MUTEXMATCH: SEMI-SUPERVISED LEARNING WITH MUTEX-BASED CONSISTENCY REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The core issue in semi-supervised learning (SSL) lies in how to effectively leverage unlabeled data, whereas most existing methods usually concentrate on the utilization of high-confidence samples yet seldom fully explore the usage of low-confidence samples. Early SSL methods mostly require low-confidence samples to optimize the same loss function as high-confidence samples, but this setting might largely challenge the low-confidence samples especially at the early training stage. In this paper, we aim to utilize low-confidence samples in a novel way, which is realized by our proposed mutex-based consistency regularization, namely MutexMatch. To be specific, the high-confidence samples are required to exactly predict "What it is" by conventional True-Positive Classifier, while the low-confidence samples, for a much simpler goal, are employed to predict "What it is not" by True-Negative Classifier with ease. In this way, we not only mitigate the pseudo-labeling errors but also make full use of the low-confidence unlabeled data in the training stage. The proposed MutexMatch achieves superior performance on multiple benchmark datasets, *i.e.*, CIFAR-10, CIFAR-100, SVHN, and STL-10. Particularly, our method shows further superiority under few quantities of labeled data, *e.g.*, 91.77% accuracy with only 20 labeled data on CIFAR-10.

1 INTRODUCTION

Aiming to escape from time-consuming and laborious labeling tasks, semi-supervised learning (SSL) (Chapelle et al., 2009; Zhu, 2017) has been a longstanding yet important direction to leverage a large quantity of unlabeled data along with few labeled data during training. Recent SSL models could be categorized into *consistency regularization* based or *entropy minimization* based methods where the utilization of unlabeled data is crucial in both. In particular, consistency regularization based methods like (Laine & Aila, 2016; Tarvainen & Valpola, 2017) intend to utilize all unlabeled data together with the supervision on labeled data, which is at the risk of strong confirmation bias. Although recent holistic methods such as Sohn et al. (2020) realize consistency regularization by combining entropy minimization via pseudo labeling, a fatal limitation is that they set a confidence threshold to control whether unlabeled data should participate in training, preventing low-confidence unlabeled data from being effectively involved. Different from consistency based methods, recent entropy minimization based methods Rizve et al. (2021) employ pseudo labeling to iteratively incorporate a part of low-confidence samples into training process. However, in this way, in addition to possible error accumulation, some low-confidence unlabeled samples are still neglected. In a nutshell, the waste of unlabeled samples with low confidence causes the model difficult to learn the potential pattern from all unlabeled data, which might deteriorate the final performance.

Being aware of the aforementioned limitations, we try to answer — if we could treat these low-confidence unlabeled samples in a novel way? As shown in Figure 1, imaging a low-confidence sample with its actual label of a "horse", it might be hard for a trained model to propose an accurate prediction, *i.e.*, *"it is a horse"*. On the contrary, it can be much easier for the model to "*guess what it is not*, *e.g.*, *it is not a cat*". This drives us to consider a straightforward yet feasible direction ***— for low-confidence samples, can we design a paradigm to exclude "what it is not" to benefit the learning of "what it is"***. Intuitively, by introducing this paradigm, the space of searching the optimal classifier could be largely reduced since the most impossible classes are initially excluded.

That is to say, for a low-confidence image, it is unnecessary to get a certain class, and its complementary pseudo-label is easier to obtain. Thus, we could learn less error information when using unlabeled data with low confidence.

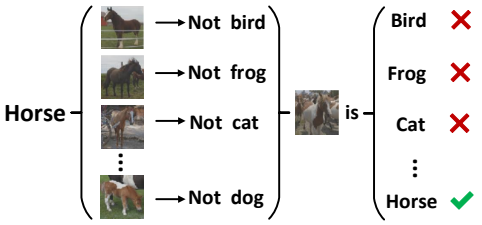


Figure 1: Graphical explanations of the "Exclusion Method" for classification task. For an image, the proposed method excludes some wrong classes via different images of the same class, so as to eliminate the wrong answers.

As aforementioned, to leverage all unlabeled data in a novel way, we propose **MutexMatch**, a new framework of SSL using mutex-based consistency regularization. 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 feature representation of unlabeled data from a mutex perspective. An improvement of MutexMatch compared with the existing methods is that it allows low-confidence unlabeled samples to participate in training by optimizing a much simpler objective compared with that of high-confidence unlabeled samples. Inspired by FixMatch (Sohn et al., 2020), the weakly-augmented unlabeled samples are used to generate pseudo-labels, and we use RandAugument (Cubuk et al., 2020) for strong augmentation. We set a threshold to control the high-confidence portion and low-confidence portion of pseudo-labels. In high-confidence portion and low-confidence portion, we enforce the consistency regularization on the output of TPC and TNC, respectively. A diagram of MutexMatch is shown in Figure 2.

In this work, the key contributions include three aspects: (1) We propose mutex-based consistency regularization for SSL, which can make full use of unlabeled data in a more effective way; (2) We use two classifiers (TPC and TNC) to construct MutexMatch, a novel framework using pseudo-label and complementary label to learn an informative representation of unlabeled samples; (3) By exploiting all unlabeled data, we can obtain better classification results in a label-scarce setting than recently-proposed SSL algorithm. For example, on the most commonly-studied SSL benchmark CIFAR-10, the accuracy of MutexMatch using only 20 labels can reach up to 91.77$\pm$2.60%.

2 Related Work

Consistency regularization is a significant branch of recent state-of-the-art (SOTA) SSL methods, which is proposed in Bachman et al. (2014). Such methods encourage the classifier to output same class probability distribution after different versions of augmentation for the same unlabeled data. Generally, the consistency regularization based models are trained with unlabeled data using the loss function: $\|p(y|\alpha(x) - p(y|\alpha(x)))\|_2^2$, where x is the input image and $\alpha(\cdot)$ is an kind of transformation that does not change the image label. Particularly, $\alpha(\cdot)$ can adopt different augmentation methods, *e.g.*, Mixup (Zhang et al., 2017) in Berthelot et al. (2019), RandAugment (Sohn et al., 2020) in Sohn et al. (2020) and CTAugment in Berthelot et al. (2020). Laine & Aila (2016) enforces a loss of consistency on the predictions of two augmented variants of unlabeled data. In Tarvainen & Valpola (2017), a teacher model is maintained to generate more stable targets for unlabeled data, and the mean squared error is used to encourage same predictions of the student and teacher models. Xie et al. (2020) adopts automatic augmentation for data perturbation and enforces a loss of consistency by the KL divergence. Recently, some holistic methods (Sohn et al., 2020; Berthelot et al., 2020) have been proposed to combine consistency regularization with pseudo-labeling for better SSL performance. Differently, in MutexMatch, in addition to enforcing prediction consistency on TPC, we propose a novel mutex-based consistency to effectively leverage all unlabeled samples.

Pseudo-labeling is widely leveraged for entropy minimization by constructing one-hot labels from predictions of unlabeled data with high-confidence and makes use of them based on cross-entropy loss (Lee et al., 2013; Shi et al., 2018; Sohn et al., 2020; Xie et al., 2020). However, these methods have a significant limitation, *i.e.*, using confidence thresholds to select pseudo-labels results in that all unlabeled data is not sufficiently exploited. Recent pseudo-labeling based method (Rizve et al., 2021) proposes an uncertainty-aware pseudo-label selection framework to use both high and low-confidence samples. However, it introduces two thresholds to control the pseudo-label generation, thus some unlabeled samples are still not utilized.

