SimMatch: Semi-supervised Learning with Similarity Matching

Mingkai Zheng[1,2]   Shan You[2*]

Lang Huang[3]   Fei Wang[4]   Chen Qian[2]   Chang Xu[1]

[1]School of Computer Science, Faculty of Engineering, The University of Sydney

[2]SenseTime Research   [3] The University of Tokyo

[4]University of Science and Technology of China

Abstract

Learning with few labeled data has been a longstanding problem in the computer vision and machine learning research community. In this paper, we introduced a new semi-supervised learning framework, SimMatch, which simultaneously considers semantic similarity and instance similarity. In SimMatch, the consistency regularization will be applied on both semantic-level and instance-level. The different augmented views of the same instance are encouraged to have the same class prediction and similar similarity relationship respected to other instances. Next, we instantiated a labeled memory buffer to fully leverage the ground truth labels on instance-level and bridge the gaps between the semantic and instance similarities. Finally, we proposed the unfolding and aggregation operation which allows these two similarities be isomorphically transformed with each other. In this way, the semantic and instance pseudo-labels can be mutually propagated to generate more high-quality and reliable matching targets. Extensive experimental results demonstrate that SimMatch improves the performance of semi-supervised learning tasks across different benchmark datasets and different settings. Notably, with 400 epochs of training, SimMatch achieves 67.2%, and 74.4% Top-1 Accuracy with 1% and 10% labeled examples on ImageNet, which significantly outperforms the baseline methods and is better than previous semi-supervised learning frameworks.

1. Introduction

Benefiting from the availability of large-scale annotated datasets and growing computational resources in the last decades, deep neural networks have demonstrated their success on a variety of visual tasks [19, 20, 22, 23, 27, 38, 50, 58–60]. However, a large volume of labeled data is very expensive to collect in a real-world scenario. Learning with few labeled data has been a longstanding problem in

---

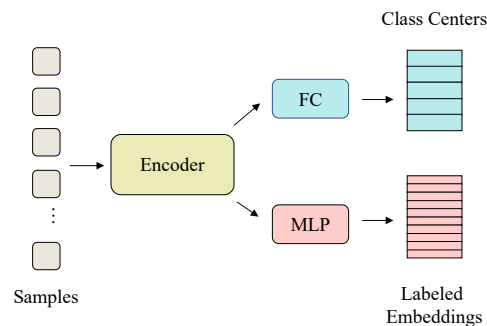*Correspondence to: Shan You <youshan@sensetime.com>.

Figure 1. A sketch of SimMatch. The Fully-Connected layer vectors can be viewed as a semantic representative or class center for each category. However, due to the limited labeled samples, the semantic-level information is not always reliable. In SimMatch, we consider the instance-level and semantic-level information simultaneously and adopt a labeled memory buffer to fully leverage the ground truth label on instance-level.

the computer vision and machine learning research community. Among various methods, semi-supervised learning (SSL) [12, 46, 53, 67] serves as an effective solution by dint of the help of massive unlabeled data, and achieves remarkable performance.

A simple but very effective semi-supervised learning method is to pretrain a model on a large-scale dataset and then transfer the learned representation by fine-tuning the pretrained model with a few labeled samples. Thanks to the recent advances in self-supervised learning [10, 14, 15, 21, 25, 26, 30, 54], such pretraining and fine-tuning pipeline have demonstrated its promising performance in SSL. Most self-supervised learning frameworks focus on the design of pretext tasks. For example, instance discrimination [55] encourages different views of the same instance to share the same features, and different instances should have distinct features. Deep clustering based methods [3, 9, 10] expect different augmented views of the same instance should be classified into the same cluster. However, most of these pretext tasks are designed in a completely unsupervised manner, without considering the few labeled data at hand.

Instead of standalone two-stage pretraining and fine-

tuning, current popular methods directly involve the labeled data in a joint feature learning paradigm with pseudo-labeling [35] or consistency regularization [47]. The main idea behind these methods is to train a semantic classifier with labeled samples and use the predicted distribution as the pseudo label for the unlabeled samples. In this way, the pseudo-labels are generally produced by the weakly augmented views [5, 48] or the averaged predictions of multiple strongly augmented views [6]. The objective will be constructed by the cross entropy loss between an different strongly augmented views and the pseudo-labels. It might also be noted that the pseudo-labels will generally be sharpened or operated by $argmax$ since every instance is expected to be classified into a category. However, when there are only very limited annotated data, the semantic classifier is no longer reliable; applying the pseudo-label method will cause the "overconfidence" issue [13,63], which means the model will fit on the confident but wrong pseudo-labels, resulting in poor performance.

In this paper, we introduce a novel semi-supervised learning framework, SimMatch, which has been shown in Figure 1. In SimMatch, we bridge both sides and propose to match the similarity relationships of both semantic and instance levels simultaneously for different augmentations. Specifically, we first require the strongly augmented view to have the same semantic similarity (*i.e.* label prediction) with a weak augmented view; besides, we also encourage the strong augmentation to have the same instance characteristics (*i.e.* similarity between instances) with the weak one for more intrinsic feature matching. Moreover, different from the previous works that simply regard the predictions of the weakly augmented views as pseudo-labels. In SimMatch, the semantic and instance pseudo-labels are allowed to interact by instantiating a memory buffer that keeps all the labeled examples. In this way, these two similarities can be isomorphically transformed with each other by introducing *aggregating* and *unfolding* techniques. Thus the semantic and instance pseudo-labels can be mutually propagated to generate more high-quality and reliable matching targets. Extensive experiments demonstrate the effectiveness of SimMatch across different settings. Our contribution can be summarized as follows:

- We proposed SimMatch, a novel semi-supervised learning framework that simultaneously considers semantics similarity and instance similarity.

