SpikeMatch: Semi-Supervised Learning with Temporal Dynamics of Spiking Neural Networks

Jini Yang[1,2]   Beomseok Oh[2]   Seungryong Kim[1]   Sunok Kim[2†]

[1]KAIST AI   [2]Korea Aerospace University

https://cvlab-kaist.github.io/SpikeMatch

Abstract

Spiking neural networks (SNNs) have recently been attracting significant attention for their biological plausibility and energy efficiency, but semi-supervised learning (SSL) methods for SNN-based models remain underexplored compared to those for artificial neural networks (ANNs). In this paper, we introduce SpikeMatch, the first SSL framework for SNNs that leverages the temporal dynamics through the leakage factor of SNNs for diverse pseudo-labeling within a co-training framework. By utilizing agreement among multiple predictions from a single SNN, SpikeMatch generates reliable pseudo-labels from weakly-augmented unlabeled samples to train on strongly-augmented ones, effectively mitigating confirmation bias by capturing discriminative features with limited labels. Experiments show that SpikeMatch outperforms existing SSL methods adapted to SNN backbones across various standard benchmarks.

1. Introduction

Spiking neural networks (SNNs) have recently attracted growing interest for their biological plausibility and energy efficiency. Unlike artificial neural networks (ANNs), SNNs communicate through discrete spikes generated when the membrane potential exceeds a threshold. This mechanism introduces intrinsic temporal dynamics, as the membrane potential evolves over time. The leakage factor in leaky integrate-and-fire (LIF) neurons determines how much past information is retained, resulting in time-varying outputs even from static inputs. This temporal variability has led researchers to interpret SNNs as implicit ensemble models [18, 53], and they have been successfully applied to various computer vision tasks [21, 43].

Despite these successes, current approaches typically rely on abundant labeled data, which can be costly and labor-intensive to acquire. Thus, it is needed to develop
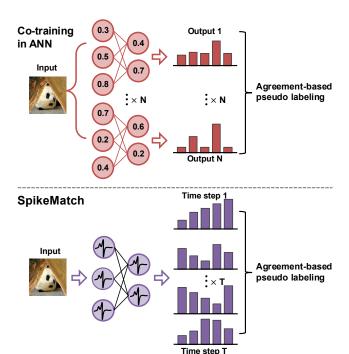

Figure 1. **Illustration comparing SNN-based SpikeMatch with ANN-based co-training.** As SNN-based training generates multiple predictions across time steps through binary spike signals, our approach leverages this temporal dynamic in predictions for agreement-based pseudo-labeling (**bottom**), in contrast to traditional ANN-based networks, which require additional parameters to produce diverse predictions (**top**).

suitable SNN method to address the challenge of label scarcity. Semi-supervised learning (SSL) is one of the powerful solutions to this issue, as it leverages large amounts of unlabeled data in combination with a small set of labeled data to improve model generalization.

Previous studies [23, 33, 54] used spike-timing-dependent plasticity (STDP) based two-stage SSL for SNNs, but these approaches struggle with deep networks

---

[†]Corresponding author.

and underperform compared to direct training. More recent work [39] applies curriculum learning, outperforming STDP-based methods but remains limited to shallow networks and simple digit tasks without comparison to modern SSL algorithms.

On the other hand, ANNs have benefited from well-established semi-supervised learning techniques, particularly self-training with pseudo-labels [8, 32]. While effective, these methods suffer from confirmation bias, where early errors propagate over time and degrade performance. To address this, recent methods combine pseudo-labeling with consistency regularization. FixMatch [45] introduces a confidence threshold and applies consistency between weak and strong augmentations. Building on this framework, recent methods further improve performance [9, 48, 52], and recent theoretical work [34] emphasizes the importance of learning semantic variations through augmentation.

Another approach is co-training [7], originally developed for multi-view settings, has also been adapted to enhance prediction diversity in single-view scenarios. Recent studies use multiple classification heads or differently initialized models to mitigate confirmation bias through cross-labeling [10]. Li et al. [35] show that diversity is a critical factor in co-training, often outperforming FixMatch. However, these methods typically require extra computational cost from multiple initializations or architectural modifications [20, 27].

In this paper, we leverage the temporal dynamics of SNNs within a co-training framework without requiring additional training resources. By simply increasing the leakage factor, we promote the SNN model to produce more diverse predictions and utilize this diversity for co-training through agreement-based pseudo-labeling.

With this approach, each temporal output of the SNN is trained using multiple perspectives, preventing extreme or dominant inaccurate pseudo-labels from being directly selected, thereby mitigating confirmation bias. Additionally, we incorporate successful SSL techniques from ANNs, such as distribution alignment and weak and strong augmentation, demonstrating their effectiveness in SNNs through experiments.

Our contributions are as follows:

- We propose an agreement-based pseudo-labeling method that treats the temporal outputs of an SNN model as multiple predictions within a co-training framework. This approach addresses common challenges in semi-supervised learning, such as confirmation bias, by effectively leveraging distinct semantic features.
- We show that a higher leakage factor in the Leaky Integrate-and-Fire (LIF) neuron enhances prediction diversity, enabling the model to capture a more diverse set of discriminative features.
- We evaluate the proposed methods against other state-of-the-art semi-supervised learning methods used with ANNs by implementing them with an SNN-based backbone on the CIFAR, STL, and ImageNet datasets.

2. Related Work

Semi-supervised learning. Semi-supervised learning (SSL) leverages large amounts of unlabeled data with limited labeled data. Two primary approaches are consistency regularization, which encourages consistent predictions under perturbations like augmentation [5], dropout [41], or random max-pooling [41], and pseudo-labeling, which assigns pseudo-labels to unlabeled samples for supervised training [32].

Recent approaches combine both methods for improved performance. ReMixMatch [4] uses multiple augmentations and distribution alignment to address class imbalance issues of pseudo-labels, but using multiple strong augmentations can yield inconsistent predictions. FixMatch [45] introduces a fixed threshold for confident pseudo-labeling using weak and strong augmentations, leading to a widely adopted confidence-aware framework [6, 26, 46, 52]. Motivated by FixMatch, FreeMatch [48] uses self-adaptive thresholding, considering both dataset-specific and class-specific factors, while SoftMatch [9] proposes a Gaussian function to adjust the threshold, balancing the trade-off between pseudo-label quality and quantity. Recently, Li et al. [34] theoretically proved that the success of FixMatch-like methods comes from strong augmentation, which encourages the model to learn multiple feature views.

Co-training is another SSL strategy that originally leverages mutually exclusive features extracted from multiple views and distinct networks [7]. However, since many datasets do not consist of multiple views, recent single-view co-training methods encourage diverse rather than independent features [47]. Li et al. [35] employ different architectures and augmentations to promote diversity and demonstrate that co-training outperforms FixMatch [45] under various conditions in segmentation tasks. Multi-head co-training [10] utilizes multiple classification heads and stochastic augmentation to generate diverse predictions for pseudo-labeling. JointMatch [55] trains two differently initialized networks for cross-pseudo-labeling, avoiding direct error accumulation. Similarly, UCC [20] employs two heads for cross-labeling in co-training to improve generalization ability for semi-supervised segmentation.

Direct training of spiking neural networks. Spiking neural networks (SNNs) use the Heaviside step function as the non-linear activation of leaky integrate-and-fire (LIF) neurons, which receive continuous inputs, accumulate membrane potential with leakage, and generate binary spikes when the threshold is exceeded at each time step, resulting in dynamic properties [11].

With these temporal characteristics, several studies have

explored the unique information propagation in SNNs. Kim et al. [29] analyze Fisher information concentration, showing that the first few time steps exhibit similar characteristics and are designated as early time steps, while the last few time steps share similar traits and are classified as late time steps. Deng et al. [15] optimize the outputs of each time step separately for generalization ability. Furthermore, Dong et al. [18] leverage the distinct output distributions at each time step as an ensemble-modeled prediction for self-distillation training, whereas Zhao et al. [53] enforce consistency across these output differences to improve classification performance. More recently, Ding et al. [17] investigate the differences in membrane potential at each time step and apply smoothing methods to enhance consistency.