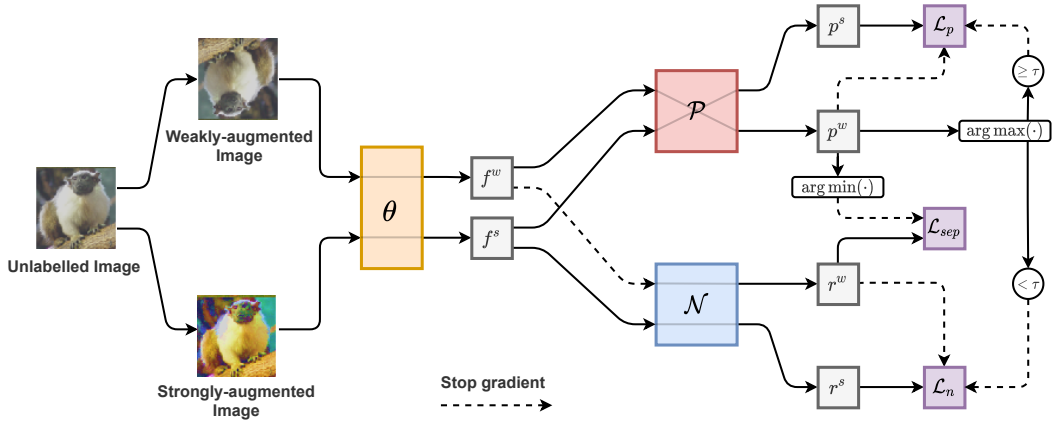


Figure 2: Diagram of the proposed MutexMatch. Given a batch of unlabeled samples, TPC $\mathcal{P}$ uses their weakly-augmented variants to generate pseudo-labels. Then we adopt the classes with the lowest confidence as the complementary labels to train TNC $\mathcal{N}$ separately. Meanwhile, TPC and TNC are used for mutex-based consistency regularization in the high and low-confidence portion of TPC's predictions respectively. f denotes output features and p, r denote predictions of TPC and TNC. Superscripts w and s represent corresponding outputs for the weakly-augmented variant and strongly-augmented variant, respectively.

Complementary label is used to help the model learn which class the input image does not belong to (Ishida et al., 2017; 2018; Yu et al., 2018). Considering a c-class classification task, we denote $x \in \mathcal{X}$ as an input image and $y \in \mathcal{Y} = \{1, ..., c\}$ as its label. Complementary label $\overline{y}$ is generated by select from $\mathcal{Y} \backslash \{y\}$ at random. Inspired by Rizve et al. (2021) and Kim et al. (2019)[1], in MutexMatch, we design a novel way (detailed in Section 3.2) to propose complementary labels, so as to ensure their effectiveness in semi supervised learning. Experiments about using standard complementary label selections are discussed in Section 5.2.

3 MutexMatch

3.1 Overview

Different from existing SSL approaches, in addition to a feature extractor $\theta(\cdot)$, MutexMatch jointly trains two distinct classifiers, a True-Positive Classifier (TPC) $\mathcal{P}(\cdot)$ and a True-Negative Classifier (TNC) $\mathcal{N}(\cdot)$. To be specific, TPC is used to predict which class the instance belongs to (*i.e.*, true positive), while TNC is employed to indicate which class the instance is not (*i.e.*, true negative). To mitigate pseudo-labeling errors, a pre-defined high-confidence threshold τ 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.

In a mini-batch, we have B labeled data $\mathcal{X} = \{(x_b^{lb}, y_b^{lb})\}_{b=1}^{B}$ and μB unlabeled data $\mathcal{U} = \{(x_b^{ulb}, y_b^{ulb})\}_{b=1}^{\mu B}$, where μ represents the relative size of $\mathcal{X}$ and $\mathcal{U}$. Following (Sohn et al., 2020), we perform weak and strong augmentations for data perturbations, denoted by $\alpha_w(\cdot)$ and $\alpha_s(\cdot)$, respectively. Given weakly-augmented instance x^w and strongly-augmented instance x^s, MutexMatch simultaneously optimizes four losses: the supervised loss $\mathcal{L}_{sup}$, the separated negative loss $\mathcal{L}_{sep}$, the positive consistency loss $\mathcal{L}_p$ and the negative consistency loss $\mathcal{L}_n$. In summary, the total loss is

$$\mathcal{L} = \mathcal{L}_{sup} + \lambda_{sep}\mathcal{L}_{sep} + \lambda_p\mathcal{L}_p + \lambda_n\mathcal{L}_n, \tag{1}$$

where λ_{sep}, λ_p and λ_n are scalar hyper-parameters to adjust the relative importance of corresponding losses. The supervised loss $\mathcal{L}_{sup}$ is simply defined as the cross-entropy between y^{lb} and the

[1]In Kim et al. (2019), complementary label based negative learning shows great potential for noisy labels. Through our method using complementary label, we also achieve amazing performance under the more difficult setting of semi-supervised with noisy labels and the results can be found in Section C in Appendix.

predictions of TPC on labeled data x^{lb}, calculated as follows:

$$\mathcal{L}_{sup} = \frac{1}{B}\sum_{n=1}^{B} H(y_n^{lb}, \mathcal{P}(\theta(\alpha_w(x_n^{lb})))), \tag{2}$$

where $H(p, q)$ denotes the standard cross-entropy loss between distribution q and p.

3.2 True-Negative Classifier

In multi-class classification tasks, for a specific instance, it is easier to predict which class it does not belong to than to know which class it exactly is. For example, given an image of airplane in CIFAR-10, we can predict which class it does not belong to with a probability 90% at random, whereas we only have the probability of 10% to correctly predict it is an airplane. To this end, we design a True-Negative Classifier to predict which class it is not. Compared to TPC, it is much easier to obtain correct labels for TNC. Thus we exploit TNC to provide more guidance information on unlabeled data. We then propose a mutex-based prediction consistency on TPC and TNC to make full use of unlabeled data, which is described in Section 3.3. The high-level training process of TNC is shown in Figure 3. Unlike the standard complementary label generations (Ishida et al., 2017; Yu et al., 2018), we use the class with the lowest confidence in TPC's predictions as the complementary label to train TNC. The training loss $\mathcal{L}_{sep}$ can be calculated as

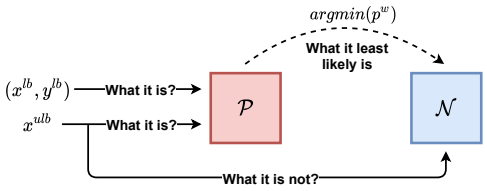


Figure 3: Training of TNC

$$\mathcal{L}_{sep} = \frac{1}{\mu B}\sum_{n=1}^{\mu B} H(\arg\min(\mathcal{P}(\theta(x_n^w)), \mathcal{N}(\hat{\theta}(x_n^w)))), \tag{3}$$

where $\hat{\theta}$ represents that θ is considered constant for the generation of this loss, *i.e.*, stop back-propagating gradients. 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. We extensively investigate the effectiveness of TNC in Section 4.3.

