MutexMatch: Semi-Supervised Learning With Mutex-Based Consistency Regularization

Yue Duan, Zhen Zhao, Lei Qi, Lei Wang, Senior Member, IEEE, Luping Zhou, Senior Member, IEEE, Yinghuan Shi, and Yang Gao, Senior Member, IEEE

Abstract—The core issue in semi-supervised learning (SSL) lies in how to effectively leverage unlabeled data, whereas most existing methods tend to put a great emphasis on the utilization of high-confidence samples yet seldom fully explore the usage of low-confidence samples. In this article, we aim to utilize low-confidence samples in a novel way with our proposed mutex-based consistency regularization, namely MutexMatch. Specifically, the high-confidence samples are required to exactly predict “what it is” by the conventional true-positive classifier (TPC), while low-confidence samples are employed to achieve a simpler goal—to predict with ease “what it is not” by the true-negative classifier (TNC). In this sense, we not only mitigate the pseudo-labeling errors but also make full use of the low-confidence unlabeled data by the consistency of dissimilarity degree. MutexMatch achieves superior performance on multiple benchmark datasets, i.e., Canadian Institute for Advanced Research (CIFAR)-10, CIFAR-100, street view house numbers (SVHN), self-taught learning 10 (STL-10), and mini-ImageNet. More importantly, our method further shows superiority when the amount of labeled data is scarce, e.g., 92.23% accuracy with only 20 labeled data on CIFAR-10. Code has been released at https://github.com/NJUyued/MutexMatch4SSL.

Index Terms—Mutex-based consistency regularization, semi-supervised classification.

I. INTRODUCTION

TO AVOID time-consuming and laborious labeling tasks, semi-supervised learning (SSL) [1], [2], [3], [4] is adopted to leverage a large quantity of unlabeled data along with a small quantity of labeled data during training. Recent SSL models could be categorized into two types: 1) consistency regularization-based methods and 2) entropy minimization-based methods, while the utilization of unlabeled data is crucial in both. In specific, consistency regularization-based methods, e.g., [5], [6], tend to utilize all unlabeled data together with the supervision from labeled data, which is at the risk of strong confirmation bias [7]. Although recent holistic methods, such as [8], [9], realize consistency regularization by integrating entropy minimization into the training process via pseudo-labeling [10], they share a non-negligible limitation: they use only certain unlabeled data (i.e., with high confidence) to participate in the training while preventing uncertain unlabeled data (i.e., with low confidence) from being fully exploited. This waste of unlabeled samples may greatly deteriorate the outcome of the final performance. In comparison, recent entropy minimization-based method [11] employs pseudo-labeling to incorporate partial samples selected by a low-confidence threshold into the training process, while certain low-confidence samples are still neglected. The reason for discarding low-confidence samples in previous methods lies in these samples could be easily mis-predicted. Thus, the core problem is to tradeoff informatively learning low-confidence samples and preventing possible error accumulation.

Given these limitations, we try to answer—if we could treat these low-confidence unlabeled samples in an appropriate way? When imaging a low-confidence sample with its actual label of a “horse,” in some cases, it might be difficult for a trained model to make an accurate prediction, i.e., “it is a horse.” On the contrary, it should be much easier for the model to “guess what it is not, e.g., it is not a cat” or “it is not an elephant.” This drives us to consider a more straightforward yet feasible direction: for low-confidence samples, can we design a paradigm to learn from “what it is not” so as to benefit the learning of “what it is.” In specific, for different augmented versions of an image with the same semantic label, though their accurate predictions might be totally different, their predictions of the degree of dissimilarity could share some overlap. As illustrated in Fig. 1 where there are many alternatives to answer what it is not, we believe the consistency of this dissimilarity degree can benefit the model to learn an informative representation especially for low-confidence samples. Thus, we impose consistent prediction on two different augmented versions of the same low-confidence sample, both toward its complementary
label, which is analogous to the consistency loss imposed on high-confidence samples commonly employed in consistency-based SSL. In this sense, when using low-confidence unlabeled data with more easily obtained complementary labels, less error information could be expected to benefit the learning for “what it is.”

To achieve the aforementioned goal, we propose a novel framework, MutexMatch, to leverage all unlabeled data, especially low-confidence unlabeled data. We utilize true-positive classifier (TPC): to predict “what it is,” and true-negative classifier (TNC): to predict “what it is not,” which is designed to learn the feature representation of unlabeled data from a mutex perspective. Like the conventional way in previous works [8], the weakly-augmented unlabeled samples are also used to generate pseudo-labels, while strong augmentation is achieved by using RandAugment [12]. Specifically, we set a threshold to control the high-confidence portion and low-confidence portion of pseudo-labels. In the high-confidence portion, we enforce the consistency regularization on the output of TPC. Compared with the existing methods, an improvement of MutexMatch is that it allows low-confidence unlabeled samples to participate in training in a manner of complementary label (i.e., a class of “not belonging”) outputted from TNC, which helps model learn better data representation to further boost TPC. We find that the class with the smallest probability predicted by TPC already serves as a simple classifier, i.e., we encourage the model to maintain a consistent degree of dissimilarity in the top-k dissimilar classes where a larger k represents a stricter consistency. By the above design, our approach is able to provide additional information from low-confidence samples to the model in a relaxed manner. Finally, we theoretically analyze an objective function involving mutex-based consistency regularization. The diagram of the proposed MutexMatch is shown in Fig. 2.

In this work, the key contributions include threefold as follows.

1) We use two classifiers (i.e., TPC and TNC) to construct MutexMatch, a novel framework using pseudolabel and complementary label to learn an informative representation of unlabeled samples.

2) We propose mutex-based consistency regularization for SSL, which aims to make full use of unlabeled data, especially low-confidence unlabeled samples, in an effective way. We also theoretically show our error bound is lower than that of the conventional consistency-based model without TNC.

3) We obtain superior results in a label-scarce setting than recently-proposed SSL algorithms, e.g., on the most commonly studied SSL benchmark Canadian Institute for Advanced Research (CIFAR)-10, MutexMatch using only 20 labels can achieve an accuracy of 92.23 ± 3.23%.

In the rest of this article, Section II provides the background for our approach; Section III introduces MutexMatch, the proposed method for image classification; Section IV shows the experimental results of MutexMatch on various SSL benchmarks; Section V provides extensive ablation studies for each component in our method; Section VI further discusses the consistency regularization on the proposed TNC; Section VII conclude our contribution in this work.

II. RELATED WORK

In this section, we review the related works from SSL, consistency regularization, pseudolabeling, and complementary label.

A. Semi-Supervised Learning (SSL)

SSL aims to exploit ample unlabeled data for mitigating the lack of labeled data. In a word, we need to utilize the guidance information provided by limited labeled data to mine potential data representation of unlabeled data. Which can be summarized as the following optimization task:

$$\min \mathcal{L} = \mathcal{L}_s(x_{lb}, y_{lb}; \theta) + \mathcal{L}_u(x_{ub}; \theta)$$

where \( \theta \) is the parameter of model, \( \mathcal{L}_s \) is the supervised loss for labeled data \( x_{lb} \) (the corresponding label is \( y_{lb} \)) and \( \mathcal{L}_u \) is the unsupervised loss for unlabeled data \( x_{ub} \). SSL is a very promising technique in many fields of machine learning, e.g., classification [8], [11], [13], [14], [15], detection [16], [17], and segmentation [18], [19], [20] because it breaks the dilemma that labeling data consumes a lot of time and energy. In addition, SSL could be applied to a broader scene, e.g., domain adaptation [21], deep hashing [22], person reidentification [23], distributed learning [24], and so on. In addition, generative models also play a significant role in SSL, which including 1) semi-supervised variational auto-encoder (VAE) [25], [26], [27], [28]; and 2) semi-supervised generative adversarial net (GAN) [29], [30], [31], [32]. Although generative models are often considered unsupervised methods, they can be used to facilitate SSL since they can learn the distribution of real data from unlabeled samples, e.g., one way that is most relevant to this article is to regularize classifiers using samples produced by GANs. In this article, we focus on SSL for image classification. A similar strategy is prevalent in current semi-supervised classification work, the model is trained on labeled data and pseudo-labels are generated for unlabeled data.
based on the model’s predictions [8], [10], [33], [34], [35], which boosts the performance of SSL combining consistency regularization (described below).

B. Consistency Regularization

Consistency regularization is a significant branch of recent state-of-the-art (SOTA) SSL methods, which is proposed in [36]. Such methods encourage the classifier to output the same class probability distribution after different versions of augmentation for the same unlabeled data. Generally speaking, the consistency regularization-based models are trained with unlabeled data using the loss function: \( \| p(y|x_\alpha(x)) - p(y|a(x)) \|^2 \), where \( x \) is the input image, \( p(\cdot) \) and \( a(\cdot) \) are stochastic functions [36]. Particularly, \( a(\cdot) \) can adopt different augmentation methods, e.g., RandAugment [8] in [8] and CTAugment in [34].

Specifically, Laine and Aila [5] enforce a loss of consistency in the predictions of two augmented variants of unlabeled data. In [6], a teacher model is maintained to generate more stable targets for unlabeled data, while the mean squared error is used to encourage the same predictions of both the student and teacher models. Xie et al. [37] adopts automatic augmentation for data perturbation and enforces a loss of consistency by the KL divergence. Some other models seek to improve the consistency of predicted outputs over the perturbed data to train semi-supervised models. The generative model mentioned above can also be applied to consistency regularization. GAN-based methods (e.g., [32], [38]) train a multiclass discriminator to distinguish fake samples from the real classes, improving the consistency of predictions over the perturbed data. Recently, some holistic methods [8], [34], [35], [39] have been proposed to combine consistency regularization with pseudo-labeling for better SSL performance.

