Self-Attentive Spatio-Temporal Calibration for Precise Intermediate Layer Matching in ANN-to-SNN Distillation

Di Hong[1,2,3], **Yueming Wang**[1,2,3]*

[1]The College of Computer Science and Technology, Zhejiang University, China
[2]Nanhu Brain-computer Interface Institute, Hangzhou, China
[3]The State Key Laboratory of Brain-Machine Intelligence, Zhejiang University, China
{hongd, ymingwang} @zju.edu.cn

Abstract

Spiking Neural Networks (SNNs) are promising for low-power computation due to their event-driven mechanism but often suffer from lower accuracy compared to Artificial Neural Networks (ANNs). ANN-to-SNN knowledge distillation can improve SNN performance, but previous methods either focus solely on label information, missing valuable intermediate layer features, or use a layer-wise approach that neglects spatial and temporal semantic inconsistencies, leading to performance degradation. To address these limitations, we propose a novel method called *self-attentive spatio-temporal calibration (SASTC)*. SASTC uses self-attention to identify semantically aligned layer pairs between ANN and SNN, both spatially and temporally. This enables the autonomous transfer of relevant semantic information. Extensive experiments show that SASTC outperforms existing methods, effectively solving the mismatching problem. Superior accuracy results include 95.12% on CIFAR-10, 79.40% on CIFAR-100 with 2 time steps, and 68.69% on ImageNet with 4 time steps for static datasets, and 97.92% on DVS-Gesture and 83.60% on DVS-CIFAR10 for neuromorphic datasets. This marks the first time SNNs have outperformed ANNs on both CIFAR-10 and CIFAR-100, shedding the new light on the potential applications of SNNs.

Code — https://github.com/ieveresthd/SASTC.git

Introduction

Spiking Neural Networks (SNNs), considered the third generation of neural networks (Maass 1997), offer a promising advancement in low-power computing. Unlike artificial neural networks (ANNs), which use continuous-valued activations, SNNs emulate the brain's discrete, spike-based information transmission, making them ideal for event-driven and energy-efficient neuromorphic hardware (Akopyan et al. 2015). Two main approaches have emerged for developing supervised deep SNNs: 1) direct training from scratch using surrogate gradients to approximate the discontinuous derivatives of spiking neurons, and 2) ANN-to-SNN conversion, which aligns ANN neuron functions with spiking neurons. Despite progress, a performance gap persists between ANNs and SNNs. To address this, ANN-to-SNN knowledge distillation has been employed to transfer relevant knowledge from ANNs to SNNs.

However, previous distillation methods have either failed to transfer sufficient knowledge or have faced spatial and temporal disparities in semantic information, resulting in degraded performance. This paper introduces a Self-Attentive mechanism to address the semantic mismatch problem by autonomously identifying the most relevant semantic layer patterns across spatial and temporal dimensions and allocating attention based on semantic relevance. The key contributions of this work are summarized as follows:

1. We propose a self-attention mechanism to address semantic mismatching during ANN-to-SNN knowledge distillation by autonomously aligning the most relevant layer patterns between ANN and SNN both spatially and temporally.
2. Through extensive experiments across various settings and prevalent network architectures, our method significantly boosts SNN performance in ANN-to-SNN distillation, surpassing current benchmarks across various datasets, including both static and neuromorphic ones.
3. Our analysis demonstrates that SASTC successfully achieves semantic matching in ANN-to-SNN distillation, advancing its applications in robust representation.

Related Work

Direct Training from Scratch

We briefly summarize some significant achievements in direct training. Lee et al. directly train SNNs in terms of spikes by regarding the membrane potential as the combination of differentiable signals and discontinuous noisy (Lee, Delbruck, and Pfeiffer 2016). Wu et al. use an approximate derivative to construct an iterative LIF neuron model and propose a spatio-temporal backpropagation (STBP) method to train SNNs from scratch (Wu et al. 2018). Zheng et al. propose a threshold-dependent batch normalization (tdBN) method for tuning the loss function (Zheng et al. 2021). Rathi et al. propose to optimize the leakage and threshold in the LIF neuron model. Furthermore, many direct training method have been proposed based on designing various surrogate gradients and coding schemes to achieve SNNs with low latency and high performance (Wu et al. 2019).

*corresponding author.

ANN-to-SNN Conversion

Pérez-Carrasco et al. first map sigmoid neuron model of ANNs into LIF neuron model by utilizing scaling factor, which is determined according to neuron parameters and modified manually (Pérez-Carrasco et al. 2013). Diehl et al. propose to regulate firing rates of SNNs through weight normalization (Diehl et al. 2015). Cao et al. adopt only one hyperparamter, which is the firing threshold of spiking neurons, to approximate the rectified linear unit (ReLU) function of ANNs (Cao, Chen, and Khosla 2015). Based on the great success achieved in previous conversion schemes, many subsequent studies are devoted to minimizing various errors in conversion process (Sengupta et al. 2019).

ANN-to-SNN Distillation

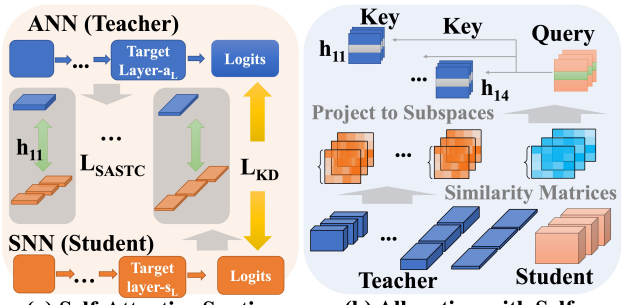
Typically, SNNs employ the spike frequency of the output layer or the average membrane potential increment as inference indicators. Analogous to ANN distillation, the conventional ANN-to-SNN knowledge distillation minimizes the Kullback-Leibler (KL) divergence between these SNN inference indicators and the predictive class probability distributions of ANNs (Lee et al. 2021). Recent efforts explore the transfer of enriched information from feature maps to enhance performance (Hong et al. 2023).