3.3 Mutex-based Consistency Regularization

In recent consistency-regularization based SSL methods, only samples with high-confidence predictions are leveraged to train models. However, it could lead to inefficient utilization of unlabeled data, especially at the early stage of the training process. Differently, MutexMatch can also effectively exploit low-confidence unlabeled samples via introducing a novel mutex-based consistency regularization. A high-confidence threshold τ on TPC's predictions is defined to split the unlabeled samples into two portions with mutex confidence intervals, *i.e.*, the high-confidence one ($> \tau$) and the low-confidence one ($\leq \tau$). 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, because it is difficult for us to obtain its real class information.

On the one hand, we use weakly-augmented example x^w to generate pseduo-labels from TPC and enforce positive consistency against its corresponding strongly-augmented variant x^s. We can then obtain their predictions, $p^w = \mathcal{P}(\theta(x^w))$ and $p^s = \mathcal{P}(\theta(x^s))$. Let $\hat{p}^w = \arg\max(p^w)$, such consistency can be achieved by minimizing the loss $\mathcal{L}_p$:

$$\mathcal{L}_p = \frac{1}{\mu B}\sum_{n=1}^{\mu B} \mathbb{1}(\max(p_n^w) \geq \tau) H(\hat{p}_n^w, p_n^s), \tag{4}$$

where $\mathbb{1}(\max(p^w) > \tau)$ retains the predictions whose maximum probabilities are larger than τ. On the other hand, for these low-confidence samples, we enforce consistency regularization against TNC's predictions by minimizing the $\mathcal{L}_n$:

$$\mathcal{L}_n = \frac{1}{\mu B}\sum_{n=1}^{\mu B} \mathbb{1}(\max(p_n^w) < \tau) H(r_n^w, r_n^s), \tag{5}$$

Table 1: Accuracy for CIFAR-10, CIFAR100 and SVHN averaged on 5 different folds. Results with * were reported in CoMatch (Li et al., 2020), while results with † are using our own reimplementation. Other results were reported in FixMatch (Sohn et al., 2020). Results with DA are achieved by combining the *distribution alignment technique* (Berthelot et al., 2020).

Method	CIFAR-10				CIFAR-100			SVHN	
	10 labels	20 labels	40 labels	80 labels	200 labels	400 labels	2500 labels	40 labels	250 labels
UDA	-	-	70.95±5.93	-	-	40.72±0.88	66.87±0.22	47.37±20.51	94.31±2.76
MixMatch	-	27.84±10.63*	52.46±11.50	80.79±1.28*	-	33.39±1.32	60.06±0.37	57.45±14.53	96.02±0.23
ReMixMatch w. DA	-	-	80.90±9.64	-	-	55.72±2.06	**72.57±0.31**	96.66±0.20	97.08±0.48
FixMatch	64.08±20.33†	82.32±9.77*	88.61±3.35	92.06±0.88*	38.87±2.50†	51.15±1.75	71.71±0.11	96.04±2.17	97.52±0.38
FixMatch w. DA*	-	83.81±9.35	86.98±3.40	92.29±0.86	-	-	-	-	-
CoMatch*	69.87±11.82†	87.67±8.47	93.09±1.39	93.97±0.62	-	-	-	96.47 ± 1.29†	**97.75 ± 0.19**†
MutexMatch	**78.73±11.21**	**91.77±2.60**	**93.49±0.22**	**94.34±0.81**	**40.38±2.36**	**56.14±1.46**	71.80±0.23	**97.19±0.26**	97.73±0.18

where $r^w = \mathcal{N}(\theta(x^w))$ and $r^s = \mathcal{N}(\theta(x^s))$ are the label predictions of TNC for x^w and x^s, respectively. For the purpose of entropy minimization (Lee et al., 2013), we adopt hard pseudo-label $\hat{p}^w$ to enforce the consistency regularization on TPC. Differently, we use soft pseudo-label r^w for consistency regularization of TNC, so as MutexMatch can know more information of impossible class for classification. We further discuss this soft-label setting in Section 5.2. The whole algorithm is presented in Section 1 of Appendix.

4 EXPERIMENTS

Following Tarvainen & Valpola (2017); Sohn et al. (2020), we perform evaluation on four benchmark datasets, including STL-10, CIFAR-10/100 and SVHN. We also conduct ablation studies in Section 5 to investigate the efficacy of MutexMatch. Other experiments, *e.g.*, the impact of learning rate and hyper-parameters, are shown in Section D in the Appendix.

4.1 CIFAR-10, CIFAR-100 AND SVHN

We evaluate our method and baselines on three widely used SSL datasets: (1) CIFAR-10, consisting of 50,000 images from 10 classes, (2) CIFAR-100, consisting of 50,000 images from 100 classes, and (3) SVHN, consisting of more than 70,000 street view house number images from 10 classes.

Baselines. We introduce recent state-of-the-art SSL methods, *i.e.*, CoMatch (Li et al., 2020), FixMatch (Sohn et al., 2020) and FixMatch with distribution alignment (Berthelot et al., 2020) to compare with MutexMatch. Moreover, we compare our method with SSL methods such as UDA (Xie et al., 2020), MixMatch (Berthelot et al., 2019) and ReMixMatch (Berthelot et al., 2020).

Settings. For all experiments, in MutexMatch, we adopt Wide ReseNet (Zagoruyko & Komodakis, 2016) as the backbone (WRN-28-2 for CIFAR-10, SVHN and WRN-28-8 for CIFAR-100) following Sohn et al. (2020). In our implementation, TNC is the same two-layer MLP as TPC. For fair comparison, We follow these baseline methods (Sohn et al., 2020; Li et al., 2020) 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 without the decay schedule for CIFAR-10, and with cosine decay schedule for CIFAR-100 and SVHN. For hyper-parameters in MutexMatch, we set $\tau = 0.95, \mu = 7, B = 64$ for all experiments. Particularly, we set $\tau = 0.5$ on CIFAR-10 with 80 labels and train the model with cosine decay schedule for learning rate. In our method, RandAugment (Cubuk et al., 2020) is used for strong augmentation. Also, λ_{sep}, λ_p and λ_n are set to 1 for simplicity. To reduce the influence from random data partition, we report the mean and variance of accuracy on five different folds of labeled/unlabeled data.

Results. Table 1 shows the comparison between MutexMatch and baselines. With only 4 labeled data per class, MutexMatch achieves an accuracy of 93.49±0.22% on CIFAR-10, 56.14±1.46% on CIFAR-100 and 97.19±0.26% on SVHN, yielding improvement over prior SSL results. Especially, we demonstrate the superiority of MutexMatch under the extremely label-scarce setting. *e.g.*, achieving an average accuracy of 91.77% on CIFAR-10 with only 20 labels, 40.38% on CIFAR-100 with 200 labels. In addition, details on barely supervised learning can be found in

Section B of Appendix. The fewer labels will lead to the accumulation of more noise pseudo-labels in training, whereas MutexMatch uses all unlabeled data while introducing little error information.

Moreover, we report additional results on CIFAR-10 with different backbone CNN-13 and more available labels. We compare MutexMatch with MT (Tarvainen & Valpola, 2017), ICT (Verma et al., 2019), DualStudent (Ke et al., 2019) and UPS Rizve et al. (2021). We conduct experiments using the same setting as CIFAR-10 with 80 labels. In Table 2, we find that MutexMatch is not backbone dependent, and achieves performance improvement when more labels are given, outperforming all baseline methods. More discussion of experimental results can be found in Section 4.3.