Differently, in MutexMatch, we propose a novel mutex-based consistency to learn an informative representation of all unlabeled data. Precisely, our method has a large distinction from previous methods that involved consistency regularization: most previous works either 1) utilizing distinction from previous methods that involved consistency regularization between predictions on different augmentations, or 2) utilizing mutex-based consistency to learn an informative representation with pseudo-labeling for better SSL performance. Moreover, our method has a large mutex-based consistency to learn an informative representation of all unlabeled data using the loss function: \( \| p(y|x_\alpha(x)) - p(y|a(x)) \|^2 \), where \( x \) is the input image, \( p(\cdot) \) and \( a(\cdot) \) are stochastic functions [36]. Particularly, \( a(\cdot) \) can adopt different augmentation methods, e.g., RandAugment [8] in [8] and CTAugment in [34].

C. Pseudo-Labeling

Pseudo-labeling is widely leveraged to achieve entropy minimization [40] by constructing one-hot labels from predictions of unlabeled data and making use of them based on cross-entropy (CE) loss [10]. However, recent pseudolabeling-based methods [8], [35] all have a significant limitation, i.e., using a threshold to select high-confidence pseudo-labels results in the waste of low-confidence samples. We notice [11] proposes an uncertainty-aware pseudo-label selection (UPS) framework to use both high and low-confidence samples. However, it introduces two thresholds to control the pseudo-label generation, leading partial samples to remain unutilized. Moreover, the way it utilizes low-confidence samples is not very informative, which can be demonstrated in Section V-A.

D. Complementary Label

Complementary label [11], [41], [42], [43] is used to help the model learn which class the input image does not belong to. Considering a \( c \)-class classification task, we denote \( x \in \mathcal{X} \) as an input image and \( y \in \mathcal{Y} = \{1, \ldots, c\} \) as its positive label. Complementary label \( \overline{y} \) is generated by selecting from \( \mathcal{Y}\setminus\{y\} \) at random. Different from [43], we design a novel way (detailed in Section III-B) to propose complementary labels, so as to ensure their effectiveness in SSL, while experiments on using the standard generation of complementary label will be discussed in Section VI-B.

III. MutexMatch

A. Overview

Different from existing SSL approaches, besides a feature extractor \( \theta(\cdot) \), MutexMatch jointly trains two distinct classifiers, a TPC \( \mathcal{P}(\cdot) \) and a TNC \( \mathcal{N}(\cdot) \). In specific, TPC is used to predict which class the instance belongs to (i.e., used for the test phase), while TNC is employed to indicate which class the instance does not belong to (i.e., true-negative). To mitigate pseudo-labeling errors, a predefined high-confidence threshold \( \tau \) is utilized to split the unlabeled data into high-confidence and low-confidence portions. Besides training TPC on the high-confidence portion, we explore complementary labels on low-confidence samples to train TNC. In this way, all the unlabeled data could be effectively exploited.

We have \( B \) labeled data \( \mathcal{X} = \{ (x^b_n, y^b_n) \}_{n=1}^B \) and \( B \) unlabeled data \( \mathcal{U} = \{ (x^ul_n)^\mu \}_{n=1}^B \), where \( \mu \) represents the relative size of \( \mathcal{X} \) and \( \mathcal{U} \). Following [8], we perform weak and strong augmentations for data perturbations, denoted by \( a_w(\cdot) \) and \( a_s(\cdot) \), respectively. Given weakly-augmented instances \( x^w = a_w(x^ul) \) and strongly-augmented instances \( x^s = a_s(x^ul) \), MutexMatch simultaneously optimizes four losses: 1) the supervised loss \( \mathcal{L}_{sup} \); 2) the separated negative loss \( \mathcal{L}_{sep} \); 3) the positive consistency loss \( \mathcal{L}_{p} \); and 5) the negative consistency loss \( \mathcal{L}_{n} \). To sum up, the total loss is

\[
\mathcal{L} = \mathcal{L}_{sup} + \lambda_{sep} \mathcal{L}_{sep} + \lambda_{p} \mathcal{L}_{p} + \lambda_{n} \mathcal{L}_{n}
\]

where \( \lambda_{sep}, \lambda_{p}, \) and \( \lambda_{n} \) are weight hyper-parameters to balance the relative importance of corresponding losses. The supervised loss \( \mathcal{L}_{sup} \) is the CE between \( y^b_n \) and the predictions of TPC on labeled data \( x^b_n \), calculated as follows:

\[
\mathcal{L}_{sup} = \frac{1}{B} \sum_{n=1}^{B} H(y^b_n, \mathcal{P}(\theta(a_w(x^b_n))))
\]

where \( H(p, q) \) denotes the standard CE loss between distribution \( q \) and \( p \).
Fig. 2. Diagram of MutexMatch. Given a batch of unlabeled samples, TPC $P$ first uses their weakly-augmented variants to generate pseudo-labels. Then, the classes with the lowest confidence are adopted as the complementary labels to train TNC $N$ separately. Meanwhile, TPC and TNC are used for mutex-based consistency regularization in the high and low-confidence portions of TPC’s predictions, respectively. $z$ denotes output features while $p$ and $r$ denote predictions of TPC and TNC. Superscripts $w$ and $s$ represent corresponding outputs for the weakly-augmented variants and strongly-augmented variants, respectively.

Fig. 3. Training models (WRN-28-2 [44]) on CIFAR-10/100 with only 40/400 labels are shown in (a)/(b). We show the error rate of complementary label with various $m$, where $m$ indicates that the class index with $m$th smallest probability in the prediction of TPC is chosen to serve as the complementary label and we choose $\arg \min (P(\theta(x^w)))$ in MutexMatch.

B. True-Negative Classifier

In multiclass classification tasks, it is easier to predict “what it is not” than to decide exactly “what it is.” For example, given an image of an airplane in CIFAR-10, we can predict which class it does not belong to with a 90% probability at random, whereas there is merely a 10% probability to correctly predict it indeed is an airplane. To this end, we design a TNC to predict what it is not. Considering that compared to TPC, it is much easier to obtain the correct labels for TNC, we first exploit TNC to provide more guidance information on unlabeled data.

Unlike the standard way of generating complementary label [7], [41], we use the class with the lowest confidence in TPC’s prediction as the complementary label to train TNC, which is a simple strategy to obtain a robust TNC. Intuitively, using the class with the lowest probability in the prediction is less likely to be the correct class of the image, i.e., the complementary label is more likely to be correct. This claim can be demonstrated in Fig. 3. The separate training loss $L_{sep}$ for TNC can be calculated as

$$L_{sep} = \frac{1}{\mu_B} \sum_{n=1}^{\mu_B} H\left(\arg \min (P(\theta(x^w))), N(\hat{\theta}(x^w))\right)$$

where $\hat{x}$ represents we stop back-propagating gradient on $x$.

Since our downstream task is to accurately classify images, we adopt such gradient-blocking operation to ensure that the feature extractor will not be affected by the training of TNC (because it predicts complementary label). Furthermore, we extensively investigate the effectiveness of TNC in Section VI-A.

C. Mutex-Based Consistency Regularization

In recent consistency-regularization-based SSL methods, only samples with high-confidence predictions are leveraged to train models, which could lead to inefficient utilization of unlabeled data, especially at the early stage of the training process. By contrast, MutexMatch can also learn an informative representation of low-confidence unlabeled samples by introducing a novel mutex-based consistency regularization. A high-confidence threshold $\tau$ on TPC’s predictions is defined to separate the unlabeled samples into two portions with mutex confidence intervals, i.e., the high-confidence one ($\geq \tau$) and the low-confidence one ($< \tau$). In the high-confidence portion, we use TPC to learn what the unlabeled data is, while in the low-confidence portion, we employ TNC to learn what it is not, as it is difficult for us to obtain its real class information.

