Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.039600
Article

Attentive Neighborhood Feature Augmentation for Semi-supervised Learning

Qi Liu[1,2], Jing Li[1,2,*], Xianmin Wang[1,*] and Wenpeng Zhao[1]

[1]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, 510002, China
[2]Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou, 350121, China
*Corresponding Authors: Jing Li. Email: lijing@gzhu.edu.cn; Xianmin Wang. Email: xianmin@gzhu.edu.cn
Received: 07 February 2023; Accepted: 14 April 2023; Published: 23 June 2023

Abstract: Recent state-of-the-art semi-supervised learning (SSL) methods usually use data augmentations as core components. Such methods, however, are limited to simple transformations such as the augmentations under the instance's naive representations or the augmentations under the instance's semantic representations. To tackle this problem, we offer a unique insight into data augmentations and propose a novel data-augmentation-based semi-supervised learning method, called Attentive Neighborhood Feature Augmentation (ANFA). The motivation of our method lies in the observation that the relationship between the given feature and its neighborhood may contribute to constructing more reliable transformations for the data, and further facilitating the classifier to distinguish the ambiguous features from the low-dense regions. Specially, we first project the labeled and unlabeled data points into an embedding space and then construct a neighbor graph that serves as a similarity measure based on the similar representations in the embedding space. Then, we employ an attention mechanism to transform the target features into augmented ones based on the neighbor graph. Finally, we formulate a novel semi-supervised loss by encouraging the predictions of the interpolations of augmented features to be consistent with the corresponding interpolations of the predictions of the target features. We carried out experiments on SVHN and CIFAR-10 benchmark datasets and the experimental results demonstrate that our method outperforms the state-of-the-art methods when the number of labeled examples is limited.

Keywords: Semi-supervised learning; attention mechanism; feature augmentation; consistency regularization

1 Introduction

Deep neural networks have achieved favorable performance on a wide variety of tasks [1–5]. Training deep neural networks commonly requires a large amount of labeled training data. However, since collecting labeled data necessarily involves expert knowledge, labeled data is usually unavailable for many learning tasks. To address this problem, numerous semi-supervised learning (SSL) methods have been developed, which exploit abundant unlabeled data effectively to improve the performance of deep models and relieve the pressure brought by the lack of labeled data.

Existing SSL methods are mainly based on a low-density separation assumption, that is, the decision boundary learned by the model is supported to lie in low-density regions of the instances. Consistency regularization is a typical measure to implement the low-density separation assumption, which has been widely used on many benchmarks. The main idea of consistency regularization is to enforce the model to produce the same output distribution for an input instance and its perturbed version. The conventional consistency-regularization-based methods mainly focus on how to construct effective perturbations. For instance, Laine et al. [6] generated different perturbations by two network models to make them predict agreement. Miyato et al. [7] produced the worst perturbations according to the adversarial direction when adversarial training [8,9], and then enforced the outputs from the original example and its perturbed version to be consistent.

Recently, data augmentation has quickly turned into the mainstream technique of consistency regularization in SSL due to its powerful capability of expanding the examples' feature representations. The essence of data augmentation is to expand the feature representations from the given training dataset. To this end, numerous data-augmentation-based SSL methods are developed. For instance, Verma et al. [10] proposed an interpolation consistency training (ICT) algorithm to train deep neural networks in the semi-supervised learning paradigm. This algorithm enforced the prediction at an interpolation of unlabeled points to be consistent with the interpolation of the predictions at those points. Xie et al. [11] presented a new perspective on how to effectively noise unlabeled examples. This work verifies that the quality of noising produced by advanced data augmentation methods is very important for semi-supervised learning. Berthelot et al. [12] presented a MixMatch approach to guess the low-entropy labels for data-augmented unlabeled examples and mixes labeled and unlabeled data using the MixUp strategy. Sohn et al. [13] presented the FixMatch algorithm to simplify existing SSL approaches. The model's predictions on weakly-augmented unlabeled pictures are used to construct pseudo-labels, which are then trained to predict the pseudo-label when fed a strongly-augmented version of the same image.

The aforementioned methods commonly generate augmented instances on their naive representations, which are unable to derive abstract semantic representations for the learning of semi-supervised models. Accordingly, inspired by the idea of feature fusion [14–18], several works focus on augmenting data by merging the feature representations of the instances from the semantic layer. Verma et al. [19] proposed a manifold mixup method to encourage neural networks to predict less confidently on interpolations of hidden representations. This method leveraged semantic interpolations as an additional training signal, obtaining neural networks with smoother decision boundaries at multiple levels of representation. Upchurch et al. [20] proposed a deep feature interpolation (DFI) method for automatic high-resolution image transformation. DFI can be used as a new baseline to evaluate more complex algorithms and provides a practical answer to the question of which image transformation tasks are still challenging after the advent of deep learning. Kuo et al. [21] proposed a novel learned feature-based refinement and augmentation method which produces a varied set of complex transformations. The transformations, combined with traditional image-based augmentation, can be used as part of the consistency-based regularization loss.

The existing feature fusion methods boost the capability of the feature representations to some extent [22]. However, they only consider the information of a single given example when merging the features. The overlooking of the relationship between the given feature and its neighborhood may lead to false predictions for the unlabeled examples and further limit the performance of SSL. To clarify this phenomenon and put forward our motivation, we take a simple example as shown in Fig. 1. As shown in Fig. 1a, we can see that in low-density feature embedding regions, it is difficult for the classifier to distinguish the ambiguous unlabeled features. Here, the ambiguous unlabeled

features are the features that are derived from the unlabeled examples and have approximately identical margins to the boundaries in the embedding spaces. The ambiguous unlabeled features have similar representations, thus they may yield similar outputs from the SSL model and then generate unreliable pseudo labels for the unlabeled examples. This fact leads to false decision boundaries during the training process. Whereas, as shown in Fig. 1b, if the neighborhood of the given feature is considered, the representation of the feature can be strengthened and refined. Based on the cluster characteristics of the neighborhood, the ambiguous unlabeled features have discriminative representations, which contributes to yielding more reliable pseudo labels for the training. Therefore, it is reasonable to generate diverse and abstract transformations by exploiting the neighborhood information of examples on their semantic feature spaces. To this end, we use the self-attention [23] mechanism to aggregate the neighbor features, and then apply a neighbor graph to refine and augment the target features. By creating such a neighborhood graph, it is possible to obtain more discriminative feature representations, which help to produce more trustworthy decision boundaries for the SSL model and more private pseudo labels (as shown in Fig. 1c).