Table 2: Accuracy on CIFAR-10 with larger amounts of labels and CNN-13 backbone. Results of baseline methods are reported in UPS (Rizve et al., 2021).

Method	CIFAR-10	
	1000 labels	4000 labels
MT	80.96±0:51	88.59±0.25
ICT	84.52±0.78	92.71±0.02
DualStudent	85.83±0.38	91.11±0.09
UPS	91.82±0.15	93.61±0.02
MutexMatch	**93.01±0.32**	**94.10±0.24**

Table 3: Accuracy on STL-10 averaged on 5 pre-defined folds with ResNet-18 backbone. Results of baseline methods are reported in CoMatch (Li et al., 2020).

Method	STL-10
	ResNet-18
MixMatch	38.02±8.29
FixMatch	65.38±0.42
FixMatch w. DA	66.53±0.39
CoMatch	79.80±0.38
MutexMatch	**83.36±0.22**

4.2 STL-10

STL-10 contains 10 classes of 5,000 labeled and 100,000 unlabeled images extracted from a similar but broader distribution. The challenge of STL-10 lies in other unlabeled images contains out of distribution images and this distribution shift enables us to test the robustness of SSL algorithm.

Settings. For STL-10, we evaluate MutexMatch on the 5 pre-defined folds. Each fold contains 1,000 labeled data and 100,000 unlabeled data. Therefore, we trained five models and averaged their performance as the final result. Following Li et al. (2020), we use ResNet-18 as the backbone because it consumes less computing resources than WRN-28-8 used in Sohn et al. (2020). We use the same hyperparameters and learning rate as CIFAR-100 in Section 4.1, and train the models using SGD with a momentum of 0.9 and a weight decay of 0.0005.

Results. Table 3 shows the results, averaged on all 5 runs. In this setting, MutexMatch achieves accuracy improvement from 79.80±0.38% to 83.36±0.22% compared with CoMatch. The performance of MutexMatch on STL-10 is much better than that of the existing methods, showing TNC is less sensitive to data distribution shift between labeled and unlabeled data, so that MutexMatch can maintain robust performance like on other datasets.

4.3 Effectiveness Analysis of TNC

As shown in Figure 1, the TNC of MutexMatch uses exclusion method to help the model deal with unlabeled data with low confidence. Ideally, we think that for one class, the distribution of complementary pseudo-labels from TNC should be evenly dispersed or diverse unlike pseudo-label from TPC, so MutexMatch can exclude more error classes as much as possible. We conduct experiments on CIFAR-10 with 40 labels using the same setting as in Section 4.1. We observe that the class prediction from TNC is indeed generally consistent with our hypothesis. As shown in Figure 4, during training, for each class of CIFAR-10, the prediction of TNC is gradually dispersed to several classes (*i.e.*, far away from the main diagonal of heat map), instead of gathering at a single class, indicating that TNC could play the role of exclusion method. On the contrary, the prediction outputted by TPC is gradually concentrated to the correct class (*i.e.*, gathered to the main diagonal of the heat map). Note that, TNC uses soft labels for consistency regularization, which can explore more complementary information to help TPC classify correctly.

Compared with other baseline methods shown in Table 1, MutexMatch performs better on CIFAR-10 with extremely scarce labels. We believe that confirmation bias (Yu et al., 2018) leads to the poor

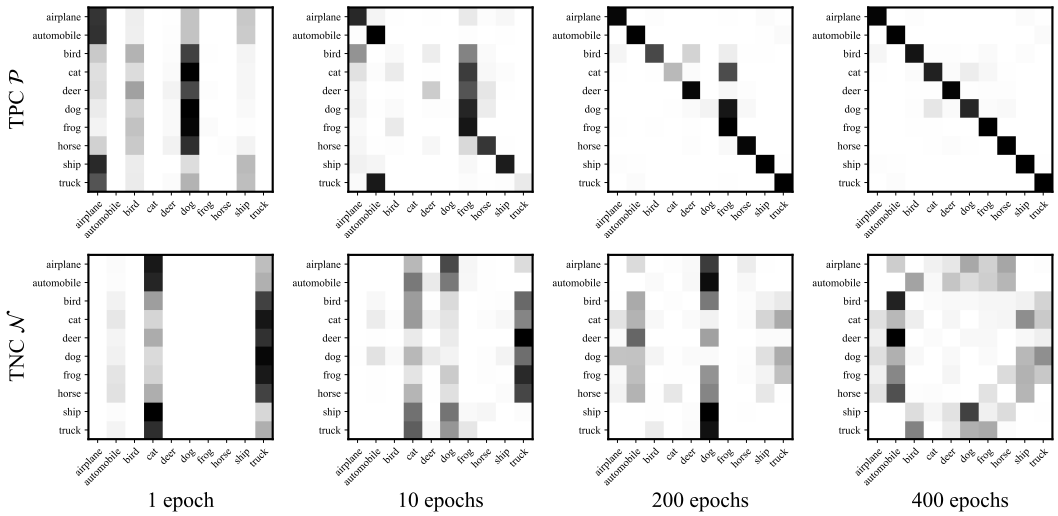


Figure 4: The rate (%) of each class (column in heat map) in the pseudo-label and complementary pseudo-label outputted by TPC and TNC respectively corresponding to each class (row in heat map) in CIFAR-10. The darker, the higher. Results are reported in a run on CIFAR-10 with 40 labels.

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 an exclusive manner by TNC, introducing few noisy pseudo-labels. As shown in Figure 5, our experiments on CIFAR-10 show MutexMatch produces more accurate pseudo-labels than FixMatch (Sohn et al., 2020), especially when there are very few labeled samples. In this figure, M indicates the results of MutexMatch and F indicates the results of FixMatch.

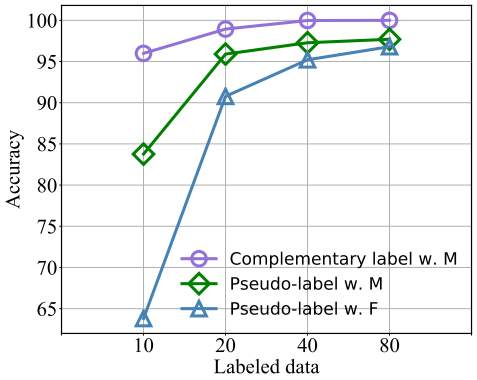


Figure 5: Accuracy of pseudo-label and complementary label on CIFAR-10 with different amount of labeled data.

The accuracy of complementary pseudo-label is very crucial. An important premise for MutexMatch to work is that the complementary pseudo-label outputted by TNC is easy to predict, so it will introduce less error information into the model. Figure 5 shows that the complementary pseudo-labels outputted by TNC achieves high accuracy. Compared with the pseudo-label outputted by TPC, complementary label is more insensitive to the change of the number of labels. Even with only one label per class, it can maintain a high accuracy.

5 Ablation Study

We conduct an extensive ablation study to verify the effectiveness of MutexMatch. The experiments are mainly conducted on CIFAR-10 and SVHN using four labels per class, where MutexMatch achieves 93.49±0.22% and 97.19±0.26% accuracy using default setting. In the following experiments, we keep the supervised loss as Equation (2) and positive consistency loss as Equation (4).