Despite these advances in achieving high accuracy in image classification, most direct training methods focus on scenarios with abundant labeled data. Recently, some research has explored algorithms for direct training SNNs in label-scarce settings. For example, augmentation-based self-supervised learning to SNNs is applied in [3], but only achieved success on small datasets. Additionally, Nguyen et al. [39] implemented pseudo-labeling for semi-supervised learning but tested it only on simple digit-recognition tasks. Extending this method to deeper SNNs for complex tasks substantially increases computational cost and training time due to the use of Monte-Carlo dropout for pseudo-labeling. Moreover, it relies on early firing time for pseudo-labeling, which becomes challenging in deeper networks due to their complex temporal dynamics. This makes pseudo-label generation less reliable and increases the risk of confirmation bias.

3. Preliminaries

Leaky integrate-and-fire model. We utilize the leaky integrate-and-fire (LIF) model [1] for SNN training. In the LIF model, the membrane potential at time step $t \in \{1, ..., T\}$ and layer l before spiking, denoted by $u^{l,pre}(t)$, changes when the pre-synaptic input $W^l s^{l-1}(t)$ is injected as follows:

$$u^{l,pre}(t) = \tau u^l(t-1) + W^l s^{l-1}(t), \qquad (1)$$

where τ represents the leakage factor ranging within $(0, 1)$ and $u^l(t-1)$ denotes the membrane potential after spiking at $t-1$. The pre-synaptic input $W^l s^{l-1}(t)$ is calculated by multiplying the synaptic weight W^l by the input at time step t, $s^{l-1}(t)$. The neuron generates spike output $s^l(t)$, when the $u^{l,pre}(t)$ exceed the firing threshold V_{th}:

$$s^l(t) = \Theta(u^{l,pre}(t) - V_{th}) = \begin{cases} 1, & if\ u^{l,pre}(t) \geq V_{th} \\ 0, & otherwise \end{cases}$$
$$(2)$$
$$u^l(t) = u^{l,pre}(t)(1 - s^l(t)), \qquad (3)$$

where Θ represents the Heaviside step function, and $u^l(t)$ is reset to zero if $s^l(t)$ is 1. In SNNs, this process is applied iteratively. Let $O^t(x)$ define the integrated pre-synaptic input $W^l s^{l-1}(t)$ of the last output layer for input x without triggering a spike [40]. $p_t(y \mid x) = \text{softmax}(O^t(x))$ denote the predicted class distribution produced by applying the softmax function to $O^t(x)$. This distribution $p_t(y \mid x)$ is used for classification [15].

Leakage factor of LIF neuron. In [2], an ensemble model leveraging diversity solely through different random seeds improves performance by aggregating individual learned semantic views, enabling the model to capture multiple discriminative features. Motivated by this observation, we employ a different approach from existing ensemble-like SNN methods [17, 53], which typically utilize temporal dynamics with abundant data by enforcing consistency. Instead, we leverage this diversity to learn multiple distinctive views in a limited-data setting with a single spiking network. We achieve this by using a higher the leakage factor unlike ANN require additional training or multi-view data. To analyze the role of the leakage factor in membrane potential dynamics, we assume that neurons do not emit spikes between time steps. From Equation (3), we can expand this recursively as for $t \in [2, T)$, where $t' \in [1, t)$ [16]:

$$u^l(t) = \tau^{t-t'} u^l(t') + \sum_{i=t'+1}^{t} \tau^{t-i} W^l s^{l-1}(i), \qquad (4)$$

To quantify the diversity of membrane potential, we select a time step $t' \in [1, t)$ and subtract its membrane potential from $u^l(t)$:

$$|u^l(t) - u^l(t')| =$$
$$|(\tau^{t-t'} - 1)u^l(t') + \sum_{i=t'+1}^{t} \tau^{t-i} W^l s^{l-1}(i)|. \qquad (5)$$

For larger τ, past spike information persists for a longer duration, increasing the difference between time steps. This enhances temporal diversity, providing the model with more opportunities to learn distinct features in the agreement-based co-training framework. Based on this property, we design our SSL method to leverage the inherent temporal dynamics of SNNs, enabling effective pseudo-labeling without additional diversity constraints.

4. SpikeMatch

4.1. Problem statement and motivation

In this section, we introduce our detailed methods utilizing an SNN-based semi-supervised learning (SSL) approach that leverages co-training inspired by the temporal dynamics of SNNs. For a C-class classification problem, let a
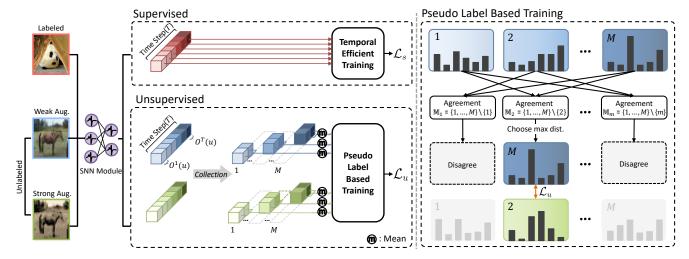
Figure 2. **An Overview of SpikeMatch (left) and pseudo-label based training (right). (left)**: The weakly augmented labeled image (red) produces T outputs through the SNN model and optimized by TET loss. For unlabeled data, both weakly (blue) and strongly (green) augmented images pass through the same SNN model. The T outputs produced by each image are partitioned and averaged into M predictions. The predictions from the weakly augmented image serve as the soft pseudo-label, using their agreement to optimize the strongly augmented image via cross-entropy. **(right)**: A pseudo-label from one collection (green) is used when the remaining $M - 1$ collections (blue) agree on the same class with the highest probability, and the prediction with the highest confidence is selected as the soft pseudo-label.

batch of B labeled samples be denoted as $\mathcal{X} = \{(x_b, y_b) : b \in (1, ..., B)\}$ and a batch of μB unlabeled samples as $\mathcal{U} = \{u_b : b \in (1, ..., \mu B)\}$, where μ is a hyperparameter representing the relative size of unlabeled samples $\mathcal{U}$ to labeled samples $\mathcal{X}$, and y_b is the one-hot label.

Unlike traditional ANNs, which process static images spatially, SNNs propagate information through both spatial and temporal dimensions, producing multiple predictions over time, as shown in Figure 1. Based on these distinctive characteristics of the SNN model, we propose a novel SNN-based SSL model that combines the temporal dynamics of spiking neural networks with a co-training mechanism to generate agreement-based pseudo-labels, utilizing consistency regularization. The overall architecture is visualized in Figure 2.

4.2. Temporal efficient supervised learning

In SSL, the amount of labeled data $|\mathcal{X}|$ is comparatively small compared to the unlabeled data $|\mathcal{U}|$, which can degrade the generalization ability of the model [26]. We address this problem by leveraging the temporal dynamics of SNNs without pre-training or adding extra parameters. For supervised learning, the loss from labeled data is defined by the Temporal Efficient Training (TET) loss [15], which is known to enhance the generalization ability of the model by optimizing the output distribution at each time step $p_t(y \mid x)$ individually. The supervised loss $\mathcal{L}_s$ is de-

fined as:

$$\mathcal{L}_s = \frac{1}{BT} \sum_{b=1}^{B} \sum_{t=1}^{T} \mathrm{H}(y_b, p_t(y \mid \alpha(x_b))), \qquad (6)$$

where $\mathrm{H}(\cdot)$ denotes the cross-entropy loss and $\alpha(\cdot)$ represents a weak augmentation.

4.3. Agreement-based pseudo-labeling

Pseudo-labeling. With limited labeled data, consistency regularization with weak and strong augmentations effectively helps to avoid overfitting [45]. Following this approach, we incorporate consistency regularization along with pseudo-labeling. However, using pseudo-labels from a single prediction can lead to error accumulation [22]. To address this challenge, motivated by co-training [10, 55], our method for unsupervised training focuses on generating complementary predictions through temporal variations, treating these variations as distinct output distributions, as visualized in Figure 2.