On the one hand, we first use weakly-augmented example $x^w$ to generate pseudo-labels from TPC and enforce positive consistency against its corresponding strongly-augmented variant $x^s$. Then we obtain their predictions, $p^w = \hat{P}(\hat{\theta}(x^w))$ and...
Algorithm 1 MutexMatch: Semi-Supervised Learning With Mutex-Based Consistency Regularization

Input: batch of labeled data $\mathcal{X} = \{(x_n^{lb}, y_n^{lb})\}_{n=1}^{B}$, batch of unlabeled data $\mathcal{U} = \{x_n^{ulb}\}_{n=1}^{B}$, feature extractor $\theta$, TPC $\mathcal{P}$, TNC $\mathcal{N}$, hyper-parameter $k$ for a top-$k$ algorithm

for iteration $t$ do
  \[ L_{sup} = \frac{1}{B} \sum_{n=1}^{B} H(y_n^{lb}, \mathcal{P}(x_n^{lb})) \] // Supervised loss for $x^{lb}$

for iteration $n = 1$ to $\mu B$ do
  \[ p_n^w = \hat{\mathcal{P}}(\theta(x_n^{ulb})) \] // Compute TPC's prediction for weakly-augmented $x_n^{ulb}$
  \[ p_n^s = \mathcal{P}(\theta(x_n^{ulb})) \] // Compute TPC's prediction for strongly-augmented $x_n^{ulb}$
  \[ r_n^w = \hat{\mathcal{N}}(\theta(x_n^{ulb})) \] // Compute TNC's prediction for weakly-augmented $x_n^{ulb}$
  \[ g_n(i) = 1_{r_n(i),(i) \in \mathcal{T}_r(r_n^w)} \] // Control the intensity of consistency regularization on $\mathcal{N}$
  \[ r_n^s = \mathcal{N}(\theta(x_n^{ulb})) \] // Compute TNC's prediction for strongly-augmented $x_n^{ulb}$
  \[ \hat{p}_n^w = \text{arg max}(p_n^w) \] // Select pseudo-label for $x_n^{ulb}$
  \[ \hat{q}_n^w = \text{arg min}(p_n^w) \] // Select complementary pseudo-label for $x_n^{ulb}$

end for

\[ L_{sep} = \frac{1}{\mu B} \sum_{n=1}^{\mu B} H(\hat{q}_n^w, \mathcal{N}(x_n^{ulb})) \] // Stop back-propagating gradient on $\theta$

\[ L_{p} = \frac{1}{\mu B} \sum_{n=1}^{\mu B} \mathbb{1}(\text{max}(p_n^w) \geq \tau) H(\hat{p}_n^w, p_n^s) \] // Positive consistency loss for $x_n^{ulb}$

\[ L_{n} = \frac{1}{\mu B} \sum_{n=1}^{\mu B} \mathbb{1}(\text{max}(p_n^w) < \tau)(-\frac{1}{k} \sum_{i=1}^{C} g_n(i) p_n^w(i) \log(r_n^s(i))) \] // Negative consistency loss for $x_n^{ulb}$

update $\theta, \mathcal{P}, \mathcal{N}$ by SGD to optimise $L_{sup} + \lambda L_{sep} + \lambda_p L_p + \lambda_n L_n$

$p^t = \mathcal{P}(\theta(x^t))$. Let $\hat{p}_n^w = \text{arg max}(p_n^w)$, such consistency can be achieved by minimizing the positive consistency loss $L_p$

\[ L_p = \frac{1}{\mu B} \sum_{n=1}^{\mu B} \mathbb{1}(\text{max}(p_n^w) \geq \tau) H(\hat{p}_n^w, p_n^s) \] (4)

where $\mathbb{1}(\text{max}(p_n^w) \geq \tau)$ retains the predictions whose maximum probabilities are larger than $\tau$. For entropy minimization [10], we adopt hard pseudo-label $\hat{p}_n^w$ to enforce the consistency regularization on TPC.

On the other hand, for the augmented images $x_n^{w}$ and $x_n^{s}$ of these low-confidence samples, TNC computes predictions $r_n^{w} = \hat{\mathcal{N}}(\theta(x_n^{w}))$ and $r_n^{s} = \mathcal{N}(\theta(x_n^{s}))$ for them. We simply treat the probability component $r_n^{w}$ in $r_n^{w}$ as the degree of dissimilarity from $t$th class, where $i \in (1, \ldots, C)$ and $C$ is the total amount of classes. We enforce consistency regularization on the degree of dissimilarity, which is achieved by minimizing the negative consistency loss $L_n$

\[ L_n = \frac{1}{\mu B} \sum_{n=1}^{\mu B} \mathbb{1}(\text{max}(p_n^w) < \tau) \times \left( -\frac{1}{k} \sum_{i=1}^{C} g_n(i) p_n^w(i) \log(r_n^{s}(i)) \right) \] (5)

where $k$ is the hyper-parameter used to control the intensity of consistency. The $g_n$ is a $C$-dimensional binary vector indicating the selected components in prediction $r_n^{w}$, which is defined as: $g_n(i) = 1_{r_n^{w}(i) \in \mathcal{T}_r(r_n^{w})}$. $\mathcal{T}_r$ represents the top-$k$ algorithm used to select the top $k$ largest probability components in prediction. When $k = C$, we use $H(r_n^{w}, r_n^{s})$ to replace term $-\frac{1}{k} \sum_{i=1}^{C} g_n(i) p_n^w(i) \log(r_n^{s}(i))$ in (5) for simplicity (i.e., all components in $r_n^{w}$ are selected). We use soft pseudo-label $r^w$ for consistency regularization on TNC, i.e., we encourage TNC to maintain consistent prediction probability distributions for different augmented versions of the same image, while the setting of consistency using soft-label will be discussed in Section VI-B. The pseudo-code of the proposed MutexMatch is presented in Algorithm 1.

D. Theoretical Analysis

Following [45], we study MutexMatch and prove a lower error bound than conventional consistency-based methods without TNC, e.g., current SOTA SSL method FixMatch [8]. We show that minimizing both positive consistency loss and negative consistency loss in a mutex-based way leads to high accuracy on pseudo-labels, which in turn boosts the performance of SSL.

Firstly, we denote $P$ as the distribution of unlabeled samples $x^{ulb}$ over input space $\mathcal{U}$. We let $G_{pl}$ denote a classifier trained on labeled data $x^{lb} \in \mathcal{X}$. Next, we define the population losses of positive consistency (for a positive label) and negative consistency (for a complementary label), respectively. Given transformation $a$, following [45], we define the population positive consistency loss as:

\[ R_a(P) = \mathbb{E}_P[\mathbb{1}(x : p^w \neq p^s)]. \] (6)

Considering that a class has multiple complementary labels, we can’t directly define the population negative consistency loss as (6). Thus, we consider using a similarity metric function $\zeta(T_r^{w}, T_r^{s})$, e.g., CE, to define the population negative consistency loss

\[ R_a(N) = \mathbb{E}_P[\mathbb{1}(x : \zeta(T_r^{w}, T_r^{s}) \leq t)] \] (7)

where $t$ is a threshold used to determine whether $r^w$ and $r^s$ (i.e., predictions of TNC) are consistent. Choosing an appropriate $t$ can enable us to obtain a reliable estimation for negative consistency loss $L_n$ that uses soft complementary labels. Next, we introduce the following two reasonable assumptions to help investigate our analysis.

We require the following assumption that it is easier for the classifier to achieve consistency on complementary
pseudo-labels than that on pseudo-labels. For a classifier that has not been implemented with any consistency regularization, it is very difficult to keep consistent predictions on different augmented versions of the same image. However, it is simpler to achieve a tolerant consistency on a set of complementary labels, i.e., as long as there is a partial overlap in the degree of dissimilarity as described in Section I.

Assumption 1: We assume $P$ satisfies Assumption 4.1 in [45] for some expansion factor $c$, and satisfies Separation Assumption 3.3 in [45] for some $\mu$ and $\omega$. For positive label, $P$ is $a$-separated with probability $1 - \mu$ by ground-truth classifier $G^*$, as follows: $R_a(G^*) \leq \mu$. For complementary label, $P$ is $a$-separated with probability $1 - \omega$ by complementary ground-truth classifier $G^\circ$, as follows: $R_a(G^\circ) \leq \omega$. And we suppose $\omega < (\mu/2)$, which is based on our claim that consistency on complementary labels is much easier to achieve.

We define robust set of $P$: $S_a(P) = \{x : p^w = p^x\}$, and robust set of $N$: $S_a(N) = \{x : \zeta(T_k(r^n), T_k(r^x)) > t\}$, which means $P$ and $N$ are robust for images under transformation $\alpha$, respectively. Now we introduce our key assumption as follows: there are few samples that are capable of achieving prediction consistency on both pseudo-labels and complementary labels.

Assumption 2: We assume $P(S_a(P) \cap S_a(N)) \geq P(S_a(P) \cap S_a(N))$.

In MutexMatch, we perform mutex-based consistency regularization on disjoint sets of samples (i.e., high-confidence portion and low-confidence portion) indicating the right-hand side of Assumption 2 is relatively small. Thus, it is reasonable to believe this assumption always holds.

Given a classifier $\mathcal{F}$ of the conventional consistency-based model (e.g., FixMatch), we design the following objective over $\mathcal{F}$: $\min_{\mathcal{F}} \mathcal{L}(\mathcal{F}) = ((c + 1)/(c - 1))L_{0,1}(\mathcal{F}, G_{pl}) + (2c/(c - 1))R_{a}(\mathcal{F}) - \text{Err}(G_{pl})$, where $L_{0,1}(G, G') = \mathbb{E}_p[\Pi(G(x) \neq G'(x))]$ is defined to be the disagreement between $G$ and $G'$. According to Lemma 4.2 in [45], we have $\text{Err}(\mathcal{F}) \leq \mathcal{L}(\mathcal{F})$. For any minimizer $\mathcal{F}$ of $\min_{\mathcal{F}} \mathcal{L}(\mathcal{F})$, we denote its error bound as $B_{\mathcal{F}}$. Given $P$ and $N$ in MutexMatch, denoting $L_{0,1}(P, G_{pl}) + R_{a}(P) - (c - 1)\text{Err}(G_{pl})$ as $\varphi$, we design the following objective over $\mathcal{P}$:

$$\min_{\mathcal{P}} \mathcal{L}(\mathcal{P}) = \begin{cases} \frac{2}{c - 1}L_{0,1}(P, G_{pl}) + \frac{c + 1}{c - 1}R_{a}(P) + 2R_{a}(N), & \varphi \leq 0 \\ \frac{c + 1}{c - 1}L_{0,1}(P, G_{pl}) + \frac{2c}{c - 1}R_{a}(P) - \text{Err}(G_{pl}), & \varphi > 0. \end{cases}$$

For any minimizer $\hat{\mathcal{P}}$ of (8), we denote its error bound as $B_{\mathcal{F}}$. The next theorem shows that the error bound of SSL methods with mutex-based consistency is lower than that of conventional consistency-based methods.

