# HyperMatch: Noise-Tolerant Semi-Supervised Learning via Relaxed Contrastive Constraint

Beitong Zhou*, Jing Lu*, Kerui Liu, Yunlu Xu, Zhanzhan Cheng†, Yi Niu
Hikvision Research Institute
zhoubt@hust.edu.cn, {lujing6, liukerui, xuyunlu, chengzhanzhan, niuyi}@hikvision.com

## Abstract

*Recent developments of the application of Contrastive Learning in Semi-Supervised Learning (SSL) have demonstrated significant advancements, as a result of its exceptional ability to learn class-aware cluster representations and the full exploitation of massive unlabeled data. However, mismatched instance pairs caused by inaccurate pseudo labels would assign an unlabeled instance to the incorrect class in feature space, hence exacerbating SSL's renowned confirmation bias. To address this issue, we introduced a novel SSL approach, HyperMatch, which is a plug-in to several SSL designs enabling noise-tolerant utilization of unlabeled data. In particular, confidence predictions are combined with semantic similarities to generate a more objective class distribution, followed by a Gaussian Mixture Model to divide pseudo labels into a 'confident' and a 'less confident' subset. Then, we introduce Relaxed Contrastive Loss by assigning the 'less-confident' samples to a hyper-class, i.e. the union of top-$K$ nearest classes, which effectively regularizes the interference of incorrect pseudo labels and even increases the probability of pulling a 'less confident' sample close to its true class. Experiments and in-depth studies demonstrate that HyperMatch delivers remarkable state-of-the-art performance, outperforming FixMatch on CIFAR100 with 400 and 2500 labeled samples by* $11.86\%$ *and* $4.88\%$*, respectively.*

## 1. Introduction

Semi-supervised learning (SSL) [2, 3, 5, 20, 21, 26, 29, 38, 39, 42] has become a promising solution for leveraging unlabeled data to save the expensive annotation cost and simultaneously improve model performance, especially in applications where large amounts of annotated data are required to obtain a model with high performance [4, 18]. Modern SSL algorithms generally fall into two categories: the *Pseudo Label*-based [22, 30, 38] focuses on generating reliable pseudo labels for unlabeled data, whereas the *Consistency Regularization*-based [2, 3, 20, 29] constrains the model to make consistent predictions on perturbed samples.

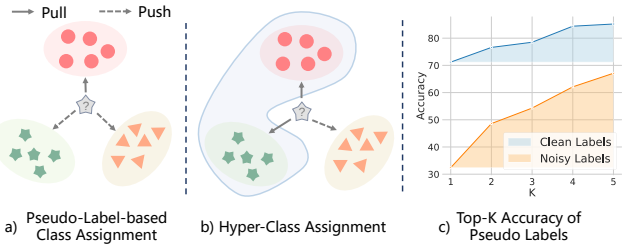



Figure 1. Illustration of our ideas. (a) Pseudo-label-based assignment: the instance is pulled close to the wrong pseudo label class and pushed away from ground-truth class (green pentagon). (b) Hyper-class assignment: the instance is assigned to its hyper-class (the union of top-$K$ nearest classes), which includes the ground truth class. (c) Top-$K$ accuracy for clean and noisy pseudo labels (divided by our Gaussian Mixture Model) in CIFAR100@400 experiment. As $K$ grows, noisy labels benefit more than clean labels.

Recently, a prominent advance is the combination of Contrastive Learning [6, 8, 11, 13] with SSL techniques [19, 21, 24, 25, 39, 42], which sets the remarkable state-of-the-art performance. The naive Self-supervised Contrastive Learning [6, 13] in pre-training tasks pushes instance-level features away, thereby potentially driving samples within the same class apart, its class-agnostic nature has been proved to conflict with the class-clustering property of SSL [24, 39], hence most recent studies [21, 24, 39] turn to Class-Aware Contrastive Learning [16]. The general routine is to first assign each unlabeled instance to the top-1 nearest class based on their pseudo labels, then two unlabeled instances from the same class are randomly selected to form a positive pair, followed by a class-aware contrastive loss to compel instances from positive pairs to share similar features while pushing away the representations of different classes.

The precision of pseudo labels has a direct impact on the class assignment of the aforementioned methods. By confidence threshold, unlabeled data can be roughly grouped into 'confident' and 'less confident' [39]. For 'confident'

*Equal contribution. †Corresponding author.

data that tends to yield accurate pseudo labels for class assignment, contrastive loss could constrain the model to acquire better clustered feature representations for each class, hence facilitating SSL learning. But for 'less confident' data with a much higher probability to provide incorrect pseudo labels, mismatched instance pairs tends to be induced and the contrastive constraint will pull the features of different classes closer while pushing the features from the same class further apart, which will inevitably degrade the learning. As shown in Fig. 1 (a), the 'less confident' instance could be drawn close to a false class (*i.e.* the red circle) while being pushed away from the true class, represented by green pentagons. For convenience, we also refer to the two kinds of unlabeled data as 'clean' and 'noisy' data.

To mitigate the detrimental effects of 'less confident' (*i.e.*, noisy) data, existing studies can be generally categorized into two categories: (1) Discarding low-quality pseudo labels [21, 24] with a low threshold, leaving some unlabeled data unused; (2) Adopting re-weighting strategies [39, 42] to lessen the effect of noisy labels. However, these approaches continue to closely adhere to the the paradigm of assigning noisy data to a single error-prone class, therefore they can only reduce the errors to a limited amount. Also, the accuracy of pseudo labels suffers from confirmation bias [1] in SSL, which is the accumulation of false predictions during training.

The aforesaid interference of 'less confident' data is caused by the inaccurate class assignment, it is more effective to devise a class assignment approach that can resist the distraction of wrong pseudo labels in order to mitigate their effects. In light of this, we propose to relax the conventional class assignment. Instead of assigning a noisy sample to a single class, we relax the assignment by grouping it into a 'hyper-class', which is the union of top-$K$ ($K > 1$) nearest classes. As depicted in Fig. 1 (b), the chance of the ground-truth class slipping into the hyper-class is dramatically enhanced by implementing the relaxation (as $K$ steadily grows). It's also worth noting that the marginal gain brought by relaxation for noisy data is significantly greater than that for clean data, as the top-1 pseudo label accuracy of clean data is already adequate. This suggests that the relaxation is more suitable for applying on noisy data.