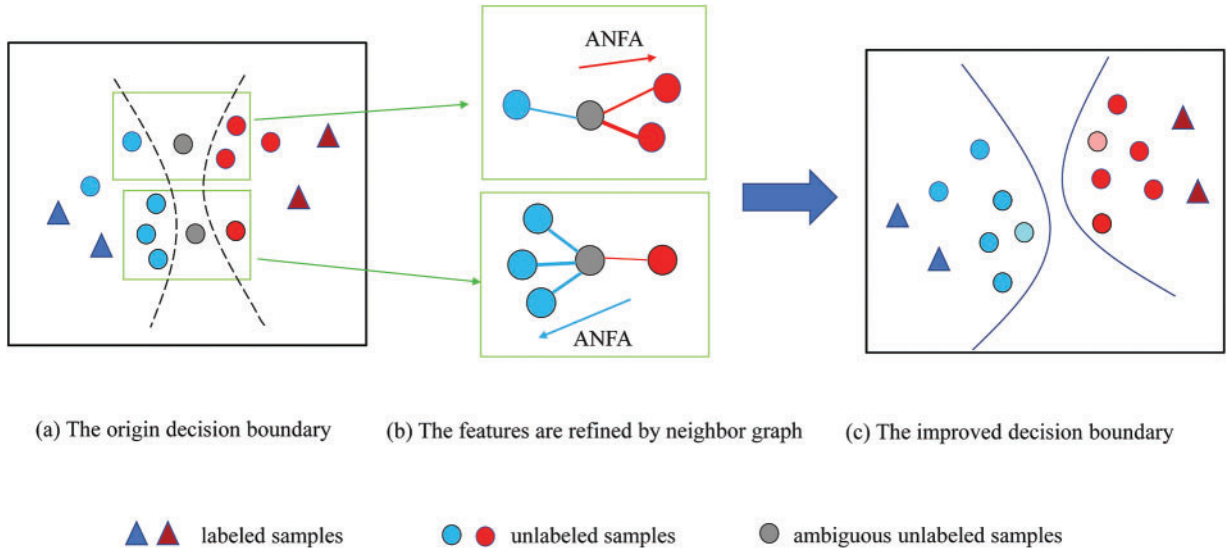

Figure 1: A simple case to clarity the motivation of our method. The blue and red circles represent represents unlabeled samples that have been divided into different clusters by the SSL model. Triangles represent labeled samples. The gray circles represent feature representations that are difficult to be discriminated by the classifier. (a) illustrates that the ambiguous unlabeled samples are difficult to be classified by their representations. (b) indicates the use of ANFA to aggregate neighboring samples, the thicker the line, the greater the attention weight. (c) shows that the SSL model can correctly classify unlabeled samples with refined features

According to the foregoing analysis, this paper proposes a novel feature augmentation framework called Attentive Neighborhood Feature Augmentation (ANFA) for SSL. First, given labeled and unlabeled data examples, we project them to an embedding space and construct a neighborhood graph based on the similarity of representations on their embedding spaces. Second, we refined the features via weighting the neighbor representations of the target features, where the weights are adaptively acquired relying upon the similarity between the target features and the neighborhood graph. Finally, we mix up the target and refined features to obtain the interpolated features and then propose a novel consistency regularization loss that encourages the predictions of the interpolated features to

be consistent with their corresponding interpolated pseudo-labels. Moreover, we test our method on standard SSL datasets such as SVHN [24] and CIFAR-10 [25] and neural network architectures CNN-13 and WRN28-2 [26], and the experimental results demonstrate that our approach outperforms the baseline methods.

This paper is organized as follows. First, we survey the related work and analyze their advantage and disadvantage in Section 2. Then, we elaborate the proposed method in Section 3. Next, we conduct experiments and analyze the results in Section 5.

2 Related Work

In the past, many semi-supervised deep learning methods have been developed. In this section, we focus on some related works, including the consistency regularization methods, augmentation methods, and the attention scheme.

2.1 Consistency Regularization Methods

Current state-of-the-art SSL methods mostly use this technique. The key idea of consistent regularization methods is that the model should be robust to local perturbations in the input space, which requires the deep neural network to be consistent with the original samples and the prediction results after adding small perturbations. In image classification tasks, the approach is to make the model's predictions invariant to texture or geometric changes in the image.

Different consistency regularization techniques differ in how they choose perturbations δ. One simple alternative is to use random perturbations δ, which is to add Gaussian noise to the image. However, random perturbation is inefficient in high dimensions because only a small fraction of the input perturbation can push the decision boundary to low-density regions. To alleviate this problem, Virtual Adversarial Training [7] searches for adversarial perturbation directions that maximize the change in model predictions. This involves computing the gradient of the classifier input [27–29], which can be very expensive for large neural network models. In addition to adding perturbations to the image, we can also add perturbations to the model. Laine et al. simply implemented this approach by training two perturbed neural network models. They used dropout [30] to randomly drop a part of the network parameters as a perturbation process. In supervised learning, de et al. proposed the Mixup [31] method, which encourages the model's prediction of a linear combination of two samples to be close to the linear combination of their labels, and interpolates and obtains different samples between the two samples to enhance the generalization ability of the model. Verma et al. [10] proposed interpolation consistency training (ICT) to introduce Mixup into semi-supervised learning by using pseudo-labels of unlabeled data. ICT encourages predictions on interpolated sample pairs to be consistent with their interpolated predictions. Wei et al. [32] proposed FMCmatch to further develop the method of sample mixing enhancement and improved the Cutout and Mixup methods to generate samples to effectively smooth the training space. However, simply cutting and mixing in the image space may produce meaningless samples. Introducing Noise, which makes the image out of the low-dimensional manifold in the high-dimensional embedding space. Chen et al. [33] proposed attention-based label consistency regularization, which uses channel and sample attention to describe the similarity of different samples, maintaining label consistency across samples and enhancing the smoothness of label prediction between data. However, this approach is limited to the similarity of samples in the same batch and cannot describe the similarity in global samples.

Recently, a series of methods that combine consistent regularization techniques with other semi-supervised learning methods have achieved the best performance, such as MixMatch [12],

ReMixMatch [34], and FixMacth [13], using strong data augmentation to create perturbations, while also using pseudo-labels, entropy Minimization, sharpening, and other techniques improve the confidence of the model. At the same time, several works have improved some graph-based methods to better extract intrinsic features from raw data. Yang et al. [35] used self-paced regularization to better factorize matrices and introduced adaptive graphs using dynamic neighbor assignment to find low-dimensional manifolds. Chen et al. [36] improved the Graph non-negative matrix factorization (GNMF) method, introduced the l_0 norm to enhance the sparsity of factorized matrices and improved the robustness of feature extraction using GNMF.

We summarize the advantages and disadvantages of some consistency regularization methods shown in Table 1.

Table 1: Key findings and limitations of some typical consistency regularization methods

Method	Key findings	Limitations
TE [1]	Better prediction by ensembling the outputs of the network in previous epochs.	Expensive calculation in a huge dataset.
VAT [7]	Better generalization by learning adversarial perturbations.	Additional backpropagation to compute the adversarial direction.
ICT [10]	Reduce overfitting to labeled points under high confidence.	Random interpolation may generate unreal samples leading to prediction bias.
MixMatch [12]	Unifying the dominant approaches of semi-supervised learning.	Multiple forward and back propagation calculations.
FeatMatch [21]	Better feature learning by exploiting category information.	Neighborhood information is ignored during feature learning.
FMCmatch [32]	Smoothed the training space using more diverse image transformations.	Random Cutout and Mixup introduce noise.
ALC [33]	Smoothed label predictions across data using channel and sample attention.	Similarity measurement limited to batch samples.

2.2 Data Augmentation

For SSL with the deep model, most recent works incorporate different data augmentation methods into their baseline models to achieve higher performance. Data augmentation alleviates the problem of limited data by performing diverse but reasonable transformations on the data and has been widely used in the training of deep models [37]. Data augmentation increases data diversity and prevents overfitting in the training of deep models. Simple data augmentation methods include random flips, blurs, transitions, geometric transformations, changing the contrast and color of images, and so on. In addition, complex augmentation operations also exist. Mixup enforces interpolation smoothness between every two training samples by generating new training samples through a convex combination of two images and their corresponding labels. It has been shown that models trained with Mixup are robust to out-of-distribution data and facilitate the uncertainty calibration of the network. In recent years, an SSL data augmentation strategy for strong image processing has attracted attention.