**Theorem 3:** Suppose Assumptions 1 and 2 hold, we have $\text{Err}(\mathcal{P}) \leq \mathcal{L}(\mathcal{P})$ and $B_{\mathcal{F}} \leq B_{\mathcal{F}}$.

To prove this theorem, we give the proof of the error bound of TPC $\mathcal{P}$ in MutexMatch first. Then Theorem 3 follows immediately by holding Assumption 1. The proofs follow the analysis of [45]. We first give the main notations of [45] which are used in the following proofs. $\mathcal{M}(\mathcal{P}) = \{x : \mathcal{P}(x) \neq G^*(x)\}$ denotes the set of examples pseudolabeled mistakenly by $\mathcal{P}$, where $G^*(x)$ is the ground-truth of unlabeled sample $x$.

Following Theorem A.2 in [45], we define three disjoint sets of $\mathcal{M}(\mathcal{P}) \cap S_a(\mathcal{P})$: $\mathcal{M}_1 = \{x : \mathcal{P}(x) = G_{pl}(x), G_{pl}(x) \neq G^*(x), x \in S_a(\mathcal{P})\}$, $\mathcal{M}_2 = \{x : \mathcal{P}(x) \neq G_{pl}(x), G_{pl}(x) \neq G^*(x), G^*(x) \neq G'(x), x \in S_a(\mathcal{P})\}$, $\mathcal{M}_3 = \{x : \mathcal{P}(x) \neq G_{pl}(x), G_{pl}(x) = G^*(x), x \in S_a(\mathcal{P})\}$.

$q$ is a factor defined in Definition A.1 in [45] and we have $P(\mathcal{M}_1 \cup \mathcal{M}_2) \leq q$ by Lemma A.3 in [45]; $\beta$ is a factor defined in Lemma A.7 in [45] for choice of $q$; $c$ is the expansion factor defined in Definition 3.1 in [45]. Moreover, given $\mathcal{F}$, a classifier of the conventional consistency-based method, and the objective over $\mathcal{F}$:

$$\min_{\mathcal{F}} \mathcal{L}(\mathcal{F}) = \frac{c + 1}{c - 1}L_{0,1}(\mathcal{F}, G_{pl}) + \frac{2c}{c - 1}R_{a}(\mathcal{F}) - \text{Err}(G_{pl})$$

we have $\text{Err}(\mathcal{F}) \leq \mathcal{L}(\mathcal{F})$ by Lemma 4.2 in [45]. And in the setting of Theorem 4.3 in [45], for any minimizer $\mathcal{F}$ of $\min_{\mathcal{F}} \mathcal{L}(\mathcal{F})$, we have its error bound: $\text{Err}(\mathcal{F}) \leq B_{\mathcal{F}} = (2/(c - 1))\text{Err}(G_{pl}) + (2c/(c - 1))\mu$.

**Lemma 4:** In the setting of Theorem A.2 in [45], we have $P(\mathcal{M}_1 \cap S_a(\mathcal{N})) \leq q + R_a(N)$.

**Proof:** Given (A.2) and (A.3) in [45], we know $\mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3 = \{x : \mathcal{P}(x) \neq G_{pl}(x), x \in S_a(\mathcal{N})\}$ and $\mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3 \cup \mathcal{M}_4 = \{x : \mathcal{P}(x) = G_{pl}(x), x \in S_a(\mathcal{N})\}$ are unions of disjoint sets. Thus we have

$$P(\mathcal{M}_1 \cap S_a(\mathcal{N})) = P(\mathcal{M}_1 \cap S_a(\mathcal{N})) + P(\{x : \mathcal{P}(x) = G_{pl}(x), x \in S_a(\mathcal{N})\} \cap S_a(\mathcal{N}))$$

**rearranging we obtain**

$$P(\mathcal{M}_3 \cap S_a(\mathcal{N})) = P(\mathcal{M}_1 \cap S_a(\mathcal{N})) + P(\{x : \mathcal{P}(x) = G^*(x), x \in S_a(\mathcal{N})\} \cap S_a(\mathcal{N}))$$

Now we complete the proof of Theorem 3 by combining the Lemma 4, Assumptions 1 and 2.

**Proof of Theorem 3:** Given three aforementioned disjoint subsets of $\mathcal{M}(\mathcal{P}) \cap S_a(\mathcal{P})$: $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$, we know...
\[ P(\mathcal{M}(P) \cap S_a(P)) = P(M_1) + P(M_2) + P(M_3). \] Thus, we obtain
\[ P(\mathcal{M}(P) \cap S_a(P) \cap \overline{S_a(N)}) = P(M_1 \cap \overline{S_a(N)}) + P(M_2 \cap \overline{S_a(N)}) + P(M_3 \cap \overline{S_a(N)}). \] (9)

Then, holding Assumption 2 which is tightly related to mutex-based consistency regularization, we compute
\[ \text{Err}(P) = P(\mathcal{M}(P)) \leq P(\mathcal{M}(P) \cap S_a(P) \cap \overline{S_a(N)}) + P(S_a(P) \cap \overline{S_a(N)}) \leq P(M_1) + P(M_2) + P(M_3 \cap \overline{S_a(N)}) \]
\[ = P(M_1 \cap \overline{S_a(N)}) + P(M_2 \cap \overline{S_a(N)}) + P(M_3 \cap \overline{S_a(N)}) \quad \text{(by Equation (9))} \]
\[ \leq 2q + R_a(N) + P(S_a(P) \cap \overline{S_a(N)}) \quad \text{(by Lemma 4)} \]
\[ \leq 2q + R_a(N) + P(\overline{S_a(P)} \cap \overline{S_a(N)}) \quad \text{(by Assumption 2)} \]
\[ = 2q + R_a(N) + P(\overline{S_a(P)} \cap \overline{S_a(N)}) \]
\[ \leq 2q + R_a(N) + R(N) + R_a(P) + R_a(N) \]
\[ \leq 2q + R_a(P) + 2R_a(N). \] (10)

Following the proof of Lemma A.8 in [45], we prove the class-conditional variant of \( \text{Err}(P) \leq \mathcal{L}(P) \) in Theorem 3. We denote the class index as \( i \), where \( i \in \{1, \ldots, C\} \). Considering the case where \( L_{\omega}^{(i)}(P, G_{pl}) + R_a^{(i)}(P) \leq (c-1)\text{Err}(G_{pl}) \), firstly, we have \( q = (\beta/(c-1))\text{Err}(G_{pl}) \) from (A.8) in [45], and \( \beta = ((L_{\omega}^{(i)}(P, G_{pl}) + R_a^{(i)}(P))/\text{Err}(G_{pl})) \) from (A.7) in [45]. By Equation (10), we obtain
\[ \text{Err}_{i}(P) \leq 2q + R_a^{(i)}(P) + 2R_a^{(i)}(N) \]
\[ = \frac{2\beta}{c-1}\text{Err}(G_{pl}) + R_a^{(i)}(P) + 2R_a^{(i)}(N) \]
\[ \quad \text{using } q = \frac{\beta}{c-1}\text{Err}(G_{pl}) \]
\[ = \frac{2(L_{\omega}^{(i)}(P, G_{pl}) + R_a^{(i)}(P))}{c-1} + R_a^{(i)}(P) + 2R_a^{(i)}(N) \]
\[ \quad \text{using } \beta = \frac{L_{\omega}^{(i)}(P, G_{pl}) + R_a^{(i)}(P)}{\text{Err}(G_{pl})} \]
\[ = \frac{2}{c-1}L_{\omega}^{(i)}(P, G_{pl}) + \frac{c+1}{c-1}R_a^{(i)}(P) + 2R_a^{(i)}(N) \]
\[ = \mathcal{L}_{i}(P). \] (11)

Considering the case where \( L_{\omega}^{(i)}(P, G_{pl}) + R_a^{(i)}(P) > (c-1)\text{Err}(G_{pl}) \), according to (A.13)–(A.17) in [45], we have
\[ \text{Err}(P) \leq \frac{c+1}{c-1}L_{\omega}^{(i)}(P, G_{pl}) + \frac{2c}{c-1}R_a^{(i)}(P) - \text{Err}(G_{pl}) \]
\[ = \mathcal{L}_{i}(P). \] (12)

Then \( \text{Err}(P) \leq \mathcal{L}(P) \) follows simply by taking the expectation of \( \text{Err}(P) \leq \mathcal{L}(P) \) over all the classes.

Following the proof of Lemma 4.2 in [45], noting that \( G^* \) satisfies \( L_{0.1}(G^*, G_{pl}) = \text{Err}(G_{pl}) \) and \( R_a(G^*) \leq \mu, R_a(G^*) \leq \omega \) by Assumption 1, for any minimizer \( \hat{P} \) of \( (8) \), in the case where \( L_{0.1}(P, G_{pl}) + R_a(P) \leq (c-1)\text{Err}(G_{pl}) \), by (11), we have
\[ \text{Err}(\hat{P}) \leq \frac{2}{c-1}\text{Err}(G_{pl}) + \frac{c+1}{c-1}\mu + 2\omega = B_f \]
\[ \leq \frac{2}{c-1}\text{Err}(G_{pl}) + \frac{2c}{c-1}\mu = B_f \]
\[ \text{(using } \omega \leq \frac{1}{2}\mu \text{ in Assumption 1).} \]

Likewise, when \( L_{0.1}(P, G_{pl}) + R_a(P) > (c-1)\text{Err}(G_{pl}) \), by (12), we have
\[ \text{Err}(\hat{P}) \leq \frac{2}{c-1}\text{Err}(G_{pl}) + \frac{2c}{c-1}\mu = B_f = B_f. \]

Our results show that by minimizing (8) to bound \( \text{Err}(P) \), mutex-based consistency regularization (corresponding to Assumption 2) could help TPC improve the accuracy on pseudo-labels.