In conjunction with the hyper-class assignment, a Relaxed Contrastive Loss is intended to restrain the feature of noisy samples being close to their corresponding hyper-class while increasing the distantce from the remaining classes. The simple yet effective relaxing has two benefits: (1) the likelihood of 'less confident' data being pushed away from its ground truth class can be lowered, and (2) the likelihood of data being pulled close to its ground truth class can be successfully raised. As seen in Fig. 1 (b), the ground truth class for the noisy unlabeled instance falls within the hyper-class, thus its feature will no longer be driven away from the actual class, but rather drawn close to it.

In order to manage the effective exploitation of bother clean and noisy unlabeled data, we proposed **HyperMatch**. First, predicted class probabilities are integrated with semantic similarities to produce unbiased per-sample class distributions. Next, a Gaussian Mixture Model (GMM) model is fitted on the calibrated distribution to separate clean unlabeled data from the noisy ones. The common class-aware contrastive loss is applied to clean data to constrain their features to approach the corresponding class. For noisy unlabeled data, a Relaxed Contrastive Loss is carefully developed to drive the noisy unlabeled data falling into their corresponding hyper-class. In summary, we contribute in three ways:

- We propose an enhanced contrastive learning method, HyperMatch, to handle the effective separation and exploitation of both clean and noisy pseudo labels for learning better-clustered feature representations. It is a plug-in that can be used to various SSL architectures to increase resilience while utilizing noisy data.
- Unlike previous studies that assign a 'less confident' sample to an error-prone class, we relax the assignment by categorizing the noisy sample into a hyper-class (a union of top-$K$ nearest classes), followed by the proposed Relaxed Contrastive Loss, which is effective at mitigating the problematic confirmation bias.
- With thorough experiments on SSL benchmarks, our HyperMatch demonstrates competitive performance and establishes the new state-of-the-art on multiple benchmarks. In-depth investigation reveals its effectiveness in handling the noisy pseudo labels.

## 2. Related Work

### 2.1. Semi-Supervised Learning

Semi-supervised learning (SSL) methods can be roughly categorized into two groups: pseudo label based and consistency regularization based. Pseudo label approaches [22, 36, 38], also referred as self-training, generate pseudo labels through a classification model trained on labeled data. The predictions of randomly augmented unlabeled data are utilized to minimize the classification loss [38]. However, they suffer from the insufficient use of unlabeled data as the predictions are constant throughout training.

Consistency regularization based methods [2, 29, 34] aim to regularize the model to output consistent predictions for different perturbations of the same input. Essential perturbations consist of network regularization techniques [31], adversarial training [26] and domain-specific data augmentations [29]. Mean Teacher [34] employs an exponential moving average (EMA) model for more accurate forecasts.

MixMatch [3] sharpens the averaged predictions of multiple strongly augmented views. ReMixMatch [2] further introduces augmentation anchoring by adopting RandAugment [10] and aligns distributions between labeled and unlabeled predictions. FixMatch [29] achieves state-of-the-art performance by integrating the highlights of above methods. The key idea is to regularize the predictions of the strongly-augmented view to match the one-hot pseudo label generated on the weakly-augmented view only when the confidence exceeds a given threshold.

### 2.2. Contrastive Learning Based SSL

With the development of unsupervised pre-training tasks, Self-Supervised Learning [6, 7, 13, 35, 37] has been in the limelight, bringing Contrastive Learning to the forefront. Self-supervised contrastive learning [37], which classifies different augmented views of the same instance as a positive pair and views of different instances as a negative pair, can provide useful visual representations for subsequent tasks. SimCLRv2 [7] utilizes the pre-trained model and validates that the SSL task can also benefit from the self-supervised pre-training in a two-stage framework. A comprehensive discussion of self-supervised contrastive learning methods can be found in [15].

Recently, works have been devoted to integrating contrastive learning with SSL in a unified framework [16, 24, 39, 42] and set the state-of-the-art. While a classification model aims to group intra-class samples together and learns class-level clustered features, self-supervised contrastive loss tends to separate each instance from others, thus will separate samples within the same class apart. To overcome this problem, pseudo labels are incorporated in representation learning to provide class-level priors, which can be referred as class-aware contrastive learning. Lee *et.al.* [21] adopt supervised contrastive learning loss [16] using filtered pseudo labels as groud truth annotations with FixMatch. CoMatch [24] combines memory-smoothed pseudo labels with the graph-based contrastive learning using a large size of memory bank. CCSSL [39] improves class-wise clustering on in-distribution data with pseudo labels.

Despite the recent progress of class-aware contrastive learning, the mismatched instance pairs brought by wrong pseudo labels of noisy data still limit their performance. Lee *et.al* [21] and Li *et.al* [24] only apply the class-aware constrative loss on samples above a given threshold, but this leaves a certain proportion of unlabeled data unused. In [39], self-supervised contrastive loss is imposed on the samples with the lowest confidence, but this conflicts with the class-clustering nature of SSL task. Yang *et.al* [39] and Zheng *et.al* [42] design re-weighting strategies to reduce the impact of mismatched pairs by assigning a smaller weight, which could only relieve the errors to some extent. All the above strategies strictly follow the paradigm of assigning the unlabeled instance to a single cluster, even when the assignment is error-prone. Instead, we relax the class assignment constraint by categorizing noisy unlabeled samples to a hyper-class, which proves to be more effective for regularizing the bias induced by wrong pseudo labels.

## 3. Method

### 3.1. Overview

In a standard semi-supervised classification task, a training batch consists of labeled instances $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{B}$ and unlabeled instances $\mathcal{U} = \{u_i\}_{i=1}^{\mu B}$, where $\mu$ is a hyper-parameter controlling the relative ratio of $\mathcal{U}$ to $\mathcal{X}$. Inputs $x_i$ and $y_i$ represent the $i$-th image and its one-hot label.

