to generate activation maps, but the gradient estimation error leads to performance degradation. Additionally, CATKD is only applicable to CNN-base architectures. In contrast, SAM is designed on the characteristics of SNNs, enabling compatibility with various architectures.

Effect of the logits noise-smoothing in NLD. As show in Fig. 5 (a), our adaptive noise strategy significantly outperforms random noise with the fixed standard deviation: small noise yields poor results, increasing noise improves performance but still lags behind our method, while excessive noise leads to performance degradation.

Qualitative Analysis

Spiking Activation Maps. Fig. 6 visualizes the learned spike activation maps (SAMs) of our method compared to other KD-based approaches. Our SAM aligns more closely with the teacher model's class activation maps (CAMs). For instance, it more accurately captures key regions such as the 'sharpener' and 'chainsaw', indicating that spikes are precisely emitted on salient areas and the teacher's CAM knowledge is effectively distilled into the student's SAM.

Spiking Features. We extract spiking features from the final layer of our CKDSNN model and visualize them using

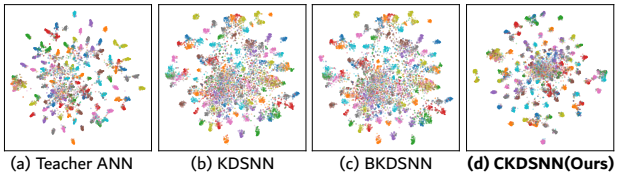

Figure 7: The t-SNE visualization of feature distributions.

Methods	Fire Rate ↓ (%)	SOPS ↓ (G)	Power ↓ (mJ)	Acc↑ (%)	Time Step
SEW-*R* (Fang et al. 2021)	18.0	4.14	4.03	67.04	4
KDSNN (Xu et al. 2023)	16.0	4.13	4.01	67.18	4
BKDSNN (Xu et al. 2024)	15.0	4.02	3.98	71.24	4
CKDSNN (Ours)	10.0	3.92	3.88	72.71	3
	13.0	4.01	3.96	**73.05**	4

Table 5: Comparison of energy efficiency on the ResNet34 with the IN-1k dataset.

t-SNE (Van der Maaten and Hinton 2008). Image samples from the CIFAR-100 dataset are used. As shown in Fig. 7, the spike features generated by CKDSNN demonstrate improved separability and discriminability compared to other KD methods and are close to the teacher's feature distribution. This superiority is largely contributed to the proposed activation map KD, which guides the spiking representations to focus on salient regions, resulting in better feature discrimination.

Efficiency Analysis

Energy Efficiency. We evaluate the energy efficiency of our method by calculating the Fire rate, SOPS, and Power consumption following prior methods (Zhou et al. 2023; Xu et al. 2024; Fang et al. 2023). As shown in Tab. 5, compared with prior KD methods, our CKDSNN not only achieves state-of-the-art performance but offers higher energy efficiency. *e.g.*, CKDSNN with 4 time steps outperforms BKDSNN by 1.81% while using slightly less fire rate and Power. This demonstrates that our method has a better trade-off between the energy-efficiency and accuracy.

Training overhead. We analyze the training overhead of our CKDSNN method. As reported in Fig. 5, CKDSNN incurs a training overhead that is higher than KDSNN but lower than BKDSNN, while achieving the best performance.

Conclusion

This paper takes a closer look at current SNN training methods using knowledge distillation (KD) techniques and finds that the discrepancies of features and logits between teacher ANNs and student SNNs are largely overlooked. We propose two novel KD strategies. The saliency-scaled activation map distillation aligns spike activation map from student SNN with the class activation map from teacher ANN. The noise-smoothed logits distillation aligns the teacher ANN's classification logits with student SNN's logits softened by Gaussian noise. In this way, the saliency activation map and the logits from the teacher and student are more semantic- and distribution-consistent, guaranteeing more effective knowledge distillation in SNN training. Extensive experiments demonstrate the effectiveness and robustness of our method.

