CoMatch: Semi-supervised Learning with Contrastive Graph Regularization

Junnan Li Caiming Xiong Steven C.H. Hoi
Salesforce Research
{junnan.li,cxiong,shoi}@salesforce.com

Abstract

Semi-supervised learning has been an effective paradigm for leveraging unlabeled data to reduce the reliance on labeled data. We propose CoMatch, a new semi-supervised learning method that unifies dominant approaches and addresses their limitations. CoMatch jointly learns two representations of the training data, their class probabilities and low-dimensional embeddings. The two representations interact with each other to jointly evolve. The embeddings impose a smoothness constraint on the class probabilities to improve the pseudo-labels, whereas the pseudo-labels regularize the structure of the embeddings through graph-based contrastive learning. CoMatch achieves state-of-the-art performance on multiple datasets. It achieves substantial accuracy improvements on the label-scarce CIFAR-10 and STL-10. On ImageNet with 1% labels, CoMatch achieves a top-1 accuracy of 66.0%, outperforming FixMatch [32] by 12.6%. Furthermore, CoMatch achieves better representation learning performance on downstream tasks, outperforming both supervised learning and self-supervised learning. Code and pre-trained models are available at https://github.com/salesforce/CoMatch/.

1. Introduction

Semi-supervised learning (SSL) – learning from few labeled data and a large amount of unlabeled data – has been a long-standing problem in computer vision and machine learning. Recent state-of-the-art methods mostly follow two trends: (1) using the model's class prediction to produce a pseudo-label for each unlabeled sample as the label to train against [19, 2, 1, 32]; (2) unsupervised or self-supervised pre-training, followed by supervised fine-tuning [5, 14, 13, 3] and pseudo-labeling [6].

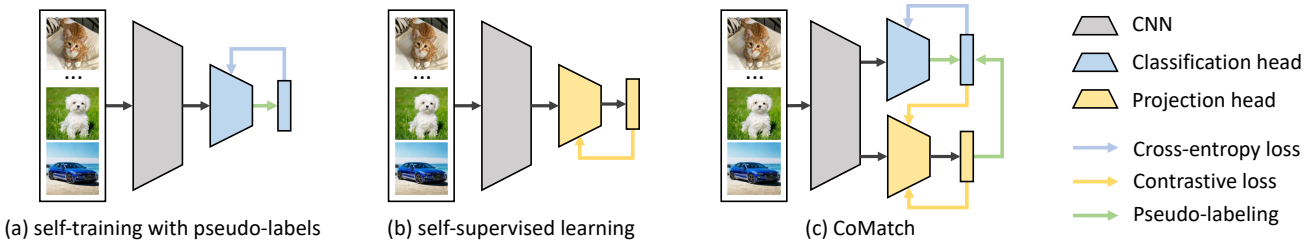
However, existing methods have several limitations. Pseudo-labeling (also called self-training) methods heavily rely on the quality of the model's class prediction, thus suffering from confirmation bias where the prediction mistakes would accumulate. Self-supervised learning methods are task-agnostic, and the widely adopted contrastive learning [5, 14] may learn representations that are suboptimal for the specific classification task. Another branch of methods explore graph-based semi-supervised learning [24, 17], but have yet shown competitive performance especially on larger datasets such as ImageNet [9].

We propose CoMatch, a new semi-supervised learning method that addresses the existing limitations. A conceptual illustration is shown in Figure 1. In CoMatch, each image has two compact representations: a class probability produced by the classification head and a low-dimensional embedding produced by the projection head. The two representations interact with each other and jointly evolve in a co-training framework. Specifically, the classification head is trained using memory-smoothed pseudo-labels, where pseudo-labels are refined by aggregating information from nearby samples in the embedding space. The projection head is trained using contrastive learning on a pseudo-label graph, where samples with similar pseudo-labels are trained to have similar embeddings. CoMatch unifies dominant ideas including consistency regularization, entropy minimization, contrastive learning, and graph-based SSL.

We perform experiments on multiple datasets and compare with state-of-the-art semi-supervised and self-supervised methods. CoMatch substantially outperforms all baselines across all benchmarks, especially in label-scarce scenarios. On CIFAR-10 with 4 labeled samples per class, CoMatch outperforms FixMatch [32] by 6.11% in accuracy. On STL-10, CoMatch outperforms FixMatch by 13.27%. On ImageNet with only 1% of labels, CoMatch achieves a top-1 accuracy of 66.0% (67.1% with self-supervised pre-training), whereas the best baseline (MoCov2 [7] followed by FixMatch [32]) has an accuracy of 59.9%. Furthermore, we demonstrate that CoMatch achieves better representation learning performance on down-stream image classification and object detection tasks, outperforming both supervised learning and self-supervised learning.

2. Background

To set the stage for CoMatch, we first introduce existing SSL methods, mainly focusing on current state-of-the-art methods that are relevant. More comprehensive reviews



(a) self-training with pseudo-labels (b) self-supervised learning (c) CoMatch

Figure 1: Conceptual illustration of different methods that leverage unlabeled data. (a) Task-specific self-training: the model predicts class probabilities for the unlabeled samples as the pseudo-label to train against [19, 2, 1, 32]. (b) Task-agnostic self-supervised learning: the model projects samples into low-dimensional embeddings, and performs contrastive learning to discriminate embeddings of different images [35, 5, 14]. (c) CoMatch: class probabilities and embeddings interact with each other and jointly evolve in a co-training framework. The embeddings impose a smoothness constraint on the class probabilities to improve the pseudo-labels. The pseudo-labels are used as the target to train both the classification head with a cross-entropy loss, and the projection head with a graph-based contrastive loss.

can be found in [42, 34]. In the following, we refer to a deep encoder network (a convolutional neural network) as $f(\cdot)$, which produces a high-dimensional feature $f(x)$ given an input image x. A classification head (a fully-connected layer followed by softmax) is defined as $h(\cdot)$, which outputs a distribution over classes $p(y|x) = h(f(x))$. We also define a non-linear projection head (a MLP) $g(\cdot)$, which transforms a feature $f(x)$ into a normalized low-dimensional embedding $z(x) = g(f(x))$.