In image classification, unsupervised data Augmentation (UDA) [11] uses AutoAugment [38], which uses reinforcement learning [39,40] to search for the best combination of different image augmentation operations based on the confidence of a validated model. In addition, the CTAugment proposed by Remixmatch [34] and the RandAugment [41] used in Fixmatch [13] use different strategies to maximize the effect of data enhancement.

We summarize the key findings and limitations of some data augmentation methods shown in Table 2.

Table 2: Key findings and limitations of some typical data augmentation methods

Method	Key findings	Limitations
Mixup [31]	A linear combination between two samples and their corresponding labels can improve generalization.	Simple interpolation may produce meaningless samples.
AutoAugment [38]	Automatically search for the best data augmentation policy.	Using reinforcement learning as a search algorithm requires additional training.
CTAugment [34]	Using control theory to dynamically infer the magnitude of the transition during training.	Dynamic updates require additional computational cost
RandAugment [41]	Only two augmentation parameters are needed the number and magnitude of augmentation transformations.	For different data sets, two augmentation parameters still need to be determined, which still has a large experimental cost.

2.3 Attention

Vaswani et al. [23] define scaled dot-product attention as an operation that maps a query and a set of key-value pairs to an output that computes a dot product of the query and key and scales it, using a softmax function for normalization and computing attention weights. It can be expressed as follows:

$$\text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where d_k denote the dimension of keys. Attention mechanisms can pay more attention to the characteristics of more attention to task correlation between input information, reduce the attention to irrelevant characteristics, and even filter out irrelevant features, thereby improving the efficiency and accuracy of task processing.

In recent years, attention mechanisms have been successfully applied to various computer vision tasks [42,43]. SENet [44] obtains the weight of each channel of the input feature layer and uses its weight to make the network focus on more important information [45]. Residual attention networks [46] are built by stacking attention modules that generate attention-aware features. As the modules go deeper, the attention-aware functions from different modules change adaptively. CBAM [47]

sequentially infers the attention map along two independent dimensions of channel and space and then multiplies the attention map with the input feature map for adaptive feature refinement [48].

We summarize the key findings and limitations of some attention methods shown in Table 3.

Table 3: Key findings and limitations of some attention methods

Method	Key findings	Limitations
SENet [44]	Automatically obtain the importance of each channel.	Ignoring the importance of spatial information.
Residual attention networks [46]	Multiple attention modules can be stacked.	Can only effectively capture local information, but cannot establish remote channel dependencies.
CBAM [47]	Simultaneously calculate the attention map of the two dimensions of channel and space.	Only consider the calculation of the local area, ignoring the information of other global areas.

3 Methodology

In this section, we present our work for semi-supervised deep learning. A glimpse of our approach is shown in Fig. 2. Our approach consists of three parts, namely neighbor graph representation, feature augmentation, and consistency regularization. We first construct a neighborhood feature graph that represents the relationship between the target feature and its neighbors. Then, based on the neighborhood feature graph, we augment the features by attention mechanism. Finally, we propose a new loss that encourages the prediction at an interpolation of features to be consistent with the interpolation of the predictions at those features.

3.1 Preliminary

In SSL tasks, we given a labeled dataset $\mathrm{D_l} = \{\mathrm{x_i}, \mathrm{y_i}\}_{\mathrm{i=1}}^{\mathrm{L}}$ and a unlabel dataset $D_u = \{x_i\}_{i=1}^{U}$, where L and U denote the number of $\mathrm{D_l}$ and $\mathrm{D_u}$. Formally, a feature extractor $f_\theta(\cdot)$ with parameter θ is used to extract input image features $\mathrm{f_i} = \mathrm{f_\theta}(\mathrm{x_i})$, a classifier h_ϕ, and the memory bank $\mathcal{M} = \left\{\mathrm{f_i}, \hat{\mathrm{y}}_\mathrm{i}\right\}_{\mathrm{i=1}}^{k}$, where $\mathrm{f_i}$ is the extract features of input sample x_i, for labeled data, $\hat{y}_i$ is the ground-truth label, while for the unlabeled, $\hat{y}_i$ is pseudo-label, and k is the size of $\mathcal{M}$.

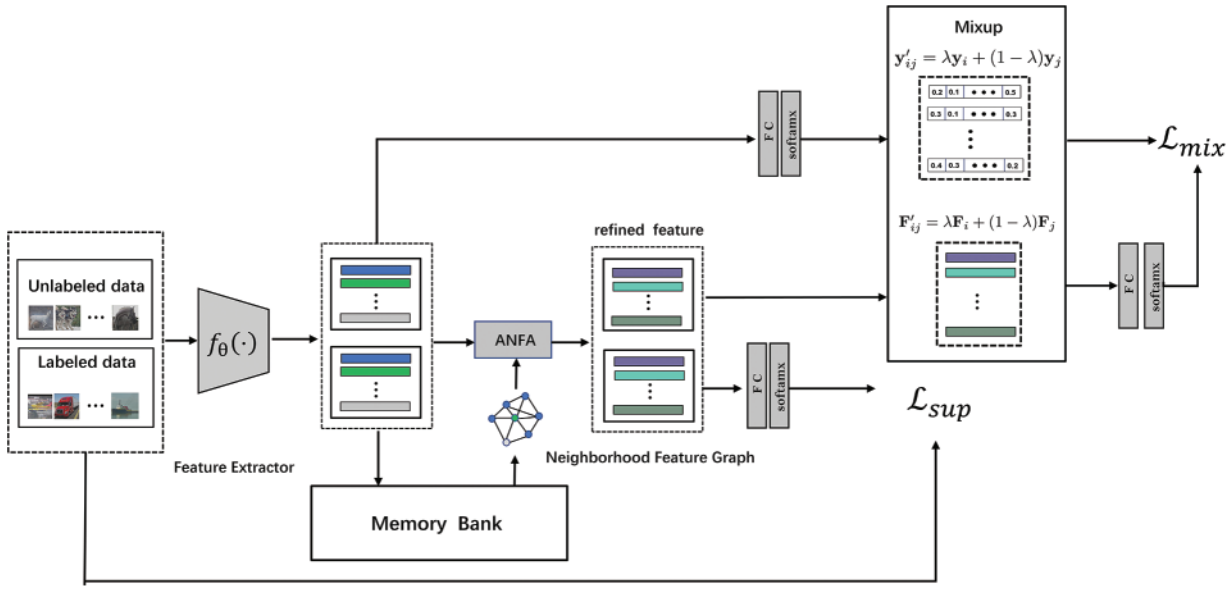

Figure 2: The pipeline of attentive neighborhood feature augmentation for semi-supervised learning

3.2 Neighborhood Feature Graph

In order to efficiently leverage the knowledge of neighbors for regularization, we propose to construct a graph among the samples and their neighborhoods in the feature semantic space. To select suitable neighbors from the dataset, we propose to use k-nearest neighbor representation in the feature space to extract neighbors for each sample.