IV. EXPERIMENTS

In this section, we evaluate MutexMatch on five benchmark datasets, including CIFAR-10/100 [46], street view house numbers (SVHN) [47], self-taught learning 10 (STL-10) [48], mini-ImageNet [49] and Tiny-ImageNet. We also conduct ablation studies in Section V to investigate the efficacy of MutexMatch. Other experiments, e.g., the impact of consistency regularization on TNC, are shown in Section VI.

A. CIFAR-10, CIFAR-100, and SVHN

Firstly, we evaluate our method and baselines on three widely used SSL datasets as follows. 1) CIFAR-10/100, consisting of 10/100 classes, 50,000 images for training, 10,000 images for evaluation and 2) SVHN, consisting of more than 70,000 SVHN images from ten classes.

1) Baselines: We introduce recent SOTA SSL methods, i.e., SLA [13], CoMatch [35], FixMatch [8], and FixMatch with distribution alignment [34] to compare with MutexMatch. Moreover, we compare with recently-proposed SSL methods such as CUDA [37], MixMatch [33] and ReMixMatch [34].

2) Settings: For all experiments in MutexMatch, Wide ResNet (WRN) [44] is adopted as the backbone (WRN-28-2 for CIFAR-10 and SVHN, WRN-28-8 for CIFAR-100) following [8]. In our implementation, TNC is the same two-layer MLP as TPC. For fair comparison, we follow these baseline methods [8], [35] using SGD with a momentum of 0.9 and a weight decay of 0.0005 during training. Also, we train the model for 1024 epochs, using a learning rate of 0.03 with cosine decay schedule. For \( k \) in MutexMatch, two versions of the experiment are provided for comparison. One, we simply set \( k \) as the total amount of classes \( C \) (i.e., \( k = 10 \) for CIFAR-10/100, \( k = 100 \) for CIFAR-100), which is denoted as \( \text{MutexMatch} \) (this works as the setting for all the following experiments, unless noted otherwise); The other, we set \( k = 6 \) for CIFAR-10 and SVHN, \( k = 60 \) for
TABLE I
ACCURACY (%) ON CIFAR-10/100 AND SVHN AVERAGED ON FIVE RUNS. RESULTS WITH † ARE REPORTED IN FIXMATCH [8] AND RESULTS WITH ‡ ARE REPORTED IN SLA [13], WHILE RESULTS WITH ∗ ARE USING OUR OWN REIMPLEMENTATION. RESULTS OF REMIXMATCH AND COMATCH ARE ACHIEVED BY COMBINING THE DISTRIBUTION ALIGNMENT (DA) [34]. WE MARK OUT THE BEST AND SECOND BEST ACCURACY

| Method       | CIFAR-10 (10 labels) | CIFAR-100 (200 labels) | SVHN (40 labels) |
|--------------|----------------------|------------------------|------------------|
|              | 10 labels | 20 labels | 40 labels | 80 labels | 200 labels | 2500 labels | 40 labels | 250 labels |
| MixMatch [33]† | -       | -         | 52.46±11.50 | -       | -         | -         | -       | -         |
| UDA [37]†    | -       | -         | 70.95±5.93  | -       | -         | -         | -       | -         |
| ReMixMatch [34]† | -    | -         | 80.92±9.64  | -       | -         | -         | -       | -         |
| SLA [13]†    | -       | -         | 65.87±10.83 | 81.91±6.77 | 94.83±0.32 | 98.98±0.28 | -       | -         |
| FixMatch [8]† | 45.91±28.46 | 84.97±10.37 | 89.18±1.54  | 91.99±0.71 | 38.87±2.50  | 52.20±0.88 | 71.63±0.21 | 96.54±0.50 | 97.44±0.26 |
| CoMatch [35]† | 69.87±11.82 | 88.43±2.72  | 93.21±1.55  | 94.08±0.31 | 40.10±2.99  | 58.04±1.39 | 72.45±0.44 | 96.47±1.29 | 96.99±1.29 |
| MutexMatch   | 76.06±18.28† | 91.77±2.60  | 93.22±2.32  | 93.23±0.81 | 40.38±2.36  | 56.14±1.46 | 71.80±0.23 | 97.19±0.26 | 97.73±0.18 |
| MutexMatch (k = 0.6C) | 57.52±21.17 | 92.23±3.23  | 94.21±0.84  | 95.80±0.34 | 41.59±1.86  | 55.59±0.42 | 72.82±0.40 | 96.55±1.46 | 97.47±0.17 |

†This result is achieved by DA. Notice that CoMatch also integrates DA technique into training, and we find that DA will greatly improve performance when the amount of labels is very small (e.g., 10 labels). Default MutexMatch achieves 66.45±30.52% accuracy.

CIFAR-100, which is denoted as MutexMatch (k = 0.6C). For other hyper-parameters in MutexMatch, we set τ = 0.95, μ = 7, B = 64 for all experiments. In our method, RandAugment [12] is used for strong augmentation. Moreover, λ sep, λ p and λ n are set to 1 for simplicity. Lately, to reduce the influence of random data partition, we report the mean and standard deviation of accuracy on five different folds of labeled/unlabeled data.

3) Results: As shown in Table I, our approach obtains the highest accuracy under most settings. With only four labeled data per class, our method achieves an accuracy of 94.21 ± 0.84% on CIFAR-10, and 97.19 ± 0.26% on SVHN, yielding improvement over prior SSL results. Especially, we show the superiority of MutexMatch under the extremely label-scarce setting. For example, achieving an average accuracy of 92.23% on CIFAR-10 with 20 labels, 41.59% on CIFAR-100 with 200 labels. Usually fewer labels will result in the accumulation of more noisy pseudo-labels in training, yet, MutexMatch is able to utilize all unlabeled data while introducing little noise. In another word, our approach helps the model to be on track as early in the training as possible due to the benefit of a more reliable information gain, i.e., consistent regularization on complementary labels with high accuracy, which leads to a robust performance in the end.

Further, the comparisons with more baseline methods are reported on CIFAR-10/100 with more backbones (CNN-13 [53] is provided) and more available labels. The experimental settings are the same as that of CIFAR-100 mentioned above except that τ = 0.5. Table II shows MutexMatch is backbone independent, and it achieves performance improvement when more labels are given, outperforming most baselines. In Table II, the comparisons with UPS [11], a recently proposed pseudo-labeling based SSL method, are worth focusing on, because UPS also implicitly benefits from low-confidence samples, i.e., introducing negative CE (NCE) loss to learn the information of complementary label. Holding the same experimental setup as UPS, we show MutexMatch’s exploitation of low-confidence samples has advantages over UPS. More discussions on this exploitation can be found in Section V-A.

B. STL-10, Mini-ImageNet, and Tiny-ImageNet

STL-10 contains 5000 labeled from ten classes and 100 000 unlabeled images extracted from a similar but broader distribution. mini-ImageNet and Tiny-ImageNet are subsets of ImageNet [54], which contain 60 000/100 000 images evenly distributed across 100/200 classes, respectively. The challenge of the three datasets lies in the fact that unlabeled images contains out-of-distribution images or with a larger number of categories, which enables us to test the robustness of MutexMatch.

1) Settings: For STL-10, MutexMatch is evaluated on the five folds predefined in the original dataset, with each fold containing 1000 labeled data, and the averaged results are reported. Meanwhile, the results using total fivefold (i.e., 5000 training samples) as labeled data are also reported. For mini-ImageNet and Tiny-ImageNet, we report the results with 1000 labels and 5000 labels averaged on fivefold, respectively. As same as Section IV-A, firstly, we adopt WRN for the three datasets, i.e., WRN-37-2 for STL-10, WRN-28-8 (23M PARAMETERS) WE USED
for mini-ImageNet, and Tiny-ImageNet. In addition, following [35], we report results with a lighter backbone network: ResNet-18 [55] to provide more comprehensive comparisons with baseline methods. We use the same hyper-parameters and learning rate as in Section IV-A, and train the models using SGD with a momentum of 0.9 and a weight decay of 0.0005.

2) Results: As shown in Table III, MutexMatch’s performance advantage is maintained across all backbones. For example, with 1000 labels and ResNet-18 backbone, compared with CoMatch, MutexMatch achieves accuracy improvement from 79.80 ± 0.42% to 88.29 ± 1.00% on STL-10 and from 43.72 ± 0.58% to 50.36 ± 0.22% on mini-ImageNet. From Table III, we find our approach is more prominent in two scenarios: 1) when lighter models are employed (e.g., ResNet-18 in Table III), the models themselves have less capacity and are less able to learn from unlabeled data. And using large models is already able to learn enough information from high-confidence samples. Therefore, when using small models, the additional information that can be provided by learning low-confidence samples using complementary labels is more substantial and can help improve the performance of the model more effectively; and 2) when the dataset is harder (e.g., mini-ImageNet and Tiny-ImageNet in Table III), it is more difficult for the model to confidently predict the pseudo-label. After training for a considerable period of time, there are still most of the samples possessing low confidence levels. This means that the approaches of using a predefined threshold to filter pseudo-labels will waste a large number of unlabeled samples and eventually lead to poor performance. In contrast, MutexMatch does not spare a single unlabeled data and allows the model to learn as much information as possible.