Consistency regularization is a crucial piece for many state-of-the-art SSL methods. It utilizes the assumption that a classifier should output the same class probability for an unlabeled sample even after it is augmented. In the simplest form, prior works [31, 18] add the following consistency regularization loss on unlabeled samples:

$$\|p(y|\mathrm{Aug}(x)) - p(y|\mathrm{Aug}(x))\|_2^2\,, \tag{1}$$

where $\mathrm{Aug}(\cdot)$ is a stochastic transformation that does not alter the label of the image. Mean Teacher [33] replaces one of the terms in eq.(1) with the output of an EMA model. VAT [26] uses an adversarial transformation in place of Aug. MixMatch [2] averages predictions across multiple augmentations to produce $p(y)$. UDA [36], ReMixMatch [1], and FixMatch [32] use a cross-entropy loss in place of the squared error, and apply stronger augmentation.

Entropy minimization is a common method in many SSL algorithms, which encourages the classifier's decision boundary to pass through low-density regions of the data distribution. It is either achieved explicitly by minimizing the entropy of $p(y|x)$ on unlabeled samples [12], or implicitly by constructing low-entropy **pseudo-labels** on unlabeled samples and using them as training targets in a cross-entropy loss [19, 2, 1, 32]. Some methods [36, 2, 1] post-process the "soft" pseudo-labels with a sharpening function to reduce entropy, whereas FixMatch [32] produces "hard" pseudo-labels for samples whose largest class probability fall above a predefined threshold. Most methods [32, 1, 36] use weakly-augmented samples to produce pseudo-labels and train the model on strongly-augmented samples. However, since the pseudo-labels purely rely on the classifier, such self-training strategy suffers from the confirmation bias problem, where the error in the pseudo-labels would accumulate and harms learning.

Self-supervised contrastive learning has attracted much attention, due to its ability to leverage unlabeled data for model pre-training. The widely adopted contrastive learning [35, 28, 5, 6, 14, 20] optimizes for the task of instance discrimination, and formulates the loss using the normalized low-dimensional embeddings z:

$$-\log \frac{\exp(z(\mathrm{Aug}(x_i)) \cdot z(\mathrm{Aug}(x_i))/t)}{\sum_{j=1}^{N} \exp(z(\mathrm{Aug}(x_i)) \cdot z(\mathrm{Aug}(x_j))/t)} \tag{2}$$

where $\mathrm{Aug}(\cdot)$ is a stochastic transformation similar as in eq.(1), and x_j include x_i and $N-1$ other images (*i.e.* negative samples). Self-supervised contrastive learning can be interpreted as a form of class-agnostic consistency regularization, which enforces the same image with different augmentations to have similar embeddings, while different images have different embeddings. Among recent methods, SimCLR [5] uses images from the same batch to calculate pairwise similarity, whereas MoCo [14] maintains a queue of embeddings from an EMA model.

Self-supervised pre-training followed by supervised fine-tuning has shown strong performance on semi-supervised learning tasks [5, 14, 13, 21, 3]. SimCLR v2 [6] further utilizes larger models for distillation. However, since self-supervised learning is a task-agnostic process, the contrastive loss in eq.(2) optimizes for an objective that partially contradicts with task-specific learning. It enforces images from the same class to have different representations, which is undesirable for classification tasks.

Graph-based semi-supervised learning defines the similarity of data samples with a graph and encourages smooth predictions with respect to the graph structure [40, 41]. Recent works use deep networks to generate graph representations. [17, 22] perform iterative label propagation and

network training. [24, 4] connect data samples that have the same pseudo-labels, and perform metric learning to enforce connected samples to have similar representations. However, these methods define representations as the high-dimensional feature $f(x)$, which leads to several limitations: (1) since the features are highly-correlated with the class predictions, the same types of errors are likely to exist in both the feature space and the label space; (2) due to the curse of dimensionality, Euclidean distance becomes less meaningful; (3) computation cost is high which harms the scalability of the methods. Furthermore, the loss functions in [24, 4] consider the absolute distance between pairs, whereas CoMatch optimizes for relative distance.

3. Method

3.1. Overview

In this section, we introduce our proposed semi-supervised learning method. Different from most existing semi-supervised and self-supervised learning methods, CoMatch jointly learns the encoder $f(\cdot)$, the classification head $h(\cdot)$, and the projection head $g(\cdot)$. Given a batch of B labeled samples $\mathcal{X} = \{(x_b, y_b)\}_{b=1}^{B}$ where y_b are one-hot labels, and a batch of unlabeled samples $\mathcal{U} = \{u_b\}_{b=1}^{\mu B}$ where μ determines the relative size of $\mathcal{X}$ and $\mathcal{U}$, CoMatch jointly optimizes three losses: (1) a supervised classification loss on labeled data $\mathcal{L}_x$, (2) an unsupervised classification loss on unlabeled data $\mathcal{L}_u^{cls}$, and (3) a graph-based contrastive loss on unlabeled data $\mathcal{L}_u^{ctr}$. Specifically, $\mathcal{L}_x$ is defined as the cross-entropy between the ground-truth labels and the model's predictions:

$$\mathcal{L}_x = \frac{1}{B}\sum_{b=1}^{B} \mathrm{H}(y_b, p(y|\mathrm{Aug_w}(x_b))), \tag{3}$$

where $\mathrm{H}(y, p)$ denotes the cross-entropy between two distributions y and p, and $\mathrm{Aug_w}$ refers to weak augmentations.

