

# Revisiting Pretraining for Semi-Supervised Learning in the Low-Label Regime

Xun Xu<sup>1</sup>, Jingyi Liao<sup>1</sup>, Lile Cai<sup>1</sup>, Manh Cuong Nguyen<sup>1</sup>, Kangkang Lu<sup>1</sup>,  
Wanyue Zhang<sup>3</sup>, Yasin Yazici<sup>1</sup>, Chuan Sheng Foo<sup>1,2</sup>

---

## Abstract

Semi-supervised learning (SSL) addresses the lack of labeled data by exploiting large unlabeled data through pseudolabeling. However, in the extremely low-label regime, pseudo labels could be incorrect, a.k.a. the confirmation bias, and the pseudo labels will in turn harm the network training. Recent studies combined finetuning (FT) from pretrained weights with SSL to mitigate the challenges and claimed superior results in the low-label regime. In this work, we first show that the better pretrained weights brought in by FT account for the state-of-the-art performance, and importantly that they are universally helpful to off-the-shelf semi-supervised learners. We further argue that direct finetuning from pretrained weights is suboptimal due to covariate shift and propose a contrastive target pretraining step to adapt model weights towards target dataset. We carried out extensive experiments on both classification and segmentation tasks by doing target pretraining then followed by semi-supervised finetuning. The promising results validate the efficacy of target pretraining for SSL, in particular in the low-label regime.

## *Keywords:*

Semi-Supervised Learning, Transfer Learning, Self-Supervised Learning, Contrastive Learning

---

<sup>1</sup>Affiliated with the Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A\*STAR)

<sup>2</sup>Affiliated with the Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A\*STAR)

<sup>3</sup>Affiliated with the Max Planck Institute, Germany

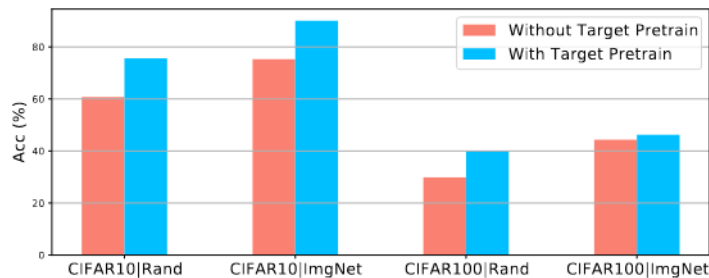


Figure 1: Illustration of FixMatch finetuning from different initial weights on CIFAR-10 (40 labeled data) and CIFAR-100 (400 labeled data). Target pretraining on target datasets (CIFAR-10 & CIFAR-100) improves semi-supervised learning performance respectively.

## 1. Introduction

Deep neural networks trained on large collections of labeled data have achieved unprecedented performance on many computer vision tasks. Unfortunately, acquiring these large amounts of labeled data is expensive and laborious. One effective strategy to mitigate this dependence on labels is exploiting unlabeled data for training, as done in semi-supervised learning (SSL).

Semi-supervised learning tackles the challenge of limited labeled data by exploiting a large amount of unlabeled data. Recent consistency-based SSL methods obtain pseudo-labels for unlabeled data through ensembles of predictions [1], network parameters [2] or weaker augmentations [3]. A student network is further trained on both labeled data and unlabeled data with pseudo-labels. Since these methods are directly trained on a specific task (dataset), we refer to this overall approach as task-specific SSL [4]. Task-specific SSL has demonstrated very competitive performance on simpler datasets like CIFAR-10 [5]. However, the performance of task-specific SSL in more challenging settings, for instance, fine-grained classification [6], the extremely low-label regime [6] and segmentation [7], is still far from satisfactory. This is partly because when few labels are available, there is simply not enough supervision to generate accurate pseudo-labels, thus misleading the training of student network [8].