**Data Augmentations.** We use both weak and strong augmentations as is typical in current SSL methods [29, 39, 41]. For a labeled image $x_i$, a weak transformation function $\mathcal{A}_w(\cdot)$ is applied for supervised training. For an unlabeled instance $u_i$, we generate one weak augmentation view with $\mathcal{A}_w(\cdot)$ and two strong augmentation views with $\mathcal{A}_s(\cdot)$. The weak augmentation is fed to generate a reliable pseudo label while strong augmentations are used both for consistency regularization and contrastive learning.

**Architecture.** A convolutional neural network $\mathcal{F}(\cdot)$ is used to extract the feature, *i.e.*, $h = \mathcal{F}(\mathcal{A}(x))$. Then, a linear classification head $\phi_{cls}(\cdot)$ generates the class probability $p = \phi_{cls}(h)$. Another 2-layer projection head $\phi_{proj}(\cdot)$ is adopted to map embedding $h$ into a lower-dimensional feature $z$, *i.e.*, $z = \phi_{proj}(h)$, which is $l_2$ normalized.

HyperMatch jointly optimizes a combination of three losses: 1) a supervised classification loss $\mathcal{L}_{cls}$ on labeled data, 2) a consistency regularization loss $\mathcal{L}_{reg}$ on unlabeled data and 3) our proposed relaxed contrastive loss $\mathcal{L}_{rel}$. The supervised classification loss $\mathcal{L}_{cls}$ is directly imposed on the weakly augmented labeled samples, *i.e.*, $\mathcal{L}_{cls} = \frac{1}{B}\sum_{i=1}^{B} H(y_i, \phi_{cls}(\mathcal{F}(\mathcal{A}_w(x_i))))$, where $H$ indicates the cross entropy loss.

For unlabeled samples, pseudo labels are first generated from their weakly augmented views. Then the consistency regularization loss $\mathcal{L}_{reg}$ is implemented following the general form in FixMatch [29], which is the cross entropy loss between the filtered pseudo labels and the predictions of corresponding strongly perturbed samples:

$$\mathcal{L}_{reg} = \frac{1}{\mu B}\sum_{i=1}^{\mu B} \mathbb{1}(max(p_i) \geq \tau_u) H(\hat{y_i}, \phi_{cls}(\mathcal{F}(\mathcal{A}_s(u_i)))) \tag{1}$$

where $p_i = \phi_{cls}(\mathcal{F}(\mathcal{A}_w(u_i)))$ represents the model's prediction on the weakly augmented sample $u_i$ and the pseudo label $\hat{y_i} = argmax(p_i)$ is the class with maximum probability. A fixed high-confidence threshold $\tau_u$ is used to filter low-quality pseudo labels, which however ignores a large proportion of unlabeled data.

The overall training objective is defined as:

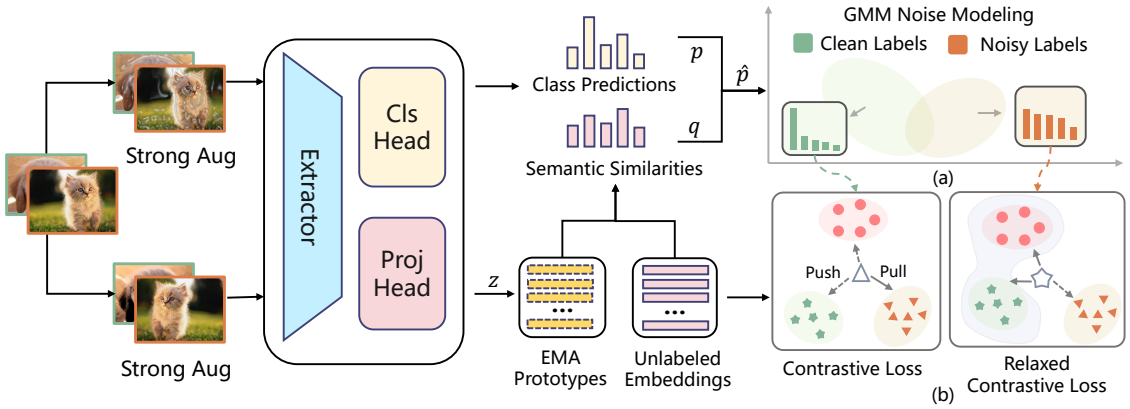



Figure 2. The framework of HyperMatch. (a) Pseudo label partition by sorting calibrated class distribution then fitting GMM on it; (b) Relaxed contrastive loss is imposed on noisy data by assigning them to the hyper-class (union of top $K$ nearest classes). By setting $K = 1$, it becomes naive class-aware contrastive loss, which is applied on clean data.

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{reg} + \lambda_2 \mathcal{L}_{rel} \tag{2}$$

where $\lambda_1$ and $\lambda_2$ are hyper-parameters to control the weight of corresponding loss. The proposed relaxed contrastive loss $\mathcal{L}_{rel}$ will be detailed in the following sections.

### 3.2. Pseudo Label Partition

To handle the noisy data, we first have to divide them from the clean ones. In [21, 24], a naive strategy that directly compares the maximum class probability with a fixed threshold is adopted to separate the clean labels from noisy ones. On the other hand, DivideMix [23] learn with noisy labels and fits a two-component Gaussian Mixture Model (GMM) based on the per-sample loss distribution to divide training data by setting a threshold. However, in a semi-supervised task, model's obsession with incorrect labels would lead to an over-confident class probability distribution and small losses even for noisy data, making it harder for a GMM to distinguish the clean instances from noisy ones if solely referring to the predicted probabilities.

**Class Distribution Calibration.** While the predicted classification probabilities may be biased towards certain categories, the semantic similarities usually exhibit a more uniform distribution [12, 27]. Hence to remedy the bias of predicted probabilities, we integrate semantic similarities to calibrate the class distributions.

First, we build a memory buffer to keep a set of class prototypes $\{\mathcal{T}^k\}_{k=1}^M$ from labeled data, where $M$ is the number of classes. In detail, we update the prototypes with the projection embeddings of labeled samples in a training batch using exponential moving average (EMA), *i.e.*, $\mathcal{T}^k = \rho\mathcal{T}^k + (1-\rho)\frac{1}{m_k}\sum_{l=1}^{m_k} z_l$, where $m_k$ is the number of $k$-th class labeled samples in the batch. Instead of maintaining a memory queue [12, 27], this implementation requires less GPU memory usage and computation time.