Acknowledgements

This work was supported in part by the National Key R&D Program of China (Grant No. 2024YFB3311602); the National Natural Science Foundation of China (Grants 61971178, 62501224, 62272144, and U20A20183); the Ordos Science and Technology Major "Open Bidding" Project (Grant No. JBGS-2023-002); the State Grid Co., Ltd. Headquarters Technology Project (Grant No. 5500-202140127); the Major Project of Anhui Province (Grant No. 202423k09020001); the Anhui Provincial Natural Science Foundation for Distinguished Young Scholars (Grant No. 2408085J040); the Fundamental Research Funds for the Central Universities (Grants JZ2024HGTG0309, JZ2024AHST0337, JZ2024AKKG0507); and the Anhui Provincial Water Conservancy Science and Technology Project (Grant No. slky202501-06). Wuhu Anpu Intelligent Production Line Co., Ltd. under Grant W2024JSKF0177.

References

Bu, T.; Fang, W.; Ding, J.; Dai, P.; Yu, Z.; and Huang, T. 2022. Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks. In *ICLR*.

Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S. H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1): 82–99.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Deng, S.; and Gu, S. 2021. Optimal conversion of conventional artificial neural networks to spiking neural networks. In *ICLR*.

Deng, S.; Li, Y.; Zhang, S.; and Gu, S. 2022. Temporal efficient training of spiking neural network via gradient reweighting. In *ICLR*.

Deng, S.; Wu, Y.; Du, K.; et al. 2024. Spiking token mixer: an event-driven friendly former structure for spiking neural networks. In *NeurIPS*.

Eshraghian, J. K.; Ward, M.; Neftci, E.; Wang, X.; Lenz, G.; Dwivedi, G.; Bennamoun, M.; Jeong, D. S.; and Lu, W. D. 2023. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9): 1016–1054.

Fang, W.; Chen, Y.; Ding, J.; Yu, Z.; Masquelier, T.; Chen, D.; Huang, L.; Zhou, H.; Li, G.; and Tian, Y. 2023. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40): 1480.

Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021. Deep residual learning in spiking neural networks. In *NeurIPS*.

Guo, R.; Ying, X.; Chen, Y.; Niu, D.; Li, G.; Qu, L.; Qi, Y.; Zhou, J.; Xing, B.; Yue, W.; et al. 2025. Audio-visual instance segmentation. In *CVPR*, 13550–13560.

Guo, Y.; Liu, X.; Chen, Y.; Zhang, L.; Peng, W.; Zhang, Y.; Huang, X.; and Ma, Z. 2023a. RMP-Loss: Regularizing membrane potential distribution for spiking neural networks. In *CVPR*, 17391–17401.

Guo, Y.; Peng, W.; Liu, X.; Chen, Y.; Zhang, Y.; Tong, X.; Jie, Z.; and Ma, Z. 2024. Enof-snn: Training accurate spiking neural networks via enhancing the output feature. *NeurIPS*, 51708–51726.

Guo, Y.; Zhang, Y.; Chen, Y.; Peng, W.; Liu, X.; Zhang, L.; Huang, X.; and Ma, Z. 2023b. Membrane potential batch normalization for spiking neural networks. In *CVPR*, 19420–19430.

Guo, Z.; Yan, H.; Li, H.; and Lin, X. 2023c. Class attention transfer based knowledge distillation. In *CVPR*, 11868–11877.

Han, B.; Srinivasan, G.; and Roy, K. 2020. RMP-SNN: Residual Membrane Potential Neuron for Enabling Deeper High-Accuracy and Low-Latency Spiking Neural Network. In *CVPR*.

Hinton, G.; Vinyals, O.; and Dean, J. 2014. Distilling the Knowledge in a Neural Network. In *NeurIPS*.

Hu, Y.; Tang, H.; and Pan, G. 2023. Spiking Deep Residual Networks. *TNNLS*, 34(8): 5200–5205.

Huang, Y.; Lin, X.; Ren, H.; Zhou, Y.; Liu, Z.; Fu, H.; Pan, B.; and Cheng, B. 2024. CLIF: Complementary leaky integrate-and-fire neuron for spiking neural networks. In *ICML*.

Jiang, H.; Zoonekynd, V.; Masi, G. D.; Gu, B.; and Xiong, H. 2024. TAB: Temporal accumulated batch normalization in spiking neural networks. In *ICLR*.