In [29], the authors investigate the temporal information in SNNs using Fisher information concentration and claim that early and late time steps focus on different feature information. Since pseudo-labeling with agreement across predictions at every time step is time-consuming, for simplicity and diversity in predictions, we reduce the total T temporal outputs to M by sequentially grouping every T/M outputs across the time steps $\{1, ..., T\}$. By averaging each of these groups, we obtain M predictions, denoted as g_m

4

Algorithm 1 SpikeMatch algorithm

Require: Labeled batch $\mathcal{X} = \{(x_b, y_b) : b \in (1, ..., B)\}$, unlabeled batch $\mathcal{U} = \{u_b : b \in (1, ..., \mu B)\}$, time step T, number of collected predictions M, unsupervised loss weight λ, weak and strong augmentations $\alpha(\cdot)$, $\mathcal{A}(\cdot)$

1: // Supervised learning on $\mathcal{X}$
2: $\mathcal{L}_s = \frac{1}{BT} \sum_{b=1}^{B} \sum_{t=1}^{T} \mathrm{H}(y_b, p_t(y \mid \alpha(x_b)))$ ▷ Compute TET loss for labeled samples
3: // Unsupervised learning on $\mathcal{U}$
4: **for** $b = 1$ to μB **do**
5:      **for** $m = 1$ to M **do**   ▷ Calculate the class distributions for m
6:          $q_b^m = \frac{1}{|t_m|} \sum_{t \in t_m} O^t(\alpha(u_b))$
7:          $g_m(y \mid \mathcal{A}(u_b)) = \frac{1}{|t_m|} \sum_{t \in t_m} O^t(\mathcal{A}(u_b))$
8:      **for** $m = 1$ to M **do**   ▷ Select the confident pseudo-label
9:          $\hat{q}_b^m = q_b^i$,   where $i = \arg\max_{j \in \mathbb{M}_m} (\max_c g_j(y = c \mid \alpha(u_b)))$
10:      **for** $m = 1$ to M **do**   ▷ Generate pseudo-label with agreement
11:          Set $\hat{c}^m = \arg\max_c g_i(y = c \mid \alpha(u_b))$ for any $i \in \mathbb{M}_m$
12:          $\ell_b = \sum_{m=1}^{M} \prod_i \mathbb{1}(\arg\max_c g_i(y = c \mid \alpha(u_b)) = \hat{c}^m) \times \mathrm{H}(\hat{q}_b^m, g_m(y \mid \mathcal{A}(u_b))), \quad i \in \mathbb{M}_m$
13: $\mathcal{L}_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \ell_b$
14: **return** $\mathcal{L}_s$ and $\mathcal{L}_u$

---

where $m \in \{1, \ldots, M\}$, which serve as the final set of predictions for unsupervised training:

$$g_m(y \mid u_b) = \frac{1}{|t_m|} \sum_{t \in t_m} O^t(u_b), \qquad (7)$$

where $t_m = \{t \mid (m - 1) \cdot \frac{T}{M} + 1 \leq t \leq m \cdot \frac{T}{M}\}$, $|t_m| = \frac{T}{M}$. Each temporal output $O^t(u_b)$ over T is divided into M collections and averaged, resulting in M predicted class distributions $g_m(y \mid u_b)$, where $m \in \{1, \ldots, M\}$. In our method, these predictions interact with others through pseudo-labels on unlabeled data, leveraging temporal prediction diversity to generate more diverse pseudo-labels. This partitioning approach, similar to other co-training methods [10, 20, 35, 55], also demonstrates robustness to confirmation bias that can arise from incorrect predictions in early training stages.

To facilitate interactions among predictions with consistency regularization, we denote $q_b^m = g_m(y \mid \alpha(u_b))$ as the predicted class distribution for each m in the M predictions on weakly augmented unlabeled data. To benefit from diversity and alleviate confirmation bias, we generate the pseudo-label $\hat{q}_b^m$ from each q_b^m using the other $M - 1$ predictions in $\mathbb{M}_m = \{1, ..., M\} \setminus \{m\}$, where $\mathbb{M}_m$ represents the set of indices excluding m. The pseudo-label $\hat{q}_b^m$ is defined as:

$$\hat{q}_b^m = q_b^i, \text{ where } i = \arg\max_{j \in \mathbb{M}_m} \left(\max_c g_j(y = c \mid \alpha(u_b)) \right), \qquad (8)$$

where i indicates the index within the set $\mathbb{M}_m$ that yields the highest value of $g_i(y = c \mid \alpha(u_b))$ over the class c. These pseudo-labels are trained with additional constraints based on the agreement of other predictions $q_b^i (i \in \mathbb{M}_m)$,

utilizing temporal dynamics similar to co-training methods, which are known to reduce error accumulation [19, 55]. If the maximal classes among $q_b^i (i \in \mathbb{M}_m)$ are the same, the $\hat{q}_b^m$ is used as a pseudo-label for training with the output prediction on m of the strongly augmented unlabeled data, $g_m(y \mid \mathcal{A}(u_b))$ as shown in Figure 2 (right). This strategy effectively integrates co-training principles with the temporal nature of SNNs to enhance learning from both labeled and unlabeled data, while preserving prediction diversity and utilizing a large set of unlabeled samples in self-training.

Loss function. The cross-entropy loss on the unlabeled data is then calculated as follows:

$$\mathcal{L}_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \sum_{m=1}^{M} \prod_{i \in \mathbb{M}_m} \mathbb{1}(\arg\max_c g_i(y = c \mid \alpha(u_b)) = \hat{c}^m)$$
$$\times \mathrm{H}(\hat{q}_b^m, g_m(y \mid \mathcal{A}(u_b))), \qquad (9)$$

where $\hat{c}^m$ denotes the class with the maximum probability in q_b^i, with i representing any element in $\mathbb{M}_m$, and $\mathcal{A}$ represents a strong augmentation. Finally, the overall loss is written as:

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_u. \qquad (10)$$

where λ is a hyper-parameter that balances the supervised and unsupervised loss.

4.4. Multi-view feature learning

Originally, co-training required multi-view data to achieve diversity. However, recent co-training methods generate diversity by using differently initialized models trained on the same dataset. These approaches have been shown to outperform FixMatch-based architectures in semi-supervised

Dataset	CIFAR-10			CIFAR-100		
Labels	4	25	400	4	25	100
UDA [50]	$69.72_{\pm 6.65}$	$87.76_{\pm 0.21}$	$90.06_{\pm 0.06}$	$44.66_{\pm 1.63}$	$60.35_{\pm 0.67}$	$66.51_{\pm 0.15}$
FixMatch [45]	$67.97_{\pm 4.42}$	$86.84_{\pm 0.91}$	$89.40_{\pm 0.23}$	$38.39_{\pm 0.91}$	$58.40_{\pm 0.48}$	$64.75_{\pm 0.17}$
AdaMatch [6]	$82.13_{\pm 6.98}$	$87.15_{\pm 0.62}$	$89.44_{\pm 0.11}$	$44.54_{\pm 2.52}$	$59.52_{\pm 0.24}$	$65.86_{\pm 0.44}$
FreeMatch [48]	$85.65_{\pm 4.16}$	$87.86_{\pm 0.34}$	$89.64_{\pm 0.27}$	$46.49_{\pm 2.75}$	$62.01_{\pm 0.27}$	$67.10_{\pm 0.35}$
SoftMatch [9]	$84.27_{\pm 7.69}$	$87.97_{\pm 0.18}$	$89.91_{\pm 0.28}$	$45.99_{\pm 3.06}$	$61.97_{\pm 0.53}$	$67.35_{\pm 0.08}$
RegMixMatch [25]	$82.59_{\pm 4.02}$	$87.36_{\pm 0.55}$	$90.08_{\pm 0.65}$	$38.49_{\pm 3.09}$	$60.27_{\pm 1.96}$	$66.69_{\pm 0.24}$
SpikeMatch w/o TET	$87.98_{\pm 0.51}$	$88.13_{\pm 0.37}$	$\mathbf{91.00}_{\pm 0.002}$	$47.45_{\pm 1.79}$	$63.18_{\pm 0.09}$	$69.37_{\pm 0.003}$
SpikeMatch	$\mathbf{88.13}_{\pm 0.79}$	$\mathbf{88.36}_{\pm 0.06}$	$90.93_{\pm 0.04}$	$\mathbf{47.52}_{\pm 2.18}$	$\mathbf{63.34}_{\pm 0.04}$	$\mathbf{69.50}_{\pm 0.17}$

Table 1. **Accuracy (%) on CIFAR-10 and CIFAR-100 using 3 different random seeds.** The best results are in bold.