5.1 Utilization of Low-confidence Samples

In order to fairly verify the effectiveness of TNC, we use the same settings as Section 4.1. We believe that the reason why the performance of MutexMatch is better than other earlier SSL algorithms is that the existence of TNC enables the model to learn from all unlabeled data. For example, in

FixMatch, with a predefined confidence threshold, the unlabeled samples whose confidence is less than this threshold will not participate in the training. Therefore, we use the three most intuitive ways to use all the unlabeled data. We first use TPC to compute prediction $p^w = \mathcal{P}(x^w)$ of weakly-augmented unlabeled data x^w and then:

(i) We use $\hat{p}^w = \arg\max(p^w)$ as a hard pseudo-label, and enforce the cross-entropy loss against the model's prediction $p^s = \mathcal{P}(x^s)$ of strongly-augmented unlabeled data x^s:

$$\mathcal{L}_{ab1} = \frac{1}{\mu B}\sum_{n=1}^{\mu B} \mathbb{1}(\max(p_n^w) < \tau) H(\hat{p}_n^w, p_n^s). \tag{6}$$

(ii) We use p^w as a soft pseudo-label and enforce the cross-entropy loss against p^s:

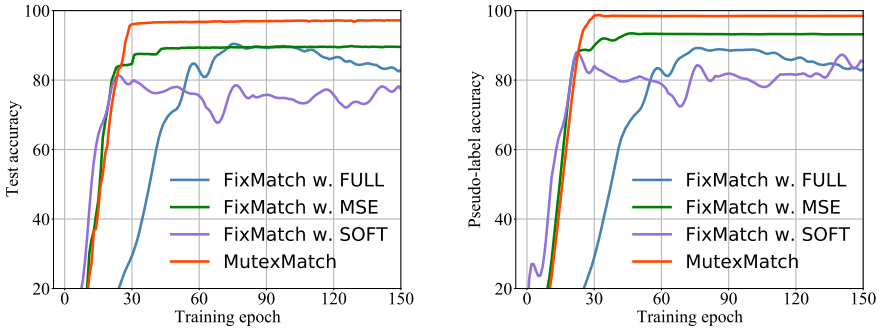
$$\mathcal{L}_{ab1} = \frac{1}{\mu B}\sum_{n=1}^{\mu B} \mathbb{1}(\max(p_n^w) < \tau) H(p_n^w, p_n^s). \tag{7}$$

(iii) We use the features $f^w = \theta(x^w)$ of weakly-augmented image and the features $f^s = \theta(x^s)$ of strongly-augmented image extracted by the feature extractor:

$$\mathcal{L}_{ab1} = \frac{1}{\mu B}\sum_{n=1}^{\mu B} \mathbb{1}(\max(p_n^w) < \tau) E(f_n^w, f_n^s), \tag{8}$$

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

The loss given above 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 we show the results of all experiments in Figure 6. In this figure, the *FULL* indicates the setting of (i), the *SOFT* indicates the setting of (ii) and the *MSE* indicates the setting of (iii). On this dataset, the default MutexMatch achieves an accuracy of 97.19±0.26%, outperforming all other experiments. Other ways using low-confidence samples will introduce more noisy pseudo-labels, resulting in the decline and instability of the accuracy of pseudo-label, which is not conducive to consistency regularization.



(a) (b)

Figure 6: The learning curve of ablation study on SVHN. The x-axis represents the training epoch and y-axis represents the test accuracy in (a) and the pseudo-label accuracy in (b).

5.2 Evaluation on Learning Scheme of TNC

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

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

$$\mathcal{L}_{sep} = \frac{1}{B}\sum_{n=1}^{B} H(\hat{q}_n^w, r_n^w),\ \mathcal{L}_{ab2} = \frac{1}{\mu B}\sum_{n=1}^{\mu B} \mathbb{1}(\max(p_n^w) < \tau) H(\hat{r}_n^w, r_n^s), \tag{9}$$

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

(ii) We use hard complementary pseudo-label $\hat{\gamma}^w$, 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 back propagation on θ, and enforce consistency regularization against soft complementary pseudo-label:

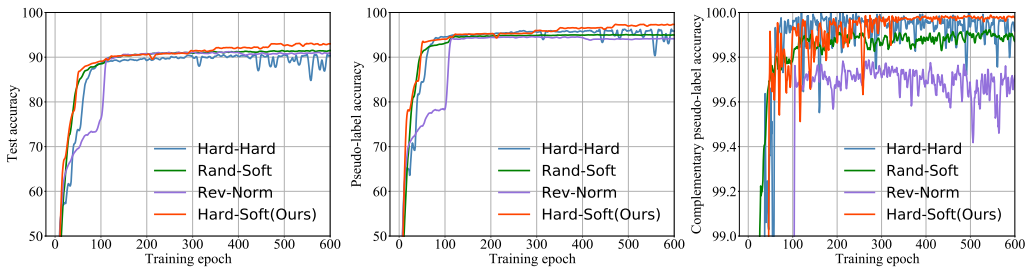
$$\mathcal{L}_{sep} = \frac{1}{B}\sum_{n=1}^{B} H(\hat{\gamma}_n^w, r_n^w),\ L_{ab2} = \frac{1}{\mu B}\sum_{n=1}^{\mu B} \mathbb{1}(\max(p_n^w) < \tau) H(r_n^w, r_n^s). \tag{10}$$

(iii) We remove the separately training part of TNC. The complementary pseudo-label for TNC is obtained directly by $q^w = \texttt{Norm}(1 - p^w)$ where $\texttt{Norm}(\cdot)$ is operation normalizing q^w into interval $[0, 1]$. We enforce consistency regularization against soft complementary pseudo-label:

$$\mathcal{L}_{ab2} = \frac{1}{\mu B}\sum_{n=1}^{\mu B} \mathbb{1}(\max(p_n^w) < \tau) H(q_n^w, r_n^s). \tag{11}$$

The loss given above minimized by experiments is simply $\mathcal{L}_{sup} + \mathcal{L}_p + \mathcal{L}_{sep} + \mathcal{L}_{ab2}$ in (i), (ii) and $\mathcal{L}_{sup} + \mathcal{L}_p + \mathcal{L}_{ab2}$ in (iii).

All models are trained on CIFAR-10 using four labels per class, and we show the results of all experiments in Figure 7. In the figure, the *Hard-Hard* indicates setting of (i), the *Rand-Soft* indicates setting of (ii) and the *Rev-Norm* indicates setting of (iii). (i), (ii) and (iii) achieve accuracy of 90.56%, 91.53% and 91.03% respectively. The default MutexMatch achieved an accuracy of 93.49% which outperforms other settings. Furthermore, experiments show that the accuracy and stability of the complementary pseudo-labels which are outputted by TNC of default MutexMatch are dominant. TNC uses hard pseudo-label for separate training to ensure the accuracy and stability of complementary pseudo-label, and uses soft pseudo-label to participate in mutex-based consistency regularization to exclude more potential error classes.



(a) (b) (c)

Figure 7: The learning curve of ablation study on CIFAR-10 with 40 labels. The x-axis represents the training epoch and the y-axis represents the test accuracy in (a), the pseudo-label accuracy in (b), and the complementary pseudo-label accuracy in (c).

6 Conclusion

In this paper, we propose MutexMatch, a novel SSL algorithm using a mutex-based consistency regularization derived by 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, especially under label-scarce conditions. Last but not least, we validate that low-confidence samples could still be well utilized in training from a novel way. We believe this usage of low-confidence samples could be borrowed to other semi-supervised tasks, *e.g.*, segmentation and detection.