V. ABLATION STUDY

Extensive ablation studies have been conducted to verify the effectiveness of MutexMatch. In the experiments mainly conducted on CIFAR-10 and SVHN using four labels per class, MutexMatch achieves 93.22 ± 2.52% and 97.19 ± 0.26% accuracy, respectively, using the default setting. In the following experiments (as well as in Section VI), we use the same settings as in Section IV-A (k = C) and average the results over multiple runs, where we keep the supervised loss as (2) and the positive consistency loss as (4). More additional results are available in Section VI.

### Table III

| Method     | STL-10 1000 labels | mini-ImageNet 5000 labels | Tiny-ImageNet 1000 labels | 5000 labels |
|------------|-------------------|--------------------------|--------------------------|------------|
| FixMatch   | 65.38 ± 0.42%     | 88.29 ± 1.00%            | 43.72 ± 0.58%            | 50.36 ± 0.22% |
| CoMatch [35] | 79.80 ± 0.38%     | 90.56 ± 0.22%            | 58.55 ± 0.20%            | 59.88 ± 0.39% |
| MutexMatch | **83.36 ± 0.22%** | **91.15 ± 0.17%**        | **48.04 ± 0.52%**        | **52.35 ± 0.21%** |

### Table IV

| λ_{ab1} | 0.1 | 0.2 | 0.5 | 1    | 2    | Ours |
|---------|-----|-----|-----|------|------|------|
| Accuracy | 96.54 | 92.89 | 89.26 | 89.16 | 88.40 | 87.57 | **97.19** |

**A. Utilization of Low-Confidence Samples**

We believe that the reason MutexMatch outperforms other earlier SSL algorithms is because TNC enables the model to learn from all unlabeled data. For example, in FixMatch, due to a predefined confidence threshold, the unlabeled samples with smaller confidence than this threshold will not participate in the training. Therefore, we use the three most intuitive ways to use all the unlabeled data for the ablation study. We first use TPC to compute prediction \( p^w = \hat{P}(\theta(x^w)) \) of weakly-augmented unlabeled data \( x^w \) and then:

1) We use \( \hat{p}^w = \arg \max (p_n^w) \) as a hard pseudo-label and enforce the CE loss against the model’s prediction \( p^t = \hat{P}(\theta(x^t)) \) of\( x^t \)

   \[
   \mathcal{L}_{ab1} = \frac{1}{|B|} \sum_{n=1}^{\mu B} \mathbb{1}(\max (p_n^w) < \tau) H(\hat{p}_n^w, p_n^t). \tag{13}
   \]

2) We use \( p^w \) as a soft pseudo-label and enforce the CE loss against \( p^t \)

   \[
   \mathcal{L}_{ab1} = \frac{1}{|B|} \sum_{n=1}^{\mu B} \mathbb{1}(\max (p_n^w) < \tau) H(p_n^w, p_n^t). \tag{14}
   \]

3) We first obtain the feature \( z_n^w = \theta(x_n^w) \) of \( x_n^w \) and the feature \( z_t^w = \theta(x_t^w) \) of \( x_t^w \) extracted by the feature extractor \( \theta \), and then we compute

   \[
   \mathcal{L}_{ab1} = \frac{1}{|B|} \sum_{n=1}^{\mu B} \mathbb{1}(\max (p_n^w) < \tau) E(z_n^w, z_t^w) \tag{15}
   \]

where \( E(p, q) \) denotes the mean squared loss between the two distributions \( p \) and \( q \).

The above loss minimized by experiments is simply \( \mathcal{L}_{sup} + \mathcal{L}_p + \mathcal{L}_{ab1} \). All models are trained on SVHN using four labels per class, and the results of all experiments are shown in Fig. 4, where \( FULL \) indicates the setting of 1), \( SOFT \) indicates the setting of 2) and \( MSE \) indicates the setting of 3), respectively. On this dataset, the default MutexMatch achieves an accuracy of 97.19 ± 0.26%, outperforming experiments with other settings. Other ways of using low-confidence samples could introduce more noisy pseudolabels, which are undetectable to consistency regularization. In addition, we conduct further experiments on SVHN for the setting of 1) where a weight \( \lambda_{ab1} \) is used to adjust the importance of \( \mathcal{L}_{ab1} \). As shown in Table IV, despite varying \( \lambda_{ab1} \) using regular consistency loss to learn low-confidence samples also impacts the performance of the original FixMatch (i.e., \( \lambda_{ab1} = 0 \)).

Meanwhile, we note that UPS [11] also utilizes complementary labels to exploit low-confidence samples. Rizve et al. [11] introduce NCE loss to learn the information of complementary...
labels. We use TNC to decouple the part where complementary labels are directly involved in training. And then we use soft labels to enforce consistency regularization on TNC so as TNC could help the model learn a better data representation of unlabeled data. For fair comparison, we first conduct the following experiments on CIFAR-10 with 40 labels, which is built on the paradigm of FixMatch. Then we provide a further discussion on the reason for not using NCE to learn low-confidence samples.

4) We remove TNC and select the positive label and the complementary label as in [11]. We introduce NCE to learn the information of complementary label (also in the same way as in [11]). This is equivalent to the following setting: keeping TPC, for the image \( x \), TPC outputs prediction \( p^w \) on the weakly-augmented version of \( x \). Then we select the positive label and the complementary label through the threshold \( \tau_p \) and \( \tau_n \) [11]. We enforce consistency regularization between the positive label and prediction \( p^* \) on strongly augmented version of \( x \), and then enforce NCE between the complementary label and \( p^w \) at the same time. We set the same \( \tau_n \) as [11], then we set \( \tau_p = 0.95 \) (i.e., \( \tau \) in MutexMatch), \( \tau_p = 0.7 \) (i.e., \( \tau_p \) in [11]) for two experiments, which are denoted as \( F \) w. NCE (1) and \( F \) w. NCE (2), respectively.

5) Instead of using NCE to learn complementary label directly, we enforce consistency between \( p^w \) and \( p^* \) by NCE [denoted as \( F \) w. NCE (3)].

As shown in Table V, we can see MutexMatch outperforms other settings utilizing samples with low-confidence in a simpler manner NCE. We analyze that the direct use of NCE to learn complementary label like [11] will lead to the “homogenization” of the learned information, i.e., the CE loss on the positive label and the NCE loss on the complementary labels share predictions of the same classifier. Moreover, the probability of the element (corresponding to complementary labels selected by the threshold) in prediction is very close to 0. The threshold \( \tau_n \) is small because [11] needs to ensure the accuracy of the complementary label. In this case, the impact of the loss item of NCE is very small. This way of using low-confidence samples may not be informative, hence not helpful for the model, which means that the final performance is still at the FixMatch level (even lower), as can be confirmed by the results of \( F \) w. NCE (1)\sim(3) shown in Table V. Even if [11] uses a very small threshold \( \tau_n \), the accuracy of the complementary labels selected in the same way as in [11] is lower than that of complementary labels output by TNC in MutexMatch. Considering that [11] treats complementary label as hard label and we treat that as a soft label, for a fair comparison, we provide the results of MutexMatch (Hard-Hard), which is described in Section VI-B (hard complementary labels are used for separate training of TNC and consistency regularization on TNC).

### B. Ablation on TPC and TNC

Since TPC is not well trained in the early stage of learning due to excessive noisy pseudo-labels, directly introducing complementary labels in this process is not helpful and even harmful. Therefore, we use consistency regularization on TNC to “decouple” the part where complementary labels are directly involved in training, and stop gradient on \( \theta \) during the separate training of TNC described in Section III-B.

For further discussion on the effectiveness of TPC and TNC, we consider removing TPC, TNC, and the stop-gradient setting, respectively. As mentioned above, our design of the following experiments is shown in Table VI: 1) explores what happens when we train TNC with complementary labels generated by TPC without stopping the gradient on \( \theta \); 2) indicates the results when we abandon the consistency on TNC. Restoring the back-propagation on \( \theta \) in (3) is a more reasonable setting, because if we set \( \lambda_n = 0 \), then TNC will not participate in training at all, i.e., the setting of (3); 4) represents that we abandon the consistency on TPC; and 5)\sim8) together present further ablation studies of each component in MutexMatch.