Seed	0	1	2	Mean
FreeMatch [48]	56.73	49.94	57.00	$54.55_{\pm 4.00}$
SoftMatch [9]	58.40	50.58	58.14	$55.70_{\pm 4.44}$
SpikeMatch	62.36	64.69	61.36	$62.80_{\pm 1.70}$

Table 2. **Accuracy (%) on STL-10.**

learning [55]. This diversity allows the model to learn distinct features in an ensemble-like manner [2]. We leverage this diversity through the leakage factor of the LIF neuron, where increasing it generates more diverse predictions over time without requiring additional constraints for diversity.

Our agreement-based method leverages the co-training framework to effectively capture diverse features under limited labeled data. Unlike FixMatch-like methods that rely on confidence thresholding, our approach does not impose such constraints. In the early stages of training, extreme predictions from a single view are filtered out if they are not in agreement with others. This process prevents the temporal outputs from being influenced by a single extreme prediction, reducing their impact and improving robustness against confirmation bias. Additionally, each temporal output is optimized based on complementary outputs, providing an opportunity for the model to learn different features from other temporal predictions.

5. Experiments

5.1. Implementation details

We evaluate our approach on CIFAR [30], STL-10 [12], and ImageNet [14], which are widely used benchmark datasets for semi-supervised image classification. Additional results including neuromorphic dataset are included in the Appendix. Since many SNN methods adopt VGG-based architectures [15], we use VGG networks [44] for our main experiments. We set the number of time steps T to 4 and

generate $M = 3$ predictions for each unlabeled sample. When T is not divisible by M, we divide the outputs into segments of sizes $1, 1, 2$ and average the outputs within each segment to produce the M predictions. Further training details are provided in the Appendix.

5.2. Main results

Analysis on CIFAR. We evaluate SpikeMatch on CIFAR-10, CIFAR-100, and STL-10, comparing it against recent SSL methods originally designed for ANNs, including UDA [50], FixMatch [45], AdaMatch [6], FreeMatch [48], SoftMatch [9], and RegMixMatch [25]. To ensure a fair comparison, we re-implement these methods with an SNN backbone using their original settings. All results are averaged over three seeds with standard deviations reported. As shown in Table 1, our approach achieves superior performance compared to other methods, including recent state-of-the-art methods FreeMatch, SoftMatch, and Reg-MixMatch, across various label regimes on CIFAR-10 and CIFAR-100. Furthermore, SpikeMatch demonstrates a notably low standard deviation when using only four examples per class.

Analysis on STL-10. We further evaluate our approach on the more challenging STL-10 dataset, which contains high-resolution examples. We compare our method against recent state-of-the-art approaches, FreeMatch and SoftMatch, using a randomly selected subset of 40 labeled samples. As shown in Table 2, our method achieves 7.10% higher accuracy than SoftMatch while maintaining a low standard deviation. For visualization, we utilize spiking activation map (SAM) [28], to generate temporal localization maps that highlight regions of interest. Figure 3 illustrates that under scarce label conditions, directly adapting existing methods to SNNs often leads to class imbalance. These results suggest that our agreement-based temporal dynamic approach effectively captures diverse discriminative features, enhancing the distinction between frequently confused classes.
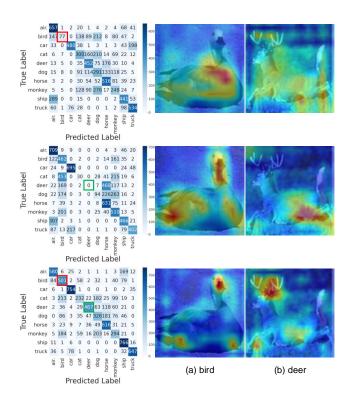
6

Figure 3. **Confusion matrix and SAM on STL-10.** Unlike FreeMatch (**first row**), which is confused by the bird class, and SoftMatch (**second row**), which struggles with the deer class, our method (**last row**) captures more discriminative features, such as the beak of (a) bird and the antlers of (b) deer. This results in a more diagonal confusion matrix, making the model less prone to confusion among similar classes.

Methods	Top-1	Top-5
SoftMatch [9]	45.66	71.44
SpikeMatch	**49.68**	**75.07**

Table 3. **Accuracy (%) on ImageNet using 100 labels per class.**

Methods	Accuracy	Energy efficiency
FixMatch (ANN)	$78.69_{\pm 9.13}$	$1\times$
SoftMatch (ANN)	$79.04_{\pm 12.54}$	$1\times$
SpikeMatch	$81.92_{\pm 3.06}$	$20.8\times$

Table 4. **Accuracy (%) and energy consumption (including MAC) on CIFAR10 with only one label per class.**

Results on ImageNet. To evaluate our method on a deeper network and a larger dataset, we conduct experiments using SEW-ResNet50 [21] on ImageNet [14] with 100 labels per class. The results presented in Table 3 show that our method outperforms SoftMatch even with a deep backbone network on a challenging task.
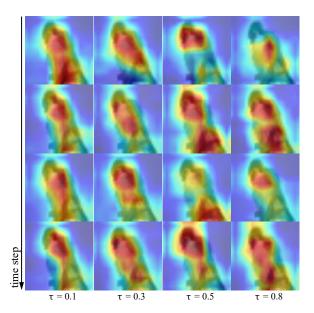

Figure 4. **SAM on leakage factor and time step variance.** As the leakage factor increases, SAM visualizations at each time step become more diverse, providing more opportunities for the model to learn discriminative features.

Energy and label efficiency. As we leverage the inherent temporal dynamics of SNNs in a co-training manner, Spike-Match achieves remarkable performance in the most challenging setting of CIFAR-10 with only one labeled sample per class as shown in Table 4. Unlike earlier setups that simply replace the backbone of existing methods with an SNN, this section compares against SoftMatch with its original ANN backbone. Despite this, SpikeMatch not only outperforms SoftMatch, a state-of-the-art ANN-based method, but also demonstrates the ability to extract highly discriminative features from unlabeled data under extreme label scarcity. Moreover, our method fully preserves the energy efficiency of SNNs, achieving over 20.8 times higher energy efficiency compared to the ANN-based counterpart. This makes SpikeMatch well suited for deployment in scenarios constrained by both energy and labeled data.

5.3. Configuration and ablation study

Leakage factor of LIF. To demonstrate that a higher leakage factor induces more diverse predictions over time, we conduct an ablation study by varying the leakage factor. As shown in Figure 4, increasing the leakage factor results in more diverse SAM visualizations at last layer. Additionally, we calculate the cosine similarity for every pair of time steps from the last layer. As shown in Figure 5, without any explicit diversity constraints, the mean cosine similarity at each leakage factor consistently decreases across all seeds. This indicates that the model produces more diverse temporal features as τ increases. This increased diversity allows the model to learn a wider range of features, ultimately im-
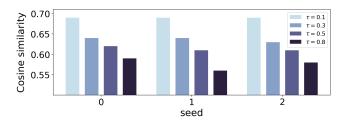
Figure 5. **Cosine similarity at varying leakage factors (τ).** Each column represents the average of 6 cosine similarity values computed across 4 time steps for each τ, at seeds 0, 1, and 2. The cosine similarity is calculated from 6 combinations of features from the 4 time steps at the last layer.

