Enhancing Representation of Spiking Neural Networks via Similarity-Sensitive Contrastive Learning

Yuhan Zhang*, Xiaode Liu*, Yuanpei Chen, Weihang Peng, Yufei Guo†, Xuhui Huang, Zhe Ma†

Intelligent Science & Technology Academy of CASIC, Beijing, 100144, China
yfguo@pku.edu.cn, mazhe_thu@163.com

Abstract

Spiking neural networks (SNNs) have attracted intensive attention as a promising energy-efficient alternative to conventional artificial neural networks (ANNs) recently, which could transmit information in the form of binary spikes rather than continuous activations thus the multiplication of activation and weight could be replaced by addition to save energy. However, the binary spike representation form will sacrifice the expression performance of SNNs and lead to accuracy degradation compared with ANNs. Considering improving feature representation is beneficial to training an accurate SNN model, this paper focuses on enhancing the feature representation of the SNN. To this end, we establish a similarity-sensitive contrastive learning framework, where SNN could capture significantly more information from its ANN counterpart to improve representation by Mutual Information (MI) maximization with layer-wise sensitivity to similarity. In specific, it enriches the SNN’s feature representation by pulling the positive pairs of SNN’s and ANN’s feature representation of each layer from the same input samples closer together while pushing the negative pairs from different samples further apart. Experimental results show that our method consistently outperforms the current state-of-the-art algorithms on both popular non-spiking static and neuromorphic datasets.

Introduction

Recent developments in deep neural networks (DNNs) have achieved great success in a variety of computer vision tasks including pattern recognition (Simonyan and Zisserman 2014; He et al. 2016), semantic image segmentation (Chen et al. 2018), object detection (Girshick 2015; Ren et al. 2015), and so on. However, the increasing complexity of these full-precision DNN models brings high energy consumption, which makes them difficult to deploy in real-world resource-constrained environments. The spiking neural network (SNN), inspired by how the brain represents reality of these full-precision DNN models brings high energy consumption, which makes them difficult to deploy in real-world resource-constrained environments. The spiking neural network (SNN), inspired by how the brain represents...
The overall workflow of the proposed method. To enhance the representation of SNN, a similarity-sensitive contrastive loss is introduced for transmitting the important information from ANN’s representation to SNN’s representation, where contrastive losses at different layers are weighted by CKA.

Our training framework is shown in Fig. 1. For clarity, the main contribution of this paper is summarized below.

- We propose a novel contrastive learning framework to train SNNs directly via maximizing the mutual information between SNN’s representation and its real-valued counterparts of the well-trained ANN. The SNN is optimized based on both joint distribution and the product of marginal distributions of the ANN’s and SNN’s representation.
- Furthermore, a series of layer-wise similarity indexes are introduced into the total loss to weigh the mutual information maximization at different layers.
- We evaluate our method on both static and neuromorphic datasets. Extensive experimental results under various experimental settings show that our method significantly outperforms state-of-the-art methods, e.g. 78.79% on CIFAR-100, 66.78% on ImageNet, and 80.00% on CIFAR10-DVS.

**Related Work**

Based on the classification of the review (Guo, Huang, and Ma 2023), the representative ability decreasing of SNNs can be mitigated on the neuron level, network structure level, and training technique level. **On the neuron level**, introducing learnable hyperparameters into the spiking neuron is a common way. LSNN (Bellec et al. 2018) and LTMD (Wang, Cheng, and Lim 2022) proposed adaptive threshold spike neurons to improve the computing and learning capabilities of SNNs. Besides this, introducing a learnable membrane time constant is another commonly used method (Yin, Corradi, and Bohte 2020; Luo et al. 2022). Furthermore, DietSNN (Rathi and Roy 2020a) adopted the learnable membrane leak and firing threshold simultaneously.