Self-Attentive Spatio-Temporal Calibration

Notations and Background

In this section, we provide a concise overview of fundamental concepts and establish necessary notations for subsequent illustration or clarity, the term "teacher model" denotes the ANN model, while the "student model" refers to the SNN model unless explicitly specified. Let $\mathcal{X} = \{x_i, y_i\}_i^n$ represent the training dataset consisting of n instances and N categories, with x_i as the input vector and y_i as the corresponding target in the form of a one-hot encoding vector. The number of output channels and spatial dimensions represented as c, h and w, respectively. For a mini-batch data of size b, the output of each SNN layer s_l at time step t is denoted as $f_{s_l}^t \in \mathbb{R}^{b\times c_{s_l}\times h_{s_l}\times w_{s_l}}$, where the superscript t signifies the index of the current time step, and T represents the total number of time steps. Simultaneously, the output of each ANN (teacher) layer a_l is denoted as $f_{a_l} \in \mathbb{R}^{b\times c_{a_l}\times h_{a_l}\times w_{a_l}}$. The layer indices s_l and a_l traverse from 1 to s_L and a_L, respectively. Notably, s_L and a_L typically differ due to the intrinsic heterogeneity inherent in the teacher and student models. The output representations at the penultimate layer of the teacher and student models are labeled as f_{a_L} and f_{s_L}. Furthermore, we define the feature pattern F as the set of outputs from intermediate feature layers. F_s^t represents the feature pattern of the student model at time step t, $F_s^t = \{f_{s_l}^t \mid \forall\, l \in [1, \ldots, L]\}$, while F_a denotes the feature pattern of the teacher model, $F_a = \{f_{a_l} \mid \forall\, l \in [1, \ldots, L]\}$. It is crucial to note that this collection is a permutation rather than a combination. In other words, multiple collections with the same intermediate layers but in different orders signify distinct feature patterns.

Concerning the student model, the outputs of the final layer $f_{end}(\cdot)$ are represented as the averaged membrane potentials over all time steps, $O_s^i = \frac{1}{T}\sum_{t=1}^{T} f_{end}(f_{s_L}^t[i]) \in \mathbb{R}^N$, where the notation i refers to the i-th input instance. We have added this clarification in the revised version. Predicted probabilities are derived through a softmax layer built on these outputs O_s^i, denoted as $P_s^i = \sigma(O_s^i/\alpha)$. Similarly, the logits of the teacher model are designated as $O_a^i = f_{end}(f_{a_L}[i]) \in \mathbb{R}^N$, and the corresponding predicted probabilities are denoted as $P_a^i = \sigma(O_a^i/\alpha)$, commonly referred to as soft targets. In both the student and teacher models, the hyperparameter α is typically set to 1.



(a) Self-Attentive Spatio-Temporal Calibration (b) Allocation with Self-attention

Figure 1: An overview of the proposed Self-Attentive Spatio-Temporal Calibration.

Spatio-Temporal Mismatch Problem on Existing ANN-to-SNN Knowledge Distillation

Prior studies have assumed that: 1) the distributions of semantic information embedded in ANNs and SNNs are similar (spatially matched), and 2) this distribution within SNNs remains constant across different time steps (temporally matched). We introduce a metric named Spatio-Temporal Mismatch Score (STM score) to assess the extent of semantic disparity between associated ANN-SNN layer pairs over time steps. STM score is computed as the Average Euclidean Distance between the generated similarity matrices of each corresponding ANN-SNN feature map pair, as expressed in Equation (1):

$$STMscore = \frac{1}{T}\frac{1}{|C^t|}\sum_{t=1}^{T}\sum_{C^t} MSE(A_{s_l}^t, A_{a_l}). \qquad (1)$$

where T represents the number of time steps, $|C^t|$ denotes the number of candidate layer pairs, and $A_{s_l}^t$ and A_{a_l} are the similarity matrices of ANN layer $_{a_l}$ and SNN layer $_{s_l}$, respectively. MSE measures the extent of semantic mismatches between the student SNN and the teacher ANN. A lower $STMscore$ signifies fewer mismatched association semantics, equating to superior model performance. Practically, we calculate the $STMscore$ (log-scale) values for each approach across training epochs and average them over the last 10 epochs, where they remain nearly unchanged.

Contrary to previous assumption, we find that existing ANN-to-SNN knowledge distillation methods either achieve

very small improvements or result in degradation effects on STM scores, as shown in Table 1. In other words, spatio-temporal mismatch of semantic information results in the loss of valuable knowledge during the knowledge distillation process. Our proposed approach diverges from the traditional paradigm by introducing self-attentive calibration. This innovative method aims to effectively transfer spatio-temporal semantic information by dynamically selecting suitable layer associations at each time step, departing from dependence on fixed teacher-student feature patterns.

SNN	Dataset	Time Step	STM score (↓)		
			Baseline	KD	FT
VGG-11	CIFAR-100	3	16.58	16.49	16.46
ResNet-18	CIFAR-100	3	16.97	16.85	16.73
ResNet-18	ImageNet	4	22.81	22.97	22.68

Note: teacher ANNs for CIFAR-100 and ImageNet are ResNet-32x4 and ResNet-34, respectively. The symbol (↓) indicates the smaller the better.