we first extract the feature $f_\theta(x_i)$ and label predictions $\widehat{y}_i$ for unlabeled sample x_i at each iteration of the training loop, and collected and recorded them in a memory bank $\mathcal{M}$ as $(f_\theta(x_i), \widehat{y}_i)$ pairs. We first pre-train a feature extractor, and then use the extracted features and pseudo-labels to initialize the memory bank. During each forward pass in the training loop, we separate the features and pseudo-labels and push them into the memory bank. Since the training of the model will influence the extracted features, we update the features corresponding to the current training sample after each iteration. To gain a more accurate prediction, we use target prediction generated by the teacher model [49]. Based on the features in the memory bank, we calculate the cosine similarity between the features and construct a similarity matrix $\mathbf{S} \in R^{n \times n}$ with

$$S_{ij} = \frac{f_\theta(x_i)^T f_\theta(x_j)}{|f_\theta(x_i)|_2 |f_\theta(x_j)|_2} \tag{2}$$

where $S_{ij} \in S$ is a measurement of the similarity between the samples x_i and x_j. n is the number of training samples. Compared with other similarity metrics, such as Euclidean distance, we find that cosine similarity has a better performance. Higher similarity indicates the two samples are closer in the feature space, so we choose the k samples with the highest similarity as neighbors for an input sample and construct a neighborhood graph $\mathcal{N} \in \mathbf{R}^{n \times n}$ as follows:

$$\mathcal{N}_{ij} = \begin{cases} S_{ij} & S_{ij} \geq S_i^k \\ 0 & S_{ij} < S_i^k \end{cases} \tag{3}$$

where $\mathcal{N}_{ij}$ is the weight of node i (sample x_i) and j (sample x_j). S_i^k is denoted k-th value in the i-th row of S where the values of elements in i-th row of S are ranked in ascending order from small to large. The embedding of a sample can take advantage of a neighborhood graph to exploit more abundant information. When we go over the whole dataset, we use the features saved in the memory bank to calculate the global similarity matrix and build a neighbor graph through the k-nearest neighbor algorithm.

3.3 Feature Augmentation

With a neighborhood feature graph built by the process described above, we propose a learned feature augmentation module via self-attention to improve target feature embedding by aggregating the neighborhood features. The proposed module refines input image features in the feature space by leveraging important neighborhood information.

Formally, Given a neighborhood feature graph $\mathcal{N}$, for an input sample with extracted feature f_x and the i-th neighbor feature $f_{x,i}$. we linearly project them into an embedding space as:

$$p_x = \phi_a\left(f_x; w_a\right), \quad p_{x,i} = \phi_b\left(f_{x,i}; w_b\right) \tag{4}$$

where w_a and w_b are the learned parameters of FC layer ϕ_a and ϕ_b, respectively. We define the attention function using a softmax function as:

$$w\left(p_x, p_{n,i}\right) = \frac{\exp\left(p_x^T p_{x,i}\right)}{\sum_{i=1}^{k} \exp\left(p_x^T p_{x,i}\right)} \tag{5}$$

In detail, we first compute the dot product similarity between p_x and $p_{x,i}$, and get the final attention weights by normalizing the similarity with the softmax operation. Then, we aggregate neighborhood information for input sample feature augmentation can be denoted as:

$$F_x = p_x + \psi_t\left(\sum_{i=1}^{k} w\left(p_x, p_{x,i}\right) p_{x,i}\right) \tag{6}$$

where ψ_t is a non-linear transformation. In this work, ψ_t is implemented by a Multi-Layer Perceptrons (MLP) layer, this layer contains two-layer with ReLU, i.e., FC-ReLU-FC-ReLU. Fig. 3 shows the detailed architecture of the proposed module.

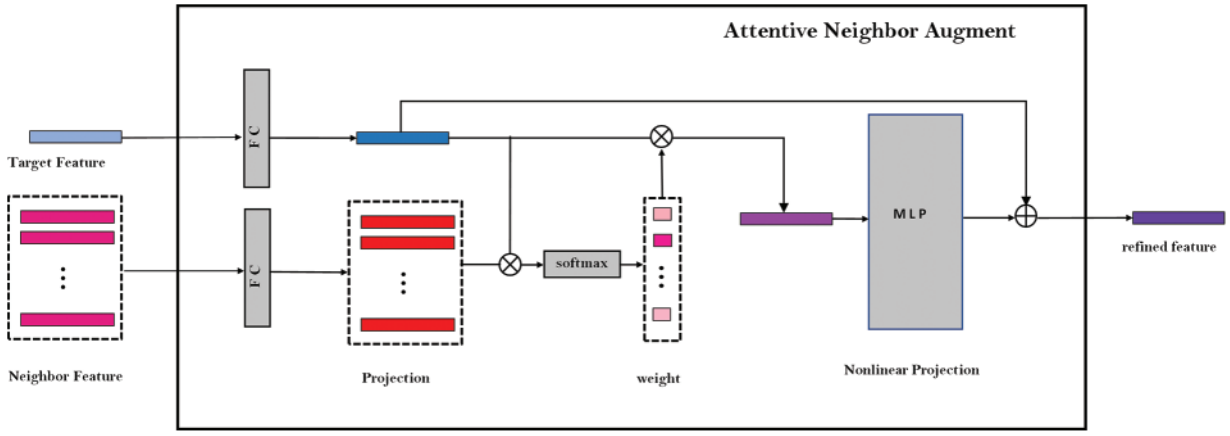

Figure 3: Illustration of the proposed attentive neighborhood feature augmentation module

3.4 Consistency Regularization

We obtain refined features by aggregating neighborhood information via the module described above. To relate refined features containing knowledge of neighbors to each other, we employ the Mixup strategy, which encourages predictions based on linear combinations of two features to approximate linear combinations of their pseudo-labels.

Formally, given two random refined features F_i, F_j and their pseudo labels y_i, y_j, the Mixup can be written as follows:

$$\begin{aligned} \hat{\boldsymbol{F}}_{ij} &= \lambda F_i + (1-\lambda) F_j \\ \hat{y}_{ij} &= \lambda y_i + (1-\lambda) y_j \end{aligned} \tag{7}$$

where $\hat{\boldsymbol{F}}_{ij}$ is the interpolation between the refined feature of F_i and F_j, and $\lambda \in [0, 1]$ is sampled from the distribution Beta (α, α).

The goal of Feature Mixup Model is minimizing the divergence between the model prediction on the interpolated feature $h_\phi(F_x)$ and the soft label $\hat{y}_{ij}$, which on an unlabeled minibatch B_u of size U can be formulated as:

$$\mathcal{L}_{mix} = \frac{1}{|U|} \sum\nolimits_{i \in \hat{B}_u} |h_\phi\left(\hat{\boldsymbol{F}}_{ij}\right) - \hat{y}_{ij}|^2 \tag{8}$$

3.5 Loss Function

Given a labeled data minibatch B_l of size L and the unlabeled data minibatch B_u of size U. The loss function for our approach consists of two terms: a supervised loss $\mathcal{L}_{sup}$ applied to labeled data and a consistency regularization term $\mathcal{L}_{mix}$. Specifically, for labeled data x with label y, the cross-entropy loss can be applied $\mathcal{L}_{sup}$ is the cross-entropy loss [50] on labeled data x:

$$\mathcal{L}_{sup} = \frac{1}{|L|} \sum\nolimits_{i \in B_l} H\left(y, h_\phi\left(\mathbf{F}_x\right)\right) \tag{9}$$

where y is the label of x and F_x is an augmented feature.

Therefore, the total loss can be written as:

$$\mathcal{L} = \mathcal{L}_{sup} + \alpha \mathcal{L}_{mix} \tag{10}$$

where α is weight for consistency regularization term.

Our propose method for SSL is summarized in Algorithm 1.