**On the network structure level**, SEW-ResNet (Fang et al. 2021a) and DS-ResNet (Feng et al. 2022) replaced the standard ResNet backbone with activation before the addition form-based ResNet. The representation capability will be increased due to that this kind of ResNet will fire positive integer spikes. However, the multiplication-addition transformation will be lost at the same time. To handle this problem, MS-ResNet (Hu et al. 2021) proposed a pre-activation form-based ResNet, where the spike-based convolution can be retained. **On the training technique level**, IM-Loss (Guo et al. 2022a) proposed an information maximization loss to improve the activation information entropy. Recently, some
works (Takuya et al. 2021; Xu et al. 2023) introduced the distillation method in the SNN domain. In these methods, an ANN teacher is used to guide SNN-student learning with KLD or distance between their outputs or features, aligning ANN’s and SNN’s output or features obtained from the same sample.

Contrastive learning considers both positive and negative sample pairs to achieve expression learning, which has validated its superiority in many computer vision tasks (Wu et al. 2018; He et al. 2020; Chen et al. 2020; Tian, Krishnan, and Isola 2020, 2019). It pulls the representations of positive sample pairs closer using a contrastive loss function, as well as pushes the representations of positive and negative sample pairs away. This study combines the knowledge distillation method and the contrastive learning method to introduce a robust learning approach with the push-and-pull scheme to effectively enhance the representation of SNNs by the guidance of the ANNs.

Preliminary and Methodology

This section first introduces the spiking neuron model and why its feature’s expression ability is limited, then, why and how to align the SNN’s representation with the ANN’s representation via similarity-sensitive contrastive learning. Finally, the process to train an SNN with the proposed methods will be given in detail by pseudocode.

Spiking Neuron Model

The primary computing neuron of an SNN is much different from that of an ANN. The neuron of an ANN only plays the role of nonlinear transformation and can output real-valued values. While the neuron in an SNN enjoys rich spatially-temporal dynamics. Take the well-known Leaky Integrate-and-Fire (LIF) neuron model as an example. The LIF neuron updates its membrane potential based on the input and its membrane potential at the previous moment as follows,

\[ U_{t,\text{pre}} = \tau U_{t-1} + WZ_t, \]

where \( U_{t,\text{pre}} \) denotes the pre-membrane potential at \( t \)-th timestep, \( U_t \) denotes the membrane potential at \( t \)-th timestep, \( \tau \) is the membrane time constant which represents the leakage effect of the membrane potential, \( W \) is the weight, and \( Z_t \) is the binary map comes from the previous layer at \( t \)-th timestep.

When \( U_{t,\text{pre}} \) exceeds the firing threshold \( U_{\text{th}} \), the neuron would fire a spike and reset the \( U_t \) to zero, otherwise, the membrane potential would be presented to the next timestep with a leak according to Eq. 2, given by

\[ O_t = \begin{cases} 1, & \text{if } U_{t,\text{pre}} \geq U_{\text{th}} \\ 0, & \text{otherwise} \end{cases}, \quad U_t = U_{t,\text{pre}} \cdot (1 - O_t), \]

where \( U_{\text{th}} \) is a given firing threshold and \( O_t \) denotes the output of the LIF neuron. \( O_t \) is a binary feature map.

The Representation of SNN

Though the binary spike information processing paradigm is highly energy efficient, it will result in unsatisfactory performance too, since the binary spike activation maps cannot carry enough information compared with ANNs. Take the information entropy concept to analyze it, given a set, \( \mathbf{M} \), its representation capability, \( \mathcal{R}(\mathbf{M}) \), can be measured by the information entropy of \( \mathbf{M} \), as follows

\[ \mathcal{R}(\mathbf{M}) = \max \mathcal{H}(\mathbf{M}) = \max \left( - \sum_{m \in \mathbf{M}} p_M(m) \log_2 p_M(m) \right), \]