REFERENCES

Phil Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems 27*, volume 27, pp. 3365–3373, 2014.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 32, pp. 5049–5059, 2019.

David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *Eighth International Conference on Learning Representations*, 2020.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18613–18624, 2020.

Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. Learning from complementary labels. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5639–5649, 2017.

Takashi Ishida, Gang Niu, Aditya Krishna Menon, and Masashi Sugiyama. Complementary-label learning for arbitrary losses and models. In *International Conference on Machine Learning*, pp. 2971–2980, 2018.

Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson Lau. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *IEEE International Conference on Computer Vision*, pp. 6728–6736, 2019.

Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *2019 IEEE International Conference on Computer Vision*, pp. 101–110, 2019.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896, 2013.

Junnan Li, Caiming Xiong, and Steven C. H. Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. *arXiv preprint arXiv:2011.11183*, 2020.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *Fifth International Conference on Learning Representations*, 2017.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2233–2241, 2017.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *The Ninth International Conference on Learning Representations*, 2021.

Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision*, pp. 299–315, 2018.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, 2017.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 3635–3641, 2019.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33, 2020.

Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *Proceedings of the European Conference on Computer Vision*, pp. 69–85, 2018.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Xiaojin Zhu. Semi-supervised learning. *Encyclopedia of Machine Learning and Data Mining*, pp. 1142–1147, 2017.

A ALGORITHM

Algorithm 1: MutexMatch algorithm

Input: batch of labeled data $\mathcal{X} = \{(x_b^{lb}, y_b^{lb})\}_{b=1}^{B}$, batch of unlabeled data $\mathcal{U} = \{x_b^{ulb}\}_{b=1}^{\mu B}$, feature extractor θ, TNC $\mathcal{P}$, TNC $\mathcal{N}$

for *iteration* t **do**
- $\mathcal{L}_{sup} = \frac{1}{B}\sum_{n=1}^{B} H(y_n^{lb}, \mathcal{P}(x_n^{lb}))$ // *Supervised loss for* x^{lb}
- **for** *iteration* $b = 1$ *to* μB **do**
  - $p_b^w = \mathcal{P}(\theta(\alpha_w(x_b^{ulb})))$ // *Compute TPC's prediction for weakly-augmented* x^{ulb}
  - $p_b^s = \mathcal{P}(\theta(\alpha_s(x_b^{ulb})))$ // *Compute TPC's prediction for strongly-augmented* x^{ulb}
  - $r_b^w = \mathcal{N}(\theta(\alpha_w(x_b^{ulb})))$ // *Compute TNC's prediction for weakly-augmented* x^{ulb}
  - $r_b^s = \mathcal{N}(\theta(\alpha_s(x_b^{ulb})))$ // *Compute TNC's prediction for strongly-augmented* x^{ulb}
  - $\hat{p}_b^w = \arg\max(p_b^w)$ // *Select pseudo-labels for* x^{ulb}
  - $\hat{q}_b^w = \arg\min(p_b^w)$ // *Select complementary pseudo-labels for* x^{ulb}
- **end**
- $\mathcal{L}_{sep} = \frac{1}{\mu B}\sum_{n=1}^{\mu B} H(\hat{q}_n^w, \mathcal{N}(\hat{\theta}(x_n^w)))$ // *Stop back-propagating gradients on* θ
- $\mathcal{L}_p = \frac{1}{\mu B}\sum_{n=1}^{\mu B} \mathbb{1}(\max(p_n^w) \geq \tau) H(\hat{p}_n^w, p_n^s)$ // *Positive consistency loss for* x^{ulb}
- $\mathcal{L}_n = \frac{1}{\mu B}\sum_{n=1}^{\mu B} \mathbb{1}(\max(p_n^w) < \tau) H(r_n^w, r_n^s)$ // *Negative consistency loss for* x^{ulb}
- update $\theta, \mathcal{P}, \mathcal{N}$ by SGD to optimise $\mathcal{L}_{sup} + \lambda_{sep}\mathcal{L}_{sep} + \lambda_p\mathcal{L}_p + \lambda_n\mathcal{L}_n$

end

B BARELY SUPERVISED LEARNING

The experimental protocol of barely supervised learning (BSL) described in Sohn et al. (2020) assume a limited availability (*e.g.*, 1 or 5) of labeled data from categories of interest. In order to test the performance of our method in extreme cases, we conduct experiments on CIFAR-10 with only one label per class, and consider developing a simple method to use our TNC in the test phase. As shown in Table 4, we use five different random seeds to extract one label of each class from CIFAR-10, and use MutexMatch to achieve test accuracy reaching between 65.30% and 93.07% with a mean of 78.73%. Compared with FixMatch (Sohn et al., 2020) reaching between 48.58%

and 85.32%, the performance of MutexMatch is more superior. Then we consider using TNC to complete the test phase under this setting to obtain the test accuracy. We assume that in the ideal case, according to Equation (3), for test data x, the prediction of TNC $r_x = \mathcal{N}(x)$ and the prediction of TPC $p_x = \mathcal{P}(x)$ should satisfy $\arg\max(r_x) = \arg\min(p_x)$.

According to negative learning proposed in Kim et al. (2019), we hypothesis TNC is trained to classify what input image does not belong to its complementary label, so that we can use $\hat{r}_x = \arg\min(r_x)$ to classify an input image x. Compared with TPC, TNC may learn less error information when the label is extremely scarce, so as to obtain better test performance. In order to verify this idea, we used TNC to participate in the test phase showed in Figure 8. For test sample x, we set a confidence threshold T, if $p_x > T$ we uses TPC to predict, if $p_x < T$ uses TNC instead, that is, the leftmost point ($T = 0$) in the figure represents only TNC for test, and the rightmost point ($T = 1$) represents only TNC for test. Taking 20 labels as the dividing line, we can see that using TNC for prediction has more advantages in the case of fewer labels.

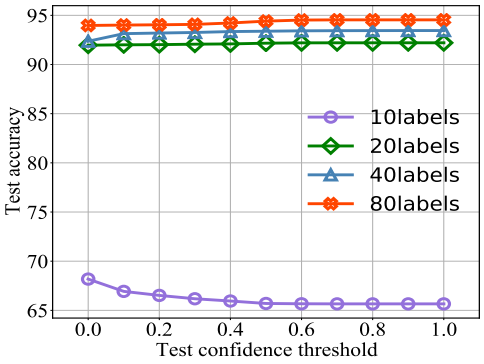


Figure 8: Test accuracy on CIFAR-10 in single run with various amount of labels using TNC to participate test phase. The x-axis represents confidence threshold T and y-axis represents test accuracy.

Table 4: Accuracy of MutexMatch on a single 1-label split of CIFAR-10 with different random seeds. Results are ordered by accuracy.

Fold	1	2	3	4	5
Accuracy	65.30	71.12	77.83	86.33	93.07

C SEMI-SUPERVISED LEARNING WITH NOISY LABELS

To evaluate the robustness of MutexMatch, we conduct our experiments following settings of semi-supervised learning with noisy labels on CIFAR-10. Semi-supervised learning and noise labels are challenging problems, and semi-supervised learning with noise labels is much more because the ability of the model to resist noise labels will be greatly weakened when there is only a small amount of labeled data.