The class-wise semantic distribution $q_i = [q_i^1, ..., q_i^M]$ for an unlabeled instance $u_i$ can be measures as:

$$q_i = Norm([sim(z_i, \mathcal{T}^1), \ldots, sim(z_i, \mathcal{T}^M)]) \tag{3}$$

where $sim(\cdot,\cdot)$ is the cosine similarity metric and $Norm$ denotes $l_1$ normalization. We further blend the semantic similarities $q_i$ with probabilities $p_i$ by:

$$\hat{p}_i = Norm(p_i \circ q_i) \tag{4}$$

to obtain more unbiased class distributions $\hat{p}_i = [\hat{p}_i^1, ..., \hat{p}_i^M]$, where $\circ$ is element-wise multiplication.

**GMM Noise Modeling**. We rearrange the calibrated class distribution $\hat{p}$ by sorting each class's probabilities in a descending order, as in Fig. 2 (a), then fit the sorted distribution $sort(\hat{p})$ with a two-component GMM model on the entire unlabeled dataset.

It can be easily observed that clean pseudo labels exhibit a more sharp class distribution than noisy ones as they are more confident in their decisions. Thus we first compute the variance of each mixture component's mean distribution, then choose the one with larger variance to represent the clean data and the other component to model noisy data, denoted as $g_{clean}$ and $g_{noisy}$. On top of that, by comparing the posterior probability $p(g_{clean}|sort(\hat{p}))$ of the clean GMM component with a threshold $\tau_g$, we obtain the clean subset $\mathcal{U}_{clean}$ by:

$$\mathcal{U}_{clean} = \{u_i | u_i \in \mathcal{U}, p(g_{clean}|sort(\hat{p}_i)) > \tau_g\}, \tag{5}$$

and $\mathcal{U}_{noisy} = \mathcal{U} \setminus \mathcal{U}_{clean}$. In such a way, we reduce the negative effect of confirmation bias caused by over-confident prediction probabilities by incorporating more uniform semantic similarity distributions. It's noteworthy that our algorithm is quite insensitive to the selection of $\tau_g$, as is further discussed in Section 4.4.

### 3.3. Relaxed Contrastive Loss

After the partition, we assign a hyper-class for each instance $u_i$ from $\mathcal{U}_{noisy}$, then apply our Relaxed Contrastive

Loss for representation learning. First, we review two popular forms of contrastive loss, then introduce our new design.

**Self-supervised Contrastive Loss.** For an unlabeled image $u_i$ from a training batch containing $N = \mu B$ unlabeled images, a stochastic strong augmentation function $\mathcal{A}_s(\cdot)$ is applied to generate two augmented views, resulting in $2N$ samples. We define an affinity matrix $S \in \mathbb{R}^{2\mathbb{N}\times 2\mathbb{N}}$ where each element represents the dot product of embeddings with a temperature factor $\tau$:

$$s_{ij} = exp(z_i \cdot z_j/\tau) \tag{6}$$

The self-supervised contrastive loss InfoNCE [6, 13] for a positive pair of samples $(i, j)$ is generally formulated as:

$$\mathcal{L}_i^{\text{InfoNCE}} = -log\frac{exp(z_i \cdot z_j^*/\tau)}{\sum_{b=1}^{2N} \mathbb{1}_{[b\neq i]} exp(z_i \cdot z_b/\tau)} \tag{7}$$

where $z_j^*$ is from the other augmented view of the same image as $z_i$, $\mathbb{1}_{[b\neq i]} \in \{0, 1\}$ is an indicator function that equals 1 if $b \neq i$. Here a positive pair is defined as the two augmented views of the same instance and a negative pair consists of any two views originating from different instances. The loss is similar to minimizing the cross entropy loss between the $l_1$-normalized affinity matrix $S$ and a target contrastive matrix $W^{\text{self}} \in \mathbb{R}^{2N\times 2N}$ [24, 39], where each element in $W^{\text{self}}$ is defined as:

$$w_{ij}^{\text{self}} = \begin{cases} 1, & \text{if } z_i \text{ and } z_j \text{ are from the same sample} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The goal of self-supervised contrastive loss is to pull the features of one sample's two augmented views close and push the embeddings of different samples apart, regardless of whether samples belong to the same category. Hence the formulation contradicts the classification task as it requires well-clustered feature representations for each class [24,39]. A visualization of $W^{self}$ is given in Fig. 3 (a), where the connection between two views from the same image indicates a positive pair.

**Class-Aware Contrastive Loss**. To address this issue, *Class-Aware Class Assignment* is commonly adopted in existing methods [19, 21, 39, 42], inspired by supervised contrastive learning [16]. The contrastive matrix $W^{\text{self}}$ indicating positive and negative pair sampling turns into $W^{\text{class}}$:

$$w_{ij}^{\text{class}} = \begin{cases} w_{i,j}^{\text{re}}, & \text{if } \hat{y}_i = \hat{y}_j \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $\hat{y}_i$ and $\hat{y}_j$ represent the pseudo label for the $i$-th and $j$-th unlabeled sample, and $w_{i,j}^{re}$ is the re-weighted learning target that considers both samples' prediction confidence, *i.e.*, $w_{i,j}^{\text{re}} = \max(p_i) * \max(p_j)$. A clear illustration is given in Fig. 3 (b). By minimizing the difference between affinity matrix $S$ and $W^{\text{class}}$, two unlabeled instances with the same pseudo label are restrained to approach each other in the feature space, while instances with different pseudo labels are pushed to drift apart. This leads to better-clustered feature representations for each class.

Although showing appealing improvements, when the quality of the pseudo label is poor, the class-aware contrastive loss still suffers from the confirmation bias and performance degradation, which we show in Section 4.5. It will pull instances from two different classes close under the guidance of incorrect pseudo labels and deteriorate the representation learning.

**Relaxed Contrastive Loss**. To alleviate the negative impact of incorrect pseudo labels, we introduce the Relaxed Contrastive Loss. Instead of assigning a noisy unlabeled instance to its untrustworthy pseudo label class, we loose the assignment by categorizing it into a more reliable hyper-class, which is the union of top-$K$ nearest classes.