- To channel both similarities, we leverage a labeled memory buffer so that semantic and instance pseudo-labels can be mutually propagated with the *aggregating* and *unfolding* techniques.

- SimMatch establishes a new state-of-the-art performance for semi-supervised learning. With only 400 epochs of training, SimMatch achieves 67.2% and

74.4% Top-1 accuracy with 1% and 10% labeled examples on ImageNet.

2. Related Work

2.1. Semi-Supervised Learning

Consistency Regularization is a widely adopted method in semi-supervised learning. The main idea is to enforce the model to output a consistent prediction for the different perturbed versions of the same instance. For example, [34, 47] achieved such consistency requirement by minimizing the mean square difference between the predicted probability distribution of the two transformed views. In this case, the transformation could be either domain-specific data augmentations [5, 6, 48], or some regularization techniques in the network (*e.g.* drop out [49] and random max-pooling [47]). Moreover, [34] also proposed a temporal ensemble strategy to aggregate the predictions of multiple previous networks, which makes the predicted distribution more reliable. Mean Teacher [52] further extends this idea which replaced the aggregated predictions with the output of an exponential moving average (EMA) model.

MixMatch [6], ReMixMatch [5], and FixMatch [48] are three augmentation anchoring based methods that fully leverage the augmentation consistency. Specifically, Mix-Match adopts a sharpened averaged prediction of multiple strongly augmented views as the pseudo label and utilizes the MixUp trick [64] to further enhance the pseudo label. ReMixMatch improved this idea by generating the pseudo label with weakly augmented views and also introduced a distribution alignment strategy that encourages the pseudo label distribution to match the marginal distribution of ground-truth class labels. FixMatch simplified these ideas, where the unlabeled images are only retained if the model produces a high-confidence pseudo label. Despite its simplicity, FixMatch achieved state-of-the-art performance among the augmentation anchoring-based methods.

2.2. Self-supervised Pretraining

Apart from the typical semi-supervised learning method, self-supervised and contrastive learning [14, 26, 55] has gained much attention in this research community since fine-tuning the pre-trained model with labeled samples has shown promising classification results, especially SimCLR v2 [15] shows that a big (deep and wide) pre-trained model is a strong semi-supervised learner. Most of the contrastive learning framework adopts the instance discrimination [55] as the pretext task, which defines different augmented views of the same instance as positive pairs, while negative pairs are formed by sampling views from different instances. However, because of the existence of similar samples, treating different instances as negative pairs will result in a class collision problem [2], which is not conducive to the down-
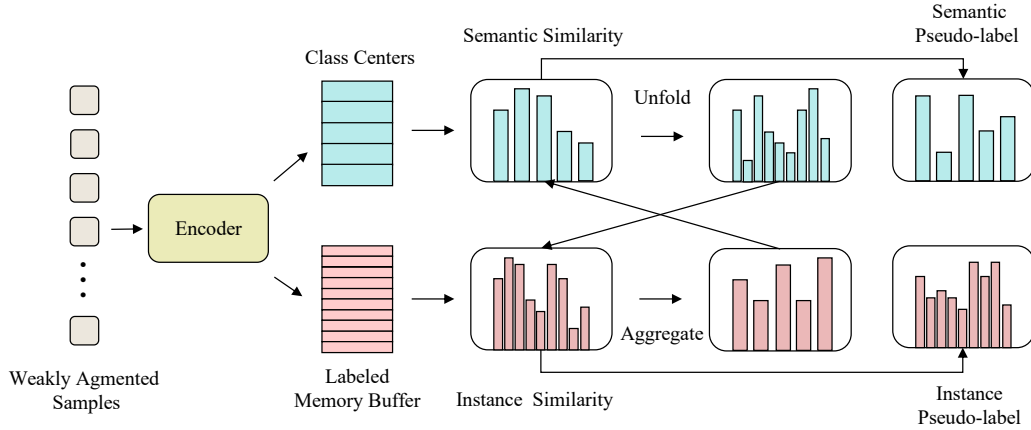
Figure 2. An overview of the SimMatch pseudo-label generation process. SimMatch will use the weak augmented view to generate a semantic pseudo-label and an instance pseudo-label. Specifically, we will first compute the semantic and instance similarity by the class centers and labeled embeddings, then use the unfolding and aggregation operations to fuse these two similarities and finally get the pseudo-label. Please see more details in our method section below.

stream tasks (especially classification tasks). Some previous works addressed this issue by unsupervised clustering [9, 10, 37, 65], where similar samples will be clustered into the same class. There are also some other methods designed various negative free pretext tasks [17, 25, 29, 31, 66] to avoid the class collision problem. Both cluster-based methods and negative-free based methods have shown significant improvements for downstream classification tasks.

CoMatch [36] combines the idea of consistency regularization and contrastive learning, where the target similarity of two instances is measured by the similarity between two class probability distributions, which achieves the current state-of-the-art performance on semi-supervised learning. However, it is very sensitive to the hyper-parameters, the optimal temperature and threshold is different for various datasets and settings. Compared to CoMatch, SimMatch is faster, more robust, and has higher performance.

3. Method

In this section, we will first revisit the preliminary work on augmentation anchoring based semi-supervised learning frameworks; then, we will introduce our proposed method SimMatch. After that, the algorithm and the implementation details will also be explained.