The unsupervised classification loss $\mathcal{L}_u^{cls}$ is defined as the cross-entropy between the pseudo-labels q_b and the model's predictions:

$$\mathcal{L}_u^{cls} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}(\max q_b \geq \tau)\mathrm{H}(q_b, p(y|\mathrm{Aug_s}(u_b))), \tag{4}$$

where $\mathrm{Aug_s}$ refers to strong augmentations. Following FixMatch [32], we retain pseudo-labels whose largest class probability are above a threshold τ. Different from FixMatch, our soft pseudo-labels q_b are not converted to hard labels for entropy minimization. Instead, we achieve entropy minimization by optimizing the contrastive loss $\mathcal{L}_u^{ctr}$. Section 3.2 explains the details of pseudo-labelling and contrastive learning.

Our overall training objective is:

$$\mathcal{L} = \mathcal{L}_x + \lambda_{cls}\mathcal{L}_u^{cls} + \lambda_{ctr}\mathcal{L}_u^{ctr}, \tag{5}$$

where λ_{cls} and λ_{ctr} are scalar hyperparameters to control the weight of the unsupervised losses.

3.2. CoMatch

In CoMatch, the high-dimensional feature of each sample is transformed to two compact representations: its class probability p and its normalized low-dimensional embedding z, which reside in the label space and the embedding space, respectively. Given a batch of unlabeled samples $\mathcal{U}$, we first perform memory-smoothed pseudo-labeling on weak augmentations $\mathrm{Aug_w}(\mathcal{U})$ to produce pseudo-labels. Then, we construct a pseudo-label graph W^q which defines the similarity of samples in the label space. We use W^q as the target to train an embedding graph W^z, which measures the similarity of strongly-augmented samples $\mathrm{Aug_s}(\mathcal{U})$ in the embedding space. An illustration of CoMatch is shown in Fig 2, and a pseudo-code is given in the appendix. Next, we first introduce the pseudo-labeling process, then we describe the graph-based contrastive learning algorithm.

Memory-smoothed pseudo-labeling aims to mitigate confirmation bias by leveraging the structure of the embeddings to refine pseudo-labels. Given each sample in $\mathcal{X}$ and $\mathcal{U}$, we first obtain its class probability. For a labeled sample, it is defined as the ground-truth label: $p^w = y$. For an unlabeled sample, it is defined as the model's prediction on its weak-augmentation: $p^w = h \circ f(\mathrm{Aug_w}(u))$. Following [1], we perform distribution alignment (DA) on unlabeled samples: $p^w = \mathrm{DA}(p^w)$. DA prevents the model's prediction from collapsing to certain classes. Specifically, we maintain a moving-average $\tilde{p}^w$ of p^w during training, and adjust the current p^w with $p^w = \mathrm{Normalize}(p^w/\tilde{p}^w)$, where $\mathrm{Normalize}(p)_i = p_i/\sum_j p_j$ renormalizes the scaled result to a valid probability distribution.

For each sample in $\mathcal{X}$ and $\mathcal{U}$, we also obtain its embedding z^w by forwarding the weakly-augmented sample through f and g. Then, we create a memory bank to store class probabilities and embeddings of the past K weakly-augmented samples: $\mathrm{MB} = \{(p_k^w, z_k^w)\}_{k=1}^{K}$. The memory bank contains both labeled samples and unlabeled samples and is updated with first-in-first-out strategy.

For each unlabeled sample u_b in the current batch with p_b^w and z_b^w, we generate a pseudo-label q_b by aggregating class probabilities from neighboring samples in the memory bank. Specifically, we optimize the following objective:

$$J(q_b) = (1-\alpha)\sum_{k=1}^{K} a_k \left\|q_b - p_k^w\right\|_2^2 + \alpha \left\|q_b - p_b^w\right\|_2^2 \tag{6}$$

The first term is a smoothness constraint which encourages q_b to take a similar value as its nearby samples' class probabilities, whereas the second term attempts to maintain its