Take an unlabeled instance $u_i$ from $\mathcal{U}_{noisy}$ for example, by referring to its sorted class distribution $\text{sort}(\hat{p}_i)$, we select the first $K$ classes that own the largest probabilities to represent its nearest $K$ classes, denoted as $c_i = \{c_{i,1}, ..., c_{i,K}\}$. Then the hyper-class set is constructed as $\mathcal{HS}_i = \mathcal{S}_{i,1} \cup ...\mathcal{S}_{i,K}$, where $S_{i,k}$ includes all samples whose calibrated pseudo label $\text{argmax}(\hat{p}) == c_{i,k}$. Samples from $\mathcal{HS}_i$ are chosen to form positive pairs with $u_i$, and our relaxed contrastive matrix $W^{relax}$ is defined as:

$$w_{ij}^{\text{relax}} = \begin{cases} \hat{p}_i^{c_{i,k}} * \hat{p}_j^{c_{i,k}}, & \text{if } u_j \in \mathcal{HS}_i \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $c_{i,k} = argmax(\hat{p}_j)$. A re-weight strategy is adopted to comprehensively consider the unbiased semantic relations between two instances by combing the calibrated probability of $u_i$ and $u_j$ being classified into the class $c_{i,k}$. The larger probability indicates stronger semantic relations within the pair, hence the corresponding weight in $W^{\text{relax}}$ increases. A comparison between the relaxed target matrix with previous variants is shown in Fig. 3 (c).

By bridging the gap between affinity matrix $S$ and $W^{relax}$, we obtain the relaxed contrastive loss:

$$\mathcal{L}_{rel,i} = -\sum_{u_j\in\mathcal{HS}_i} w_{ij}^{\text{relax}} * log\frac{exp(z_i \cdot z_j/\tau)}{\sum_{b=1}^{2N} \mathbb{1}_{[b\neq i]} exp(z_i \cdot z_b/\tau)} \tag{11}$$

We anticipate the embedding of a noisy sample to fall into the top-$K$ nearest classes instead of a single mismatched class. On one hand, this improves the chance that two noisy instances from the same ground truth class are matched as a positive pair, thereby their embedding can be restrained to be similar. Besides, confirmation bias induced by incorrect pseudo labels can be regularized as we propose a partially different learning target compared to the consistency regularization in Eq. 1. For samples from clean subset $\mathcal{U}_{clean}$, we directly set $K = 1$ and relaxed contrastive loss degenerates to the naive class-aware cotrastive loss.

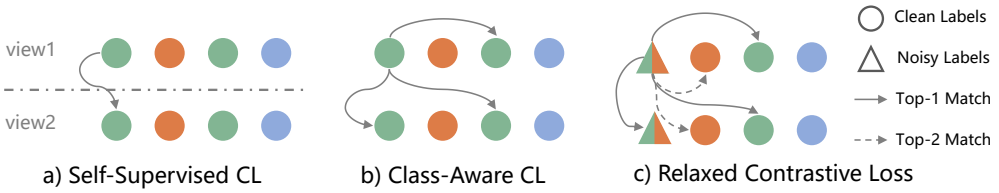



Figure 3. Illustrations of positive pairs in different contrastive loss. Pseudo label class is marked in different colors, arrows indicate positive pairs and CL stands for contrastive loss. (a) Only two views of the same image are considered as a positive pair in Self-Supervised CL. (b) Any two views with same pseudo label comparise a positive pair in Class-Aware CL. (c) The triangle containing two colors indicates a noisy sample matched with 2 nearest classes as positive pairs. Relaxed contrastive loss builds more positive pairs and increases the chance of finding the correct positive pair, meanwhile regularizing the overfitting to a single mismatched pair.

### 3.4. Discussion of Top-$K$ Selection

Top-$K$ is the most crucial hyper-parameter as it determines the covered range of a hyper-class. A larger choice of $K$ would increase the probabilities of an unlabeled instance matching with its groud-truth class, while at the same time, it would also add additional noise induced by those mismatched instance pairs. Thus, the selection of $K$ is a trade-off between the class clustering and noise resistance. We give detailed analysis in Section 4.4 about its selection.

## 4. Experiments

### 4.1. Experimental Setup

We first evaluate our method on mainstream SSL benchmarks including CIFAR10/CIFAR100 [17] and STL-10 [9] dataset, under different settings where the amount of labeled data varies. The best performance of an EMA model with a decay rate $0.999$ is reported in every single run [29], 5 runs for each dataset are conducted with different random seeds.

**Datasets.** CIFAR10 and CIFAR100 consist of 50K train and 10K test images from 10 and 100 classes, respectively. Following the widely used setting [29] on CIFAR10, we conduct experiments on $40, 250, 4000$ randomly selected labeled images in a class-balanced way and use all training samples as the unlabeled set. For CIFAR100, we randomly select $400, 2500, 10000$ images as labeled data. STL10 contains 5K labeled images (10 predefined folds) of size $96\times96$ from 10 classes and 100K unlabeled images. We use the first 5 predefined folds following [29].

**Implementation Detail.** As for model architectures, we use a Wide ResNet-28-2 [40] for CIFAR10 and WRN-28-8 for CIFAR100. A ResNet-18 network [14] is used for STL10 following [24]. We build a 2-layer MLP projection head to compute 64-dimensional embeddings for feature alignment. SGD optimizer with a momentum of $0.9$ is used. Instead of setting training epochs to 1024 in [29], we train for only 512 epochs with an initial learning rate $0.03$ and a cosine decay schedule to show our efficiency in convergence. By default, we set $K = 2$, $\tau_u = 0.95$, $\tau_g = 0.6$, $\mu = 7$, $B = 64$, $\lambda_1 = \lambda_2 = 1$ while only $\lambda_2 = 5$ in STL10.

**Baseline Methods**. We include current state-of-the-art methods adopting the consistency regularization technique such as FixMatch [29], MixMatch [3], and ReMixMatch [2]. Also, comparisons with previous methods that also combine contrastive learning like CoMatch [24], CCSSL [39], and SimMatch [42] is also given, to verify the improvements of HyperMatch.