Algorithm 1: The proposed Attentive Neighborhood Feature Augmentation (ANFA) Algorithm for semi-supervised learning

3.6 Complexity Analysis

Because we need to build a global neighborhood graph, the computational complexity and memory overhead for our proposed method will unavoidably rise. We must specifically pre-train the feature extractor on labeled data before calculating the similarity matrix. We retrieve the neighborhood of each test sample from the memory bank created in the training phase and directly construct the neighborhood subgraph in the test phase. Although additional calculations are required, the convergence rate of our proposed method is much quicker than that of strong enhancement-based methods such as FixMatch and ReMixMatch, which typically require thousands of training epochs, whereas our method only requires 500 training epochs to converge.

Algorithm 1: The proposed Attentive Neighborhood Feature Augmentation (ANFA) Algorithm for semi-supervised learning

Require: labeled training set $D_l\,(x, y)$, unlabeled set $D_u\,(x)$, initial memory bank $\mathcal{M}$
Require: Feature Extractor $f\,(\theta)$, Classifier h_ϕ
Require: Attentive Neighborhood Feature Augmentation ANFA $(\cdot)$
for $t = 1, \ldots, T$ **do**
initialize the memory bank $\mathcal{M}$
using Eq. (2) calculate the similarity matrix S
Sample a labeled batch:
$B_l = \{(x_i, y_i)\}_{i=1}^{L} \sim \mathcal{D}_{\mathcal{L}}\,(x, y)$
Sample an unlabeled batch:
$B_u = \{x_i\}_{i=1}^{U} \sim \mathcal{D}_{\mathcal{U}}$
$B = Concat\,(B_l \cup B_u)$
$f_i = f_\theta\,(x_{i\in B})$
$z_i = h_\phi\,(f_i)$
construct the neighborhood graph $\mathcal{N}$ according to Eq. (3):
$\mathcal{N}_i = \mathrm{CONSTRUCTNEIGHBORSGRAPH}\,(f_i, S)$
Craft a feature augmentation batch:
$B_l = \{(\boldsymbol{F}_i = \mathrm{FEATUREAUGMENTATION}\,\left(f_{i\in B_l}, \mathcal{N}_i\right), y_i)\}_{i=1}^{L}$
Craft a feature augmentation batch with soft labels:
$B_u = \{(\boldsymbol{F}_i = \mathrm{FEATUREAUGMENTATION}\,\left(f_{i\in B_u}, \mathcal{N}_i\right), z_i)\}_{i=1}^{U}$
compute the labeled feature augmentation loss:
$\mathcal{L}_{\mathrm{sup}} = \dfrac{1}{|L|}\sum_{i\in B_l} H\left(y_i, h_\phi\,(F_i)\right)$
Shuffle $\mathrm{B_u}$ as $\mathrm{B_s}$
Sample $\lambda \sim \mathrm{Beta}\,(\alpha, \alpha)$
Compute interpolation $\hat{B}_u = \left\{\left(\hat{F}_i, \hat{y}_i\right)\right\}_{i=1}^{u}$ with:
$\hat{\boldsymbol{F}}_{ij} = \lambda F_i^1 + (1-\lambda)\,F_i^2$
$\hat{y}_{ij} = \lambda y_i^1 + (1-\lambda)\,y_i^2$
where $\left(F_i^1, y_i^1\right) = B_u[i]$, $\left(F_i^2, y_i^2\right) = B_s[i]$
Compute the consistency regularization term:
$\mathcal{L}_{mix} = \dfrac{1}{|U|}\sum_{i\in\hat{B}_u} |h_\phi\left(\hat{\boldsymbol{F}}_{ij}\right) - \hat{y}_{ij}|^2$
$\mathcal{L} = \mathcal{L}_{sup} + \alpha\mathcal{L}_{mix}$
$\theta \leftarrow \nabla_\theta\mathcal{L}$
end for
return $\boldsymbol{\theta}$

4 Experiments

In this section, we evaluate the proposed framework on commonly used SSL benchmark datasets, CIFAR-10 [25] and SVHN [24], and discuss the experimental results. We report the error rates are averaged over 5 runs with different seeds for data splitting. Specifically, we first briefly introduce the SSL benchmark datasets. Then, we show the implementation details of our proposed framework. In the end, we conduct ablation studies to validate the effectiveness of our proposed framework for SSL.

4.1 Datasets

4.1.1 SVHN

SVHN is a street view house numbers dataset, which has 73,257 training samples and 26,032 test samples from 10 number classes. The samples are 32×32 pixel RGB images. In SVHN, each sample is a number in a street View house number, and the class represents the identity of the digit in the image. Following the standard approach in SSL, we randomly select a certain number of training samples as labeled data and discard the labels of the remaining data as unlabeled data. In SVHN, we randomly select 25, 50, 100 labeled samples from each class as the labeled samples, respectively. The batch size is set to 64 for labeled data and 128 for unlabeled data.

4.1.2 CIFAR10

CIFAR10 is a natural image dataset, which has 50,000 training samples and 10,000 test samples belonging to 10 natural classes. The samples are RGB images of 32×32 size. The images in the CIFAR10 dataset are from real natural objects with large differences between categories and a certain degree of recognition difficulty, which is a classic dataset in image classification tasks. For the semi-supervised experiment, we randomly select 25, 50, 100 labeled samples from each class as the labeled samples, respectively. The batch size is set to 64 for labeled data and 128 for unlabeled data.

4.2 Implementations

Data Augmentation. We adopt standard data augmentation and data normalization in the preprocessing phase following our baselines. On the CIFAR10 dataset, we first augment the training data by random horizontal flipping and random translation (in the range of $[-2, 2]$ pixels), and then apply global contrast normalization and ZCA normalization based on statistics of all training samples. On the SVHN dataset, we first augment the training data with random translations. Inspired by [11,13], we also employ RandAugment [41] strategy to augment the training samples, which gives us a strong baseline.

Model Architecture. We conduct our experiments using a 13-layer CNN network and Wide-Resnet-28-2 architectures. For CNN-13, we adopt the exactly same 13-layer convolution neural network architecture as in [10], which eliminates the dropout layers compared to the variants in other SSL methods. The Wide-Resnet-28-2 architecture [51] is a specific residual network architecture, with extensive hyperparameter search to compare the performance of various consistency-based semi-supervised algorithms, which has been adopted as the standard benchmark architecture in recent state-of-the-art SSL methods.

Training. We use an SGD optimizer with a momentum of 0.9 and a weight decay factor 1×10^{-4}; the batch size is 64 for labeled data and 128 for unlabeled data. We conduct a hyperparameter search over the hyperparameters introduced by our method: the value of the consistency coefficient α (we searched over the values in $\{0.1, 0.2, 0.5, 1.0\}$). During the training, we set an initial learning rate of 0.1 and then decayed using the cosine annealing strategy and obtain the final results after 500 epochs. We adopt standard data augmentation such as random cropping and horizontal flipping. As our method relied on the feature representation to build the neighborhood feature graph, we pre-train the model only on labeled training samples for 10 epochs.

4.3 Results

We show our results on the CIFAR10 and SVHN datasets in Tables 4 and 5 and we have the following observations.