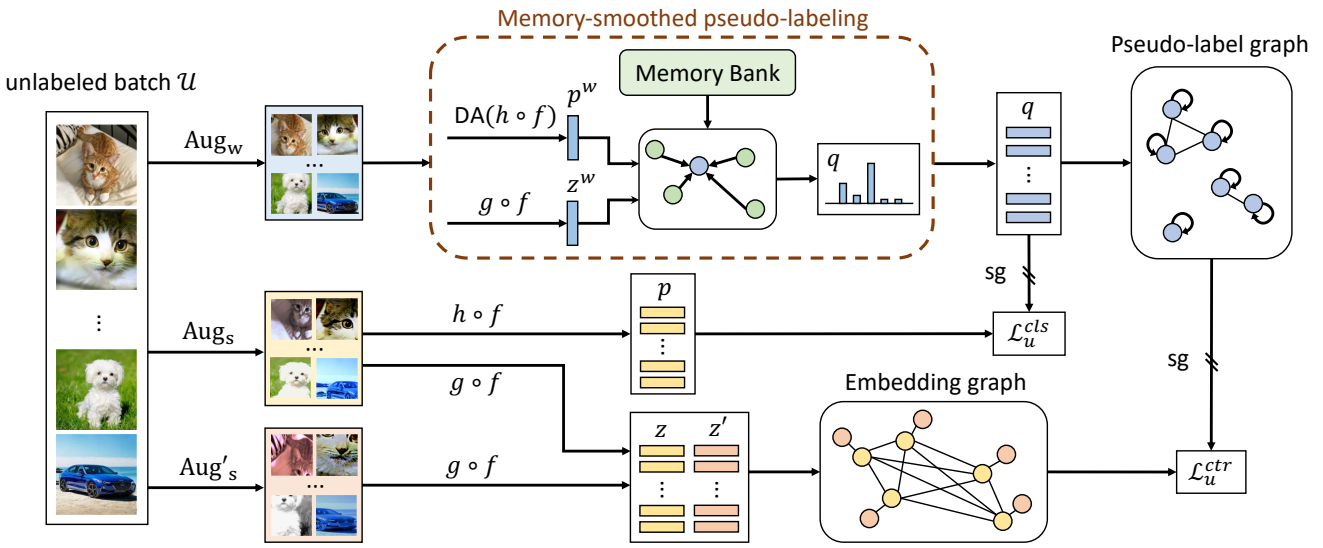


Figure 2: Framework of the proposed CoMatch. Given a batch of unlabeled images, their weakly-augmented images are used to produce memory-smoothed pseudo-labels, which are used as targets to train the class prediction on strongly-augmented images. A pseudo-label graph with self-loop is constructed to measure the similarity between samples, which is used to train an embedding graph such that images with similar pseudo-labels have similar embeddings. sg means stop-gradient. $\circ$ means that two functions are applied consecutively.

original class prediction. a_k measures the affinity between the current sample and the k-th sample in the memory, and is computed using similarity in the embedding space:

$$a_k = \frac{\exp(z_b^w \cdot z_k^w / t)}{\sum_{k=1}^{K} \exp(z_b^w \cdot z_k^w / t)}, \tag{7}$$

where t is a scalar temperature parameter.

Since a_k is normalized (*i.e.* a_k sums to one), the minimizer for $J(q_b)$ can be derived as:

$$q_b = \alpha p_b^w + (1-\alpha) \sum_{k=1}^{K} a_k p_k^w. \tag{8}$$

Graph-based contrastive learning aims to learn representations guided by a pseudo-label graph. Given the pseudo-labels $\{q_b\}_{b=1}^{\mu B}$ for the batch of unlabeled samples, we build the pseudo-label graph by constructing a similarity matrix W^q of size $\mu B \times \mu B$:

$$W_{bj}^q = \begin{cases} 1 & \text{if } b = j \\ q_b \cdot q_j & \text{if } b \neq j \text{ and } q_b \cdot q_j \geq T \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Samples with similarity lower than a threshold T are not connected, and each sample is connected to itself with the strongest edge of value 1 (*i.e.* self-loop).

The pseudo-label graph serves as the target to train an embedding graph. To construct the embedding graph, we first perform two strong augmentations on each unlabeled sample $u_b \in \mathcal{U}$, and obtain their embeddings $z_b = g \circ f(\text{Aug}_\text{s}(u_b))$, $z'_b = g \circ f(\text{Aug}'_\text{s}(u_b))$. Then we build the embedding graph W^z as:

$$W_{bj}^z = \begin{cases} \exp(z_b \cdot z'_b / t) & \text{if } b = j \\ \exp(z_b \cdot z_j / t) & \text{if } b \neq j \end{cases} \tag{10}$$

We aim to train the encoder f and the projection head g such that the embedding graph has the same structure as the pseudo-label graph. To this end, we first normalize W^q and W^z with $\hat{W}_{bj} = W_{bj} / \sum_j W_{bj}$, so that each row of the similarity matrix sums to 1. Then we minimize the cross-entropy between the two normalized graphs. The contrastive loss is defined as:

$$\mathcal{L}_u^{ctr} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \text{H}(\hat{W}_b^q, \hat{W}_b^z) \tag{11}$$

$\text{H}(\hat{W}_b^q, \hat{W}_b^z)$ can be decomposed into two terms:

$$-\hat{W}_{bb}^q \log\left(\frac{\exp(z_b \cdot z'_b / t)}{\sum_{j=1}^{\mu B} \hat{W}_{bj}^z}\right) - \sum_{j=1, j\neq b}^{\mu B} \hat{W}_{bj}^q \log\left(\frac{\exp(z_b \cdot z_j / t)}{\sum_{j=1}^{\mu B} \hat{W}_{bj}^z}\right) \tag{12}$$

The first term is a self-supervised contrastive loss that comes from the self-loops in the pseudo-label graph. It encourages the model to produce similar embeddings for different augmentations of the same image, which is a form of consistency regularization. The second term encourages samples with similar pseudo-labels to have similar embeddings. It gathers samples from the same class into clusters, which achieves entropy minimization.

During training, a natural curriculum would occur from CoMatch. The model would start with producing low-

confidence pseudo-labels, which leads to a sparse pseudo-label graph. As training progresses, samples are gradually clustered, which in turns leads to more confident pseudo-labels and more connections in the pseudo-label graph.

Another advantage of CoMatch appears in open-set semi-supervised learning, where the unlabeled data contains out-of-distribution (ood) samples. Due to the smoothness constraint, ood samples would have low-confidence pseudo-labels. Therefore, they are less connected to in-distribution samples, and will be pushed further away from in-distribution samples by the proposed contrastive loss.

3.3. Scalable learning with an EMA model

In order to build a meaningful pseudo-label graph, the unlabeled batch of data should contain a sufficient number of samples from each class. While this requirement can be easily satisfied for datasets with a small number of classes (*e.g.* CIFAR-10), it becomes difficult for large datasets with more classes (*e.g.* ImageNet) because a large unlabeled batch would exceed the memory capacity of 8 commodity GPUs (*e.g.* NVIDIA V100). Therefore, we improve CoMatch for SSL on large-scale datasets.

Inspired by MoCo [14] and Mean Teacher [33], we introduce an EMA model $\{\bar{f}, \bar{g}, \bar{h}\}$ whose parameters $\bar{\theta}$ are the moving-average of the original model's parameters θ:

$$\bar{\theta} \leftarrow m\bar{\theta} + (1-m)\theta. \tag{13}$$

The advantage of the EMA model is that it can evolve smoothly as controlled by the momentum parameter m.