Jin, D.; Zhou, Y.; Zhou, J.; Ma, J.; Guo, R.; and Guo, D. 2025. SimToken: A Simple Baseline for Referring Audio-Visual Segmentation. *arXiv preprint arXiv:2509.17537*.

Jin, Y.; Wang, J.; and Lin, D. 2023. Multi-level logit distillation. In *CVPR*, 24276–24285.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images.

Kryklyvets, Y.; Kurpath, M. I.; Mullappilly, S. S.; Zhou, J.; Khan, F. S.; Anwer, R. M.; Khan, S.; and Cholakkal, H. 2025. MAviS: A Multimodal Conversational Assistant For Avian Species. In *EMNLP*, 28601–28627.

Li, H.; Liu, H.; Ji, X.; Li, G.; and Shi, L. 2017. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11: 309.

Li, Y.; Geller, T.; Kim, Y.; and Panda, P. 2023. SEENN: Towards temporal spiking early exit neural n xetworks. In *Proceedings of the NeurIPS*.

Li, Z.; Guo, D.; Zhou, J.; Zhang, J.; and Wang, M. 2024. Object-aware adaptive-positivity learning for audio-visual question answering. In *AAAI*, 3306–3314.

Li, Z.; Zhou, J.; Zhang, J.; Tang, S.; Li, K.; and Guo, D. 2025. Patch-level sounding object tracking for audio-visual question answering. In *AAAI*, 5075–5083.

Liu, X.; Xia, N.; Zhou, J.; Li, Z.; and Guo, D. 2025. Towards energy-efficient audio-visual classification via multimodal interactive spiking neural network. *TOMM*, 21(5): 1–24.

Mehonic, A.; and Kenyon, A. J. 2022. Brain-Inspired Computing Needs a Master Plan. *Nature*, 604(7905): 255–260.

Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; and Luo, Z. 2023. Towards memory and time efficient backpropagation for training spiking neural networks. In *CVPR*, 6166–6176.

Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; and Luo, Z.-Q. 2022. Training High-Performance Low-Latency Spiking Neural Networks by Differentiation on Spike Representation. In *CVPR*, 12444–12453.

Parchami-Araghi, A.; Böhle, M.; Rao, S.; and Schiele, B. 2024. Good teachers explain: Explanation-enhanced knowledge distillation. In *ECCV*, 293–310. Springer.

Qian, W.; Su, G.; Guo, D.; Zhou, J.; Li, X.; Hu, B.; Tang, S.; and Wang, M. 2025. PhysDiff: Physiology-based Dynamicity Disentangled Diffusion Model for Remote Physiological Measurement. In *AAAI*, 6568–6576.

Rueckauer, B.; Lungu, I.-A.; Hu, Y.; Pfeiffer, M.; and Liu, S.-C. 2017. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11: 682.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *CVPR*, 618–626.

Shen, X.; Li, D.; Zhou, J.; Qin, Z.; He, B.; Han, X.; Li, A.; Dai, Y.; Kong, L.; Wang, M.; et al. 2023. Fine-grained audible video description. In *CVPR*, 10585–10596.

Song, P.; Guo, D.; Zhou, J.; Xu, M.; and Wang, M. 2022. Memorial gan with joint semantic optimization for unpaired image captioning. *IEEE trans. on cybernetics*, 4388–4399.

Sun, S.; Ren, W.; Li, J.; Wang, R.; and Cao, X. 2024. Logit standardization in knowledge distillation. In *CVPR*, 15731–15740.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR*, 9(11): 2579–2605.

Wang, Y.; Cheng, L.; Duan, M.; Wang, Y.; Feng, Z.; and Kong, S. 2025. Improving knowledge distillation via regularizing feature direction and norm. In *ECCV*, 20–37. Springer.

Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Xie, Y.; and Shi, L. 2019. Direct training for spiking neural networks: Faster, larger, better. In *AAAI*, 1311–1318.

Xiao, R.; Wan, Y.; Yang, B.; Zhang, H.; Tang, H.; Wong, D. F.; and Chen, B. 2022. Towards energy-preserving natural language understanding with spiking neural networks. *TASLP*, 31: 439–447.

Xu, Q.; Li, Y.; Shen, J.; Liu, J. K.; Tang, H.; and Pan, G. 2023. Constructing deep spiking neural networks from artificial neural networks with knowledge distillation. In *CVPR*, 7886–7895.