where \( p_M(m) \) is the probability of a sample, \( m \) from \( \mathbf{M} \). When \( p_M(m_1) = p_M(m_2) = p_M(m_3) \cdots = p_M(m_{N_M}) \), \( \mathcal{H}(\mathbf{M}) \) reaches its maximum, \( \log_2(N_M) \), where \( N_M \) is the total number of the samples from \( \mathbf{M} \). And for the activation map of the ANN, it can be denoted as \( \mathbf{M}_R = \mathbb{R}^{C \times H \times W} \), where \( C \) is the channels and \( H \) and \( W \) are height and width of the map. Correspondingly, the activation map of the SNN can be denoted as \( \mathbf{M}_B \in \mathbb{R}^{T \times C \times H \times W} \), where \( T \) is the total timesteps. Since the binary spike output \( o \) can be expressed with 1 bit, the number of samples from \( o \) is 2. Then, the number of samples from \( \mathbf{M}_B \) is \( 2^{T \times C \times H \times W} \) and \( \mathcal{R}(\mathbf{M}_B) = \log_2 2^{T \times C \times H \times W} = T \times C \times H \times W \). While a real-valued activation for ANN needs 32 bits, thus consisting of \( 2^{32} \) samples and \( \mathcal{R}(\mathbf{M}_R) = \log_2 2^{32 \times C \times H \times W} = 32 \times C \times H \times W \). In most of the direct training SNN works, the \( T \) is smaller than 32, thus the representation capability of the SNN is much worse than that of the ANN.

Mutual Information and Centered Kernel Alignment

In this paper, we consider maximizing the mutual information (MI) of the representations of SNN and ANN to align the output features of SNN and ANN, so as to achieve the effect of enhancing the representation ability of the SNN. For ease of notation, we define random variables \( S^l \) and \( A^l \) for the SNN’s and ANN’s representation of the data at layer \( l \) respectively:

\[ S^l = f^S_l(x), \]
\[ A^l = f^A_l(x). \]

Their mutual information (MI) can be defined as (Solomon 1997):

\[ I(S^l, A^l) = \sum_{s,a} P_{S^l}() \log \frac{P_{S^l A^l}(s,a)}{P_{S^l}(s) P_{A^l}(a)}, \]

where \( P_{S^l A^l}(s,a) \) is the joint distribution, \( P_{S^l}(s) = \sum_a P_{S^l A^l}(s,a) \) and \( P_{A^l}(a) = \sum_s P_{S^l A^l}(s,a) \) are the marginals of \( S^l \) and \( A^l \), respectively. Compared with (1), mutual information (7) introduces additional information within the product of the respective marginal distributions of ANN’s and SNN’s representation \( P_{S^l}(s) P_{A^l}(a) \) to learn the SNN contrastively. It quantifies the amount of information obtained about SNN’s representation by observing ANN’s representation and can be considered as the reduction in uncertainty about SNN’s representation given knowledge of ANN’s representation. High mutual information indicates a large reduction in uncertainty and vice versa (Solomon 1997). We would like ANN’s and SNN’s representations to share as much information as possible, because the more similar they are in the feature space, the smaller the
different samples can be pushed away, which corresponds to samples can be pulled close, while representations from dif-
sensitive contrastive loss based on Noise-Contrastive Esti-
In this section, we introduce how to construct a similarity-
Similarity-Sensitive Contrastive Learning
measure how similar they are.

$$\text{HSIC}(K, L) = \frac{1}{n(n-3)} \left( \text{tr}(K\tilde{L}) + \frac{1}{n-1} \text{tr}(11^\top \tilde{L}1) - \frac{2}{n-2} \text{tr}(K\tilde{L}1) \right).$$

(9)

Similarity-Sensitive Contrastive Learning
In this section, we introduce how to construct a similarity-
sensitive contrastive loss based on Noise-Contrastive Esti-
mation (NCE) to maximize the mutual information between
the layer-wise discrete and the full-precision representations.
NCE estimates the mutual information with its lower bound
to avoid computing it directly. As shown in Figure 1, the
discrete and full-precision representations from the same
samples can be pulled close, while representations from
different samples can be pushed away, which corresponds to
the core idea of contrastive learning. Then, layer-wise NCE
losses are weighted by the corresponding CKA scale, thus
forming the similarity-sensitive contrastive loss.