As shown in Table VI, the default MutexMatch achieves overwhelmingly superior performance compared with other settings. Apparently, if TNC participates in model training directly, it will cause the collapse of training, which means that the model will be severely affected by the learning of TNC, so there is no way to learn effective information about “what it is.” Both 1) and 2) show the superiority of using consistency regularization on TNC for learning complementary label. This “decoupling” ensures that TNC allows the model to learn a better data representation without adversely affecting the learning of TPC. 3) shows that TPC in this case fails to both get adequate training and complete the classification task, i.e., the training of TPC is similar to that using only labeled data (only by \( L_{sup} \)). Meanwhile, the training of TNC
TABLE VI
ABLATION STUDY ON SVHN WITH 40 LABELS. X IN THE FIFTH COLUMN INDICATES THAT WE RESTORE THE GRADIENT BACK-PROPAGATION ON FEATURE EXTRACTOR $\theta$ IN (3). NOTABLY, THE SETTINGS OF (III) AND (VII) ARE EQUIVALENT TO ORIGINAL FIXMATCH

| Ablation | $\lambda_p$ | $\lambda_n$ | $\lambda_{\text{rep}}$ | Stop gradient on $\theta$ | Accuracy (%) |
|----------|-------------|-------------|---------------------|-----------------|--------------|
| (i)      | 1           | 1           | 1                   | $\checkmark$    | 15.84        |
| (ii)     | 1           | 0           | 1                   | $\checkmark$    | 14.79        |
| (iii)    | 1           | 0           | 1                   | $\checkmark$    | 96.54        |
| (iv)     | 0           | 1           | 1                   | $\checkmark$    | 17.71        |
| (v)      | 1           | 1           | 0                   | $\checkmark$    | 83.24        |
| (vi)     | 0           | 1           | 0                   | $\checkmark$    | 16.23        |
| (vii)    | 1           | 0           | 0                   | $\checkmark$    | 96.54        |
| (viii)   | 0           | 0           | 0                   | $\checkmark$    | 15.55        |

MutexMatch 1 1 1 $\checkmark$ 97.19

Fig. 5. Ablation study on $k$ controlling the consistency intensity on TNC. (a) Results on CIFAR-10 with 40 labels. (b) Results on CIFAR-100 with 400 labels. The results are averaged on multiple runs.

is closely related to that of TPC. In this case, TNC has not been well trained either. Finally, the combination of 4)–6) together proves the necessity of each component in MutexMatch.

C. Consistency Regularization on TNC

Firstly, we conduct ablation studies on hyper-parameter $k$ used for the intensity control of consistency on TNC. As the results in Fig. 5 indicate, an appropriate $k$ will be more conducive to learning from low-confidence samples. Excessive consistency regularization may cause the model to overly narrow the overlap of the dissimilarity degree of low-confidence samples, consequently worsening the performance of the model. Whereas, too weak consistency regularization would provide too little guidance information for the model. We note that MutexMatch is more sensitive to $k$ on CIFAR-10 than on CIFAR-100. This is easy to understand because CIFAR-10 is too simple and different settings are more likely to produce performance fluctuations on it. Actually, using the default MutexMatch (i.e., $k = C$) is sufficient to achieve good enough performance. Moreover, we conduct further ablation studies on consistency regularization on TNC as follows.

1) We enforce consistency regularization on TNC over all samples, instead of enforcing only on low-confidence samples like the default setting.

2) We use hard labels for the consistency regularization on TNC, i.e., we set the component $r_{n(i)}^{w} = 1$ in $r_n^{w}$ when $g_{n,(i)} = 1$ for (5).

As shown in Table VII, the result of 1) confirms our claim that the consistency on TNC is more suitable for low-confidence samples. For high-confidence samples, it is sufficient to use consistency regularization on TPC to learn the guidance information provided by its pseudo-labels, while consistency regularization on complementary labels is superfluous. The result of 2) proves that the consistency regularization on the TNC using hard labels is too strong, and since it does not contain the discriminative information of the degree of dissimilarity, it is not beneficial for the model performance. More discussions of consistency regularization on TNC can be found in Section VI-C. In addition, more ablation studies on learning scheme of TNC can be found in Section VI-B.

D. Hyper-Parameters

For MutexMatch, we should be very cautious about the choice of $\tau$, because different $\tau$ will lead to the division of high and low-confidence portions, which will in turn affect the impact of the mutex-based consistency regularization on the model. We vary $\tau$ to verify the sensitivity of MutexMatch to this hyperparameter. As shown in Table VIII, MutexMatch needs to select appropriate $\tau$ to divide confidence portions. We note that $\tau$ has a greater impact on performance. The more labels are available, the less confirmation bias will be when using TPC directly for classification, so the portion of TPC in mutex-based consistency regularization can be used...
E. Learning Rate and Learning Rate Schedule

We note that learning rate and learning rate schedule are very important for MutexMatch. In this section, we conduct additional ablation experiments for both. Following [56], recent works [8], [35] use a cosine learning rate decay to achieve best performance. Likewise, as shown in Table IX, we find that MutexMatch achieves better results with cosine learning rate decay on CIFAR-10. e.g., using 40 labels, MutexMatch training with cosine learning rate decay outperforms that training without learning rate decay by 1.04%.

VI. FURTHER DISCUSSIONS ON TNC

A. Effectiveness Analysis of TNC

Compared with other baseline methods shown in Table I, MutexMatch performs better on CIFAR-10 with extremely scarce labels. We believe that confirmation bias [7] leads to the poor performance of other methods. Fewer labels will introduce more noisy pseudo-labeled examples to participate in the learning process. Nevertheless, MutexMatch utilizes the unlabeled samples with low confidence, in a more reasonable manner by TNC, introducing few noisy pseudo-lables. As shown in Table X, our experiments on CIFAR-10 show that MutexMatch produces more accurate pseudo-labels than FixMatch [8], especially when labeled samples are scarce.

Ideally, we think that for every class, unlike pseudo-labels from TPC, the distribution of complementary pseudo-labels from TNC should be evenly dispersed or diverse, so MutexMatch can learn more multiclass information as much as possible. For example, considering CIFAR-100 (containing 100 classes), if the model only learns that the class “monkey” is not class “truck,” then its revenue from the complementary labels outputted by TNC will not be very informative.

B. Ablation Study on the Learning Scheme of TNC

The learning of TNC is very important for MutexMatch. In default MutexMatch, we use hard complementary pseudo-label \( \hat{q}^w = \arg\min(p^w) \) to train TNC separately while stopping gradient back-propagation on the feature extractor, and enforce consistency regularization against soft pseudo-label \( r^w = \hat{N}(\theta(x^w)) \) in the low-confidence portion of unlabeled data. In order to validate the effectiveness of learning scheme of TNC in MutexMatch, we use three modified learning schemes for experiments as follows.

1) We use hard pseudo-label \( \hat{q}^w = \arg\min(p^w) \) to train TNC separately while stopping gradient back-propagation on \( \theta \), and enforce consistency regularization against hard complementary pseudo-label

\[
\mathcal{L}_{sep} = \frac{1}{\mu B} \sum_{n=1}^{\mu B} H(\hat{q}^w_n, r^w_n) \quad (16)
\]

\[
\mathcal{L}_{ab2} = \frac{1}{\mu B} \sum_{n=1}^{\mu B} \indic(\max(p^w_n) < \tau) H(r^w_n, r^w_n) \quad (17)
\]

where \( r^w = \arg\max(r^w) \) and \( r^w = \hat{N}(\theta(x^w)) \).

2) We use hard complementary pseudo-label \( \hat{r}^w \), which is generated via randomly selecting the class without the highest confidence from \( p^w \) (just like the standard complementary label selection) to train TNC separately while stopping gradient on \( \theta \), and enforce consistency regularization against soft complementary pseudo-label

\[
\mathcal{L}_{sep} = \frac{1}{\mu B} \sum_{n=1}^{\mu B} H(\hat{r}^w_n, r^w_n) \quad (18)
\]

\[
\mathcal{L}_{ab2} = \frac{1}{\mu B} \sum_{n=1}^{\mu B} \indic(\max(p^w_n) < \tau) H(r^w_n, r^w_n). \quad (19)
\]

3) We remove the separate training part of TNC. The complementary pseudo-label for TNC is obtained directly by \( q^w = \text{Norm}(1 - p^w) \) where 1 is a all-one vector.
and \textbf{Norm}(·) is operation normalizing $q^w$ into interval $[0, 1]$. We enforce consistency against soft complementary pseudo-label

$$
L_{ab2} = \frac{1}{\mu B} \sum_{n=1}^{\mu B} \mathbb{1}\left(\max\left(p^w_n\right) < \tau\right)H\left(q^w_n, r^s_n\right).
$$

The losses minimized by experiments is simply $L_{sup} + L_p + L_{sep} + L_{ab2}$ in 1), 2) and $L_{sup} + L_p + L_{ab2}$ in 3).

All models are trained on CIFAR-10 using four labels per class, and the results of all experiments are listed in Fig. 7. In the figure, the Hard-Hard indicates setting of 1); the Rand-Soft indicates setting of 2); and the Rev-Norm indicates setting of 3). The default MutexMatch achieves an accuracy of $93.22 \pm 2.52\%$, outperforming all other settings. MutexMatch uses hard labels for separate training of TNC to ensure a robust classifier that could output more accurate complementary pseudolabels, and uses soft labels to participate in consistency regularization on TNC for informative guidance.