Setting. Following Kim et al. (2019); Patrini et al. (2017), we applied three different types of noise in experiments:
(1) *Symmetric-inc* noise is created by randomly selecting the label from all classes.
(2) *Symmetric-exc* noise is created by randomly selecting the label from all classes without ground truth label.
(3) *Asymmetric* noise is generated by mapping TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ PLANE, DEER $\rightarrow$ HORSE, and CAT $\leftrightarrow$ DOG for CIFAR-10.
We evaluate MutexMatch and baselines with noisy labels mentioned above using the same settings as in Section 4.1. All experiments use 40 labeled data for training, varying radio of noisy labels in labeled data (25%&50%).

Results. Table 5 shows the accuracy comparison between MutexMatch and baselines. All the results are reported by averaging on 5 different folds. Experiments show the robustness of MutexMatch under this setting. For example, with 2 labels and 2 noisy labels (Symmetric-inc) per class, MutexMatch achieves 88.72±3.51% accuracy, while training of FixMatch collapse reaching a lower 77.80±17.57% accuracy. MutexMatch contains the idea of negative learning. Learning from the perspective of complementary pseudo-label can prevents model from overfitting to noisy data (Kim et al., 2019) so that MutexMatch achieves superior performance in SSL with noisy labels.

Table 5: Accuracy on CIFAR-10 with noisy labels averaged on 5 different folds. All experiments were based on 40 labeled data with varying radio of noisy labels.

Method	Symmetric-inc		Symmetric-exc		Asymmetric	
	25%noisy	50%noisy	25%noisy	50%noisy	25%noisy	50%noisy
FixMatch	77.80±17.57	**81.54±18.47**	80.05±5.80	75.11±14.66	84.58±5.90	72.91±19.30
MutexMatch	**88.72±8.51**	77.18±10.55	**89.37±6.10**	**81.85±8.00**	**89.51±5.14**	**78.28±15.44**

D Additional Experimental Results

D.1 Ablation Study on Learning Rate and Learning Rate Schedule

We note that learning rate and learning rate schedule are very important for MutexMatch. In this section, we use the experimental setting in Section 4.1 to conduct additional ablation experiments for both. Following Loshchilov & Hutter (2017), recent work (Sohn et al., 2020; Li et al., 2020) use a cosine learning rate decay and achieve best performance. However, as shown in Table 6, we found that MutexMatch achieves better results without learning rate decay on CIFAR-10, outperforming cosine learning rate decay by 0.32%. When there are many labels, the pseudo-labels outputted by TPC are more likely to have high-confidence and remain stable. It is necessary for MutexMatch to use cosine learning rate decay to jump out of the local optimum.

Table 6: Ablation study on learning rate and learning rate schedule. Results are reported on CIFAR-10 varying number of labels.

Decay Schedule	Learning Rate	Labels	Backbone	Accuracy
No Decay	0.03	40	WRN-28-2	**93.54**
No Decay	0.07	40	WRN-28-2	93.02
No Decay	0.10	40	WRN-28-2	92.89
Cosine Decay	0.03	40	WRN-28-2	93.22
Cosine Decay	0.07	40	WRN-28-2	93.20
Cosine Decay	0.10	40	WRN-28-2	92.59
No Decay	0.03	80	WRN-28-2	93.95
Cosine Decay	0.03	80	WRN-28-2	**94.53**
No Decay	0.03	1000	CNN-13	91.57
Cosine Decay	0.03	1000	CNN-13	**93.46**
No Decay	0.03	4000	CNN-13	92.75
Cosine Decay	0.03	4000	CNN-13	**94.41**

D.2 Hyperparameters

For MutexMatch, the choice of τ needs to be very cautious, because different τ will lead to the division of high and low-confidence portions, which will affect the impact of the mutex-based consistency regularization on the model. We use the identical setting of experiments in Section 4.1 for MutexMatch and vary τ to verify the sensitivity of MutexMatch to this hyperparameter. As shown

Table 7: Ablation study on confidence threshold τ. Results are reported on CIFAR-10 varying number of labels.

τ	Labels	Backbone	Accuracy
0.5	40	WRN-28-2	93.52
0.75	40	WRN-28-2	93.44
0.85	40	WRN-28-2	93.28
0.95	40	WRN-28-2	**93.54**
0.99	40	WRN-28-2	92.17
0.5	80	WRN-28-2	**94.53**
0.95	80	WRN-28-2	93.64
0.5	1000	CNN-13	**93.46**
0.95	1000	CNN-13	92.07
0.5	4000	CNN-13	**94.41**
0.95	4000	CNN-13	92.94

in Table 7, MutexMatch needs to select appropriate τ to divide confidence portions. We note that when there are many labels, τ 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 directly for learning. Therefore, we guess that in general, we should choose a smaller τ to make more pseudo-labels participate in the training of TPC when the number of labels increases.

At the same time, showed in Figure 9, we vary the weight λ_{sep} of the separate training loss for TNC $\mathcal{L}_{sep}$ and λ_n of the negative consistency loss $\mathcal{L}_n$. Choosing the appropriate weight of loss is very important for MutexMatch. Larger λ_{sep} ensures the accuracy of complementary pseudo-labels, which helps TNC better participate in training. Appropriate λ_n weighs the contribution of TNC and TPC in mutex-based consistency regularization, so that the model can achieve better performance.

Additionaly, we provide more ablation studies on various λ_{sep}, λ_n and λ_p shown in Table 8. We find that increasing λ_{sep} and λ_n at the same time will cause severe performance degradation, which shows that we must carefully control the importance of TNC in the learning process, because TPC has always maintained the most important position in completing our classification tasks. Meanwhile, appropriate λ_p ensures model can benefit from learning of TNC. More results of experiments about situation where $\lambda_p = 0$ or $\lambda_n = 0$ can be found in Section D.3.

Table 8: Accuracy on CIFAR-10 with 40 labels and various λ_{sep}, λ_n, λ_p.

λ_{sep}	λ_n	λ_p	Accuracy
1	1	1	**93.49**
1	1	10	91.27
1	1	20	85.05
10	1	1	88.94
20	1	1	85.44
20	10	1	18.03
20	10	10	15.96

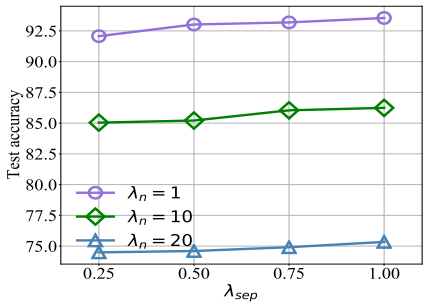


Figure 9: Accuracy on CIFAR-10 with 40 labels and various λ_{sep}, λ_n.

D.3 Ablation Study on TPC and TNC

We explain why we enforce consistency regularization on TNC as follows. Given two augmented variants derived from the same unlabeled instance, we claim that the class probability distributions

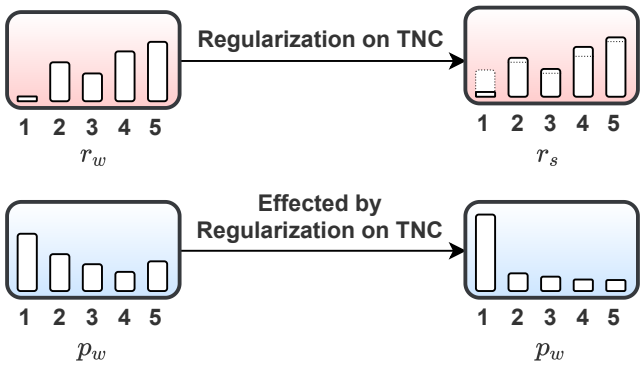


Figure 10: The correct component in prediction vector is class 1.