For a training batch with $N$ samples, the samples can be
denoted as: \{${x_i}$\}$_1^N$. Let us define a distribution \(q\)
with latent variable \(G\) which decides whether a contrastive
pair \(\{f^S(x_i), f^A(x_j)\}\) comes from the same samples \(i = j \Leftrightarrow G = 1\) or different samples \(i \neq j \Leftrightarrow G = 0\):

\begin{align*}
q(S^i, A^i | G = 1) &= p(S^i, A^i), \quad (10) \\
q(S^i, A^i | G = 0) &= p(S^i)p(A^i). \quad (11)
\end{align*}

Suppose that in the data, for every \(N - 1\) incongruent pair
(different samples input to \(f^S\) and \(f^A\) from the product
of marginal distributions), 1 congruent pair (the same sample
input to \(f^S\) and \(f^A\)) is given. Then the priors of the latent
variable \(G\) are:

$$q(G = 1) = \frac{1}{N}, q(G = 0) = \frac{N - 1}{N}. \quad (12)$$

Through simple operations and Bayes’ rule, the posterior for
\(G = 1\) can be obtained by:

$$q(S^i, A^i | G = 1) = \frac{q(S^i, A^i | G = 1)q(G = 1)}{q(S^i, A^i | G = 1)q(G = 1) + q(S^i, A^i | G = 1)q(G = 1)}$$

\begin{align*}
&= \frac{p(S^i)p(A^i)}{p(S^i)p(A^i) + (N - 1)p(S^i)p(A^i)} \\
&= \frac{1}{1 + (N - 1)}p(S^i)p(A^i). \quad (13)
\end{align*}

Next, taking the logarithm on both sides of the above equation, we have:

$$\log q(G = 1 | S^i, A^i) = \log p(S^i)p(A^i) - \log(1 + (N - 1))p(S^i)p(A^i)$$

\begin{align*}
&= - \log(1 + (N - 1))p(S^i)p(A^i) \\
&\leq \log(N - 1) - \log p(S^i)p(A^i). \quad (14)
\end{align*}

Then taking expectation on both sides w.r.t. \(q(S^i, A^i | G = 1)\) and rearranging, we obtain:

$$I(S^i; A^i) \geq \log(N - 1) +$$

\begin{align*}
E_q(S^i,A^i|G=1) \log q(G = 1 | S^i, A^i), \quad (15)
\end{align*}

where \(I(S^i; A^i)\) denotes the mutual information between
the SNN’s and ANN’s representations. Therefore, maxi-
mizing the mutual information is equivalent to maximizing
its lower bound \(E_q(S^i,A^i|G=1) \log q(G = 1 | S^i, A^i)\) w.r.t.
the parameters of the SNN. However, the true distribution
\(q(G = 1 | S^i, A^i)\) is not known, but can be estimated in-
stead by fitting a critic function \(h: \{S, A\} \rightarrow [0, 1]\) (Tian,
Krishnan, and Isola 2019) to samples from the data distri-
bution \(q(S^i, A^i | G = 1)\) and \(q(S^i, A^i | G = 0)\), where \(S\)
and \(A\) represent the domains of the representations. According
to the properties of \(h\), \((N - 1)E_q(S^i,A^i|G=0)\log(1 -
Based on inequality (16), for any \( h \), \( \arg \max_h I(S^i; A^i) \) is the mutual information. Thus, the above learning problem (18) is relatively smaller, and the contrastive loss based on mutual information similarity between ANN and SNN of a certain layer is greater, and the representation of this layer is more aligned to the same dimension by \( g \) as well as further normalized by L2 norm before inner product.