Leakage factor	0	0.1	0.3	0.5	0.8
SoftMatch	79.71	84.27	84.41	84.27	85.52
SpikeMatch	80.16	85.43	86.33	88.13	88.17

Table 5. **Averaged accuracy (%) over three seeds on CIFAR-10 with 4 labels per class under different leakage factors.**

proving SSL performance. Additional empirical analyses of this diversity are provided in the appendix. As summarized in Table 5, both SoftMatch and our method achieve higher performance with a higher leakage factor. However, the performance gap is more notable in our method. This is because our agreement-based approach is more effective at leveraging minor features that emerge from temporal diversity, leading to greater improvements in SSL performance. However, a higher leakage factor accumulates membrane potential and generally produces more spikes, leading to increased energy consumption. To balance efficiency and performance, we set τ to 0.5 by default.

Ablation Study. We evaluated the contributions of each technique in SpikeMatch through ablation studies on CIFAR with 4 labels per class, as shown in Table 6. We compared SpikeMatch with an averaged prediction model averaging temporal outputs (single prediction model), an intra prediction model without agreement-based pseudo-labeling, a version without distribution alignment (DA) [4], and one using a threshold as an additional constraint for pseudo-labeling.

The averaged prediction model showed sub-optimal results because it cannot leverage intrinsic diversity of SNN for pseudo-labeling. Using intra prediction also resulted in performance degradation, as it does not complement each prediction with distinct features. In contrast, SpikeMatch leverages cross-labeling to effectively utilize this diversity. To verify the DA in SNNs, we performed ablation experiments by applying SpikeMatch without DA. There was a performance drop because DA aligns the pseudo-labels from all collected predictions. We also added a threshold as an additional constraint in agreement-based pseudo-labeling but observed a performance decline, possibly because setting the threshold to 0.8 is not optimal for our SNN network. However, finding the optimal threshold for SNNs for each of the M predictions, considering class- and data-dependent adaptation at each epoch, would significantly increase training time. Therefore, using agreement-based pseudo-labeling without additional thresholds achieves optimal performance by leveraging sufficient and diverse predictions in our SNN setting.

Ablation	CIFAR-10	CIFAR-100
SpikeMatch	86.94	48.31
- Averaged prediction	80.81	44.05
- Intra prediction	80.46	44.02
- w/o DA	76.32	42.91
- w/ threshold	83.17	41.73

Table 6. **Ablation studies**

T	M=1	M=3
4	80.81	86.94
6	81.47	87.84
8	-	88.54

Table 7. **Ablation studies of time step (T) and the number of collections (M)**

Impact of time steps and the number of collections. Additionally, we varied the number of time steps (T) and prediction collections (M). As shown in Table 7, increasing T improves accuracy but results in higher energy consumption. Thus, we adopt $T = 4$ as a trade-off. Using multiple collections $M = 3$ consistently improves generalization over single-prediction pseudo-labeling, especially at lower T, demonstrating the strength of temporal diversity in SpikeMatch even under limited computation. Additional ablations with network variations and hyperparameter details are included in the Appendix.

6. Conclusion

In this work, we presented SpikeMatch, a semi-supervised learning (SSL) framework for spiking neural networks (SNNs) that utilize their temporal dynamics via the leakage factor to achieve diverse pseudo-labeling within a co-training framework. By utilizing agreement among multiple predictions from a single SNN, SpikeMatch produces reliable pseudo-labels from weakly augmented unlabeled samples to train on strongly augmented ones, effectively mitigating confirmation bias and enhancing feature learning with limited labeled data. Experiments show that SpikeMatch outperforms existing SSL methods adapted to SNN backbones across various benchmark.

References

[1] A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, 95:1–19, 2006. 3, 12

[2] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020. 3, 6

[3] Yeganeh Bahariasl and Saeed Reza Kheradpisheh. Self-supervised contrastive learning in spiking neural networks. In *2024 13th Iranian/3rd International Machine Vision and Image Processing Conference (MVIP)*, pages 1–5. IEEE, 2024. 3

[4] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 2, 8

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 2

[6] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. *arXiv preprint arXiv:2106.04732*, 2021. 2, 6, 14

[7] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998. 2

[8] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 6912–6920, 2021. 2

[9] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint arXiv:2301.10921*, 2023. 2, 6, 7, 12, 13, 14, 15, 16

[10] Mingcai Chen, Yuntao Du, Yi Zhang, Shuwei Qian, and Chongjun Wang. Semi-supervised learning with multi-head co-training. In *Proceedings of the AAAI conference on artificial intelligence*, pages 6278–6286, 2022. 2, 4, 5, 16

[11] Sayeed Shafayet Chowdhury, Chankyu Lee, and Kaushik Roy. Towards understanding the effect of leak in spiking neural networks. *Neurocomputing*, 464:83–94, 2021. 2

[12] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 6, 12

[13] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 13, 14

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6, 7, 12

[15] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022. 3, 4, 6, 12, 13, 14

[16] Jianhao Ding, Tong Bu, Zhaofei Yu, Tiejun Huang, and Jian Liu. Snn-rat: Robustness-enhanced spiking neural network through regularized adversarial training. *Advances in Neural Information Processing Systems*, 35:24780–24793, 2022. 3

[17] Yongqi Ding, Lin Zuo, Mengmeng Jing, Pei He, and Hanpu Deng. Rethinking spiking neural networks from an ensemble learning perspective. *arXiv preprint arXiv:2502.14218*, 2025. 3

[18] Yiting Dong, Dongcheng Zhao, and Yi Zeng. Temporal knowledge sharing enable spiking neural network learning from past and future. *IEEE Transactions on Artificial Intelligence*, 2024. 1, 3

[19] W Dong-DongChen and ZH WeiGao. Tri-net for semi-supervised deep learning. In *Proceedings of twenty-seventh international joint conference on artificial intelligence*, pages 2014–2020, 2018. 5

[20] Jiashuo Fan, Bin Gao, Huan Jin, and Lihui Jiang. Ucc: Uncertainty guided cross-head co-training for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9947–9956, 2022. 2, 5

[21] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021. 1, 7, 13

[22] Zhengyang Feng, Qianyu Zhou, Qiqi Gu, Xin Tan, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Dmt: Dynamic mutual training for semi-supervised learning. *Pattern Recognition*, 130:108777, 2022. 4

[23] Kotaro Furuya and Jun Ohkubo. Semi-supervised learning combining backpropagation and stdp: Stdp enhances learning by backpropagation with a small amount of labeled data in a spiking neural network. *Journal of the Physical Society of Japan*, 90(7):074802, 2021. 1

[24] Fuqiang Gu, Jiarui Dou, Mingyan Li, Xianlei Long, Songtao Guo, Chao Chen, Kai Liu, Xianlong Jiao, and Ruiyuan Li. Eventaugment: learning augmentation policies from asynchronous event-based data. *IEEE Transactions on Cognitive and Developmental Systems*, 16(4):1521–1532, 2024. 13

[25] Haorong Han, Jidong Yuan, Chixuan Wei, and Zhongyang Yu. Regmixmatch: Optimizing mixup utilization in semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 17032–17040, 2025. 6

[26] Zhuo Huang, Li Shen, Jun Yu, Bo Han, and Tongliang Liu. Flatmatch: Bridging labeled data and unlabeled data with cross-sharpness for semi-supervised learning. *Advances in Neural Information Processing Systems*, 36:18474–18494, 2023. 2, 4, 14

[27] Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson WH Lau. Dual student: Breaking the limits of

the teacher in semi-supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6728–6736, 2019. 2

[28] Youngeun Kim and Priyadarshini Panda. Visual explanations from spiking neural networks using inter-spike intervals. *Scientific reports*, 11(1):19037, 2021. 6, 17, 18

[29] Youngeun Kim, Yuhang Li, Hyoungseob Park, Yeshwanth Venkatesha, Anna Hambitzer, and Priyadarshini Panda. Exploring temporal information dynamics in spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8308–8316, 2023. 3, 4

[30] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6, 12

[31] Souvik Kundu, Massoud Pedram, and Peter A Beerel. Hiresnn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5209–5218, 2021. 12

[32] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, page 896. Atlanta, 2013. 2