3.1. Preliminaries

We define the semi-supervised image classification problem as following. Given a batch of B labeled samples $\mathcal{X} = \{x_b : b \in (1, ..., B)\}$, we randomly apply a weak augmentation function (*e.g.* using only a flip and a crop) $T_w(\cdot)$ to obtain the weakly augmented samples. Then, a convolutional neural network based encoder $F(\cdot)$ is employed to extract the feature information from these samples, *i.e.* $\mathbf{h} = \mathcal{F}(T(x))$. Finally, a fully connected class prediction head $\phi(\cdot)$ is utilized to map $\mathbf{h}_b$ into semantic similarities,

which can be written as: $p = \phi(\mathbf{h})$. The labeled samples could be directly optimized by the cross entropy loss with the ground truth labels:

$$\mathcal{L}_s = \frac{1}{B} \sum \mathrm{H}(y, p) \tag{1}$$

Let us define a batch of μB unlabeled samples $\mathcal{U} = \{u_b : b \in (1, ..., \mu B)\}$. By following [5, 6], we randomly apply the weak and strong augmentation $T_w(\cdot)$, $T_s(\cdot)$ and using the same processing step as the labeled samples to get the semantic similarities for weakly augmented sample p^w (pseudo label) and strongly augmented sample p^s. Then the unsupervised classification loss can be defined as the cross-entropy between these two predictions:

$$\mathcal{L}_u = \frac{1}{\mu B} \sum \mathbb{1}(\max DA(p^w) > \tau)\mathrm{H}(DA(p^w), p^s) \tag{2}$$

Where τ is the confidence threshold. Following [48], we only retain the unlabeled samples whose largest class probability in the pseudo-labels are larger than τ. $DA(\cdot)$ stands for the distribution alignment strategy from [5] which balanced the pseudo-labels distribution. We simply follow the implementation from [36] where we maintain a moving-average of p^w_{avg} and adjust the current p^w with $Normalize(p^w/p^w_{avg})$. Please also noted that we do not take the sharpened or one-hot version of p^w, $DA(p^w)$ will be directly served as the pseudo-label.

3.2. Instance Similarity Matching

In SimMatch, we also consider the instance-level similarity as we have discussed previously. Concretely, we encourage the strongly augmented view to have a similar similarity distribution with the weakly augmented view. Suppose we have a non-linear projection head $g(\cdot)$ which maps the representation $\mathbf{h}$ to a low-dimensional embedding

$\mathbf{z}_b = g(\mathbf{h}_b)$. Following the anchoring based method, we use $\mathbf{z}_b^w$ and $\mathbf{z}_b^s$ to denote the embedding from the weakly and strongly augmented view. Now, lets assume we have K weakly augmented embeddings for a bunch of different samples $\{\mathbf{z}_k : k \in (1, ..., K)\}$, we calculate the similarities between $\mathbf{z}^w$ and i-th instance by using a similarity function $sim(\cdot)$, which represents the dot product between L_2 normalized vectors $sim(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T\mathbf{v}/\|\mathbf{u}\|\|\mathbf{v}\|$. A softmax layer can be adopted to process the calculated similarities, which then produces a distribution:

$$q_i^w = \frac{\exp(sim(\mathbf{z}_b^w, \mathbf{z}_i)/t)}{\sum_{k=1}^{K} \exp(sim(\mathbf{z}_b^w, \mathbf{z}_k)/t)} \tag{3}$$

Where t is the temperature parameter that controls the sharpness of the distribution. On the other hand, we can calculate the similarities between the strongly augmented view $\mathbf{z}^s$ and $\mathbf{z}_i$ as $sim(\mathbf{z}_b^s, \mathbf{z}_i)$. The resulting similarity distribution can be written as:

$$q_i^s = \frac{\exp(sim(\mathbf{z}_b^s, \mathbf{z}_i)/t)}{\sum_{k=1}^{K} \exp(sim(\mathbf{z}_b^s, \mathbf{z}_k)/t)} \tag{4}$$

Finally, the consistency regularization can be achieved by minimizing the different between q^s and q^w. Here, we adopt the cross entropy loss, which can be formulated as:

$$\mathcal{L}_{in} = \frac{1}{\mu B} \sum \mathrm{H}(q^w, q^s) \tag{5}$$

Please noted that the instance consistency regularization will only be applied on the unlabeled examples. The overall training objective for our model will be:

$$\mathcal{L}_{overall} = \mathcal{L}_s + \lambda_u \mathcal{L}_u + \lambda_{in} \mathcal{L}_{in} \tag{6}$$

where λ_u and λ_{in} are the balancing factors that control the weights of the two losses.

3.3. Label Propagation through SimMatch

Although our overall training objective also considers the consistency regularization on instance-level, however, the generation of the instance pseudo-labels q^w are still in a fully unsupervised manner, which is absolutely a waste of the labeled information. To improve the quality of the pseudo-labels, in this section, we will illustrate how to leverage the labeled information on instance-level and also introduce a way that allows semantic similarity and instance similarity to interact with each other.

We instantiated a labeled memory buffer to keep all the annotated examples as we have shown in Figure 2 (red branch). In this way, each $\mathbf{z}_k$ that we used in Eq.(3) and Eq.(4) could be assigned to a specific class. If we interpret the vectors in ϕ as the "centered" class references, the embeddings in our labeled memory buffer can be viewed as a set of "individual" class references.

By given a weakly augmented sample, we first compute the semantic similarity $p^w \in \mathbf{R}^{1 \times L}$ and instance similarity $q^w \in \mathbf{R}^{1 \times K}$. (Noted that L is generally much smaller than K since we need at least one sample for each class.) To calibrate q^w with p^w, we need to **unfold** p^w into K dimensional space which we denote it as p^{unfold}. we achieved this by matching the corresponding semantic similarity for each labeled embedding:

$$p_i^{unfold} = p_j^w, \ where \ \ class(q_j^w) = class(p_i^w) \tag{7}$$

where $class(\cdot)$ is the function that returns the ground truth class. Specifically, $class(q_j^w)$ represent the label for the jth element in memory buffer and $class(p_i^w)$ means the ith class. Now, we regenerate the calibrated instance pseudo-labels by **scaling** q^w with p^{unfold}, which can be expressed as the following:

$$\widehat{q}_i = \frac{q_i^w p_i^{unfold}}{\sum_{k=1}^{K} q_k^w p_k^{unfold}} \tag{8}$$