Finally, for constructing the similarity-sensitive contrastive loss to enhance representations of SNNs, we weigh the NCE loss \( \mathcal{L}_{NCE}^l \) by CKA-based weight \( \lambda_{CKA}^l = 1/\text{CKA}(K, L) \) at \( l \)-th layer and combine it with classification loss. Then, the total loss \( L \) can be defined as:

\[
\mathcal{L}_{Total} = \sum_l \lambda_{CKA}^l \mathcal{L}_{NCE}^l + \mathcal{L}_{CE}^l,
\]

where \( \mathcal{L}_{CE}^l \) is the cross-entropy loss. When the representation similarity between ANN and SNN of a certain layer is relatively smaller, the CKA-based weight \( \lambda_{CKA}^l \) will be larger, and the contrastive loss based on mutual information maximization of this layer is greater, and the representation of this layer will be preferentially optimized. We adopt the spatial-temporal backpropagation (STBP) algorithm (Wu et al. 2019) to train the SNN with our method and the following STE surrogate gradients to solve the non-differentiable firing activity of the spiking neuron as other surrogate gradient (SG) methods (Rathi and Roy 2021; Guo et al. 2022b).

\[
\frac{dO}{dU} = \begin{cases} 
1, & \text{if } 0 \leq U \leq 1 \\
0, & \text{otherwise}
\end{cases}
\]

Experiments

In this section, extensive experiments were conducted to validate the effectiveness of the proposed method adopting widely-used spiking ResNet20 (Rathi and Roy 2020b; Sengupta et al. 2018), VGG16 (Rathi and Roy 2020b), ResNet18 (Fang et al. 2021a), ResNet19 (Zheng et al. 2020), and ResNet34 (Fang et al. 2021a) on both static and neuromorphic datasets including CIFAR-10 (Krizhevsky, Nair, and Hinton 2010), CIFAR-100 (Krizhevsky, Nair, and Hinton 2010), ImageNet (Deng et al. 2009), and CIFAR10-DVS (Li 2017). These networks are typically divided into four stages corresponding to four downsamples of the input feature map. The layer-wise representations used in our implementation consist of the representations after these four stages and the output feature representation after a subsequent global averaging pooling. We used the same architecture of ANN and SNN. We first train an ANN and then use it to guide the learning of the homogenous SNN. The hyperparameters for LIF neuron including the firing threshold \( U_{th} \), the membrane potential decaying \( \tau \), and reset potential \( U_{reset} \) were set as 0.5, 0.25, and 0 respectively. For static image datasets, the images were fed into the SNN model directly and encoded to 0/1 spikes using the first layer as recent works (Zheng et al. 2020; Rathi and Roy 2020b). For the neuromorphic image dataset, the 0/1 spike format was used directly. For comparison results, we list the mean top-1 accuracy and standard deviation when running three times for each experiment.

Ablation Study

To verify the effectiveness of our method, a series of ablation studies were conducted first, including the studies using spiking ResNet20 architecture with different time steps on the CIFAR-100. Table 3 lists the top-1 accuracy of these models. As can be seen, the test accuracy of our SNN models with similarity-sensitive contrastive learning (SSCL) is consistently higher than that of these vanilla SNN models. Moreover, it can be clearly seen that the proposed contrastive loss and the CKA benefit the overall improvement. Specifically, 2-timestep baseline SNN provides an accuracy of 68.27% on CIFAR-100, while SSCL could help SNN improve its accuracy to 69.81%, which is a huge improvement in the SNN field (close to 2.0%).

Comparison with SoTA Methods

We then conducted various experiments on both static and neuromorphic datasets. The details for the datasets and settings are given in the appendix.