We also introduce a momentum queue which stores the pseudo-labels and the strongly-augmented embeddings for the past K unlabeled samples: $\mathrm{MQ} = \{(\bar{q}_k, \bar{z}_k = \bar{g} \circ \bar{f}(\mathrm{Aug_s}'(u_k)))\}_{k=1}^{K}$, where $\bar{q}_k$ and $\bar{z}_k$ are produced using the EMA model. Different from the memory bank, the momentum queue only contains unlabeled samples.

We modify the pseudo-label graph W^q to have a size of $\mu B \times K$. It defines the similarity between each sample in the current batch and each sample in the momentum queue (which also contains the current batch). Different from eqn.(9), the similarity is now calculated as $\bar{q}_b \cdot \bar{q}_j$, where $b = \{1, ..., \mu B\}$ and $j = \{1, ..., K\}$.

The embedding graph W^z is also modified to have a size of $\mu B \times K$, where the similarity is calculated using the model's output embedding z_b and the momentum embedding $\bar{z}_j$: $W^z_{bj} = \exp(z_b \cdot \bar{z}_j / t)$. Since gradient only flows back through z_b, we can use a large K with only a small increase in GPU memory usage and computation time.

Besides the contrastive loss, we also leverage the EMA model for memory-smoothed pseudo-labeling, by forwarding the weakly-augmented samples through the EMA model instead of the original model. A graphical illustration of the memory bank and the momentum queue is given in the appendix.

4. Experiment

4.1. CIFAR-10 and STL-10

First, we conduct experiments on CIFAR-10 and STL-10 datasets. CIFAR-10 contains 50,000 images of size 32×32 from 10 classes. We vary the amount of labeled data and focus on the label-scarce scenario where few labels are available. We evaluate on 5 runs with different random seeds. STL-10 contains 5,000 labeled images of size 96×96 from 10 classes and 100,000 unlabeled images including ood samples. We evaluate on the 5 pre-defined folds. Following [2, 32], we report the performance of an EMA model.

Baseline methods. For fair comparison, we improve the current state-of-the-art method FixMatch [32] with distribution alignment [1] to build a stronger baseline. We also compare with the original FixMatch and MixMatch [2]. We omit previous methods such as Π-model [30], Pseudo-Labeling [19], and Mean Teacher [33] due to their poorer performance as reported in [32]. Following [27], we reimplemented the baselines and performed all experiments using the same model architecture, the same codebase (PyTorch [29]), and the same random seeds.

Implementation details. For CIFAR-10, we use a Wide ResNet-28-2 [37]. For STL-10, we use a ResNet-18 [16] due to its lower computation cost compared to the WRN-37-2 used in [32][1]. The projection head is a 2-layer MLP which outputs 64-dimensional embeddings. The models are trained using SGD with a momentum of 0.9 and a weight decay of 0.0005. We follow the original papers [2, 32] and train the baselines for 1024 epochs, using an learning rate of 0.03 with a cosine decay schedule. We train CoMatch for only 512 epochs to demonstrate its efficiency in learning. For the hyperparameters in CoMatch that also exist in [32], we follow [32] and set $\lambda_{cls} = 1$, $\tau = 0.95$, $\mu = 7$, $B = 64$. For other hyperparameters, we fix $\alpha = 0.9$, $K = 2560$, $t = 0.2$, $T = 0.8$, and $\lambda_{ctr} = 1$ for all CIFAR-10 experiments, and only changes λ_{ctr} to 5 for STL-10.

Augmentations. CoMatch uses one "weak" augmentation $\mathrm{Aug_w}$, and two "strong" augmentations $\mathrm{Aug_s}$ and $\mathrm{Aug_s'}$. The weak augmentation for all experiments is the standard crop-and-flip. For strong augmentations, we follow [32] and uses RandAugment [8] as $\mathrm{Aug_s}$. For $\mathrm{Aug_s'}$, we follow the augmentation strategy in SimCLR [5] which applies random color jittering and grayscale conversion.

Results. Table 1 shows the results. CoMatch outperforms the best baseline across all settings. The improvement is more substantial when fewer labeled samples are available. For example, CoMatch achieves an average accuracy of 93.09% on CIFAR-10 with only 4 labels per class, whereas FixMatch (w. DA) has a lower accuracy of 86.98% and a

[1]The forward-pass GFLOPs/image is 0.34 for ResNet-18 and 2.58 for WRN-37-2. Compared to ResNet-18, WRN-37-2 takes 3×GPU memory and 7×training time per epoch.

Method	CIFAR-10				STL-10
	20 labels	40 labels	80 labels	250 labels	1000 labels
MixMatch [2]	27.84±10.63	51.90±11.76	80.79±1.28	88.97±0.85	38.02±8.29
FixMatch [32]	82.32±9.77	86.12±3.53	92.06±0.88	94.90±0.67	65.38±0.42
FixMatch [32] w. DA [1]	83.81±9.35	86.98±3.40	92.29±0.86	94.95±0.66	66.53±0.39
CoMatch	**87.67**±8.47	**93.09**±1.39	**93.97**±0.62	**95.09**±0.33	**79.80**±0.38

Table 1: Accuracy for CIFAR-10 and STL-10 on 5 different folds. All methods are tested using the same data and codebase.

larger variance. On STL-10, CoMatch also improves FixMatch (w. DA) by 13.27%.

4.2. ImageNet

We evaluate CoMatch on ImageNet ILSVRC-2012 to verify its efficacy on large-scale datasets. Following [38, 5], we randomly sample 1% or 10% of images with labels in a class-balanced way (13 or 128 samples per-class, respectively), while the rest of images are unlabeled. Our results are not sensitive to different random seeds hence we use a fixed random seed.