The calibrated instance pseudo-label $\widehat{q}$ will be served as a new target and replace the old one q^w in Eq.(5). On the other hand, we can also use the instance similarity to adjust the semantic similarity. To do this, we first need to **aggregate** q into L dimensional space which we denote it as q^{agg}. We achieved this by sum over the instance similarities that share the same ground truth labels:

$$q_i^{agg} = \sum_{j=0}^{K} \mathbb{1}(class(p_i^w) = class(q_j^w))q_j^w \tag{9}$$

Now, we regenerate the adjusted semantic pseudo-label by **smoothing** p^w with q^{agg}, which can be written as:

$$\widehat{p}_i = \alpha p_i^w + (1 - \alpha)q_i^{agg} \tag{10}$$

where α is the hyper-parameter that controls the weight of the semantic and instance information. Similarly, The adjusted semantic pseudo-label will replace the old one p_i^w in Eq.(2). In this way, the pseudo-label $\widehat{p}$ and $\widehat{q}$ will both contains the semantic-level and instance-level information. As we have shown in Figure 3, when semantic and instance similarities are similar, which means these two distributions agree with the prediction of each other, then the result pseudo-label will be much sharper and produce high confidence for some classes. On the other hand, if these two similarities are different, the result pseudo-label will be much flatter and not contain high probability values. In SimMatch, we adopt the scaling and smoothing strategy for $\widehat{q}$ and $\widehat{p}$ respectively, we also have tried different combination for these two strategies, please see more details in our ablation study section.

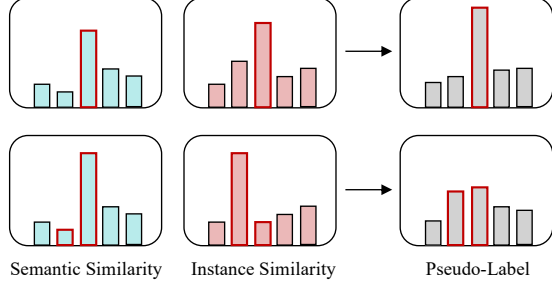Semantic Similarity — Instance Similarity — Pseudo-Label

Figure 3. The intuition behind label propagation. If the semantic and instance similarities are similar, the result pseudo-label will be much sharper and produce high confidence for some classes. When these two similarities are different, the result pseudo-label will be much flatter.

3.4. Efficient Memory Buffer

As we have mentioned, SimMatch requires a memory buffer to keep the embeddings for labeled examples. In doing so, we are required to store both feature embeddings and the ground truth labels. Specifically, we defined a feature memory buffer $Q_f \in \mathbf{R}^{K \times D}$ and a label memory buffer $Q_l \in \mathbf{R}^{K \times 1}$ where K is the number of available annotated samples, D is the embedding size. The largest K in our experiments is around 10^5 (ImageNet 10% setting), which only costs 64M GPU memories for Q_f. For Q_l, we just need to store a scalar for each label, the aggregation and unfolding operation can be easily achieved by $gather$ and $scatter\_add$ function, which should has been efficiently implemented in recent deep learning libraries [1, 44]. In this case, Q_l only costs less than 1M GPU memories ($K = 10^5$), which is almost negligible.

According to [26], the rapid changed feature in memory buffer will dramatically reduce the performance. In Sim-Match, we adopt two different implementation for different buffer size. When K is large, we follow MoCo [26] to leverage a student-teacher based framework, we denote it as $\mathcal{F}_s$ and $\mathcal{F}_t$. In this case, the labeled examples and strongly augmented samples will be passed into $\mathcal{F}_s$, the weakly augmented samples will be feed into $\mathcal{F}_t$ to generate the pseudo-labels. The parameters of $\mathcal{F}_t$ will be updated by:

$$\mathcal{F}_t \leftarrow m\mathcal{F}_t + (1-m)\mathcal{F}_s \qquad (11)$$

On the other hand, when K is small, maintain a teacher network is not necessary, We simply adopt the temporal ensemble strategy [34, 55] to smooth the features in memory buffer, which can be written as :

$$\mathbf{z}_t \leftarrow m\mathbf{z}_{t-1} + (1-m)\mathbf{z}_t \qquad (12)$$

In this case, all the samples will be directly pass into the same encoder. The student-teacher version of SimMatch has been illustrated in Algorithm 1.

Algorithm 1: SimMatch (Student-Teacher)

Input: $\mathbf{x}_l$ and :$\mathbf{x}_u$ a batch of labeled and unlabeled samples. $T_w(\cdot)$ and $T_s(\cdot)$: Weak and strong augmentation function. $\mathcal{F}_t$ and $\mathcal{F}_s$: Teacher and student encoder. ϕ_t and ϕ_s: teacher and student classifier. g_t and g_s: teacher and student projection head. Q_f and Q_l: The feature and label memory buffer.