[33] Jongseok Lee and Donggyu Sim. Semi-supervised learning for spiking neural networks based on spike-timing-dependent plasticity. *IEEE Access*, 11:35140–35149, 2023. 1

[34] Jingyang Li, Jiachun Pan, Vincent YF Tan, Kim-Chuan Toh, and Pan Zhou. Towards understanding why fixmatch generalizes better than supervised learning. *arXiv preprint arXiv:2410.11206*, 2024. 2

[35] Yijiang Li, Xinjiang Wang, Lihe Yang, Litong Feng, Wayne Zhang, and Ying Gao. Diverse cotraining makes strong semi-supervised segmentor. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16009–16021. IEEE Computer Society, 2023. 2, 5

[36] Charlotte Loh, Rumen Dangovski, Shivchander Sudalairaj, Seungwook Han, Ligong Han, Leonid Karlinsky, Marin Soljacic, and Akash Srivastava. On the importance of calibration in semi-supervised learning. *arXiv preprint arXiv:2210.04783*, 2022. 16

[37] Khanh-Binh Nguyen. Sequencematch: Revisiting the design of weak-strong augmentations for semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 96–106, 2024. 14

[38] Khanh-Binh Nguyen and Joon-Sung Yang. Boosting semi-supervised learning by bridging high and low-confidence predictions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1028–1038, 2023. 16

[39] Thao NN Nguyen, Bharadwaj Veeravalli, and Xuanyao Fong. A semi-supervised learning method for spiking neural networks based on pseudo-labeling. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2023. 2, 3

[40] Nitin Rathi and Kaushik Roy. 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*, 34(6):3174–3182, 2021. 3

[41] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29, 2016. 2

[42] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13:95, 2019. 13

[43] Xinyu Shi, Zecheng Hao, and Zhaofei Yu. Spikingresformer: Bridging resnet and vision transformer in spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5610–5619, 2024. 1

[44] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[45] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 2, 4, 6, 12, 14, 15, 16

[46] Tiberiu Sosea and Cornelia Caragea. Marginmatch: Improving semi-supervised learning with pseudo-margins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15773–15782, 2023. 2

[47] Wei Wang and Zhi-Hua Zhou. Theoretical foundation of co-training and disagreement-based algorithms. *arXiv preprint arXiv:1708.04403*, 2017. 2

[48] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2022. 2, 6, 13, 14, 15, 16

[49] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018. 12, 14

[50] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020. 6, 14

[51] Sergey Zagoruyko. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 13

[52] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. 2, 14

[53] Dongcheng Zhao, Guobin Shen, Yiting Dong, Yang Li, and Yi Zeng. Improving stability and performance of spiking neural networks through enhancing temporal consistency. *Pattern Recognition*, page 111094, 2024. 1, 3

[54] Xiaoman Zhao, Xianghong Lin, and Zhen Zhang. A semi-supervised multi-spike learning algorithm for deep spiking neural networks. In *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0976–0982. IEEE, 2023. 1

[55] Henry Peng Zou and Cornelia Caragea. Jointmatch: A unified approach for diverse and collaborative pseudo-labeling to semi-supervised text classification. In *The 2023 Conference on Empirical Methods in Natural Language Processing*. 2, 4, 5, 6

SpikeMatch: Semi-Supervised Learning with Temporal Dynamics of Spiking Neural Networks

Supplementary Material

A. Experimental Setup

A.1. Semi-supervised training details

For each dataset CIFAR [30], STL-10 [12], and ImageNet [14], we select a various number of labeled data, sampling an equal number of examples from each class. In all datasets, we randomly sample a small subset as labeled data and treat the remaining examples as unlabeled, except for STL-10, which originally includes an unlabeled dataset. The hyperparameter configurations for semi-supervised learning are summarized in Table A.1, where we generally follow the standard semi-supervised learning protocol [9, 45], with certain adjustments tailored to the SNN backbone.

A.2. Spiking-specific training

To address the non-differentiability of spike generation, we adopt a triangular-shaped surrogate gradient function [15]. As we utilize the leaky integrate-and-fire (LIF) model [1], for the direct training of spiking neural networks (SNNs), we use the spatio-temporal back-propagation (STBP) rule for gradient descent training [49]. The membrane potential before spiking at time step $t \in \{1, ..., T\}$ and layer $l \in \{1, ..., L\}$ denoted as $u^{l,pre}(t)$, is updated based on the pre-synaptic input $W^l s^{l-1}(t)$, which is calculated by multiplying the synaptic weight W^l by the input at time step t, $s^{l-1}(t)$:

$$u^{l,pre}(t) = \tau u^l(t-1) + W^l s^{l-1}(t), \qquad (11)$$

where τ is the leakage factor and it balances the contribution of membrane potential after spiking at previous time step, $u^{l,pre}(t)$. The neuron generates spike output $s^l(t)$, when the accumulated membrane potential $u^{l,pre}(t)$ exceeds the pre-defined firing threshold V_{th}, which we set it to 1:

$$s^l(t) = \Theta(u^{l,pre}(t) - V_{th}) = \begin{cases} 1, & if \ u^{l,pre}(t) \geq V_{th} \\ 0, & otherwise \end{cases}, \qquad (12)$$

$$u^l(t) = u^{l,pre}(t)(1 - s^l(t)), \qquad (13)$$

where Θ represents the Heaviside step function, and $u^l(t)$ is reset to zero (hard reset) if $u^{l,pre}(t)$ exceed the threshold V_{th}. In SNNs, this process is applied iteratively. For the gradient descent training via back-propagation, the gradient calculation with the cross-entropy loss L under the STBP rule can be expressed as:

$$\frac{\partial L}{\partial W^l} = \sum_t \frac{\partial L}{\partial s^l(t)} \frac{\partial s^l(t)}{\partial u^{l,pre}(t)} \frac{\partial u^{l,pre}(t)}{\partial W^l}, \qquad (14)$$

where $\frac{\partial s^l(t)}{\partial u^{l,pre}(t)}$ is non-differentiable, we use a surrogate gradient as an approximate derivative of the spike activity. In this paper, we use a triangular function as the surrogate gradient [15]:

$$\frac{\partial s^l(t)}{\partial u^{l,pre}(t)} = \frac{1}{\gamma^2} \max\left(0, \gamma - |u^{l,pre}(t) - V_{th}|\right). \qquad (15)$$

where γ is a hyper-parameter, which we set to 1.

A.3. Energy estimation

For the estimation of energy consumption, we follow the formulation proposed by Kundu et al. [31], based on the direct input coding scheme. The total inference energy of an L-layer SNN is computed as

$$\mathcal{E}_{SNN}^{direct} = \mathcal{F}_1 \cdot E_{MAC} + \sum_{l=2}^{L} \mathcal{F}_l \cdot E_{AC}, \qquad (16)$$

where $\mathcal{F}_l$ denotes the number of operations in layer l, and E_{MAC} and E_{AC} represent the energy cost of multiply-accumulate and accumulate operations, respectively. The first layer operates on real-valued inputs and thus incurs MAC operations, while all subsequent layers process binary spikes and use AC operations.

The operation count $\mathcal{F}_l$ for a convolutional layer is calculated as

$$\mathcal{F}_l = (k^l)^2 \cdot H^l \cdot W^l \cdot C_{in}^l \cdot C_{out}^l \cdot \zeta^l, \qquad (17)$$

where k^l is the kernel size, H^l and W^l are the height and width of the output feature map, C_{in}^l and C_{out}^l denote the number of input and output channels, and $\zeta^l \in [0, 1]$ is the average spiking activity at layer l. This activity reflects the temporal sparsity in SNNs and scales the effective number of computations accordingly.

Following [31], we assume an energy cost of $E_{MAC} = 4.6 \, pJ$ for MAC operations and $E_{AC} = 0.9 \, pJ$ for AC operations under 32-bit floating point (FP) precision.