Table 1: Evaluation of Spatio-Temporal Mismatch Score on CIFAR-100 and ImageNet

Formulation of Self-Attentive Calibration

In our methodology, each student layer at every time step seamlessly aligns itself with semantically matched target layers through attention allocation, as depicted in Figure. 1 (a) and (b). The training process, guided by calibrated associations, prompts the student model to adeptly gather and integrate information from multiple layers at each time step, fostering a more tailored regularization. Furthermore, SASTC is versatile and can be applied in scenarios where the number of candidate layers differs between the teacher and student models. The ensemble of acquired feature patterns at time step t in SASTC is denoted as $C^t = \{(f_{s_l}^t, f_{a_l}) \mid \forall\, f_{s_l}^t \in F_s^t, f_{a_l} \in F_a\}$, with the corresponding weight satisfying $\sum_{a_l=1}^{a_L} \eta^t_{(f_{s_l}^t, f_{a_l})} = 1, \forall\, f_{s_l}^t \in F_s^t$ at each time step. This weight $\eta^t_{(f_{s_l}^t, f_{a_l})} \in \mathbb{R}^{b\times T}$ signifies the degree to which the target layer f_{a_l} is considered in the calibration of spatio-temporal semantic differences during ANN-to-SNN distillation. A more detailed exploration of these self-attentive weights will be provided subsequently. The feature maps from each time step of the student model are transformed into a_L distinct forms, aligning with the spatial dimensions of each target layer for subsequent distance calculations, as indicated by

$$\begin{aligned} \hat{f}^t_{s_l,a_l} = Proj\ (f_{s_l}^t \in \mathbb{R}^{b\times c_{s_l}\times h_{s_l}\times w_{s_l}}, a_l), \\ f_{s_l}^t \in F_s^t, a_l \in [1,\ldots,a_L], \end{aligned} \tag{2}$$

with $\hat{f}^t_{s_l,a_l} \in \mathbb{R}^{b\times c_{a_l}\times h_{a_l}\times w_{a_l}}$. Each function $Proj(\cdot,\cdot)$ comprises a stack of two convolution layers with 3×3 and 1×1 to fulfill the requirement for an efficient transformation. This design choice is guided by previous research, which has illustrated the remarkable effectiveness of a 3×3 convolution layer and the pyramid architecture (Han, Kim, and Kim 2017).

Allocation with Self-attention Previous research suggests that the abstraction of feature representations is closely associated with the layer depth (Bengio, Courville, and Vincent 2013). The semantic levels of these intermediates can vary between teacher and student architectures with differing capacities. Furthermore, we observe variations in the semantic level among the feature patterns of SNNs at different time steps, resulting in spatio-temporal information loss during distillation. To address these spatio-temporal differences in the student model and enhance the performance of feature transfer during distillation, each layer of the student model should be associated with the most semantically relevant target layer to derive its own regularization. Simple approaches such as random selection or forcing feature maps from the same layer depths to align may be inadequate due to the adverse effects resulting from semantic mismatched pairs.

Inspired by the layer associations facilitated by attention mechanisms in ANNs (Vaswani et al. 2017), we expand the self-attentive scheme from spatial calibration to spatio-temporal calibration. This extension presents a potentially viable solution for addressing the semantic mismatch problem and enhancing the overall distillation performance. Given that feature maps in SNNs, generated by similar instances, tend to cluster at distinct granularities across different time steps and layers, and similarly, in ANNs, these feature maps cluster based on their depth, the proximity of pairwise similarity matrices serves as a meaningful measure of inherent semantic similarity. These similarity matrices are computed as follows

$$\begin{aligned} A^t_{s_l} = R(f^t_{s_l}) \cdot R(f^t_{s_l})',\ f^t_{s_l} \in F^t_s, \\ A_{a_l} = R(f_{a_l}) \cdot R(f_{a_l})',\ f_{a_l} \in F_a, \end{aligned} \tag{3}$$

where $R(\cdot) : \mathbb{R}^{b\times c\times h\times w} \rightarrow \mathbb{R}^{b\times chw}$ represents a reshaping operation, and the symbol $'$ denotes the $transpose$ operation. Consequently, $A^t_{s_l}$ and A_{a_l} yield $b\times b$ matrices. More importantly, incorporating similarity matrices significantly mitigates the memory cost associated with large spatio-temporal dimensions ($T\cdot c_{s_l/a_l}\cdot h_{s_l/a_l}\cdot w_{s_l/a_l} \gg b$).

Building upon the self-attention framework (Vaswani et al. 2017), we independently project the pairwise similarity matrices of each student layer, originating from individual time steps, and each target layer, into two subspaces using a Multi-Layer Perceptron (MLP). This procedure endeavors to alleviate the influence of noise and sparsity, with the resulting vectors identified as $query$ and key. To expound further, for the i-th instance, the formulation can be articulated as follows:

$$Q^t_{s_l}[i] = MLP_Q(A^t_{s_l}[i]),\ K_{a_l}[i] = MLP_K(A_{a_l}[i]). \tag{4}$$

The parameters in $MLP_Q(\cdot)$ and $MLP_K(\cdot)$ are acquired through training to produce $query$ and key vectors, shared across all instances. Subsequently, the calibrated weight $\eta^{t,i}_{(f^t_{s_l}, f_{a_l})}$ for the i-th instance is computed as follows:

$$\eta^{t,i}_{(f^t_{s_l}, f_{a_l})} = \frac{e^{Q^t_{s_l}[i]' K_{a_l}[i]}}{\sum_{j=1}^{a_L} e^{Q^t_{s_l}[i]' K_j[i]}}. \tag{5}$$

The allocation based on attention offers a viable approach to alleviate the adverse effects stemming from spatio-temporal differences (mismatch in student-teacher layer

pairs) and amalgamate beneficial guidance from multiple target layers. The complete training procedure, incorporating the proposed semantic calibration formulation, is succinctly outlined in Algorithm 1.