while *network not converge* **do**
  for *i=1 to step* **do**
    $\mathbf{h}^w = \mathcal{F}_t(T_w(\mathbf{x}_u))$ $\mathbf{h}^s = \mathcal{F}_s(T_s(\mathbf{x}_u))$
    $p^w = DA(\phi_t(\mathbf{h}^w))$ $p^s = \phi_s(\mathbf{h}^s)$
    $\mathbf{z}^w = g_t(\mathbf{h}^w)$ $\mathbf{z}^s = g_s(\mathbf{h}^s)$
    $\mathbf{h}_t^l = \mathcal{F}_t(T_w(\mathbf{x}_l))$ $\mathbf{h}_s^l = \mathcal{F}_s(T_w(\mathbf{x}_l))$
    $p^l = \phi_s(\mathbf{h}_s^l)$ $\mathbf{z}^l = g_t(\mathbf{h}_t^l)$
    Compute q^w and q^s by Eq.(3) Eq.(4)
    Compute p^{unfold} and q^{agg} by Eq.(7) Eq.(9)
    Compute $\widehat{q}$ and $\widehat{p}$ by Eq.(8) Eq.(10)
    $\mathcal{L}_s = \frac{1}{B}\sum \mathrm{H}(y, p^l)$
    $\mathcal{L}_u = \frac{1}{\mu B}\sum \mathbb{1}(\max \widehat{p} > \tau)\mathrm{H}(\widehat{p}, p^s)$
    $\mathcal{L}_{in} = \frac{1}{\mu B}\sum \mathrm{H}(\widehat{q}, q^s)$
    $\mathcal{L}_{overall} = \mathcal{L}_s + \lambda_u \mathcal{L}_u + \lambda_{in}\mathcal{L}_{in}$
    Optimize F_s, g_s and ϕ_s by $\mathcal{L}_{overall}$
    Momentum Update F_t, g_t and ϕ_t
    Update Q_f and Q_l with $\mathbf{z}^l$ and y
  end
end
Output: The well trained model $\mathcal{F}_s$ and g_s

---

4. Experiments

In this section, we will first test SimMatch on various dataset and settings to shows its superiority, then we will ablation each component to validate the effectiveness of each component in our framework.

4.1. CIFAR-10 and CIFAR-100

We first evaluation SimMatch on CIFAR-10 and CIFAR-100 [33] datasets. CIFAR-10 consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. CIFAR-100 is just like the CIFAR-10, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. For CIFAR-10, we randomly sample 4, 25, and 400 samples from the training set as the labeled data and use the rest of the training set as the unlabeled data. For CIFAR-100, we perform the same experiments but use 4, 25, and 100 samples per class.

Implementation Details. Most our implementations follows [48]. Specifically, we adopt WRN28-2, and WRN28-8 [61] for CIFAR-10 and CIFAR-100 respectively. We use a standard SGD optimizer with Nesterov momen-

Table 1. Top-1 Accuracy comparison (mean and std over 5 runs) on CIFAR-10 and CIFAR-100 with varying labeled set sizes.

	CIFAR-10			CIFAR-100		
Method	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels
Π-Model [34]	-	45.74±3.97	58.99±0.38	-	42.75±0.48	62.12±0.11
Pseudo-Labeling [35]	-	50.22±0.43	83.91±0.28	-	42.62±0.46	63.79±0.19
Mean Teacher [52]	-	67.68±2.30	90.81±0.19	-	46.09±0.57	64.17±0.24
UDA [56]	70.95±5.93	91.18±1.08	95.12±0.18	40.72±0.88	66.87±0.22	75.50±0.25
MixMatch [6]	52.46±11.50	88.95±0.86	93.58±0.10	32.39±1.32	60.06±0.37	71.69±0.33
ReMixMatch [5]	80.90±9.64	94.56±0.05	95.28±0.13	55.72±2.06	72.57±0.31	76.97±0.56
FixMatch(RA) [48]	86.19±3.37	94.93±0.65	95.74±0.05	51.15±1.75	71.71±0.11	77.40±0.12
Dash [57]	86.78±3.75	**95.44**±0.13	95.92±0.06	55.24±0.96	72.82±0.21	78.03±0.14
CoMatch [36]	93.09±1.39	95.09±0.33	-	-	-	-
SimMatch(Ours)	**94.40**±1.37	95.16±0.39	**96.04**±0.01	**62.19**±2.21	**74.93**±0.32	**79.42**±0.11

tum [45, 51] and set the initial learning rate to 0.03. For the learning rate schedule, we use a cosine learning rate decay [40] which adjust the learning rate to $0.03 \cdot cos(\frac{7\pi s}{16S})$ where s is the current training step, and S is the total number of training steps. We also report the final performance using an exponential moving average of model parameters. Note that we use an identical set of hyper-parameters for both datasets ($\lambda_u = 1, \lambda_{in} = 1, t = 0.1, \alpha = 0.9, \tau = 0.95, \mu = 7, m = 0.7, B = 64, S = 2^{20}$) . For distribution alignment, we accumulate the past 32 steps p^w for calculating the moving average p^w_{avg}. We adopt the temporal ensemble memory buffer [34] since most settings for these two datasets have a relatively small K. For the implementations of the strong and weak augmentations, we strictly follow the FixMatch [48].

Results. The result has been reported in table 1. For baseline, we mainly consider methods Π-Model [34], Pseudo-Labeling [35], Mean Teacher [52], UDA [56], Mix-Match [6], ReMixMatch [5], FixMatch [48], and CoMatch [36]. We compute the mean and variance of accuracy when training on 5 different "folds" of labeled data. As we can see that the SimMatch achieves state-of-the-art performance on various settings, especially on CIFAR-100. For CIFAR-10, SimMatch has a large performance gain on 40 labels setting, but the improvements for 250 and 4000 is relatively small. We doubt that this is due to the accuracy of $95\% \sim 96\%$ being already quite close to the supervised performance.

4.2. ImageNet-1k

We also performed SimMatch on the large-scale ImageNet-1k dataset [20] to show the the superiority. Specifically, we test our algorithm on 1% and 10% settings. We follow the same label generation process as in CoMatch [36], where 13 and 128 labeled samples will be selected per class for 1% and 10% settings respectively.