Table 4: Comparison of our ANFA with state-of-the-art methods on CIFAR-10

Method	CNN-13		WRN-28-2	
	1000	4000	1000	4000
PI-Model [1]	–	12.36 ± 0.31	23.07 ± 0.66	17.41 ± 0.37
TE [1]	–	12.16 ± 0.24	–	–
MeanTeacher [49]	21.55 ± 1.48	12.31 ± 0.28	17.32 ± 4.00	10.36 ± 0.25
SNTG [52]	18.41 ± 0.52	10.93 ± 0.14	–	–
VAT [7]	–	10.55	18.68 ± 0.40	11.05 ± 0.31
ICT [10]	15.48 ± 0.78	7.29 ± 0.02	–	7.66 ± 0.17
PLCB [53]	6.85 ± 0.15	5.97 ± 0.15	–	6.28 ± 0.30
MixMatch [12]	–	6.84	7.75 ± 0.32	6.24 ± 0.06
UDA [11]	–	–	6.39 ± 0.32	5.27 ± 0.11
DMT [54]	–	–	8.49 ± 0.90	5.79 ± 0.19
SimPLE [55]	–	–	**5.16**	**5.05**
DNLL [56]	12.13	7.94	7.97	5.71
ANFA(Ours)	$\mathbf{6.70 \pm 0.13}$	$\mathbf{5.33 \pm 0.05}$	6.52 ± 0.10	5.57 ± 0.15

Table 5: Comparison of our ANFA with state-of-the-art methods on SVHN

Method	CNN-13			WRN-28-2		
	250	500	1000	250	500	1000
PI-Model [1]	–	6.65 ± 0.53	4.82 ± 0.17	18.96 ± 1.92	–	7.54 ± 0.06
TE [1]	–	5.12 ± 0.13	4.42 ± 0.16	–	–	–
MT [49]	–	21.55 ± 1.48	12.31 ± 0.28	6.45 ± 2.43	3.82 ± 0.17	3.75 ± 0.10
SNTG [52]	4.29 ± 0.23	3.99 ± 0.24	3.86 ± 0.27	–	–	–
VAT [7]	–	–	–	8.41 ± 1.01	7.44 ± 0.79	5.98 ± 0.21
ICT [10]	4.78 ± 0.68	4.23 ± 0.15	3.89 ± 0.04	–	–	–
PLCB [53]	3.66 ± 0.12	3.64 ± 0.04	3.55 ± 0.08	–	–	–
MixMatch [12]	3.59	–	3.39	3.78 ± 0.26	$\mathbf{3.64 \pm 0.46}$	3.27 ± 0.31
SimPLE [55]	–	–	3.96 ± 0.10	–	–	2.75 ± 0.15
FixMatch [13]	–	–	–	$\mathbf{2.64 \pm 0.64}$	–	$\mathbf{2.36 \pm 0.19}$
ANFA(Ours)	$\mathbf{3.41 \pm 0.12}$	$\mathbf{3.39 \pm 0.07}$	$\mathbf{3.20 \pm 0.08}$	3.56 ± 0.11	$\mathbf{3.45 \pm 0.21}$	3.12 ± 0.05

For CIFAR10, our method achieves comparable results with state-of-the-art methods. It is worth mentioning that current methods with leading performance methods on the CIFAR-10 need require thousands of training epochs. In contrast, our approach converges more easily. Meanwhile, our method outperforms all the baselines under the CNN-13 architecture with 1 and 4 k labeled training samples.

For SVHN, this is much easier than the task on CIFAR-10 and the baselines already achieve a quite high accuracy. Nonetheless, our method still demonstrates a clear improvement over all the baselines across different numbers of labeled data. In particular, our method outperforms all of the

baselines under the CNN-13 architecture with 250, 500, and 1 K labeled training data, which already beats the results of all baselines with 500 labeled samples.

4.4 Ablation Study

Comparison with other attention functions. In the proposed method, we investigate the impact of various attention functions, and we choose classical attention functions for the experiments: dot-product attention, additive attention, hard attention, and multi-head attention. The experimental findings on the CIFAR-10 dataset are shown in Fig. 4. We can see that the proposed method has the same performance when using the additive attention function as when using the dot product attention function, but the calculation is faster when using the dot product attention function because it can be computed using highly optimized matrix multiplication. At the same time, when using the hard attention function, performance is slightly lower because using the one-hot weight loses some local information. Multi-head attention performs slightly better than dot-product attention, but it requires more memory and calculations. In conclusion, we employ dot-product attention, which has slightly lower performance but lower computational overhead.

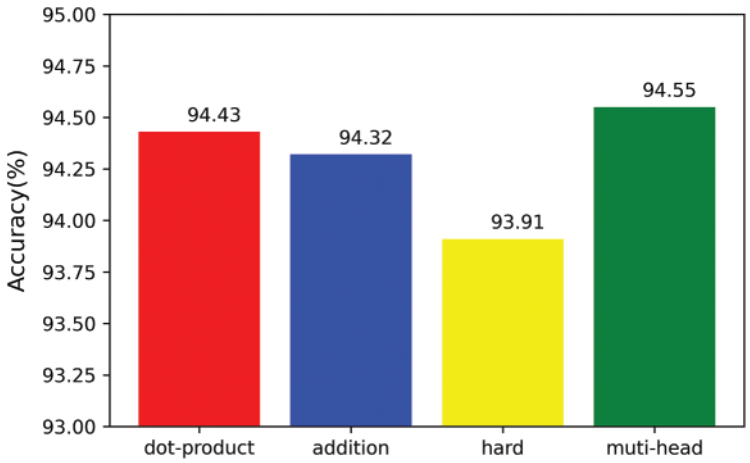

Figure 4: Comparison with other attention functions on CIFAR-10

Effectiveness of Attentive Aggregation. We propose an attention-based feature augmentation module that aggregates the neighboring features to enhance the features of the target instance, which improves the performance of the model. To show the effectiveness of attention-based aggregation, we compare the proposed attentive aggregation with the average feature aggregate method, which is the most straightforward strategy for summarizing features. We adopt ICT as the baseline model and conduct experiments on the CIFAR10 dataset with 4000 labeled samples. We conduct a baseline experiment providing the comparison results in Fig. 5. We can observe that the attention-based neighborhood feature augmentation module improves the performances of the ICT model (from 91.4% to 93.2%), and the neighborhood information helps the model to learn discriminative feature embeddings. Meanwhile, attention-based aggregation performs better than average aggregation and attentive aggregation converges faster because the adaptive weight learned by attention fully captures the neighborhood information.

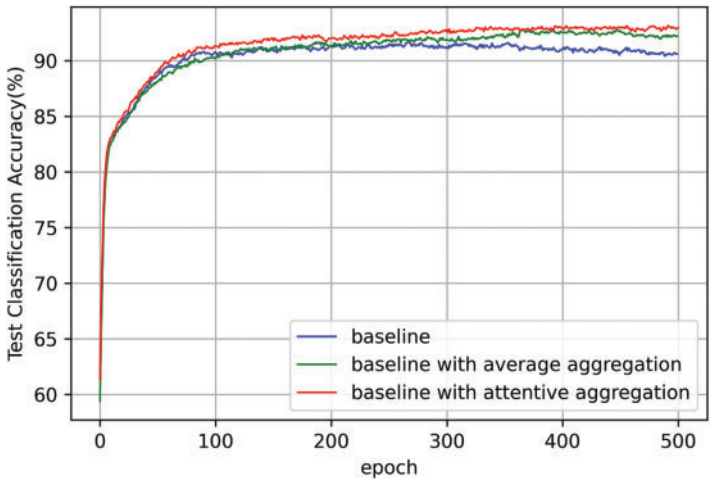

Figure 5: Test classification accuracy on CIFAR-10

Evaluation of the Neighborhood Feature Graph Size. We find that different numbers of neighbors affect the performance of the experiment. Our previous experiments on CIFAR10 fixed the size of the neighbor graph to 16. Here we explore different neighbor graph sizes for our attentive neighborhood feature augmentation. Specifically, we conduct experiments with different neighbor graph sizes on the CIFAR10 and SVHN datasets, respectively, and present the results in Fig. 6. It can be seen from the figure that the final performance will be reduced if the number of neighbors is too large or too small. This may be explained by the fact that a too-small number of neighbors will not obtain sufficient neighbor information, while a too-large number of neighbors will introduce irrelevant neighbors, which may weaken the effectiveness of neighborhood aggregation and thus impair the target features [57].