Loss Function In a mini-batch of size b, the student model generates multiple feature patterns spanning various time steps $(F_s^t, \ldots, F_s^T)$. Following dimensional projections and self-attentive calibration, we employ Mean-Square-Error (MSE) to align the raw pairwise similarity matrices of the teacher and student (referred to as the loss of SASTC),

$$\begin{aligned}\mathcal{L}_{SASTC} &= \sum_t \sum_l \eta^t_{(f^t_{s_l}, f_{a_l})} Dist(f_{a_l}, Proj(f^t_{s_l}, a_l)) \\ &= \sum_{i=1}^{b} \sum_{t=1}^{T} \sum_{a_l=1}^{a_L} \sum_{s_l=1}^{s_L} \eta^{t,i}_{(f^t_{s_l}, f_{a_l})} MSE(f_{a_l}[i], \hat{f}^t_{s_l, a_l}[i]),\end{aligned} \tag{6}$$

as it demonstrated superior empirical performance. In this process, each feature map from the student model $f^t_{s_l}$ undergoes transformation via a projection function $Trans_s = Proj(\cdot, \cdot)$, while the target layers remain unchanged through identity transformation $Trans_a(\cdot) = I(\cdot)$. Multiplying the outcomes by the learned self-attentive distributions, the total loss is computed through a weighted summation of individual distances among feature maps from candidate teacher-student layer pairs at each time step. Consequently, the total loss of SASTC is expressed as:

$$\mathcal{L}_{total} = \mathcal{L}_{KD} + \beta \mathcal{L}_{SASTC}. \tag{7}$$

$$\mathcal{L}_{KD} = \mathcal{L}_{CE}(y_i, \sigma(O^i_s)) + \alpha^2 \mathcal{L}_{KL}(\frac{\sigma(O^i_a)}{\alpha}, \frac{\sigma(O^i_s)}{\alpha}). \tag{8}$$

where $\mathcal{L}_{CE}$ represents the standard cross-entropy loss (CE) between the predicted probabilities of the student model and the one-hot target, and $\mathcal{L}_{KL}$ denotes the KL divergence between P^i_s and the soft targets of the teacher model P^i_a.

Neuron Model and Surrogate Gradient

LIF Neuron We employ the LIF neuron model, which in discrete time, is described by:

$$u^t = \lambda u^{t-1} + I^t, \; o^t = \Theta(u^t - V_{th}), \tag{9}$$

where u signifies the membrane potential, I^t denotes presynaptic inputs, $\lambda(< 1)$ represents the constant leaky factor in the membrane potential, V_{th} signifies the threshold membrane potential, Θ stands for the Heaviside step function, o denotes the spike output propagating to the next layer, and the superscript t indicates the time step. Following the emission of the spike output, the reset operation is delineated in

$$u^t = u^t \cdot (1 - o^t). \tag{10}$$

To minimize trainable parameters, all neurons share identical leak values λ and threshold potentials V_{th}. For consistency across experiments, we set the initial membrane potential u^0 to 0, the threshold V_{th} to 1, and the leaky factor λ to 0.5.

Algorithm 1: Self-Attentive Spatio-Temporal Calibration for ANN-to-SNN Knowledge Distillation

Input: Training dataset $\mathcal{X} = (x_i, y_i)^n_{i=1}$; A pre-trained ANN (teacher model) with parameter $\mathcal{W}^a$; An SNN (student model) with randomly initialized parameter $\mathcal{W}^s$;

1: **while** $\mathcal{W}^s$ is not converged **do**
2: Sample a mini-batch b size samples from $\mathcal{X}$ named $\mathcal{B}$.
3: Obtain intermediate layers' presentations $\mathbf{F}^t_s$ across time steps and $\mathbf{F}_a$ by propagating $\mathcal{B}$ into $\mathcal{W}^a$ and $\mathcal{W}^s$.
4: Construct pairwise similarity matrices $A^t_{s_l}$ and A_{a_l} as Equation (3).
5: Perform self-attention based spatio-temporal calibration as Equation (4-5).
6: Spatially align feature patterns across time steps as Equation (2).
7: Update parameters $\mathcal{W}^s$ by propagating backward the surrogate gradients as defined in Equation (11) of the loss in Equation (7) and Equation (6).
8: **end while**

Triangle Shape Surrogate Gradient In this study, prioritizing a balance between accuracy and computational efficiency, we choose the triangular surrogate gradient, as established in prior research (Deng et al. 2022). The mathematical expression for the triangular surrogate gradient is as follows:

$$\frac{\partial o}{\partial u} = \frac{1}{\gamma^2} max(0, \gamma - |u - V_{th}|), \tag{11}$$

with γ set to 0.3 based on previous works (Deng et al. 2022).

Experiments

To demonstrate the effectiveness of our proposed self-attentive spatio-temporal calibration in ANN-to-SNN knowledge distillation, we conduct comprehensive experiments. We evaluate various ANN-SNN combinations using popular network architectures like VGG (Simonyan and Zisserman 2014), ResNet (He et al. 2016), PyramidNet (Han, Kim, and Kim 2017), and WRN (Zagoruyko and Komodakis 2016) on static datasets. Meticulously designed experiments and ablation studies validate the effectiveness of SASTC in providing proper regularization for student models. We apply SASTC to neuromorphic datasets like DVS-Gesture and DVS-CIFAR10, demonstrating its robust generalization in noisy-label learning. Additionally, we offer a visual analysis of SASTC's success.