C. Analysis of Consistency Regularization on TNC

Given two augmented variants derived from the same unlabeled instance, we claim that their predictions of the degree of dissimilarity could share some overlap, which can be achieved by encouraging the class probability distributions (i.e., soft-labels) of their complementary predictions (i.e., the TNC’s outputs) to be consistent. Under the help of an independent training process of TNC, the model can be more confident on “what it is not.” As a result, such prediction consistency on TNC can effectively decrease the false-negative probability in TPC’s predictions, e.g., given an instance of class “dog,” the independent training of TNC can generate an accurate complementary prediction with extremely low probability of class “dog.” Then encouraging a similar prediction on its strongly-augmented variant can help the model to learn more discriminative features, which can in turn affect the TPC’s prediction (TNC and TPC share the same backbone) in a way that the False-Negative probabilities (to be predicted as other classes except “dog”) can be effectively lowered. Consequently, the true-positive probabilities of TPC’s
predictions are enlarged. We construct an experiment on CIFAR-10 with 40 labels to verify our findings. We denote MutexMatch without consistency regularization on TNC as $M \text{ wo. c}$ (i.e., MutexMatch degenerates to FixMatch). We take the predictions from MutexMatch and $M \text{ wo. c}$ as an example for comparison. In order to better display the results, we show the results at 100 epochs. As shown in Fig. 8(a), given the unlabeled instances belonging to “dog,” the MutexMatch’s average probability of “dog” component in prediction vectors is higher than that of $M \text{ wo. c}$. We can also obtain similar findings on other different classes, as shown in Fig. 8(b). Such observations demonstrate enforcing prediction consistency on TNC can help the model lower the false-negative probabilities, which in turn improve the true-positive probabilities in the TPC’s prediction vectors.

VII. CONCLUSION

In this article, we propose MutexMatch, a novel SSL algorithm using a mutex-based consistency regularization derived from two distinct classifiers. One is to predict “what it is” and the other is to predict “what it is not.” MutexMatch can achieve superior performance on various SSL benchmarks. Last but not least, we validate that low-confidence samples can also be well utilized in training in a novel way. We believe this utilization of low-confidence samples can be borrowed for other semi-supervised tasks, e.g., segmentation and detection.

REFERENCES

[1] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” Synth. Lect. Artif. Intell. Mach. Learn., vol. 3, no. 1, pp. 1–130, 2009.
[2] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-supervised learning,” IEEE Trans. Neural Netw., vol. 20, no. 3, p. 542, Mar. 2009.
[3] X. J. Zhu, “Semi-supervised learning literature survey,” Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, 2005.
[4] J. E. Van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” Master’s thesis, vol. 109, no. 2, pp. 373–440, 2020.
[5] S. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” 2016, arXiv:1610.02242.
[6] A. Tarvainen and H. Valpola, “Mean teachers are better role models!” Master’s thesis, vol. 109, no. 2, pp. 373–440, 2020.
[7] S. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” 2016, arXiv:1610.02242.
[8] M. Rizve, Y. S. Rawat, and M. Shah, “In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning,” in Proc. Int. Conf. Learn. Represent., 2021, pp. 1–10.
[9] D. Z. Cai, B. Zoph, I. Shlens, and Q. Le, “RA: Reciprocal distribution alignment for robust semi-supervised learning,” in Proc. Adv. Conf. Comput. Vis., 2022, pp. 533–549.
[10] J. Jeong, S. Lee, J. Kim, and N. Kwak, “Consistency-based semi-supervised learning for object detection,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 1–10.
[11] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, and T. Pfister, “A simple semi-supervised learning framework for object detection,” 2020, arXiv:2005.04757.
[12] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, “Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1742–1750.
[13] Y. Shi et al., “Inconsistency-aware uncertainty estimation for semi-supervised medical image segmentation,” IEEE Trans. Med. Imag., vol. 41, no. 3, pp. 608–620, Mar. 2022.
[14] L. Yang, W. Zhao, L. Qi, Y. Shi, and Y. Gao, “ST++: Make self-training work better for semi-supervised semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 4268–4277.
[15] X. Chen, M. Harandi, X. Jin, and Y. Yang, “Semi-supervised domain adaptation via asymmetric joint distribution matching,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 12, pp. 5708–5722, Dec. 2021.
[16] G. Wang, Q. Hu, Y. Yang, J. Cheng, and Z.-G. Hou, “Adversarial binary mutual learning for semi-supervised deep hashing,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 8, pp. 4110–4124, Aug. 2022.
[17] J. Qi, L. Wang, J. Huo, Y. Shi, and Y. Gao, “Progressive cross-camera soft-label learning for semi-supervised person re-identification,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 9, pp. 2815–2829, Sep. 2020.
[18] R. Fierionmonte, S. Scardapane, A. Uncini, and M. Panella, “Fully decentralized semi-supervised learning via privacy-preserving matrix completion,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 11, pp. 2699–2711, Nov. 2016.
[19] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, “Deep convolutional inverse graphics network,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 1–9.
[20] B. Paige et al., “Learning disentangled representations with semi-supervised deep generative models,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 1–11.
[21] Y. Li, Q. Pan, S. Wang, H. Peng, T. Yang, and E. Cambria, “Disentangled variational auto-encoder for semi-supervised learning,” Inf. Sci., vol. 482, pp. 73–85, May 2019.
[22] T. Joy, S. M. Schmon, P. H. S. Torr, N. Siddharth, and T. Rainforth, “Capturing label characteristics in VAEs,” 2020, arXiv:2006.10102.
[23] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, “Good semi-supervised learning that requires a bad GAN,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 1–11.
[24] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 1–9.
[25] G.-J. Qi, L. Zhang, H. Hu, M. Eidraki, J. Wang, and X.-S. Hua, “Global versus localized generative adversarial nets,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 1–9.
[26] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, “Improving the improved training of Wasserstein GANs: A consistency term and its dual effect,” 2018, arXiv:1803.01541.
[27] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 1–11.
[28] D. Berthelot et al., “Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring,” in Int. Conf. Learn. Represent., 2020, pp. 1–13.
[29] J. Li, C. Xiong, and S. C. Hoi, “Comatch: Semi-supervised learning with contrastive graph regularization,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Oct. 2021, pp. 9475–9484.
[30] P. Bachman, O. Alsharif, and D. Precup, “Learning with pseudo-ensembles,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 1–9.
[31] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, “Unsupervised data augmentation for consistency training,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 1–13.
[32] G.-J. Qi, Z. Zhao, L. Qi, L. Zhou, L. Wang, and Y. Shi, “Improving barely supervised learning by discriminating unlabeled samples with super-class,” in Proc. Adv. Neural Inf. Process. Syst., 2022, pp. 1–12.
Y. Grandvalet and Y. Bengio, “Semi-supervised learning by entropy minimization,” in Proc. Adv. Neural Inf. Process. Syst., 2005, pp. 1–8.

T. Ishida, G. Niu, W. Hu, and M. Sugiyama, “Learning from complementary labels,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 1–11.

T. Ishida, G. Niu, A. K. Menon, and M. Sugiyama, “Complementary-label learning for arbitrary losses and models,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 2971–2980.

Y. Kim, J. Yin, J. Yun, and J. Kim, “NLNL: Negative learning for noisy labels,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1–10.

S. Zagoruyko and N. Komodakis, “Wide residual networks,” in Proc. Int. Conf. Learn. Represent., 2016, pp. 1–15.

A. Krizhevsky et al., “An analysis of single-layer networks in unsupervised feature learning,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2011, pp. 215–223.

O. Vinyals et al., “Matching networks for one shot learning,” in Proc. Int. Conf. Artif. Intell. Statist., 2016, pp. 1–9.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, “Reading digits in natural images with unsupervised feature learning,” in Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn., 2011, pp. 1–9.

A. Coates, A. Y. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in Proc. Int. Conf. Artif. Intell. Statist., 2011, pp. 215–223.

A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, “Interpolation consistency training for semi-supervised learning,” in Proc. 28th Int. Joint Conf. Artif. Intell., Aug. 2019, pp. 1–17.

Z. Ke, D. Wang, Q. Yan, J. Ren, and R. Lau, “Dual student: Breaking the limits of the teacher in semi-supervised learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1–9.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2009, pp. 248–255.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” in Int. Conf. Learn. Represent., 2017, pp. 1–16.

Yue Duan received the B.Sc. degree from the School of Computer Science and Technology, Harbin Institute of Technology, Weihai, China, in 2017. He is currently pursuing the M.Sc. degree with the Department of Computer Science, Nanjing University, Nanjing, China. His research interests include machine learning and computer vision.

Zhen Zhao is currently pursuing the Ph.D. degree in electrical and information engineering with The University of Sydney, Sydney, NSW, Australia. His current research interests include semi/self-supervised learning algorithms in computer vision and their applications in medical image analysis.

Lei Qi is currently an Assistant Researcher with the School of Computer Science and Engineering, Southeast University, Nanjing, China. His current research interests include some ML methods, such as domain adaptation, semi-supervised learning, unsupervised learning, and meta-learning. For applications, he mainly focuses on person reidentification and image segmentation.

Lei Wang (Senior Member, IEEE) is currently a Professor with the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia. He has published more than 120 peer-reviewed articles, including those in highly regarded journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, International Journal of Computer Vision (IJCV), Computer Vision and Pattern Recognition Conference (CVPR), International Conference on Computer Vision (ICCV), and European Conference on Computer Vision (ECCV). His research interests include machine learning, pattern recognition, and computer vision.

Luping Zhou (Senior Member, IEEE) received the Ph.D. degree from The Australian National University, Canberra, ACT, Australia, in 2010. She is currently an Associate Professor with the School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW, Australia. Her current research interests include machine learning, computer vision, and medical image analysis. Dr. Zhou was a recipient of the Australian Research Council Discovery Early Career Researcher Award (DECRA Award) in 2015.

Yang Gao (Senior Member, IEEE) is currently a Professor and the Deputy Director with the Department of Computer Science and Technology, Nanjing University, Nanjing, China. He is currently directing the Reasoning and Learning Research Group, Nanjing University. He has published more than 100 papers in top-tiered conferences and journals. His current research interests include artificial intelligence and machine learning. Prof. Gao serves as the program chair and the area chair for many international conferences.