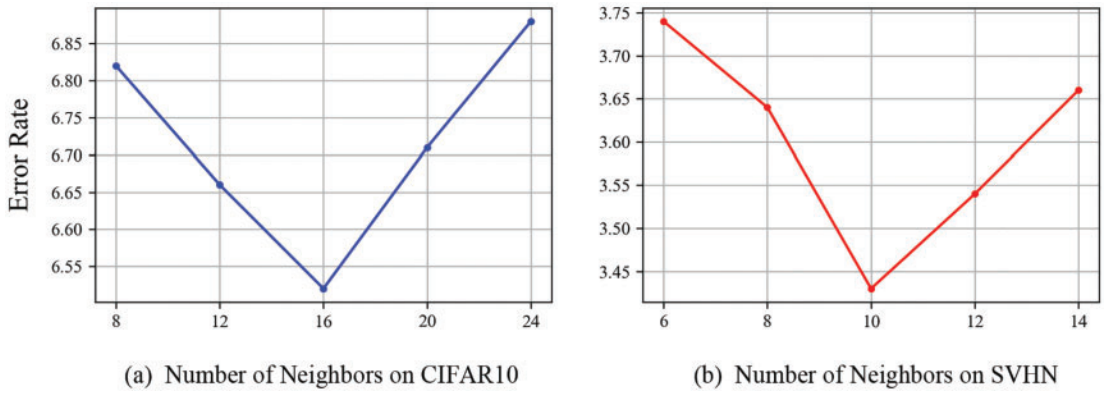

Figure 6: Evaluation of number of neighbor graph size on CIFAR-10(a) and SVHN(b)

Combination of Augmentation Strategy. Since our method employs a data augmentation strategy, we will further investigate the impact of commonly used pixel-based data augmentation strategies on the performance of the proposed method. We conduct ablation experiments on CIFAR10 datasets with WRN-28-2 architecture to study the influence of strong augmentation policies (RandAugment) and Mixup on experimental performance. The results are shown in Table 6. As we can see, excellent

data augmentation techniques give a boost to our approach. Our method can be well combined with other pixel-based augmentation strategies, as various transpositions can provide richer neighborhood information and drive our model to learn better feature representations for refinement.

Table 6: Comparison of our ANFA with data augmentation on CIFAR-10

Ablation	4000 labeled
ANFA w/o data augmentation	91.56
ANFA with Mixup	93.01
ANFA with RandAugment	94.43

5 Conclusion

In this paper, we propose a novel data augmentation method for semi-supervised learning by exploiting neighborhood information of a given instance in its semantic feature. First, for the target instance, we construct a neighbor graph based on a similarity matrix calculated by its neighbor features in the semantic layer. Second, we refine the target features with an attention-based module according to the neighbor graph. Finally, we mix up the target features and their corresponding predictions and promote a novel consistency loss as the consistency regularization. We conducted experiments on SVHN and CIFAR10 datasets. The experimental results demonstrate that our proposal is superior to the state-of-the-art SSL methods under CNN-13 neural architecture when the number of label examples is small. Moreover, the attention-based module in our method can be combined with some mainstream semi-supervised learning methods to further improve the SSL performance. Note that it might be time-consuming to create the neighborhood graph in our method when the number of training examples is large. Thus, for future work, we will focus on reducing the time complexity of constructing the neighborhood graph by exploring a parallel computation strategy. In addition, we will consider the scenario where the training dataset is unbalanced.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Nos. 62072127, 62002076, 61906049), Natural Science Foundation of Guangdong Province (Nos. 2023A1515011774, 2020A1515010423), Project 6142111180404 supported by CNKLSTISS, Science and Technology Program of Guangzhou, China (No. 202002030131), Guangdong basic and applied basic research fund joint fund Youth Fund (No. 2019A1515110213), Open Fund Project of Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University) (No. MJUKF-IPIC202101), Natural Science Foundation of Guangdong Province No. 2020A1515010423), Scientific research project for Guangzhou University (No. RP2022003).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Z. Zhou, C. Ding, J. Li, E. Mohammadi, G. Liu *et al.,* "Sequential order-aware coding-based robust subspace clustering for human action recognition in untrimmed videos," *IEEE Transactions on Image Processing*, vol. 32, pp. 13–28, 2023.