### 4.2. Main Results

Performance of HyperMatch on different SSL benchmarks is reported in Tab. 1. On CIFAR10, our performance is on par with other state-of-the-art methods and we assume that the potential of semi-supervised learning has reached its accuracy upper bound, which makes it hard to be distinguished from existing works. Meanwhile, on the more complicated CIFAR100, HyperMatch significantly outperforms other methods, achieving $11.86\%$, $4.74\%$, $3.69\%$ gain over FixMatch with 400, 2500, and 10000 labeled samples and surpassing previous best results. The performance gains increase with fewer labeled samples as HyperMatch can effectively handle the unreliable pseudo labels.

STL10 contains unlabeled data drawn from a similar but different distribution from labeled data, making it a more practical challenge. HyperMatch achieves $2.97\%$ accuracy gains over the current best result of CCSSL and shows the adaptation ability to varied distributions of unlabeled data. To sum up, HyperMatch can remarkably boost the performance of existing SSL techniques, especially when the task is challenging and the pseudo labels are unreliable.

### 4.3. Semi-iNat 2021

Furthermore, we test HyperMatch on the more complex Semi-iNat 2021 [32], a complex real-world dataset where tough challenges like imbalanced distribution, domain mismatch and out-of-distribution classes exists. The labeled training set contains 9721 images from part of 810 species and validation set contains 4050 images, while the unlabeled set has 313248 images. We follow the settings in [39]. Images are resized to $224 \times 224$ and we use a ResNet-50 backbone with a 2-layer projection head. The same setting of $\tau_u = 0.6$, $B = 64$, $\mu = 7$, $\lambda_1 = 1$, $\lambda_2 = 2$ as in [39] is used. For other parameters in HyperMatch, we keep $K = 2$ and $\tau_g = 0.6$ as on other datasets.

| Method | CIFAR10 | | | CIFAR100 | | | STL10 |
|---|---|---|---|---|---|---|---|
| | 40 | 250 | 4000 | 400 | 2500 | 10000 | |
| MixMatch [3] | $52.46 \pm 11.5$ | $88.95 \pm 0.86$ | $93.58 \pm 0.10$ | $32.39 \pm 1.32$ | $60.06 \pm 0.37$ | $71.69 \pm 0.33$ | $38.02 \pm 8.29$ |
| ReMixMatch [2] | $80.90 \pm 9.64$ | $94.56 \pm 0.05$ | $95.28 \pm 0.13$ | $55.72 \pm 2.06$ | $72.57 \pm 0.31$ | $76.97 \pm 0.56$ | - |
| SSWPL [33] | - | - | - | - | $73.48 \pm 0.45$ | $79.12 \pm 0.85$ | - |
| LaplaceNet [28] | - | - | $95.35 \pm 0.07$ | - | $68.36 \pm 0.02$ | $73.40 \pm 0.23$ | - |
| FixMatch(RA) [29] | $86.19 \pm 3.37$ | $94.93 \pm 0.65$ | $95.74 \pm 0.05$ | $51.15 \pm 1.75$ | $71.71 \pm 0.11$ | $77.40 \pm 0.12$ | $65.38 \pm 0.42$ |
| CoMatch [24] | $93.09 \pm 1.39$ | $95.09 \pm 0.33$ | $95.44 \pm 0.20$ | $58.11 \pm 2.34$ | $71.63 \pm 0.35$ | $79.14 \pm 0.36$ | $79.80 \pm 0.38$ |
| SimMatch [42] | $\mathbf{94.40 \pm 1.37}$ | $\mathbf{95.16 \pm 0.39}$ | $96.04 \pm 0.01$ | $62.19 \pm 2.21$ | $74.93 \pm 0.32$ | $79.42 \pm 0.11$ | - |
| CCSSL [39] | $90.83 \pm 2.78$ | $94.86 \pm 0.55$ | $95.54 \pm 0.20$ | $61.19 \pm 1.65$ | $75.7 \pm 0.63$ | $80.68 \pm 0.16$ | $80.01 \pm 1.39$ |
| **HyperMatch** | $93.92 \pm 1.10$ | $95.01 \pm 0.23$ | $\mathbf{96.05 \pm 0.12}$ | $\mathbf{63.01 \pm 0.57}$ | $\mathbf{76.45 \pm 0.35}$ | $\mathbf{81.09 \pm 0.28}$ | $\mathbf{82.98 \pm 0.37}$ |

Table 1. Top-1 accuracy comparisons with other methods on CIFAR10, CIFAR100 and STL10 dataset with varied labeled samples.

| Method | Semi-iNat 2021 | |
|---|---|---|
| | From Scratch | Moco Pretrain |
| Supervised | 19.09 | 34.96 |
| CoMatch [24] | 20.94 | 38.94 |
| FixMatch [29] | 21.41 | 40.3 |
| CCSSL (CoMatch) [39] | 24.12 | 39.85 |
| CCSSL (FixMatch) [39] | 31.21 | 41.28 |
| HyperMatch (FixMatch) | **33.47** | **42.57** |

Table 2. Comparisons on Semi-iNat 2021. When training from MoCo [13], the first three blocks are freezed.

| SS-CL | CA-CL | R-CL | re-weight | GMM Partition | CIFAR100 | |
|---|---|---|---|---|---|---|
| | | | | | 400 | 2500 |
| | | | | | 57.73 | 72.85 |
| ✓ | | | | | 56.62 | 72.43 |
| | ✓ | | | | 60.76 | 75.45 |
| | ✓ | | ✓ | | 61.69 | 75.82 |
| | | ✓ | | | 61.92 | 76.12 |
| | | ✓ | ✓ | | 62.35 | 76.63 |
| | | ✓ | ✓ | ✓ | **63.22** | **77.01** |

Table 3. Different combinations of our algorithms on CIFAR100.

See Tab. 2, although without tuning any hyper-parameters in relaxed contrastive loss, we still achieve better results than CCSSL that requires tuing the threshold for separating out-of-distribution data. We improve by $2.26\%$ and $0.87\%$ over CCSSL on training from scratch and from MoCo [13] pretrained weights. This verifies the robustness of HyperMatch even handling out-of-distribution data.

### 4.4. Ablation Study

**Analysis of Each Technique.** We investigate each technique in our work and results are given in Tab. 3. For simplicity, self-supervised, class-aware and relaxed contrastive loss are referred as SS-CL, CA-CL and R-CL, respectively.