Implementation Details. For ImageNet-1k, we adopt ResNet-50 [28] and use a standard SGD optimizer with Nesterov momentum. We warm up the model for five epochs until it reaches the initial learning rate 0.03 and then cosine decay it to 0. We use the same set of hyper-

parameters for both 1% and 10% settings ($\lambda_u = 10, \lambda_{in} = 5, t = 0.1, \alpha = 0.9, \tau = 0.7, \mu = 5, m = 0.999, B = 64$). We keep the past 256 steps p^w for distribution alignment. We choose the student-teacher version memory buffer and test performance on the student network. For strong augmentation, we follow the same strategy in MoCo v2 [16].

Results. We have shown the results in Table 2. As we can see, with 400 epochs training, SimMatch achieves 67.2%, and 74.4% Top-1 accuracy on 1% and 10% labeled examples, which is significantly better than the previous methods. FixMatch-EMAN [8] achieves a slightly lower performance (74.0%) on 10% setting. However, it requires 800 epochs of self-supervised pretrain (MoCo-EMAN) where SimMatch can directly train from scratch. The most recent work PAWS [4] achieves 66.5% and 75.5% Top-1 accuracy on 1% and 10% settings with 300 epochs training. Nevertheless, PAWS requires the multi-crops strategy [10] and 970×7 labeled examples to construct the support set. For each epoch, the actual training FLOPS of PAWS is 4 times that of SimMatch. Hence, the reported 300 epochs PAWS should have similar training FLOPS with 1200 epochs SimMatch. Due to the limited GPU resources, we cannot push this research to such a scale, but since Sim-Match surpassed PAWS on 1% setting with 1/3 training costs (400 epochs), we believe it can already demonstrate the superiority of our method.

Transfer Learning. We also evaluate the learned representations on multiple downstream classification tasks. We follow the linear evaluation setup described in [14, 25]. Specifically, we trained an L2-regularized multinomial logistic regression classifier on features extracted from the frozen pretrained network (400 epochs 10% SimMatch), then we used L-BFGS [39] to optimize the softmax cross-entropy objective, and we did not apply data augmentation. We selected the best L2-regularization parameter and learning rate from validation splits and applied it to the test sets. The datasets used in this benchmark are as follows: CIFAR-10 [33], CIFAR-100 [33], Food101 [7], Cars [32], DTD [18], Pets [43], Flowers [42]. The results have been shown in Table 3. As we can see, with only 400 epochs

Table 2. Experimental results on ImageNet with 1% and 10% labeled examples.

Self-supervised Pre-training	Method	Epochs	Paramters (train/test)	Top-1 Label fraction		Top-5 Label fraction	
				1%	10%	1%	10%
None	Pseudo-label [35,62]	∼100	25.6M / 25.6M	-	-	51.6	82.4
	VAT+EntMin. [24,41,62]	-	25.6M / 25.6M	-	68.8	-	88.5
	S4L-Rotation [62]	∼200	25.6M / 25.6M	-	53.4	-	83.8
	UDA [56]	-	25.6M / 25.6M	-	68.8	-	88.5
	FixMatch [48]	∼300	25.6M / 25.6M	-	71.5	-	89.1
	CoMatch [36]	∼400	30.0M / 25.6M	66.0	73.6	86.4	**91.6**
PCL [37]	Fine-tune	∼200	25.8M / 25.6M	-	-	75.3	85.6
SimCLR [14]		∼1000	30.0M / 25.6M	48.3	65.6	75.5	87.8
SimCLR V2 [15]		∼800	34.2M / 25.6M	57.9	68.4	82.5	89.2
BYOL [25]		∼1000	37.1M / 25.6M	53.2	68.8	78.4	89.0
SwAV [10]		∼800	30.4M / 25.6M	53.9	70.2	78.5	89.9
WCL [65]		∼800	34.2M / 25.6M	65.0	72.0	86.3	91.2
MoCo V2 [16]	Fine-tune	∼800	30.0M / 25.6M	49.8	66.1	77.2	87.9
	CoMatch [36]	∼1200	30.0M / 25.6M	67.1	73.7	**87.1**	91.4
MoCo-EMAN [8]	FixMatch-EMAN [8]	∼1100	30.0M / 25.6M	63.0	74.0	83.4	90.9
None	**SimMatch (Ours)**	∼400	30.0M / 25.6M	**67.2**	**74.4**	**87.1**	**91.6**

Table 3. Transfer learning performance using ResNet-50 pretrained with ImageNet. Following the evaluatiion protocal from [14, 25], we report Top-1 classification accuracy except Pets and Flowers for which we report mean per-class accuracy.

Method	Epochs	CIFAR-10	CIFAR-100	Food-101	Cars	DTD	Pets	Flowers	Mean
Supervised	-	**93.6**	78.3	72.3	66.7	74.9	91.5	94.7	81.7
SimCLR [14]	1000	90.5	74.4	72.8	49.3	**75.7**	84.6	92.6	77.1
MoCo v2 [16]	800	92.2	74.6	72.5	50.5	74.4	84.6	90.5	77.0
BYOL [25]	1000	91.3	**78.4**	**75.3**	67.8	75.5	90.4	**96.1**	82.1
SimMatch (10%)	400	**93.6**	**78.4**	71.7	**69.7**	75.1	**92.8**	93.2	**82.1**

Table 4. GPU hours per epoch for different methods. The speed is tested on 8 NVIDIA V100 GPUs.

Method	FixMatch	CoMatch	SimMatch (Ours)
GPU (Hours)	2.77	2.81	**2.34**

of training, SimMatch achieves the best performance on CIFAR-10, CIFAR-100, Cars and Flowers datasets which is comparable with BYOL and significantly better than Sim-CLR, MoCo V2, and supervised baseline. These results further validate the representation quality of SimMatch for classification tasks.