CIFAR-10. The result for CIFAR-10 is shown in Table 1. Our models provide better performance than other SoTA methods using three commonly used networks with fewer time steps. Our VGG16 model with only 2 timesteps outperforms the KDSNN (Xu et al. 2023) with 4 timesteps by 2.78% accuracy. This demonstrates that, compared with the KD method which only uses the sample pairs from the joint distribution, it can improve the accuracy better that our model also uses the sample pairs of the product of the marginal distributions to align the representation. With
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Type</th>
<th>Architecture</th>
<th>Timestep</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SpikeNorm (Sengupta et al. 2018)</td>
<td>ANN2SNN</td>
<td>VGG16</td>
<td>2500</td>
<td>91.55%</td>
</tr>
<tr>
<td></td>
<td>Hybrid-Train (Rathi et al. 2020)</td>
<td>Hybrid training</td>
<td>VGG16</td>
<td>200</td>
<td>92.02%</td>
</tr>
<tr>
<td></td>
<td>Spike-basedBP (Lee et al. 2020)</td>
<td>SNN training</td>
<td>ResNet11</td>
<td>100</td>
<td>90.95%</td>
</tr>
<tr>
<td></td>
<td>STBP (Wu et al. 2019)</td>
<td>SNN training</td>
<td>CIFARNet</td>
<td>12</td>
<td>90.53%</td>
</tr>
<tr>
<td></td>
<td>TSSL-BP (Zhang and Li 2021)</td>
<td>SNN training</td>
<td>ResNet19</td>
<td>2</td>
<td>94.44%</td>
</tr>
<tr>
<td></td>
<td>PLIF (Fang et al. 2021b)</td>
<td>SNN training</td>
<td>PLIFNet</td>
<td>8</td>
<td>93.50%</td>
</tr>
<tr>
<td></td>
<td>GLIF (Yao et al. 2023)</td>
<td>SNN training</td>
<td>VGG16</td>
<td>4</td>
<td>91.05%</td>
</tr>
<tr>
<td></td>
<td>KDSNN (Xu et al. 2023)</td>
<td>SNN training</td>
<td>VGG16</td>
<td>5</td>
<td>92.70%</td>
</tr>
<tr>
<td></td>
<td>Diet-SNN (Rathi and Roy 2020b)</td>
<td>SNN training</td>
<td>ResNet20</td>
<td>5</td>
<td>91.78%</td>
</tr>
<tr>
<td></td>
<td>STBP-tdBN (Zheng et al. 2020)</td>
<td>SNN training</td>
<td>ResNet19</td>
<td>4</td>
<td>92.92%</td>
</tr>
<tr>
<td></td>
<td>TET (Deng et al. 2022)</td>
<td>SNN training</td>
<td>ResNet19</td>
<td>4</td>
<td>94.44%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>94.50%</td>
</tr>
<tr>
<td></td>
<td><strong>Our method</strong></td>
<td>SNN training</td>
<td>VGG16</td>
<td>2</td>
<td><strong>93.83% ± 0.10</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td><strong>94.27% ± 0.09</strong></td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>Diet-SNN (Rathi and Roy 2020b)</td>
<td>SNN training</td>
<td>ResNet20</td>
<td>2</td>
<td>92.34%</td>
</tr>
<tr>
<td></td>
<td>STBP-tdBN (Zheng et al. 2020)</td>
<td>SNN training</td>
<td>ResNet19</td>
<td>4</td>
<td>92.92%</td>
</tr>
<tr>
<td></td>
<td>TET (Deng et al. 2022)</td>
<td>SNN training</td>
<td>ResNet19</td>
<td>4</td>
<td>94.44%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>94.50%</td>
</tr>
<tr>
<td></td>
<td><strong>Our method</strong></td>
<td>SNN training</td>
<td>ResNet19</td>
<td>2</td>
<td><strong>95.33% ± 0.09</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td><strong>96.08% ± 0.10</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td><strong>92.16% ± 0.07</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td><strong>94.27% ± 0.07</strong></td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>BinarySNN (Lu and Sengupta 2020)</td>
<td>ANN2SNN</td>
<td>VGG15</td>
<td>62</td>
<td>63.20%</td>
</tr>
<tr>
<td></td>
<td>Hybrid-Train (Rathi et al. 2020)</td>
<td>Hybrid training</td>
<td>VGG11</td>
<td>125</td>
<td>67.90%</td>
</tr>
<tr>
<td></td>
<td>T2FSNN (Park et al. 2020)</td>
<td>ANN2SNN</td>
<td>VGG16</td>
<td>125</td>
<td>68.80%</td>
</tr>
<tr>
<td></td>
<td>SNNThroughKD (Takuya et al. 2021)</td>
<td>SNN training</td>
<td>VGG16</td>
<td>5</td>
<td>74.42%</td>
</tr>
<tr>
<td></td>
<td>Diet-SNN (Rathi and Roy 2020b)</td>
<td>SNN training</td>
<td>ResNet20</td>
<td>5</td>
<td>64.07%</td>
</tr>
<tr>
<td></td>
<td>TET (Deng et al. 2022)</td>
<td>SNN training</td>
<td>ResNet19</td>
<td>4</td>
<td>72.87%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>74.72%</td>
</tr>
<tr>
<td></td>
<td><strong>Our method</strong></td>
<td>SNN training</td>
<td>ResNet19</td>
<td>2</td>
<td><strong>77.75% ± 0.05</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td><strong>78.79% ± 0.10</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td><strong>67.93% ± 0.12</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td><strong>69.81% ± 0.12</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td><strong>72.86% ± 0.10</strong></td>
</tr>
<tr>
<td>CIFAR-DVS</td>
<td>Rollout (Kugele et al. 2020)</td>
<td>Rollout</td>
<td>DenseNet</td>
<td>10</td>
<td>66.80%</td>
</tr>
<tr>
<td></td>
<td>STBP-tdBN (Zheng et al. 2020)</td>
<td>SNN training</td>
<td>ResNet19</td>
<td>10</td>
<td>67.80%</td>
</tr>
<tr>
<td></td>
<td>RecDis-SNN (Guo et al. 2022b)</td>
<td>SNN training</td>
<td>ResNet19</td>
<td>10</td>
<td>72.42%</td>
</tr>
<tr>
<td></td>
<td>LIAF-Net (Wu et al. 2022)</td>
<td>Conv3D</td>
<td>LIAF-Net</td>
<td>10</td>
<td>71.70%</td>
</tr>
<tr>
<td></td>
<td>LIAF-Net (Wu et al. 2022)</td>
<td>LIAF</td>
<td>LIAF-Net</td>
<td>10</td>
<td>70.40%</td>
</tr>
<tr>
<td></td>
<td>Real Spike (Guo et al. 2022c)</td>
<td>SNN training</td>
<td>ResNet19</td>
<td>10</td>
<td>72.85%</td>
</tr>
<tr>
<td></td>
<td><strong>Our method</strong></td>
<td>SNN training</td>
<td>ResNet20</td>
<td>10</td>
<td><strong>80.00% ± 0.20</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td><strong>78.50% ± 0.10</strong></td>
</tr>
</tbody>
</table>