SS-CL deteriorates the performance as it pushes away instances within the same class as discussed above. By introducing pseudo labels as priors, CA-CL helps learn well-clustered features and improves overall accuracy. Also, note that CA-CL combined with re-weighting is equivalent to CCSSL [39] on in-distribution data. Directly applying our R-CL already outperforms other losses and using a re-weighting strategy further improves the results slightly.

| $\tau_g$ | CIFAR10@250 | CIFAR100@400 |
|---|---|---|
| 0 | 94.62 | 61.2 |
| 0.2 | 94.81 | 62.34 |
| 0.4 | 95.06 | 62.23 |
| 0.6 | **95.14** | **62.84** |
| 0.8 | 95.11 | 62.44 |
| 1.0 | 94.89 | 61.8 |

Table 4. Different thresholds $\tau_g$ on CIFAR datasets.

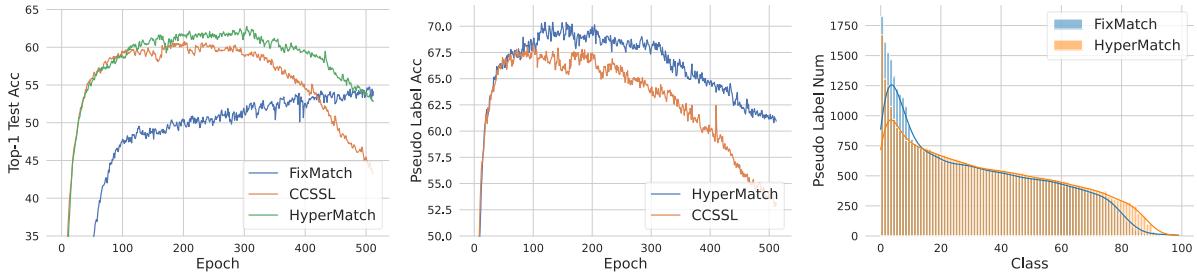
| Top-$K$ | 1 | 2 | 3 | 5 |
|---|---|---|---|---|
| CIFAR100@400 | 60.51 | **62.84** | 61.24 | 60.67 |
| CIFAR100@2500 | 75.87 | 76.62 | **76.73** | 75.86 |

Table 5. Experiments of using different $K$ in relaxed contrastive loss. When $K = 1$, it turns into class-aware contrastive loss.

Moreover, by adding GMM to split the clean and noisy pseudo labels and then only applying R-CL on noisy data, we obtain the best results. This proves each technique contributes to the final advantage of our method.

**GMM Threshold.** We analyze the effect of varied thresholds $\tau_g$ for the pseudo label partition in Tab. 4. $\tau_g$ controls the relative ratio of the clean subset to the noisy subset. When $\tau_g = 0$, the relaxed contrastive loss becomes class-aware contrastive loss as all unlabeled samples are divided into the clean subset. $\tau_g = 1$ indicates that all instances fall into the noisy set. Setting $\tau_g > 0$ consistently improves the accuracy compared to $\tau_g = 0$, which proves our algorithm is robust with the selection of threshold $\tau_g$.

**Top-K Selections.** We also test different choices of $K$ in relaxed contrastive loss in Tab. 5. $K = 1$ indicates the class-aware contrastive loss. Intuitively, a large $K$ increases the potential for an unreliable instance to find its ground-truth class while also adding noise to the loss through mismatched instance pairs. $K = 2$ or 3 achieves the best accuracy on CIFAR100 with 400 or 2500 labeled data, and further increasing $K$ leads to a performance drop as expected. Note that the performance ($60.67\%$) of a relatively large $K = 5$ is still on par with the performance ($60.51\%$) of $K = 1$, which indicates the under-exploitation of class relationships in class-aware contrastive loss.




a) Test Accuracy b) Pseudo Label Accuracy c) Pseudo Label Class Distributions

Figure 4. Qualitative Analysis of HyperMatch. (a) Top-1 test accuracy curve of different methods on CIFAR100@400; (b) Top-1 accuracy of pseudo labels above the predefined confidence threshold $\tau_u = 0.95$ on CIFAR100@400; (c) Class distributions of pseudo label numbers on the training set of CIFAR100 sorted in a descending order.

| Method | w/o $q$ | w/ $q$ |
|---|---|---|
| Pseudo Label Acc of $\mathcal{U}_{clean}$ | 65.32 | **68.13** |
| Pseudo Label Acc of $\mathcal{U}_{noisy}$ | 32.48 | **33.33** |

Table 6. Remove distribution calibration on CIFAR100@400.

| Method | CIFAR100 | |
|---|---|---|
| | 400 | 2500 |
| CoMatch | 58.11 | 71.63 |
| CoMatch + CCSSL [39] | 59.21 | 73.52 |
| CoMatch + HyperMatch | **60.85** | **75.2** |

Table 7. Deploy relaxed contrastive loss to CoMatch.

**Distribution Calibration.** To validate the effect of distribution calibration by integrating semantic similarities, we remove the calibration and report pseudo label accuracy of $\mathcal{U}_{clean}$ and $\mathcal{U}_{noisy}$ in Tab. 6. Adding semantic similarity achieves larger accuracy gap between two sets, which implies correct and wrong pseudo labels are better separated.

**Deployment to different SSL architectures.** As a variant of contrastive loss, relaxed contrastive loss can also be conveniently plugged into other SSL techniques. We further deploy on CoMatch to verify its generalization, shown in Tab. 7. Adding relaxed contrastive loss improves $2.74\%$, $3.27\%$ over CoMatch on CIFAR100@400 and CIFAR100@2500 settings and outperfroms CCSSL using naive class-aware contrastive loss.

### 4.5. Qualitative Analysis

**Convergence Speed.** It's validated in previous works [21, 24, 39, 42] that class-aware contrastive loss learns well-clustered feature representations and improves the convergence. Shown in Fig. 4 (a), HyperMatch shares the attribute of fast convergence as CCSSL [39] and reaches the higher accuracy of $63.22\%$ at epoch 300, while FixMatch [29] needs more iterations and ends up with inferior result.