B. Additional Results

B.1. Results on CIFAR-10-DVS

To validate the effectiveness of SpikeMatch beyond normal datasets, we conduct additional experiments on CIFAR10-DVS using event-based data. For strong augmentation,

Dataset	CIFAR-10	CIFAR-100	STL-10	ImageNet
Model	VGG9	VGG11	VGG11 SEW-ResNet-34	SEW-ResNet-50
Weight decay	5×10^{-4}			3×10^{-4}
Iterations	2^{18}			2^{20}
Batch size	32			16
label unlabel ratio (μ)	7			1
Learning rate	0.03			
SGD momentum	0.9			
EMA decay	0.999			
Weak Augmentation	Random Crop, Random Horizontal Flip			
Strong Augmentation	RandAugment [13]			

Table A.1. **Hyperparameter setting for semi-supervised learning.**

	DVS-CIFAR10
SoftMatch	36.40
FreeMatch	37.40
SpikeMatch	**49.80**

Table A.2. **Accuracy (%) on DVS-CIFAR10 with 1% labels.**

we adopt EventAugment [24], which avoids image-specific transformations (e.g., brightness or color jitter) that are unsuitable for DVS data, while preserving the temporal structure of events. As shown in Table A.2, SpikeMatch outperforms SoftMatch [9] and FreeMatch [48] even on this neuromorphic dataset, supporting that our multi-view agreement remains effective when the temporal diversity arises from the data itself.

C. Extended Analysis

C.1. Network configuration

By default, we use the VGG network [15, 42] in our experiments, as it is commonly used in other SNN methods. In contrast, SSL methods in ANNs typically use Wide ResNet (WRN) [51] as the backbone. To demonstrate that our method performs well across different network architectures, we also conduct experiments using WRN on CIFAR-10 and CIFAR-100, utilizing 400 and 100 labeled samples per class, respectively. As reported in Table A.3, our method achieves the highest performance on WRN for both CIFAR-10 and CIFAR-100. These results demonstrate that our agreement-based pseudo-labeling method, which leverages the temporal dynamics of SNNs, is effective for semi-supervised learning in SNNs across various network architectures.

Dataset	Network	Method	**Accuracy**
CIFAR-10	WRN-28-2	FixMatch	87.98
		AdaMatch	88.20
		FreeMatch	88.80
		SoftMatch	89.19
		Ours	**90.04**
CIFAR-100	WRN-28-8	FixMatch	55.22
		AdaMatch	56.83
		FreeMatch	57.62
		SoftMatch	57.38
		Ours	**60.84**

Table A.3. **Accuracy (%) using WRN with 4 time steps on CIFAR-10 using 400 labels per class and CIFAR-100 using 100 labels per class.**

Seed	0	1	2	Mean
SoftMatch	83.65	83.63	83.55	$83.61_{\pm 0.05}$
SpikeMatch	83.76	85.35	85.55	$\mathbf{84.88}_{\pm 0.98}$

Table A.4. **Accuracy (%) on STL-10 with 250 labeled samples.**

C.2. Results on STL-10

To demonstrate the backbone-agnostic effectiveness of our method, we evaluate it on STL-10, a more complex dataset with high-resolution images, using 250 labeled samples. We conduct experiments with SEW-ResNet [21], a commonly used residual connection architecture for deep SNNs.

Reset&SG	Method	Accuracy
	UDA [50]	74.26
	FixMatch [45]	57.62
	FlexMatch [52]	66.94
Soft	AdaMatch [6]	88.58
&Rectangular	FreeMatch [48]	89.49
	SoftMatch [9]	87.49
	SequenceMatch [37]	76.88
	SpikeMatch	**89.73**

Table A.5. **Accuracy (%) with soft reset and rectangular surrogate gradient function (SG) on CIFAR-10 using 4 labels per class with a time step 8.**

	$T = 4$ $(M = 3)$	$T = 4$ $(M = 2)$	$T = 3$ $(M = 3)$	$T = 2$ $(M = 2)$	$T = 1$
Acc.(%)	**88.13**	85.97	87.85	84.03	64.38

Table A.6. **Performance across time steps (T) and groups (M).**

For the SEW-ResNet backbone, we use 4 time steps, a triangular function as the surrogate gradient, and a hard reset mechanism after spiking, implemented by modifying the code provided by the authors. As shown in Table A.4, our method achieves higher accuracy compared to the recent SSL method, SoftMatch [9]. Even with a deeper network, our method achieves higher accuracy across all seed settings by effectively utilizing the temporal dynamics of SNNs for generalizable and reliable SSL training.

C.3. LIF configuration

We also conducted ablation experiments on different configurations of the Leaky Integrate-and-Fire (LIF) model. By default, we used a hard reset mechanism with a triangular function as the surrogate gradient to handle the non-differentiability in direct training of SNNs, and we set the number of time steps $T = 4$. In the ablation study, we employed a soft reset mechanism and a rectangular function [49] as the surrogate gradient, increasing the time steps T to 8. With $T = 8$, we partitioned the outputs into collections of sizes $\{3, 2, 3\}$.

As shown in Table A.5, by increasing the time steps and using a soft reset, which is known for higher information preservation, we achieved higher accuracy on CIFAR-10 with 4 labeled examples per class. These results demonstrate that our method can be extended to longer time steps and various network configurations in SNNs while still outperforming other SSL methods in ANNs. As shown on Table A.6, while longer time steps contribute to better feature learning, even with the same T, treating temporally grouped predictions as distinct views in our agreement-based frame-

	$n = 3$	$n = 4$	$n = 5$
Acc.(%)	**88.13**	87.15	86.95

Table A.7. **Ablation study on number of RandAugment Transforms.**

	$\lambda = 0.5$	$\lambda = 1.$	$\lambda = 2.$	$\lambda = 4.$
Acc.(%)	87.04	**88.13**	88.04	87.71

Table A.8. **Ablation study on unlabeled loss weight**

work yields more substantial performance gains. These results demonstrate that temporal dynamics are fundamental to learning discriminative features in SNNs, and that leveraging them as diverse views through agreement is especially beneficial when labeled data is limited. The noticeable performance drop when the time step is reduced to one highlights that, in the context of semi-supervised learning with scarce labels, exploiting temporal diversity is particularly important for enabling reliable pseudo-labeling.

C.4. Sensitivity analysis

In our experiments, we use random cropping and random horizontal flipping as weak augmentations, and adopt RandAugment [13] as a strong augmentation. We conduct ablation studies on both the augmentation strategy and the loss weight λ, which balances the supervised and unsupervised objectives. As shown in Table A.7, varying the number of RandAugment transformations exhibits trends consistent with ANN-based SSL methods such as UDA [50] and FixMatch [45], leading us to set it to 1 by default. Regarding λ, we observe that performance remains stable across a wide range, confirming that our default choice of assigning equal weight to the supervised and unsupervised losses is near-optimal, as shown in Table A.8.

C.5. Convergence speed

We compare the convergence speed of SpikeMatch with SoftMatch [9] and FreeMatch [48] on CIFAR-10 with 4 labeled samples per class. As shown in Figure A.1, SpikeMatch requires fewer iterations to converge while achieving higher accuracy by leveraging temporal efficient training [15] and effectively utilizing abundant unlabeled data with diverse pseudo-labels.

C.6. Loss landscape

In SSL, labeled data are typically much smaller in quantity than unlabeled data, resulting in poor generalization during labeled data training [26]. To address this, we utilize Temporal Efficient Training (TET) loss [15] to enhance generalization without requiring additional training to find flat-
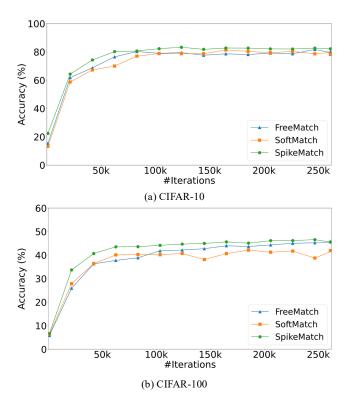
(a) CIFAR-10

(b) CIFAR-100

Figure A.1. **Convergence speed of proposed method with SoftMatch [9] and FreeMatch [48] on (a) CIFAR-10 and (b) CIFAR-100 with 4 labeled samples per class.**
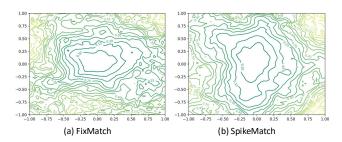

(a) FixMatch (b) SpikeMatch

Figure A.2. **Loss landscape of (a) FixMatch and (b) Spike-Match of labeled data trained on CIFAR-10 with 250 labeled samples.**

ter minima in the context of limited labeled samples. As shown in Figure A.2, SpikeMatch using TET loss achieves a smoother loss landscape compared to FixMatch [45], which uses averaged temporal outputs as prediction. This suggests that the low generalization issue associated with small labeled datasets in SSL can be moderately alleviated by leveraging the inherent characteristics of SNNs without relying on pre-training methods.