Table 1: Comparison with SoTA methods on CIFAR.
Table 2: Comparison with SoTA methods on ImageNet.

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
<th>Architecture</th>
<th>Timestep</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid-Train (Rathi et al. 2020)</td>
<td>Hybrid training</td>
<td>ResNet34</td>
<td>250</td>
<td>61.48%</td>
</tr>
<tr>
<td>SpikeNorm (Sengupta et al. 2018)</td>
<td>ANN2SNN</td>
<td>ResNet34</td>
<td>2500</td>
<td>69.96%</td>
</tr>
<tr>
<td>STBP-tdBN (Zheng et al. 2020)</td>
<td>SNN training</td>
<td>ResNet34</td>
<td>6</td>
<td>63.72%</td>
</tr>
<tr>
<td>TET (Deng et al. 2022)</td>
<td>SNN training</td>
<td>ResNet34</td>
<td>6</td>
<td>64.79%</td>
</tr>
<tr>
<td>RecDis-SNN (Guo et al. 2022b)</td>
<td>SNN training</td>
<td>ResNet34</td>
<td>6</td>
<td>67.33%</td>
</tr>
<tr>
<td>OTTT (Xiao et al. 2022)</td>
<td>SNN training</td>
<td>ResNet34</td>
<td>6</td>
<td>63.10%</td>
</tr>
<tr>
<td>MS-ResNet (Hu et al. 2023)</td>
<td>SNN training</td>
<td>ResNet18</td>
<td>6</td>
<td>63.10%</td>
</tr>
<tr>
<td>Real Spike (Guo et al. 2022c)</td>
<td>SNN training</td>
<td>ResNet18</td>
<td>4</td>
<td>63.68%</td>
</tr>
<tr>
<td>SEW ResNet (Fang et al. 2021a)</td>
<td>SNN training</td>
<td>ResNet18</td>
<td>4</td>
<td>63.18%</td>
</tr>
<tr>
<td>Our method</td>
<td>SNN training</td>
<td>ResNet18</td>
<td>4</td>
<td>66.78%±0.10</td>
</tr>
</tbody>
</table>