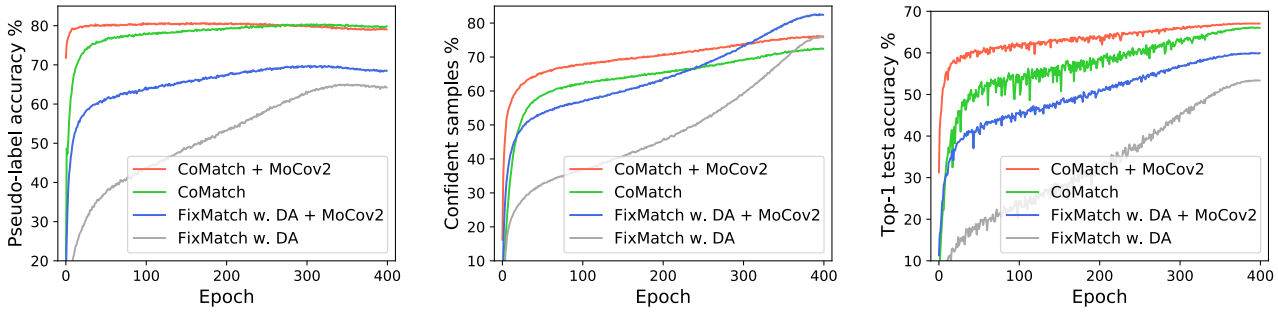
Baseline methods. The baselines include (1) semi-supervised learning methods and (2) self-supervised pre-training followed by fine-tuning. Furthermore, we construct a state-of-the-art baseline which combines FixMatch (w. DA) with self-supervised pre-training using MoCov2 [7] (pre-trained for 800 epochs). Self-supervised methods require additional model parameters during training due to the projection network. We count the number of training parameters as those that require gradient update. We also report the performance of SimCLRv2 [6]. However, SimCLRv2 uses a substantially (33×) larger pre-trained teacher model (which is itself distilled from a teacher of the same size) to produce pseudo-labels for distillation. Hence CoMatch should not be directly compared to SimCLRv2.

Implementation details. We use a ResNet-50 [16] model as the encoder. Following [7, 5], the projection head is a 2-layer MLP which outputs 128-dimensional embeddings. We train the model using SGD with a momentum of 0.9 and a weight decay of 0.0001. The learning rate is 0.1, which follows a cosine decay schedule for 400 epochs. For models that are initialized with MoCov2, we use a smaller learning rate of 0.03. The momentum parameter is set as $m = 0.996$. Other hyperparameters are shown in appendix A. We use the same strong augmentation for $\mathrm{Aug_s}$ and $\mathrm{Aug'_s}$, which applies crop-and-flip followed by color distortion. For fair comparison with baselines, we report the original model's performance instead of the EMA model's.

Results. Table 2 shows the result, where CoMatch achieves state-of-the-art performance. CoMatch obtains a top-1 accuracy of 66.0% on 1% of labels. Compared to the the

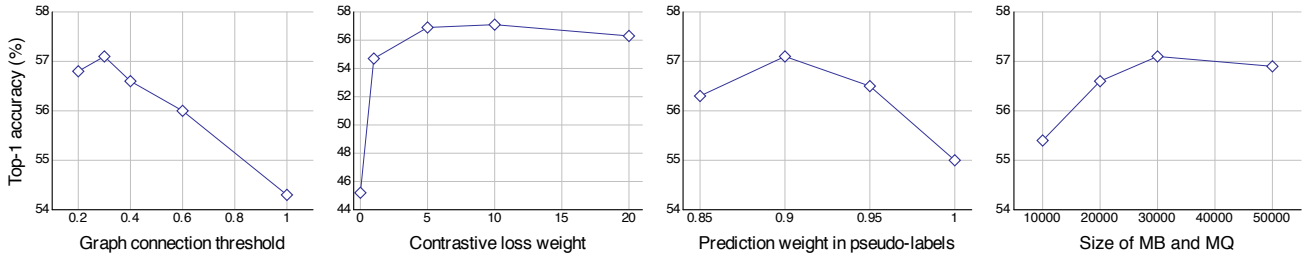
Self-supervised Pre-training	Method	#Epochs	#Paramters (train/test)	Top-1 Label fraction		Top-5 Label fraction	
				1%	10%	1%	10%
None	Supervised baseline [38]	∼20	25.6M / 25.6M	25.4	56.4	48.4	80.4
	Pseudo-label [19, 38]	∼100	25.6M / 25.6M	-	-	51.6	82.4
	VAT+EntMin. [26, 12, 38]	-	25.6M / 25.6M	-	68.8	-	88.5
	S4L-Rotation [38]	∼200	25.6M / 25.6M	-	53.4	-	83.8
	UDA (RandAug) [36]	-	25.6M / 25.6M	-	68.8	-	88.5
	FixMatch (RandAug) [32]	∼300	25.6M / 25.6M	-	71.5	-	89.1
	FixMatch w. DA	∼400	25.6M / 25.6M	53.4	70.8	74.4	89.0
	CoMatch	∼400	30.0M / 25.6M	**66.0**	**73.6**	**86.4**	**91.6**
PIRL [25]	Fine-tune	∼800	26.1M / 25.6M	30.7	60.4	57.2	83.8
PCL [21]		∼200	25.8M / 25.6M	-	-	75.3	85.6
SimCLR [5]		∼1000	30.0M / 25.6M	48.3	65.6	75.5	87.8
BYOL [13]		∼1000	37.1M / 25.6M	53.2	68.8	78.4	89.0
SwAV [3]		∼800	30.4M / 25.6M	53.9	70.2	78.5	89.9
MoCov2 [7]	Fine-tune	∼800	30.0M / 25.6M	49.8	66.1	77.2	87.9
	FixMatch w. DA	∼1200	30.0M / 25.6M	59.9	72.2	79.8	89.5
	CoMatch	∼1200	30.0M / 25.6M	**67.1**	**73.7**	**87.1**	**91.4**
SimCLRv2* [6]	Fine-tune	∼800	34.2M / 29.8M	57.9	68.4	82.5	89.2
	Teacher distillation	∼2400	829.2M / 29.8M	73.9	77.5	91.5	93.4

Table 2: Accuracy for ImageNet with 1% and 10% of labeled examples. SimCLRv2* [6] uses larger models for training and test.