Further analyses of temporal information dynamics are provided in Appendix 1, with demonstrations of robust generalization of few-shot learning in Appendix 2. Details on batch size, sensitivity, computational efficiency, running time, and memory consumption are summarized in Appendix 3. The experimental setup details are available in Appendix 4.

SNN (student)	Baseline			ANN (teacher)	KD			Feature KD			SASTC		
	T=2	T=3	T=7		T=2	T=3	T=7	T=2	T=3	T=7	T=2	T=3	T=7
CIFAR-10													
VGG-11	92.48	92.80	93.16	ResNet-19	93.01	92.92	93.22	93.28	93.48	93.60	93.55	93.73	93.92
				Pyramidnet-20	92.98	93.07	93.52	93.04	93.28	93.36	93.52	93.70	93.64
				WRN-28-4	92.14	92.83	93.22	92.64	92.80	93.48	93.07	93.10	93.70
ResNet-18	94.04	94.12	94.72	WRN-28-4	94.24	94.44	94.68	94.92	95.44	95.76	95.12	95.24	95.48
				Pyramidnet-20	94.16	94.24	94.48	94.60	95.16	95.48	94.92	95.00	95.16
WRN-16-2	89.40	89.48	90.80	ResNet-19	92.36	92.52	93.36	92.16	93.13	93.26	94.12	94.16	94.12
				Pyramidnet-20	92.04	92.60	93.28	92.86	93.18	93.51	93.16	93.76	93.96
				WRN-28-4	91.72	92.20	92.92	92.16	93.08	93.44	93.16	93.28	93.72
CIFAR-100													
VGG-11	68.70	69.76	70.00	VGG-13	73.44	74.52	75.24	74.01	74.43	75.11	74.60	74.88	76.36
				ResNet-32x4	73.28	74.32	75.16	73.76	73.88	75.07	75.40	76.80	77.08
				WRN-40-2	74.24	75.16	75.48	74.31	75.07	75.48	74.48	75.60	75.72
ResNet-18	70.56	75.72	76.40	ResNet-32x4	77.68	78.00	78.64	77.01	77.96	78.09	80.28	80.24	80.68
				WRN-40-2	77.56	78.00	79.12	77.14	77.68	78.24	77.76	78.36	78.84

Note: baseline SNNs are trained from scratch using the same surrogate gradient as our distilled student models.

Table 2: Top-1 Test Accuracy(%) of Different ANN-to-SNN Distillation Approaches on CIFAR-10 and CIFAR-100 datasets

Comparison to Conventional ANN-to-SNN Distillation Methods

Top-1 test accuracy (%) on CIFAR-10 and CIFAR-100 across seventeen distinct ANN-SNN combinations is illustrated in Table 2. Four of these combinations share similar architectures (WRN-16-2/28-4, VGG-11/13, ResNet-18/32x4), while the remaining nine are heterogeneous. Since the large memory consumption of traditional feature ANN-to-SNN KD method , we train the student SNN with different single layer combination settings and calculate their average value as the final result (*i.e.,* Feature KD in Table 2).

Table 2 illustrates that SASTC demonstrates significant relative improvement across all compared methods on both CIFAR-10 and CIFAR-100 datasets, indicating its ability to effectively leverage intermediate information across time steps for superior distillation results. The most notable enhancements occur in the "WRN-16-2 & ResNet-19" (T=2) and the "ResNet-18 & ResNet-32x4" (T=2), with a 4.72% improvement over baseline on CIFAR-10 and a 8.84% improvement over baseline on CIFAR-100. Notably, when compared to the competitive distillation method feature KD, results with combinations of WRN-16-2/ResNet-19 on CIFAR-10 and all combinations on CIFAR-100 meet performance degradation phenomenon compared to the vanilla ANN-to-SNN KD method (the detailed explanation of these negative regularization effects is illustrated in the section of mechanism analysis).

Moreover, we achieve significant improvements on ImageNet, which is illustrated in Table 3. Despite training for only 90 epochs, SASTC improves ResNet-18 performance by 2.02% over the baseline SNN. Additionally, SASTC achieves 1.15% and 1.51% improvements over vanilla and feature-based ANN-to-SNN KD on ResNet-18, respectively. Notably, feature-based method shows significant negative regularization effects during the ANN-to-SNN distillation process on ImageNet.

Method	Architecture	Time Steps	Accuracy
Baseline	ResNet-18	4	60.50
KD	ResNet-18	4	61.37
Feature KD	ResNet-18	4	61.01
SASTC	ResNet-18	4	62.52
	ResNet-34	4	68.69
Teacher (ANN)	ResNet-34	1	73.48

Table 3: Top-1 Test Accuracy(%) of ANN-to-SNN Distillation Approaches on ImageNet dataset

Comparison to Existing SNN Training Methods

In this section, we present a comparison of our experimental results with previous conventional training methods, summarized in Table 4.

On the CIFAR-10 and CIFAR-100 datasets, our SASTC outperforms all existing approaches, achieving the highest accuracy and the lowest inference latency. Notably, our method first outperforms the ANN counterpart with the spatio-temporal calibration on both CIFAR-10 and CIFAR-100, the relatively maximum increments are 0.51% and 5.33%, respectively.