Xu, Z.; You, K.; Guo, Q.; Wang, X.; and He, Z. 2024. BKDSNN: Enhancing the performance of learning-based spiking neural networks training with blurred knowledge distillation. In *ECCV*, 106–123.

Zagoruyko, S.; and Komodakis, N. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *ICLR*.

Zenke, F.; and Vogels, T. P. 2021. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural computation*, 899–925.

Zhang, J.; Shen, J.; Wang, Z.; Guo, Q.; Yan, R.; Pan, G.; and Tang, H. 2024. SpikingMiniLM: energy-efficient spiking transformer for natural language understanding. *Science China Information Sciences*, 67(10): 200406.

Zhang, Y.; Qu, P.; Ji, Y.; Zhang, W.; Gao, G.; Wang, G.; Song, S.; Li, G.; Chen, W.; Zheng, W.; et al. 2020. A System Hierarchy for Brain-Inspired Computing. *Nature*, 586(7829): 378–384.

Zhao, P.; Zhou, J.; Zhao, Y.; Guo, D.; and Chen, Y. 2025. Multimodal class-aware semantic enhancement network for audio-visual video parsing. In *AAAI*, 10448–10456.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *ICCV*, 2921–2929.

Zhou, J.; Guo, D.; Guo, R.; Mao, Y.; Hu, J.; Zhong, Y.; Chang, X.; and Wang, M. 2024a. Towards open-vocabulary audio-visual event localization. *CVPR*.

Zhou, J.; Guo, D.; Mao, Y.; Zhong, Y.; Chang, X.; and Wang, M. 2024b. Label-anticipated event disentanglement for audio-visual video parsing. In *ECCV*, 1–22.

Zhou, J.; Guo, D.; and Wang, M. 2023. Contrastive positive sample propagation along the audio-visual event line. *TPAMI*, 7239–7257.

Zhou, J.; Guo, D.; Zhong, Y.; and Wang, M. 2024c. Advancing weakly-supervised audio-visual video parsing via segment-wise Pseudo Labeling. *IJCV*, 1–22.

Zhou, J.; Li, Z.; Yu, Y.; Zhou, Y.; Guo, R.; Li, G.; Mao, Y.; Han, M.; Chang, X.; and Wang, M. 2025a. Mettle: Meta-Token Learning for Memory-Efficient Audio-Visual Adaptation. *arXiv preprint arXiv:2506.23271*.

Zhou, J.; Shen, X.; Wang, J.; Zhang, J.; Sun, W.; Zhang, J.; Birchfield, S.; Guo, D.; Kong, L.; Wang, M.; and Zhong, Y. 2024d. Audio-visual segmentation with semantics. *IJCV*, 1–21.

Zhou, J.; Wang, J.; Zhang, J.; Sun, W.; Zhang, J.; Birchfield, S.; Guo, D.; Kong, L.; Wang, M.; and Zhong, Y. 2022. Audio–visual segmentation. In *ECCV*, 386–403.

Zhou, J.; Zheng, L.; Zhong, Y.; Hao, S.; and Wang, M. 2021. Positive sample propagation along the audio-visual event line. In *CVPR*, 8436–8444.

Zhou, J.; Zhou, Y.; Han, M.; Wang, T.; Chang, X.; Cholakkal, H.; and Anwer, R. M. 2025b. Think before you segment: An object-aware reasoning agent for referring audio-visual segmentation. *arXiv preprint arXiv:2508.04418*.

Zhou, J.; Zhou, Z.; Zhou, Y.; Mao, Y.; Duan, Z.; and Guo, D. 2025c. Clasp: Cross-modal salient anchor-based semantic propagation for weakly-supervised dense audio-visual event localization. *arXiv preprint arXiv:2508.04566*.

Zhou, Z.; Zhou, J.; Qian, W.; Tang, S.; Chang, X.; and Guo, D. 2024e. Dense audio-visual event localization under cross-modal consistency and multi-temporal granularity collaboration. In *AAAI*.

Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; Yan, S.; Tian, Y.; and Yuan, L. 2023. Spikformer: When spiking neural network meets transformer. In *ICLR*.