(a) (b) (c)

Figure 3: Plots of different methods as training progresses on ImageNet with 1% labels. (a) Accuracy of the confident pseudo-labels *w.r.t* to the ground-truth labels of the unlabeled samples. (b) Ratio of the unlabeled samples with confident pseudo-labels that are included in the unsupervised classification loss. (3) Top-1 accuracy on the test data.



(a) (b) (c) (d)

Figure 4: Plots of ablation studies on CoMatch. The default hyperparameter setting achieves 57.1% (ImageNet with 1% labels, trained for 100 epochs). FixMatch with EMA pseudo-label achieves 43.9%. (a) Varying the threshold T which controls the sparsity of edges in the pseudo-label graph. $T = 1$ reduces to self-supervised contrastive learning. (b) Varying the weight λ_{ctr} for the contrastive loss. $\lambda_{ctr} = 0$ removes contrastive learning. (c) Varying α, the weight of the EMA model's prediction in generating pseudo-labels. $\alpha = 1$ reduces to pseudo-labeling with mean teacher [33]. (d) Varying K, the number of samples in both the memory bank and the momentum queue.

best baseline (MoCov2 followed by FixMatch w. DA), CoMatch achieves 6.1% improvement with 3× less training time. With the help of MoCov2 pre-training, the performance of CoMatch can further improve to 67.1% on 1% of labels, and 73.7% on 10% of labels. In Figure 3, we further show that CoMatch produces pseudo-labels that are more confident and accurate. Pre-training with MoCov2 helps speed up the convergence rate.

4.3. Ablation Study.

We perform extensive ablation study to examine the effect of different components in CoMatch. We use ImageNet with 1% labels as the main experiment. Due to the number of experiments in our ablation study, we report the top-1 accuracy after training for 100 epochs, where the default setting of CoMatch achieves 57.1%.

Graph connection threshold. The threshold T in eqn.(9) controls the sparsity of edges in the pseudo-label graph. Figure 4(a) presents the effect of T. As T increases, samples whose pseudo-labels have lower similarity are disconnected. Hence their embeddings are pushed apart by our contrastive loss. When $T = 1$, the proposed graph-based contrastive loss downgrades to the self-supervised loss in eqn.(2) where the only connections are the self-loops. Using the self-supervised contrastive loss decreases the performance by 2.8%.

Contrastive loss weight. We vary the weight λ_{ctr} for the contrastive loss $\mathcal{L}_u^{ctr}$ and report the result in Figure 4(b), where $\lambda_{ctr} = 10$ gives the best performance. With 10% of ImageNet labels, $\lambda_{ctr} = 2$ yields better performance. We find that in general, fewer labeled samples require a larger λ_{ctr} to strengthen the graph regularization.

Prediction weight in pseudo-labels. Our memory-smoothed pseudo-labeling uses α to control the balance between the EMA model's prediction and smoothness constraint. Figure 4(c) shows its effect, where $\alpha = 0.9$ results in the best performance. When $\alpha = 1$, the pseudo-labels are purely generated by the EMA model, which is the Mean-Teacher [33]. The accuracy decreases by 2.1% due to confirmation bias. When $\alpha < 0.9$, the pseudo-labels are over-smoothed. A potential improvement is to apply sharpening [2] to pseudo-labels with smaller α, but is not studied here due to the need of an extra hyperparameter.

Size of memory bank and momentum queue. K controls both the size of the memory bank for pseudo-labeling and the size of the momentum queue for contrastive learning. A larger K considers more samples to enforce a structural constraint on the label space and the embedding space. As

Method	#ImageNet labels	#Pre-train epochs	k=4	k=8	k=16	k=64	Full
Supervised	100%	90	73.51±2.12	79.60±0.61	82.75±0.34	85.55±0.12	87.12
MoCov2 [7]	0%	800	70.47±2.18	76.74±0.87	80.61±0.53	84.60±0.11	86.83
SwAV [3]		400	68.04±2.39	75.06±0.73	79.46±0.55	84.24±0.13	86.86
MoCov2 [7]	1%	800	71.82±2.09	77.35±0.83	81.33±0.50	84.98±0.14	87.05
CoMatch		400	72.81±1.50	79.18±0.51	82.30±0.46	85.65±0.17	87.66
MoCov2 [7]	10%	800	73.09±2.02	79.37±0.40	82.05±0.46	85.41±0.16	87.48
CoMatch		400	**74.56**±2.04	**80.60**±0.31	**83.24**±0.43	**86.07**±0.16	**87.91**

(a) VOC07

Method	#ImageNet labels	#Pre-train epochs	k=4	k=8	k=16	k=64	k=256
Supervised	100%	90	27.20±0.41	32.08±0.45	35.95±0.21	41.81±0.17	45.74±0.14
MoCov2 [7]	0%	800	25.34±0.51	30.64±0.39	35.08±0.34	42.18±0.10	46.96±0.06
SwAV [3]		400	25.32±0.46	31.00±0.47	35.65±0.28	42.60±0.11	**47.51**±0.20
MoCov2 [7]	1%	800	26.22±0.50	31.33±0.40	35.55±0.35	42.20±0.11	46.95±0.07
CoMatch		400	27.15±0.42	32.36±0.37	36.56±0.33	42.97±0.11	47.32±0.18
MoCov2 [7]	10%	800	27.19±0.47	32.11±0.49	36.00±0.30	42.31±0.13	46.88±0.08
CoMatch		400	**28.11**±0.33	**33.05**±0.46	**36.98**±0.28	**43.06**±0.22	47.10±0.11

(b) Places

Table 3: Linear classification on VOC07 and Places using models pre-trained on ImageNet. We vary the number of examples per-class (k) on the down-stream datasets. We report the average result with std across 5 runs.