On the ImageNet dataset, the SASTC algorithm achieves a 1.91% increment compared to SSCL-SNN (Zhang et al. 2024) with 4 time steps. Although SEW-ResNet34 deviates from a typical SNN as it adopts the IF model and modifies the Residual structure, SASTC achieves 1.65% improvement than SEW-ResNet (Fang et al. 2021)

Method	Architecture	Time Steps	Accuracy
CIFAR-10			
Norm (Sengupta et al. 2019)	VGG-16	2500	91.55
Norm (Han, Srinivasan, and Roy 2020)	VGG-16	2048	93.63
Norm (Deng and Gu 2021)	VGG-16	16	92.29
STBP (Wu et al. 2018)	CIFARNet	12	89.83
STBP NeuNorm (Wu et al. 2019)	CIFARNet	12	90.53
Hybrid (Rathi et al. 2020)	ResNet-20	250	92.22
DIET-SNN (Rathi and Roy 2021)	ResNet-20	10	92.54
TET (Deng et al. 2022)	ResNet-19	6	94.50
TSSL-BP (Zhang and Li 2020)	CIFARNet	5	91.41
		6	93.16
STBP-tdBN (Zheng et al. 2021)	ResNet-19	4	92.92
		2	92.34
GLIF (Yao et al. 2022)	ResNet-19	2	94.44
KDSNN (Xu et al. 2023)	VGG-16	4	91.05
Norm (Bu et al. 2023)	VGG-16	4	93.96
TKS (Dong, Zhao, and Zeng 2024)	ResNet-19	4	95.30
SSCL-SNN (Zhang et al. 2024)	ResNet-20	4	94.27
		7	95.48
Ours	ResNet-18	3	95.24
		2	95.12
ANN (Deng et al. 2022)	ResNet-19	1	94.97
CIFAR-100			
Hybrid (Rathi et al. 2020)	VGG-11	125	67.87
DIET-SNN (Rathi and Roy 2021)	VGG-16	5	69.67
TET (Deng et al. 2022)	ResNet-19	6	74.72
		6	71.12
STBP-tdBN (Zheng et al. 2021)	ResNet-19	4	70.86
		2	69.41
GLIF (Yao et al. 2022)	ResNet-19	2	75.48
Norm (Bu et al. 2023)	VGG-16	4	69.62
TKS (Dong, Zhao, and Zeng 2024)	ResNet-19	4	76.20
SSCL-SNN (Zhang et al. 2024)	ResNet-19	2	78.79
		7	80.68
Ours	ResNet-18	3	80.24
		2	80.28
ANN (Deng et al. 2022)	ResNet-19	1	75.35
ImageNet			
Hybrid (Rathi et al. 2020)	ResNet-34	250	61.48
SPIKE-NORM (Sengupta et al. 2019)	ResNet-34	2500	69.96
STBP-tdBN (Zheng et al. 2021)	Spiking-ResNet-34	6	63.72
SEW ResNet (Fang et al. 2021)	SEW-ResNet-34	4	67.04
TET (Deng et al. 2022)	Spiking-ResNet-34	6	64.79
SSCL-SNN (Zhang et al. 2024)	ResNet-34	4	66.78
MS-ResNet (Hu et al. 2024)	ResNet-18	6	63.10
Ours	ResNet-18	4	62.52
	ResNet-34	4	68.69

Table 4: Top-1 Test Accuracy(%) of Different SNN Methods on Static Datasets

Mechanism Analysis and Ablation Study

In this section, we delve into an experimental exploration of the negative regularization effect induced by manually specified layer associations across time steps. Furthermore, we provide evidence of the success of SASTC, supported by the proposed criterion and visual evidence.

SASTC Improves Negative Regularization Effects We conduct experiments on the CIFAR-10 dataset by training the student model exclusively with a specified teacher-student layer pair in various settings, and observe negative regularization effects that feature-pattern-based distillation with specific layer associations across time steps performs worse than vanilla ANN-to-SNN KD. The network architectures involved "VGG-11& ResNet-19", "ResNet-18 & Pyramidnet-20", and "WRN-16-2 & WRN-28-4". The numbers of candidate target layers and student layers for each case are (4, 5), (3, 4), and (4, 4), respectively.

The outcomes of student models with 20, 12 and 16 ANN-SNN layer combinations under the three settings on CIFAR-10 are illustrated in Fig. 2. Notably, all layer associations of SNN layer-2 and layer-3 in Fig. 2 (a) obtain extremely poor performance, likely due to highly sparse semantic information contained in layer-2 of VGG-11 student model. In addition, the performance of a student model significantly diminishes for certain layer associations across time steps (i.e., negative regularization effect), including SNN layer-1 to layer-4 in Fig. 2 (a) and (c), most like due to the substantial semantic mismatch. Notably, it is observed that the one-to-one layer matching scheme is non-optimal because better results can be obtained by leveraging information from a target layer with different depth, such as "SNN layer-5 & ANN-layer3" in Fig. 2 (a), "SNN layer-4 & ANN-layer3" in Fig. 2 (b) and "SNN layer-2 & ANN-layer3" in Fig. 2 (c).

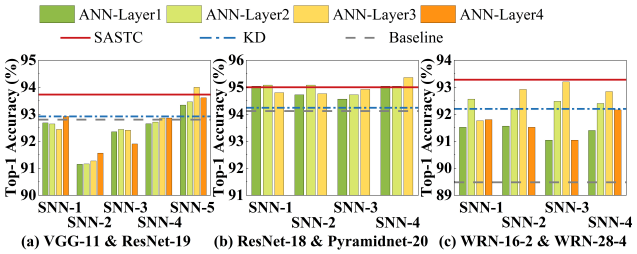


Figure 2: Illustration of negative regularization on CIFAR-10 with three model combinations. Each tick label of x-axis denotes an SNN (student) layer number. Different color bars indicate the results of different specified ANN-SNN layer combinations.

Although training with specific hand-crafted layer associations may outperform SASTC in isolated cases like "SNN layer-5 & ANN-layer3" in Fig. 2 (a) and "SNN layer-4 & ANN-layer3" in Fig. 2 (b), SASTC consistently performs well across a large number of associations. It is particularly noteworthy considering that the knowledge of the best layer association for each network combination is not available in advance. Furthermore, instances where training with SASTC is inferior to the best layer association suggest potential refinements in our association strategy.