Training Efficiency. Next, we test the actual training speed for FixMatch, CoMatch, and SimMatch. The results is shown in Table 4, SimMatch is nearly 17% faster than FixMatch and CoMatch. In FixMatch, the weakly augmented $\mathcal{U}$ will be passed into the online network, which consumes more resources for the extra computational graphs. But in SimMatch, $\mathcal{U}$ only needs to be passed into the EMA network, so the computational graph does not need to be retained. Compared with CoMatch which requires two forward passes (strongly and weakly augmented $\mathcal{U}$) for the EMA network, SimMatch only requires one pass. Moreover, CoMatch adopts 4 memory banks (258M Memory) to compute the pseudo-label; SimMatch only needs 2 memory banks with 6.4M / 64M Memory for 1% and 10% labels, thus the pseudo-label generation will also be faster.

4.3. Ablation Study

Pseudo-Label Accuracy. Firstly, we would like to show the pseudo-label accuracy of SimMatch. In Figure 4, we visualized the training progress of FixMatch and our method. SimMatch can always generate high-quality pseudo-labels and consistently has higher performance on both unlabeled samples and validation sets.

Temperature. The temperature t in Eq. (4) and Eq. (3) controls the sharpness of the instance distribution. (Noted $t = 0$ is equivalent to the $argmax$ operation). We present the result of varying different t value in Figure 5a. As can be seen, the best Top-1 accuracy comes from $t = 0.1$, and slightly decreased when $t = 0.07$. This is consist with the most recent works in contrastive learning where $t = 0.1$ is generally the best temperature [10, 11, 14, 15].

Smooth Parameter. We also show the effective of different smooth parameter α Eq. (10) in Figure 5b. Specifically, we sweep over $[0.8, 0.9, 0.95, 1.0]$ for α, it clear to see that $\alpha = 0.9$ achieves the best result. Noted that $\alpha = 1.0$ is equivalent to directly take the original pseudo-labels p^w for Eq. (2), which result in 1.8% performance drop.

Label Propagation. Next, we would like to verify the effectiveness of the label propagation. The results has been shown in Table 5. When we remove $\widehat{p}$, this is the same case with $\alpha = 1.0$, so we will not discuss this setting further. If we remove $\widehat{q}$, which means the projection head will

(a) Pseudo-Labels Accuracy   (b) Unlabeled Samples Accuracy   (c) Validation Accuracy
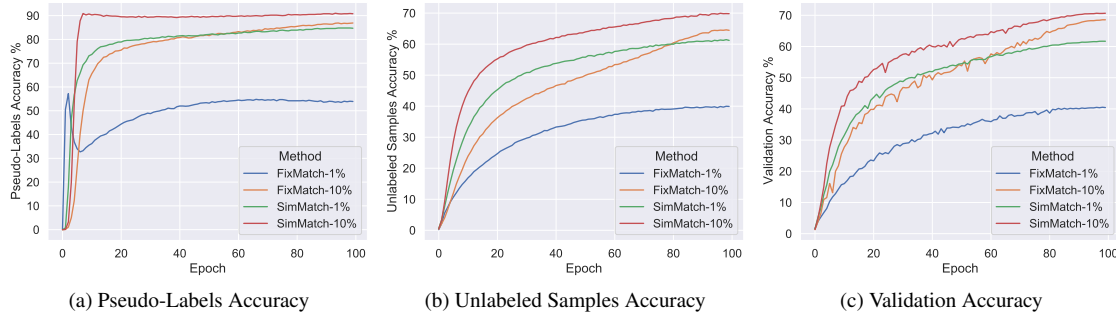
Figure 4. Visualization of (a) pseudo-labels accuracy - the accuracy of $\widehat{p}$ that has higher confidence than threshold, (b) unlabeled samples accuracy - the accuracy of all $\widehat{p}$ regardless of the threshold, (c) validation accuracy for FixMatch and SimMatch on 1% and 10% setting.
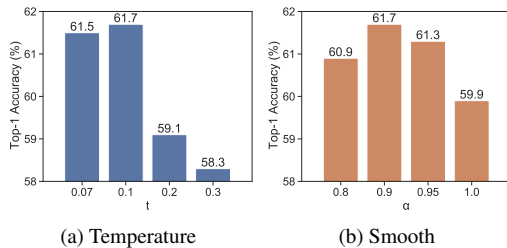

(a) Temperature   (b) Smooth

Figure 5. Results of varying t and α. (ImageNet-1k 1% - 100 ep)

Table 5. Results of removing scaling and smoothing strategy. (ImageNet-1k 1% - 100 ep)

Method	w/o $\widehat{p}$	w/o $\widehat{q}$	Standard
Top-1	59.9	52.3	**61.7**

Table 6. Results of different combinations for scaling and smoothing strategy. (ImageNet-1k 1% - 100 ep)

$\widehat{p}$ \ $\widehat{q}$	Scaling	Smoothing
Scaling	56.6	59.9
Smoothing	**61.7**	61.5

be trained in a fully unsupervised manner as in [66], as we can see the performance is significantly worse than standard SimMatch, demonstrating the importance of our label propagation strategy.

Propagation Strategy. Then, we tried different combinations of the scaling and smoothing strategy to generate the pseudo-labels $\widehat{p}$ and $\widehat{q}$. From Table 6, we can see that take smoothing for $\widehat{p}$ and scaling for $\widehat{q}$ achieves the best result. We might notice that applying smoothing to both $\widehat{p}$ and $\widehat{q}$ can achieve similar performance (61.5%). However, the smoothing strategy will introduce a smoothing parameter. Thus, for keeping our framework simple, we prefer to choose the scaling strategy for $\widehat{q}$.