Method	#ImageNet labels	1× schedule						2× schedule					
		AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Supervised	100%	38.9	59.6	42.7	35.4	56.5	38.1	40.6	61.3	44.4	36.8	58.1	39.5
MoCo [14]	0%	38.5	58.9	42.0	35.1	55.9	37.7	40.8	61.6	44.7	36.9	58.4	39.7
CoMatch	1%	39.7	61.2	43.1	36.1	57.8	38.5	41.2	62.2	44.9	37.3	59.0	39.9
CoMatch	10%	**40.5**	**61.5**	**44.2**	**36.7**	**58.3**	**39.2**	**41.5**	**62.5**	**45.4**	**37.6**	**59.5**	**40.3**

Table 4: Transfer the pre-trained models to object detection and instance segmentation on COCO, by fine-tuning Mask-RCNN with R50-FPN on `train2017`. We evaluate bounding-box AP (AP^{bb}) and mask AP (AP^{mk}) on `val2017`.

shown in Figure 4(d), the performance increases as K increases from 10k to 30k, but plateaus afterwards. We would also like to highlight that the memory bank and the momentum queue only introduce a small computation overhead because (1) low-dimensional embeddings are stored, (2) gradients are not computed *w.r.t* to the embeddings.

4.4. Transfer of Learned Representations

We further evaluate the quality of the representations learned by CoMatch by transferring it to other tasks. Following [11, 21], We first perform linear classification on two datasets: PASCAL VOC2007 [10] for object classification and Places205 [39] for scene recognition. We train linear SVMs using fixed representations from ImageNet pre-trained models. We preprocess all images by resizing them to 256 pixels along the shorter side and taking a 224×224 center crop. The SVMs are trained on the global average pooling features of ResNet-50. To study the transferability of the representations in few-shot scenarios, we vary the number of samples per-class (k) in the downstream datasets.

Table 3 shows the results. We compare CoMatch with standard supervised learning, self-supervised learning (MoCov2 [7] and SwAV [3]), and fine-tuning after self-supervised learning. CoMatch outperforms both supervised learning and self-supervised learning, which shows the efficacy of semi-supervised representation learning. It is interesting to observe that self-supervised learning methods do not perform well in few-shot transfer, and only catch up with supervised learning when k increases.

In Table 4, we also show that compared to supervised and self-supervised learning, CoMatch learns a better CNN backbone for object detection and instance segmentation on COCO [23]. We follow the exact same setting as [14] to fine-tune a Mask-RCNN model [15] for 1× or 2× schedule.

5. Conclusion

To conclude, the success of CoMatch can be attributed to three contributions: (1) co-training of class probabilities and image embeddings, (2) memory-smoothed pseudo-labeling to mitigate confirmation bias, (3) graph-based contrastive learning to learn better representations. We believe that CoMatch will help enable machine learning to be deployed in domains where labels are expensive to acquire.

References

[1] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *ICLR*, 2020. 1, 2, 3, 5, 6

[2] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019. 1, 2, 5, 6, 7

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 1, 2, 6, 8

[4] Peibin Chen, Tao Ma, Xu Qin, Weidi Xu, and Shuchang Zhou. Data-efficient semi-supervised learning by reliable edge mining. In *CVPR*, pages 9189–9198, 2020. 3

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2, 5, 6

[6] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 1, 2, 6

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 6, 8

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, pages 702–703, 2020. 5

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1

[10] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 8

[11] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, pages 6391–6400, 2019. 8

[12] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, pages 529–536, 2004. 2, 6

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila 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 preprint arXiv:2006.07733*, 2020. 1, 2, 6

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2, 5, 8

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 8

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5, 6

[17] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, pages 5070–5079, 2019. 1, 2

[18] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 2

[19] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, volume 3, page 2, 2013. 1, 2, 5, 6

[20] Junnan Li, Caiming Xiong, and Steven C.H. Hoi. Mopro: Webly supervised learning with momentum prototypes. In *ICLR*, 2021. 2

[21] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021. 2, 6, 8

[22] Suichan Li, Bin Liu, Dongdong Chen, Qi Chu, Lu Yuan, and Nenghai Yu. Density-aware graph for deep semi-supervised visual recognition. In *CVPR*, pages 13397–13406. IEEE, 2020. 2

[23] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 8

[24] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *CVPR*, pages 8896–8905, 2018. 1, 3

[25] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 6

[26] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993, 2019. 2, 6

[27] Avital Oliver, Augustus Odena, Colin Raffel, Ekin Dogus Cubuk, and Ian J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *NeurIPS*, pages 3239–3250, 2018. 5

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[29] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshop*, 2017. 5

[30] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *NIPS*, pages 3546–3554, 2015. 5

[31] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In Daniel D. Lee,

Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *NIPS*, pages 1163–1171, 2016. 2

[32] 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. In *NeurIPS*, 2020. 1, 2, 3, 5, 6

[33] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, pages 1195–1204, 2017. 2, 5, 7

[34] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. 2

[35] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 2

[36] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. 2, 6

[37] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith, editors, *BMVC*, 2016. 5

[38] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, pages 1476–1485, 2019. 6

[39] Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495, 2014. 8

[40] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *NIPS*, pages 321–328, 2003. 2

[41] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In Tom Fawcett and Nina Mishra, editors, *ICML*, pages 912–919, 2003. 2

[42] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005. 2