C.7. Pseudo-label utilization ratio

The utilization ratio indicates the number of pseudo-labels used for training: for SpikeMatch, those in agreement; for
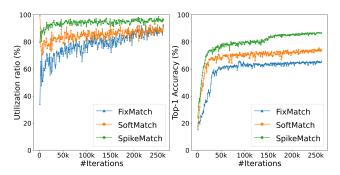

Figure A.3. **Utilization graphs (Left) comparing SpikeMatch, FixMatch, and SoftMatch on CIFAR-10 with 4 labeled samples per class and accuracy graph (Right).**
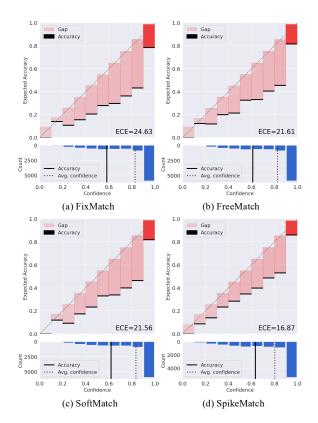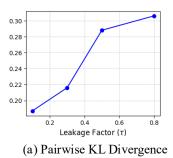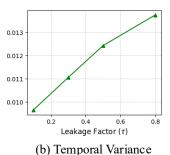

(a) FixMatch (b) FreeMatch

(c) SoftMatch (d) SpikeMatch

Figure A.4. **Reliability diagram (top) and confidence histogram (bottom).** (a) FixMatch, (b) FreeMatch, (c) SoftMatch, and (d) SpikeMatch are trained on CIFAR-100 with 2500 labeled samples and tested using 10 interval bins.

FixMatch and SoftMatch, those surpassing the confidence threshold. As shown in Figure A.3, with our agreement method, unreliable predictions are filtered out in early training, effectively reducing confirmation bias. The utilization ratio increases steadily throughout training, indicating both the stability and high quantity of reliable pseudo-labels.
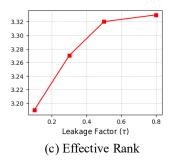
(a) Pairwise KL Divergence   (b) Temporal Variance   (c) Effective Rank

Figure A.5. **(a) Pairwise KL divergence, (b) Temporal Variance, and (c) Effective Rank at varying leakage factors (τ) on CIFAR-10 with 4 time steps.**

C.8. Expected Calibration Error (ECE)

Model calibration is crucial in semi-supervised learning (SSL) to mitigate confirmation bias [10, 36–38]. To demonstrate the effectiveness of SpikeMatch, we compare the calibration of FixMatch [45], FreeMatch [48], and Soft-Match [9] with our approach using Expected Calibration Error (ECE), confidence histograms, and reliability diagrams to visualize calibration error. In this work, we compute ECE as the expected difference between the accuracy and confidence across 10 bins for all samples. A high ECE value indicates that the model is over-confident, which can lead to unreliable pseudo-labels and confirmation bias during training. Lower ECE values indicate better calibration.

As shown in Figure A.4, SpikeMatch achieves the best calibration among all methods. Its reliability diagram aligns most closely with the ideal diagonal, and its confidence histogram indicates reduced over-confidence. The red bars show the gap between confidence and accuracy in each bin, demonstrating its effectiveness in addressing confirmation bias.

D. Empirical Insights

D.1. Leakage factor and temporal diversity

To support our claim that a higher leakage factor τ enhances temporal diversity in spiking neural networks, we conduct a supervised analysis using multiple diversity metrics. While cosine similarity is used in the main paper to measure differences between temporal predictions, we extend the analysis by incorporating additional metrics that capture different aspects of diversity. We use pairwise KL divergence to evaluate the dissimilarity of predictions over time, temporal variance to measure the dispersion of features, and effective rank to quantify the dimensional richness of temporal representations.

We vary τ from 0.1 to 0.8 and evaluate these metrics on the CIFAR-10 dataset. As shown in Figure A.5, all three metrics increase consistently with larger τ. In (a), KL divergence grows as predictions across time steps become

more distinct. In (b), temporal variance increases, indicating that features are more widely distributed. In (c), effective rank rises, showing that the temporal features span a higher-dimensional space with reduced redundancy.

These results provide strong empirical evidence from multiple perspectives that a higher leakage factor naturally induces temporal diversity. This property plays an important role in the co-training-inspired design of SpikeMatch.

E. Visualization on CIFAR-10 and STL-10.

As additional examples, Figure A.6 and Figure A.7 illustrate how our method captures multiple discriminative features by leveraging the inherent temporal diversity of SNNs. Specifically, our agreement-based temporal dynamic framework enhances class distinction by extracting diverse and complementary features across time steps. These qualitative results further support the conclusion that our approach remains effective even under severe label constraints.
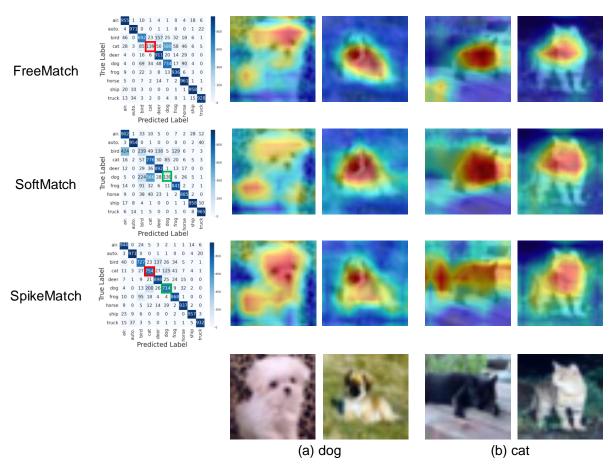
Figure A.6. **Confusion matrix and spiking attention map (SAM) [28] on CIFAR-10.** Unlike FreeMatch (**first row**) and SoftMatch (**second row**), which struggle with the cat and dog classes, our method (**last row**) captures more distinctive features of (a) dog and (b) cat. This leads to a more diagonal confusion matrix, reducing confusion among similar classes.

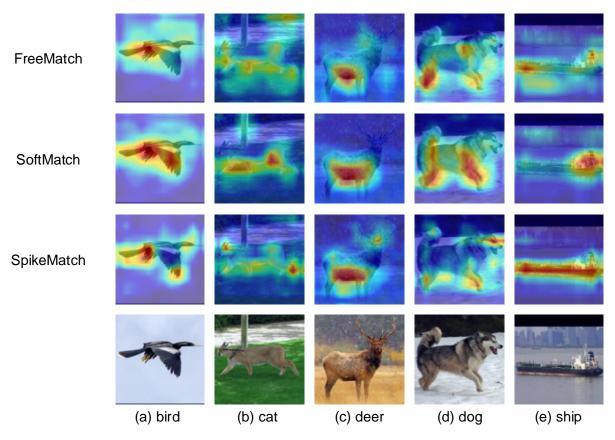| | (a) bird | (b) cat | (c) deer | (d) dog | (e) ship |

Figure A.7. **Spiking attention map (SAM) [28] on STL-10.** Unlike FreeMatch (**first row**) and SoftMatch (**second row**), our method (**last row**) more effectively captures multiple discriminative features through the agreement-based approach.