Instance Matching Loss Design. To verify the effectiveness of the instance similarity matching term $\mathcal{L}_{in}$, we simply replace it with InfoNCE and SwAV. We show the result in Table 7 When working with InfoNCE loss, we sweep the temperature over [0.07, 0.1, 0.2]. In this case, the best result we can get is 53.5%, which is 8.2% lower than Sim-Match. This is due to the natural conflict between the clas-

Table 7. Results of replacing $\mathcal{L}_{in}$ with InfoNCE and SwAV. (ImageNet-1k 1% - 100 ep)

Method	InfoNCE	SwAV	SimMatch
Top-1	53.5	49.7	**61.7**

sification problem and InfoNCE objective. To be specific, the classification problem aims to group similar samples together, but InfoNCE aims to distinguish every instance. When working with SwAV, we tried to set the number of prototypes to 1000, 3000, and 10000. Finally, the best result we can get is 49.7%, which is 12% lower than SimMatch. SwAV aims to distribute the samples equally to each prototype, preventing the model from collapsing. However, distribution alignment has a similar objective, which is adopted by SimMatch in Eq (2) . Moreover, the SwAV loss will be trained in a completely unsupervised manner, which will lose the power of the labels. The advantage of $\mathcal{L}_{in}$ is that the label information can easily cooperate with instance similarities.

5. Conclusion

In this paper, we proposed a new semi-supervised learning framework SimMatch, which considers the consistency regularization on both semantic-level and instance-level. We also introduced a labeled memory buffer to fully leverage the data annotations on instance-level. Finally, our defined *unfolding* and *aggregation* operation allows the label to propagate between semantic-level and instance-level information. Extensive experiment shows the effectiveness of each component in our framework. The results on ImageNet-1K demonstrate the state-of-the-art performance for semi-supervised learning.

Acknowledgment

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016. 5

[2] S. Arora, Hrishikesh Khandeparkar, M. Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *ArXiv*, abs/1902.09229, 2019. 2

[3] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*, 2020. 1

[4] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. *arXiv preprint arXiv:2104.13963*, 2021. 6

[5] 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. *arXiv preprint arXiv:1911.09785*, 2019. 2, 3, 6

[6] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019. 2, 3, 6

[7] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014. 6

[8] Zhaowei Cai, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Zhuowen Tu, and Stefano Soatto. Exponential moving average normalization for self-supervised and semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 194–203, 2021. 6, 7

[9] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018. 1, 3

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. 2020. 1, 3, 6, 7

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. 7

[12] Olivier Chapelle, Mingmin Chi, and Alexander Zien. A continuation method for semi-supervised svms. In *Proceedings of the 23rd international conference on Machine learning*, pages 185–192, 2006. 1

[13] Mingcai Chen, Yuntao Du, Yi Zhang, Shuwei Qian, and Chongjun Wang. Semi-supervised learning with multi-head co-training. *arXiv preprint arXiv:2107.04795*, 2021. 2

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1, 2, 6, 7

[15] 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, 7

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

[17] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 3

[18] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 6

[19] Cheng Deng, Erkun Yang, Tongliang Liu, Jie Li, Wei Liu, and Dacheng Tao. Unsupervised semantic-preserving adversarial hashing for image search. *IEEE Transactions on Image Processing*, 28(8):4032–4044, 2019. 1

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 1, 6

[21] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9588–9597, October 2021. 1

[22] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1

[23] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[24] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005. 7

[25] 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, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 1, 3, 6, 7

[26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 1, 2, 5

[27] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 6

[29] Qianjiang Hu, Xiao Wang, Wei Hu, and Guo-Jun Qi. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. *arXiv preprint arXiv:2011.08435*, 2020. 3

[30] Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: Bridging the supervised and self-supervised learning. *arXiv preprint arXiv:2101.08732*, 2021. 1

[31] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. *arXiv preprint arXiv:2105.07269*, 2021. 3

[32] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 6

[33] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5, 6

[34] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 2, 5, 6

[35] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 2, 6, 7

[36] Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9475–9484, 2021. 3, 6, 7

[37] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021. 3, 7

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[39] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989. 6

[40] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[41] 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(8):1979–1993, 2018. 7

[42] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 6

[43] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 6

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 5

[45] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964. 6

[46] V Jothi Prakash and Dr LM Nithya. A survey on semi-supervised learning techniques. *arXiv preprint arXiv:1402.4645*, 2014. 1

[47] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171, 2016. 2

[48] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 2, 3, 5, 6, 7

[49] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2

[50] Xiu Su, Tao Huang, Yanxi Li, Shan You, Fei Wang, Chen Qian, Changshui Zhang, and Chang Xu. Prioritized architecture sampling with monto-carlo tree search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10968–10977, 2021. 1

[51] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013. 6

[52] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017. 2, 6

[53] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. 1

[54] Guangrun Wang, Keze Wang, Guangcong Wang, Philip H.S. Torr, and Liang Lin. Solving inefficiency of self-supervised representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9505–9515, October 2021. 1

[55] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 5

[56] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In H. Larochelle, M. Ranzato, R. Hadsell, M. F.

Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc., 2020. 6, 7

[57] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pages 11525–11536. PMLR, 2021. 6

[58] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and Xinbo Gao. Pairwise relationship guided deep hashing for cross-modal retrieval. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. 1

[59] Erkun Yang, Tongliang Liu, Cheng Deng, Wei Liu, and Dacheng Tao. Distillhash: Unsupervised deep hashing by distilling data pairs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2946–2955, 2019. 1

[60] Shan You, Tao Huang, Mingmin Yang, Fei Wang, Chen Qian, and Changshui Zhang. Greedynas: Towards fast one-shot nas with greedy supernet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1999–2008, 2020. 1

[61] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[62] Xiaohua Zhai, A. Oliver, A. Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1476–1485, 2019. 7

[63] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 2

[64] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[65] Mingkai Zheng, Fei Wang, Shan You, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Weakly supervised contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10042–10051, October 2021. 3, 7

[66] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. ReSSL: Relational self-supervised learning with weak augmentation. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 3, 8

[67] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009. 1