Table 3: Ablation experiments for the proposed contrastive loss and the CKA.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy/Timestep=2</th>
<th>Accuracy/Timestep=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>68.27%</td>
<td>71.45%</td>
</tr>
<tr>
<td>Distillation</td>
<td>68.74%</td>
<td>71.72%</td>
</tr>
<tr>
<td>Contrastive loss(uniform weighted)</td>
<td>69.25%</td>
<td>72.16%</td>
</tr>
<tr>
<td>w/ SSCL</td>
<td>69.81%</td>
<td>72.86%</td>
</tr>
</tbody>
</table>

only 2 timesteps, our ResNet19 model also outperforms the TET (Deng et al. 2022) and the STBP-tdBN (Zheng et al. 2020) with 6 timesteps by 1.58%, and 2.92%, respectively. The same superiority can also be seen with ResNet20 backbone. These comparison results demonstrate the effectiveness and efficiency of our method.

CIFAR-100. We have also validated our approach on CIFAR-100. The results for CIFAR-100 are presented in Table 1. With 5 timesteps, our VGG16 model achieves an accuracy of 76.37%, which outperforms the SNNThroughKD (Takuya et al. 2021) by 1.95%. This once again demonstrates the superiority of our method over KD methods. Moreover, our method also achieves better accuracy than other previous works with fewer timesteps on ResNet19 and ResNet20. For instance, the accuracy of our ResNet19 model with just 1 timestep can be as high as 77.75%, while the GLIF (Yao et al. 2023) and the TEBN (Duan et al. 2022) are even less accurate at 4 steps by 0.7% and 1.62%.

CIFAR10-DVS. The result for CIFAR10-DVS is shown in Table 1. Our method achieves 80.00% and 78.50% accuracy with ResNet19 and ResNet20 as the backbone respectively. It can be observed that the accuracy of our ResNet19 model is much higher than that of ResNet20 one. This can be explained from the perspective of mitigating overfitting. Since the data for training of CIFAR10-DVS is less sufficient than that of CIFAR10, and the overfitting issue is more severe on spiking ResNet19 model than ResNet20 one. Our contrastive learning method introduces the information of contrastive pairs from the product of marginal distributions and can be regarded as a proxy of data augmentation. Thus, it can alleviate the overfitting as well as improve the accuracy of spiking ResNet19 model more noticeably on CIFAR10-DVS.

ImageNet. All training settings for ImageNet are the same as CIFAR dataset but a temperature of 0.07 ($\tau = 0.07$) and training epoches as 320. We present the result for ImageNet in Table 2. It can be seen that our ResNet18 and ResNet34 model achieve 62.95% and 66.78% top-1 accuracy with only 4 timesteps, better than most of recent SoTA methods, only relatively smaller compared with SEW ResNet (Fang et al. 2021a) and Real Spike (Guo et al. 2022c). However, SEW ResNet (Fang et al. 2021a) and Real Spike (Guo et al. 2022c) are both SEW ResNet-based models, which are not typical SNN models. SEW ResNet-based models can fire positive integer spikes with the form of activation before addition. Since they have lost SNN’s advantages of event-driven and multiplication-addition transform, we adopt the original spiking ResNet which fires standard binary spikes.

Conclusion
In order to enhance the representation of SNNs, this work proposes a new similarity-sensitive contrastive learning framework. Via Mutual Information (MI) maximization, the positive pairs of SNN’s and ANN’s representation of each layer from the same input samples will be pulled closer, while the negative pairs from different samples will be pushed apart. Furthermore, a similarity indicators (CKAs) for each layer is introduced to balance the layer-wise “push and pull” scheme. A series of ablation studies show that the proposed method can greatly increase the SNN’s accuracy and will consistently outperforms the other SoTA methods.
References