SASTC Achieves Semantic Matching during Knowledge Distillation The results in Table 5 indicate that SASTC consistently attains the lowest spatio-temporal mismatch score throughout the training process compared to other approaches owing to our spatio-temporal calibration mechanism on both CIFAR-100 and ImageNet datasets.

Moreover, we provide evidence in Appendix 1 that SASTC optimizes the temporal information dynamics of SNNs, further illustrating the success and mechanism of our proposed method.

Extension, Application and Visual Analysis

Extension to Neuromorphic Datasets Previous works focus on utilizing complex architectures to tackle the challenging DVS-CIFAR10 task. However, these sophisticated models have failed to achieve satisfied performance and are susceptible to overfitting. Recently, a temporal efficient training approach has achieved the state-of-

SNN	Dataset	Time Step	STM score (↓)			
			Baseline	KD	FT	**SASTC**
VGG-11	CIFAR-100	3	16.58	16.49	16.46	16.30
ResNet-18	CIFAR-100	3	16.97	16.85	16.73	16.11
ResNet-18	ImageNet	4	22.81	22.97	22.68	22.09

Note: teacher ANNs for CIFAR-100 and ImageNet are ResNet-32x4 and ResNet-34, respectively. The symbol (↓) indicates the smaller the better.

Table 5: Evaluation of Spatio-Temporal Mismatch Score on CIFAR-100 and ImageNet

the-art accuracy with the streamlined VGGSNN architecture. In this paper, we conduct experiments on the DVS-Gesture and DVS-CIFAR10 datasets. As shown in Table 6, our proposed method outperform the contemporary best-performing methods, the accuracy of SASTC increase to 97.92% on DVS-Gesture and 83.60% on DVS-CIFAR10 through the calibration of spatio-temporal semantic mismatches during ANN-to-SNN distillation. Consequently, our SASTC method significantly improves the processing temporal information ability of SNNs on neuromorphic tasks.

Method	Architecture	Time Step	Accuracy
DVS-Gesture			
PLIF (Fang et al. 2021)	c128k3s1-BN-PLIF-MPk2s2*5-DPFC512-PLIF-DP-FC110-PLIF-APk10s10	20	97.57
STBP-tdBN (Zheng et al. 2021)	CIFARNet	40	96.87
SLAYER (Shrestha and Orchard 2018)	8 layers	25	93.64
Ours	ResNet-18	16	97.92
DVS-CIFAR10			
STBP-tdBN (Zheng et al. 2021)	ResNet-19	10	67.80
Streaming Rollout (Kugele et al. 2020)	DenseNet	10	66.80
Conv3D (Wu et al. 2021)	LIAF-Net	10	71.70
LIAF (Wu et al. 2021)	LIAF-Net	10	70.40
TET (Deng et al. 2022)	VGGSNN	10	83.17
SSCL-SNN (Zhang et al. 2024)	ResNet-19	10	80.00
Ours	VGGSNN	10	83.60

Table 6: Top-1 Test Accuracy(%) of Different SNN Methods on Neuromorphic Datasets

Application to Robust Representation In addition to evaluating the clean test set performance, we introduce noisy-label learning datasets by randomly perturbing 10%, 20%, 30%, 40%, and 50% of labels in training images. As shown in Table 7, training the lightweight SNN with SASTC extremely enhances its robustness compared to other ANN-to-SNN knowledge distillation approaches and the teacher ANN counterpart.

Visualization Analysis of SASTC To visually elucidate the advantages of SASTC, we randomly selected several images from ImageNet, and highlight regions deemed crucial for predicting the respective labels by utilizing Spike Activation Map (SAM) (Kim and Panda 2021). As depicted in Figure 3, SASTC consistently centralizes class-discriminative regions and excels in capturing more semantically related information, resembling the teacher model, while the compared methods scatter them in the surroundings.

Percentage of Noisy Labels	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**
Teacher (ResNet-32x4)	79.42%	76.51%	73.63%	70.57%	68.08%	64.27%
Baseline	69.76%	66.84%	64.00%	61.40%	59.84%	54.48%
KD	74.32%	73.64%	72.64%	72.36%	71.36%	70.80%
Feature KD	73.96%	63.68%	63.36%	63.24%	62.96%	61.84%
SASTC	**77.08%**	**75.49%**	**74.77%**	**74.62%**	**74.32%**	**73.87%**

Note: SNNs adopt three time steps.

Table 7: Noisy-Label Learning: Top-1 Test Accuracy(%) of "VGG-11 & ResNet-32x4" Combination on CIFAR-100

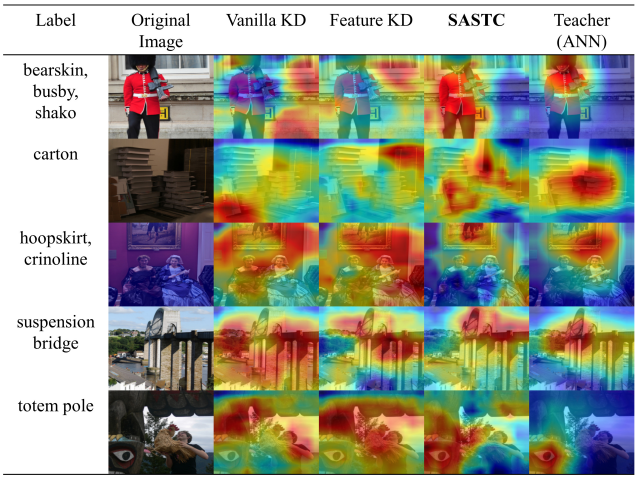


Figure 3: Spike Activation Map (SAM) visualization of ANN-to-SNN distillation approaches on ImageNet. The red regions highlight areas deemed important for model inference.