**Pseudo Label Accuracy.** We compare the accuracy of pseudo labels above the confidence threshold $\tau_u = 0.95$ of HyperMatch with CCSSL [39] in Fig. 4 (b). It is evident that in the late training stage, HyperMatch still maintains better accuration of pseudo labels than CCSSL and the accuracy drops by only $9.28\%$ from the best one while CCSSL significantly drops by $15.4\%$ in Fig. 4 (b), which implies the better alleviation of confirmation bias.

**Mitigation of Imbalanced Distributions.** As mentioned in [43], a disparate impact exists in SSL methods that a class with a higher baseline accuracy would benefit more from SSL. Here we claim the same behaviour by plotting the class distributions of pseudo labels sorted in descending order on 50K training images in Fig. 4 (c). FixMatch exhibits an obvious long-tailed pattern, which exacerbates confirmation bias as more predictions from head classes are added as pseudo labels. HyperMatch alleviates this imbalance by associating unlabeled data with multiple nearest classes as pseudo labels. This partially explains the improvement of HyperMatch on final performance.

### 4.6. Limitations

HyperMatch shows significant improvements with noisy pseudo labels and limited labeled samples. When pseudo label predictions are already accurate enough, the gains become smaller. Meanwhile, with only a fraction of labeled samples, training fluctuations could affect the results such as in CIFAR10@40 experiment. Averaging results over different runs would mitigate the problem.

## 5. Conclusion

Here, we proposed a novel SSL method, HyperMatch, to effectively handle the utilization of noisy unlabeled data while resisting the inference of wrong pseudo labels. The core of our design is to relax the previously error-prone class assignment by categorizing noisy data into the hyper-class, which is the union of top-$K$ nearest classes. Accompanied by the calibrated class distribution to find noisy data, HyperMatch achieves the state-of-the-art, and further analysis also gives insight to its intrinsic noise-tolerant abilities. **In future**, we'll explore HyperMatch under extreme long-tailed distributions and on out-of-distribution data.

## References


[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. 2

[2] David Berthelot, Nicholas Carlini, Ekin Dogus Cubuk, Alexey Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *ICLR*, 2020. 1, 2, 3, 6, 7

[3] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *ArXiv*, abs/1905.02249, 2019. 1, 3, 6, 7

[4] Mehdi Boroumand, Mo Chen, and Jessica J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14:1181–1193, 2019. 1

[5] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20, 2006. 1

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020. 1, 3, 5

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. *ArXiv*, abs/2006.10029, 2020. 3

[8] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *ArXiv*, abs/2003.04297, 2020. 1

[9] Adam Coates, A. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011. 6

[10] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3008–3017, 2020. 3

[11] Jean-Bastien Grill, Florian Strub, Florent Altch'e, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *ArXiv*, abs/2006.07733, 2020. 1

[12] Tao Han, Junyu Gao, Yuan Yuan, and Qi Wang. Unsupervised semantic aggregation and deformable template matching for semi-supervised learning. *ArXiv*, abs/2010.05517, 2020. 4

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. 1, 3, 5, 7

[14] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[15] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *ArXiv*, abs/2011.00362, 2020. 3

[16] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *ArXiv*, abs/2004.11362, 2020. 1, 3, 5

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 6

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012. 1

[19] Xin Lai, Zhuotao Tian, Li Jiang, Shu Liu, Hengshuang Zhao, Liwei Wang, and Jiaya Jia. Semi-supervised semantic segmentation with directional context-aware consistency. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1205–1214, 2021. 1, 5

[20] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *ArXiv*, abs/1610.02242, 2017. 1

[21] Doyup Lee, Sungwoong Kim, Ildoo Kim, Yeongjae Cheon, Minsu Cho, and Wook-Shin Han. Contrastive regularization for semi-supervised learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3910–3919, 2022. 1, 2, 3, 4, 5, 8

[22] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2013. 1, 2

[23] Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *ArXiv*, abs/2002.07394, 2020. 4

[24] Junnan Li, Caiming Xiong, and Steven C. H. Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9455–9464, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[25] Shikun Liu, Shuaifeng Zhi, Edward Johns, and Andrew J. Davison. Bootstrapping semantic segmentation with regional contrast. *ArXiv*, abs/2104.04465, 2022. 1

[26] Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:1979–1993, 2019. 1, 2

[27] Youngtaek Oh, Dong jin Kim, and In-So Kweon. Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9776–9786, 2022. 4

[28] Philip Sellars, Angelica I. Avilés-Rivero, and Carola-Bibiane Schönlieb. Laplacenet: A hybrid energy-neural model for deep semi-supervised classification. *ArXiv*, abs/2106.04527, 2021. 7

[29] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin Dogus Cubuk, Alexey Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *ArXiv*, abs/2001.07685, 2020. 1, 2, 3, 6, 7, 8

[30] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *ArXiv*, abs/2005.04757, 2020. 1

[31] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014. 2

[32] Jong-Chyi Su and Subhransu Maji. The semi-supervised inaturalist challenge at the fgvc8 workshop. *ArXiv*, abs/2106.01364, 2021. 6

[33] Fariborz Taherkhani, Ali Dabouei, Sobhan Soleymani, Jeremy M. Dawson, and Nasser M. Nasrabadi. Self-supervised wasserstein pseudo-labeling for semi-supervised image classification. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12262–12272, 2021. 7

[34] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, 2017. 2

[35] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018. 3

[36] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Loddon Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10852–10861, 2021. 2

[37] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 3

[38] Qizhe Xie, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2020. 1, 2

[39] Fan Yang, Kaixing Wu, Shuyi Zhang, Guannan Jiang, Yong Liu, Feng Zheng, Wei Zhang, Chengjie Wang, and Long Zeng. Class-aware contrastive semi-supervised learning. *ArXiv*, abs/2203.02261, 2022. 1, 2, 3, 5, 6, 7, 8

[40] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *ArXiv*, abs/1605.07146, 2016. 6

[41] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *NeurIPS*, 2021. 3

[42] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. *ArXiv*, abs/2203.06915, 2022. 1, 2, 3, 5, 6, 7, 8

[43] Zhaowei Zhu, Tianyi Luo, and Yang Liu. The rich get richer: Disparate impact of semi-supervised learning. *ArXiv*, abs/2110.06282, 2022. 8