[2] J. Li, J. Wu, L. Chen, J. Li and S. Lam, "Blockchain-based secure key management for mobile edge computing," *IEEE Transactionson Mobile Computing*, vol. 22, no. 1, pp. 100–114, 2023.
[3] W. Wang, X. Wang, M. Zhou, X. Wei, J. Li *et al.,* "A spatiotemporal and motion information extraction network for action recognition," *Wireless Networks*, vol. 29, pp. 1–17, 2023.
[4] X. Wei, J. Li, M. Zhou and X. Wang, "Contrastive distortion-level learning-based no-reference image-quality assessment," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 8730–8746, 2022.
[5] W. Li, Y. Wang and J. Li, "Enhancing blockchain-based filtration mechanism via IPFS for collaborative intrusion detection in IoT networks," *Journal of Systems Architecture*, vol. 127, pp. 102510, 2022.
[6] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," arXiv preprint arXiv:1610.02242, 2017.
[7] T. Miyato, S. Maeda, M. Koyama and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.
[8] X. Wang, J. Li, Q. Liu, W. Zhao, Z. Li *et al.,* "Generative adversarial training for supervised and semi-supervised learning," *Frontiers Neurorobotics*, vol. 16, pp. 859610, 2022.
[9] Z. Luo, C. Zhu, L. Fang, G. Kou, R. Hou *et al.,* "An effective and practical gradient inversion attack," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 9373–9389, 2022.
[10] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, Y. Bengio *et al.,* "Interpolation consistency training for semi-supervised learning," *Neural Networks*, vol. 145, pp. 90–106, 2022.
[11] Q. Xie, Z. Dai, E. Hovy, T. Luong and Q. V. Le, "Unsupervised data augmentation for consistency training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6256–6268, 2020.
[12] D. Berthelot, N. Carlini, I. Goodfellow, A. Oliver, N. Papernot *et al.,* "Mixmatch: A holistic approach to semi-supervised learning," *Advances in Neural Information Processing Systems*, vol. 32, pp. 1–11, 2019.
[13] K. Sohn, D. Berthelot, C. Li, Z. Zhang, N. Carlini *et al.,* "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020.
[14] Y. Chen, R. Xia, K. Zou and K. Yang, "FFTI: Image inpainting algorithm via features fusion and Two-steps inpainting," *Journal of Visual Communication and Image Representation*, vol. 93, pp. 103776, 2023.
[15] R. Xia, Y. Chen and B. Ren, "Improved anti-occlusion object tracking algorithm using unscented rauch-tung-striebel smoother and kernel correlation filter," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6008–6018, 2022.
[16] Y. Chen, R. Xia, K. Yang and K. Zou, "MFFN: Image super-resolution via multi-level features fusion network," *The Visual Computer*, vol. 39, pp. 1–16, 2023.
[17] Y. Chen, L. Liu, V. Phonevilay, K. Gu, R. Xia *et al.,* "Image super-resolution reconstruction based on feature map attention mechanism," *Applied Intelligence*, vol. 51, pp. 4367–4380, 2021.
[18] X. Wang, X. Kuang, J. Li, J. Li, X. Chen *et al.,* "Oblivious transfer for privacy-preserving in VANET's feature matching," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4359–4366, 2021.
[19] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitilagkas *et al.,* "Manifold mixup: Better representations by interpolating hidden states," in *Int. Conf. on Machine Learning*, Long Beach, CA, USA, pp. 6438–6447, 2019.
[20] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely *et al.,* "Deep feature interpolation for image content changes," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 7064–7073, 2017.
[21] C. Kuo, C. Ma, J. Huang, Z. Kira, G. Tech *et al.,* "Featmatch: Feature-based augmentation for semi-supervised learning," in *Computer Vision–ECCV 2020: 16th European Conf., Glasgow, UK, August 23–28, 2020, Proc., Part XVIII 16*, Springer Int. Publishing, pp. 479–495, 2020.
[22] X. Liu, J. Yin, J. Li, P. Ding, J. Liu *et al.,* "TrajectoryCNN: A new spatio-temporal feature learning network for human motion prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 6, pp. 2133–2146, 2020.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.
[24] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu *et al.,* "Reading digits in natural images with unsupervised feature learning," 2011.
[25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Master's Thesis, University of Tront, pp. 32–33, 2009.
[26] S. Zagoruyko and N. Komodakis, "Wide residual networks," arXiv preprint arXiv:1605.07146, 2016.
[27] X. Wang, J. Li, X. Kuang, Y. Tan and J. Li, "The security of machine learning in an adversarial setting: A survey," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12–23, 2019.
[28] T. Huang, Q. Zhang, J. Liu, R. Hou, X. Wang *et al.,* "Adversarial attacks on deep-learning-based SAR image target recognition," *Journal of Network and Computer Applications*, vol. 162, no. 12, pp. 102632. 2020.
[29] J. Lai, Y. Huo, R. Hou and X. Wang, "A universal detection method for adversarial examples and fake images," *Sensors*, vol. 22, no. 9, pp. 3445, 2022.
[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[31] H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," arXiv preprint arXiv:1710.09412, 2017.
[32] X. Wei, X. Wei, X. Kong, S. Lu, W. Xing *et al.,* "FMixcutmatch for semi-supervised deep learning," *Neural Networks*, vol. 133, pp. 166–176, 2021.
[33] J. Chen, M. Yang and J. Ling, "Attention-based label consistency for semi-supervised deep learning based image classification," *Neurocomputing*, vol. 453, pp. 731–741, 2021.
[34] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang *et al.,* "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," arXiv preprint arXiv:1911.09785, 2019.
[35] X. Yang, H. Che, M. F. Leung and C. Liu, "Adaptive graph nonnegative matrix factorization with the self-paced regularization," *Applied Intelligence*, vol. 52, pp. 1–18, 2022.
[36] K. Chen, H. Che, X. Li and M. F. Leung, "Graph non-negative matrix factorization with alternative smoothed L 0 regularizations," *Neural Computing and Applications*, vol. 34, pp. 1–15, 2022.
[37] F. Ou, Y. Wang, J. Li, G. Zhu and S. Kwong, "A novel rank learning based no-reference image quality assessment method," *IEEE Transactions on Multimedia*, vol. 24, pp. 4197–4211, 2021.
[38] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q. V. Le *et al.,* "Autoaugment: Learning augmentation strategies from data," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 113–123, 2019.
[39] X. Lu, J. Jie, Z. Lin, L. Xiao, J. Li *et al.,* "Reinforcement learning based energy efficient robot relay for unmanned aerial vehicles against smart jamming," *Science China Information Sciences*, vol. 65, no. 1, pp. 112304, 2022.
[40] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. of Int. Conf. on Learning Representations*, Toulon, France, pp. 1–16, 2017.
[41] E. D. Cubuk, B. Zoph, J. Shlens and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*, Seattle, WA, USA, pp. 702–703, 2020.
[42] C. Ji, Z. Zhu, X. Wang, W. Zhai, X. Zong *et al.,* "Task-aware swapping for efficient DNN inference on DRAM-constrained edge systems," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 8155–8169, 2022.
[43] N. Jiang, W. Jie, J. Li, X. Liu and D. Jin, "GATrust: A multi-aspect graph attention network model for trust assessment in OSNs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, pp. 1, 2022.
[44] J. Hu, S. Li and G. Sun, "Squeeze-and-excitation networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 7132–7141, 2018.

[45] X. Li, Z. Zheng, P. Cheng, J. Li and L. You, "MACT: A multi-channel anonymous consensus based on Tor," *World Wide Web*, vol. 25, pp. 1–25, 2022.
[46] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li *et al.,* "Residual attention network for image classification," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 3156–3164, 2017.
[47] S. Woo, J. Park, J. Lee and I. Kweon, "Cbam: Convolutional block attention module," in *Proc. of the European Conf. on Computer Vision (ECCV)*, Munich, Germany, pp. 3–19, 2018.
[48] Y. Chen, Y. Li, Q. Chen, X. Wang, T. Li *et al.,* "Energy trading scheme based on consortium blockchain and game theory," *Computer Standards & Interfaces*, vol. 84, pp. 103699, 2023.
[49] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1–10, 2017.
[50] Y. Chen, J. Ma, X. Wang, X. Zhang and H. Zhou, "DE-RSTC: A rational secure two-party computation protocol based on direction entropy," *Int. J. International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 8947–8967, 2022.
[51] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk and I. J. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," *Advances in Neural Information Processing Systems*, vol. 31, pp. 1–12, 2018.
[52] Y. Luo, J. Zhu, M. Li, Y. Ren and B. Zhang, "Smooth neighbors on teacher graphs for semi-supervised learning," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 8896–8905, 2018.
[53] E. Arazo, D. Ortego, P. Albert, N. E. O' Connor and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," *Int. Joint Conf. on Neural Networks*, Glasgow, United Kingdom, pp. 1–8, 2020.
[54] Z. Feng, Q. Zhou, Q. Gu, X. Tan, G. Cheng *et al.,* "Dmt: Dynamic mutual training for semi-supervised learning," *Pattern Recognition*, vol. 30, pp. 108777, 2022.
[55] Z. Hu, Z. Yang, X. Hu and R. Nevatia, "Simple: Similar pseudo label exploitation for semi-supervised classification," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 15099–15108, 2021.
[56] H. Xu, H. Xiao, H. Hao, L. Dong, X. Qiu *et al.,* "Semi-supervised learning with pseudo-negative labels for image classification," *Knowledge-Based Systems*, vol. 260, pp. 110166, 2023.
[57] W. Tang, B. Li, M. Barni, J. Li and J. Huang, "Improving cost learning for JPEG steganography by exploiting JPEG domain knowledge," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 4081–4095, 2021.