Conclusion

This study focuses on mitigating performance degradation due to spatio-temporal semantic mismatches and negative regularization in conventional ANN-to-SNN knowledge distillation methods. We propose a self-attentive mechanism to learn layer association weights across different time steps, enabling semantically aligned knowledge transfer. Qualitative and quantitative evidence validate SASTC's spatio-temporal calibration capability. Extensive experiments demonstrate that SASTC consistently outperforms various SNN training approaches and distillation schemes. SASTC also shows strong generalization across tasks and network architectures, excelling in robust representation.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62336007, in part by the Key R&D Program of Zhejiang under Grant 2022C03011, in part by the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study under Grant SN-ZJU-SIAS-002, and in part by the Fundamental Research Funds for the Central Universities.

References

Akopyan, F.; Sawada, J.; Cassidy, A.; Alvarez-Icaza, R.; Arthur, J.; Merolla, P.; Imam, N.; Nakamura, Y.; Datta, P.; Nam, G.-J.; Taba, B.; Beakes, M.; Brezzo, B.; Kuang, J. B.; Manohar, R.; Risk, W. P.; Jackson, B.; and Modha, D. S. 2015. TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10): 1537–1557.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8): 1798–1828.

Bu, T.; Fang, W.; Ding, J.; Dai, P.; Yu, Z.; and Huang, T. 2023. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. *arXiv preprint arXiv:2303.04347*.

Cao, Y.; Chen, Y.; and Khosla, D. 2015. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113: 54–66.

Deng, S.; and Gu, S. 2021. Optimal conversion of conventional artificial neural networks to spiking neural networks. *arXiv preprint arXiv:2103.00476*.

Deng, S.; Li, Y.; Zhang, S.; and Gu, S. 2022. Temporal efficient training of spiking neural network via gradient reweighting. *arXiv preprint arXiv:2202.11946*.

Diehl, P. U.; Neil, D.; Binas, J.; Cook, M.; Liu, S.-C.; and Pfeiffer, M. 2015. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks (IJCNN)*, 1–8. ieee.

Dong, Y.; Zhao, D.; and Zeng, Y. 2024. Temporal Knowledge Sharing Enable Spiking Neural Network Learning From Past and Future . *IEEE Transactions on Artificial Intelligence*, 5(07): 3524–3534.

Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; and Tian, Y. 2021. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2661–2671.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13558–13567.

Han, D.; Kim, J.; and Kim, J. 2017. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5927–5935.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hong, D.; Shen, J.; Qi, Y.; and Wang, Y. 2023. LaSNN: Layer-wise ANN-to-SNN Distillation for Effective and Efficient Training in Deep Spiking Neural Networks. *arXiv preprint arXiv:2304.09101*.

Hu, Y.; Deng, L.; Wu, Y.; Yao, M.; and Li, G. 2024. Advancing spiking neural networks toward deep residual learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Kim, Y.; and Panda, P. 2021. Visual explanations from spiking neural networks using inter-spike intervals. *Scientific reports*, 11(1): 1–14.

Kugele, A.; Pfeil, T.; Pfeiffer, M.; and Chicca, E. 2020. Efficient processing of spatio-temporal data streams with spiking neural networks. *Frontiers in neuroscience*, 14: 512192.

Lee, D.; Park, S.; Kim, J.; Doh, W.; and Yoon, S. 2021. Energy-efficient Knowledge Distillation for Spiking Neural Networks. *CoRR*, abs/2106.07172.

Lee, J. H.; Delbruck, T.; and Pfeiffer, M. 2016. Training deep spiking neural networks using backpropagation. *Frontiers in neuroscience*, 10: 228000.

Maass, W. 1997. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9): 1659–1671.

Pérez-Carrasco, J. A.; Zhao, B.; Serrano, C.; Acha, B.; Serrano-Gotarredona, T.; Chen, S.; and Linares-Barranco, B. 2013. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward ConvNets. *IEEE transactions on pattern analysis and machine intelligence*, 35(11): 2706–2719.

Rathi, N.; and Roy, K. 2021. DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*.

Rathi, N.; Srinivasan, G.; Panda, P.; and Roy, K. 2020. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. *arXiv preprint arXiv:2005.01807*.

Sengupta, A.; Ye, Y.; Wang, R.; Liu, C.; and Roy, K. 2019. Going Deeper in Spiking Neural Networks: VGG and Residual Architectures. *Frontiers in Neuroscience*, 13.

Shrestha, S. B.; and Orchard, G. 2018. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wu, Y.; Deng, L.; Li, G.; and Shi, L. 2018. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12: 323875.

Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Xie, Y.; and Shi, L. 2019. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1311–1318.

Wu, Z.; Zhang, H.; Lin, Y.; Li, G.; Wang, M.; and Tang, Y. 2021. Liaf-net: Leaky integrate and analog fire network for lightweight and efficient spatiotemporal information processing. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11): 6249–6262.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7886–7895.

Yao, X.; Li, F.; Mo, Z.; and Cheng, J. 2022. Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks. *Advances in Neural Information Processing Systems*, 35: 32160–32171.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, W.; and Li, P. 2020. Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Advances in neural information processing systems*, 33: 12022–12033.

Zhang, Y.; Liu, X.; Chen, Y.; Peng, W.; Guo, Y.; Huang, X.; and Ma, Z. 2024. Enhancing Representation of Spiking Neural Networks via Similarity-Sensitive Contrastive Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16926–16934.

Zheng, H.; Wu, Y.; Deng, L.; Hu, Y.; and Li, G. 2021. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11062–11070.