(*i.e.*, soft-labels) of their complementary predictions (*i.e.*, the TNC's outputs) should 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 on TPC's predictions. As shown in Figure 10, given a instance of class 1, the independent training of TNC can generate an accurate complementary prediction with extremely low probability of class 1. Then encouraging a similar prediction on its strongly augmented variant can help the model to learn more discriminative features. It can in turn affect the TPC's prediction, such that the False-Negative probabilities (to be predicted as a class of $2, 3, 4, 5$) can be effectively decreased. Consequently, the True-Positive probability of TPC's predictions is enlarged.

We construct an experiment on CIFAR-10 with 40 labels to verify our findings. We denote MutexMatch without consistency reguarlization on TNC as *M wo. c* (*i.e.*, MutexMatch degenerates to FixMatch). Although the confidence of the correct predictions of MutexMatch and *M wo. c* is very high (are very close to 100%), we check their wrong predictions as an example for comparison. In fact, for the correct part of the pseudo-labels, MutexMatch obtains more correct pseudo-labels than *M wo. c* (FixMatch) thanks to the use of consistency regularization on TNC (*i.e.*, MutexMatch's accuracy of pseduo-labels is higher than FixMatch, which is shown in Figure 5).

As shown in Figure 11(a), given the unlabeled instances belonging to "automobile", the MutexMatch's average probability of "automobile" component in prediction vector is higher than that of *M wo. c*. We can also obtain similar findings on other different classes, as shown in Figure 11(b). Such observations demonstrate enforcing prediction consistency on TNC can successfully help the model lower the False-Negative probability, which in turn improve the True-Positive probability in the TPC's prediction vector.

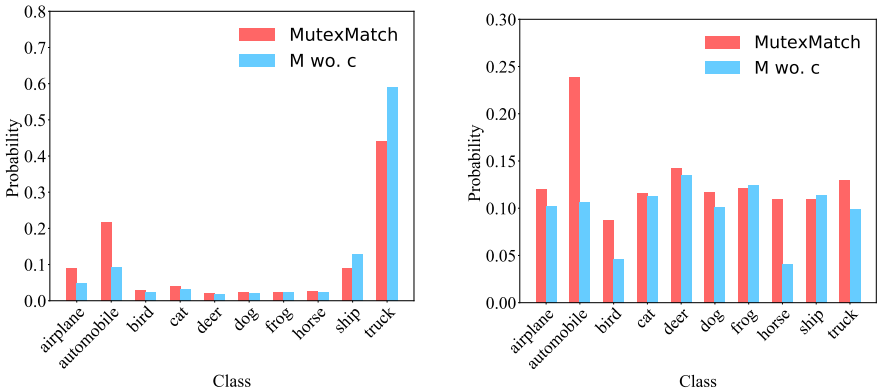


Figure 11: (a) Average probability of each component in prediction vector of automobile images. (b) Average probability of correct component in prediction vector of all classes in CIFAR-10.

Moreover, this design is based on the consideration of it's unreasonable to directly involve complementary label based negative learning in semi-supervised. In the early training stage of semi-supervised learning, the accuracy of pseudo-labels is often not very high. In this process, the introduction of complementary labels directly into the learning process will not only be helpful, but even harmful, and lead to training collapse finally. Therefore, we use consistency regularization to *"decouple"* the part where complementary label are directly involved in training. We believe that it is reasonable to use consistency regularization on TNC at the sample level, because we only need TNC to show the various results for each class at the dataset level, which is shown in Section 4.3. This demonstrates TNC has learned discriminative information from the aspect of complementary label. Implementing consistency regularization on TNC at the sample level is to help TNC learn from the perspective of complementary labels, so that feature extraction can learn better data representation of unlabeled data with low confidence. In addition, we use soft labels for consistency regularization on TNC, which also ensures TNC can learn multi-class information, which is described in Section 3.3. For further discussion on the effectiveness of TPC and TNC, we consider removing these two components respectively. Given mentioned above, we designed the following experiments:

(i) What happens if consistency regularization on TNC is abandoned (*i.e.*, $\lambda_n = 0$)? Taking into account that in the default MutexMatch, training of TNC with complementary pseudo-labels stops the gradient, so if we set $\lambda_n = 0$, then TNC is equivalent to not participating in the training of the model at all, which means that *MutexMatch degenerates into FixMatch*. So a more reasonable setting is to restore the backpropagation on feature extractor θ in Equation (3). Then we explore the effect of not using consistency regularization on TNC:

$$\mathcal{L}_{sep} = \frac{1}{\mu B}\sum_{n=1}^{\mu B} H(\arg\min(\mathcal{P}(\theta(x_n^w)), \mathcal{N}(\theta(x_n^w))), \tag{12}$$

where we restore the back propagation on θ and set $\lambda_n = 0$ in Equation (1).

(ii) What happens when we train TNC with complementary labels generated by TPC without stopping the gradient on TNC? We keep $\lambda_n = 1$ in Equation (1), and restore the backpropagation on θ like Equation (12).

(iii) At the same time, in order to explore the role of TPC component, we set $\lambda_p = 0$ in Equation (1) for ablation study.

(iv) We simply set $(\lambda_p, \lambda_n, \lambda_{sep})$ to $(0, 0, 0)$, $(0, 1, 0)$ and $(1, 1, 0)$.

As shown in Figure 12, the default MutexMatch achieves dominant performance compared with other settings. In figures, *wo. TNC* represents setting of (i), *wo. SG* represents setting of (ii), *wo. TPC* represents setting of (iii) and *w. 000*, *w. 010*, *w. 110* represent setting of (iv). Obviously, TNC participates in model training directly will cause training to collapse, which means that the model is seriously affected by the learning of TNC, and there is no way to learn effective information of "what it is". (iii) shows that TPC in this case does not get adequate training and it can't complete the classification task. *i.e.*, the training of TPC is equivalent to using only labeled data. Meanwhile, The training of TNC is closely related to TPC. In this case, TNC has not been well trained, too. And in fact, we don't use TNC to participate in the testing phase. (i) and (ii) illustrate the superiority of using consistency regularization for learning on TNC. This *"decoupling"* ensures that TNC can make the model learn a better data representation without affecting the learning of TPC. Finally, the combination of (iv) and other settings proves the necessity of each component in MutexMatch.

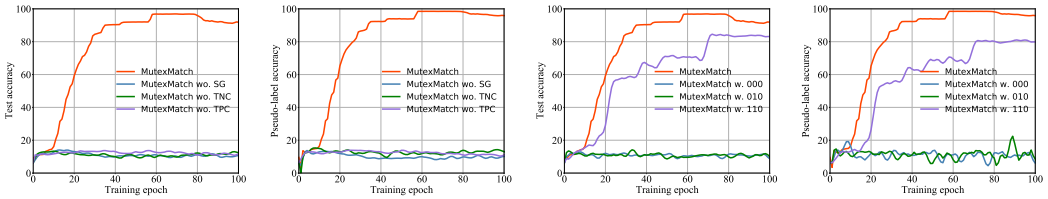


Figure 12: The learning curve of ablation study on SVHN. The x-axis represents the training epoch and y-axis represents the test accuracy in (a), (c) and the pseudo-label accuracy in (b), (d).