In contrast to task-specific SSL, the finetuning (FT) approach involves pretraining a backbone network on a separate, large dataset in either supervised [9] or unsupervised [4, 10, 11, 12] manner, and then finetuning the network on the potentially smaller target dataset. For example, it is a com-

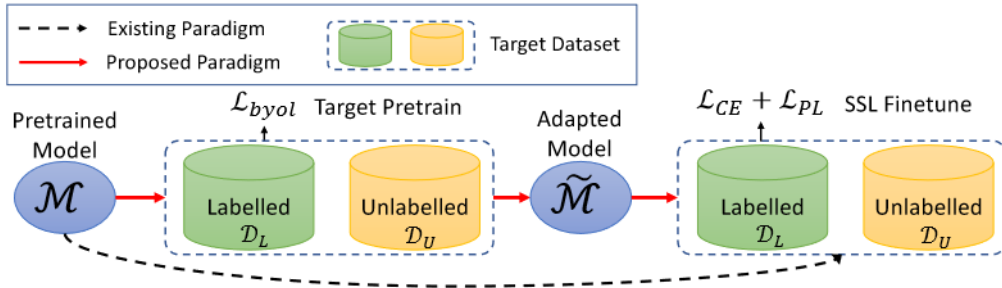


Figure 2: The pipeline for contrastive target pretraining for better initial weights. Existing works often use pretrained model weights to directly initialize semi-supervised finetuning on target dataset. In contrast, we introduce an additional adaptation step by doing contrastive target pretraining on the target dataset (both labeled and unlabeled) before semi-supervised finetuning.

mon practice in computer vision to start with a model pretrained on ImageNet [13], and then finetune it on the downstream task of interest. The FT paradigm has been successfully used across a wide range of computer vision tasks, including classification [14], segmentation [15], detection [16] etc.

The advantages of SSL and FT are often orthogonal as the former exploits unlabeled data to generate pseudo labeled data while the latter provides better initial model parameters. Therefore, synergistically combining FT with SSL, termed as finetuning + SSL, to enjoy the benefits of both strategies, has not been explored until recently. Among these attempts, SimCLR v2 [17] starts with ImageNet pretrained weights and then employs a supervised loss to finetune on a small labeled target dataset followed by distillation on unlabeled data. This initial attempt of finetuning + SSL in a non-trivial way demonstrated superior performance, compared with task-specific SSL methods, mainly in the low-label regime with very large network capacity. Along this direction, Self-Tuning [6] further proposed to combine cross-entropy loss with a multi-positive pair contrastive loss during finetuning. It was claimed that by incorporating this additional contrastive loss with pseudo-labels, confirmation bias [8] and model shift [18] issues can be mitigated, and Self-Tuning is reported to outperform state-of-the-art semi-supervised learners.

The reported large performance improvement of finetuning + SSL [17, 6] over task-specific SSL [2, 19, 20, 3], inspired us to explore the following questions: 1) What is the key ingredient that enables the success of finetuning +

SSL? 2) Does it generalize to task-specific SSL? 3) Is task-specific SSL still the most effective SSL approach? To answer these questions, we first make a critical observation that one major issue with the comparison of finetuning + SSL and task-specific SSL is the initialization of model weights. In particular, SimCLR v2 [17] and Self-Tuning [6] employed model weights pretrained on ImageNet while task-specific SSL methods are often trained with randomly initialized weights. Therefore, we hypothesize that the better *initial weights* provided by pretraining on the separate large dataset is largely responsible for the superior performance reported by finetuning + SSL. It is thus reasonable to believe, if the same initial weights are available to task-specific SSL methods, a similar performance boost could be expected. To test this hypothesis, we carry out empirical studies by comparing state-of-the-art task-specific SSL methods both with random initial weights and pretrained weights. As shown in Fig. 1, with FixMatch we observe a clear performance gap between random initialized weights “CIFAR-10/CIFAR-100—Rand” and ImageNet pretrained weights “CIFAR-10/CIFAR-100—ImgNet”. These observations suggest that the benefits of good initial weights also apply to task-specific SSL methods. With further empirical results in Sect. 4.3, we reveal that under fair comparison, task-specific SSL finetuning still outperforms state-of-the-art finetuning based methods by a large margin.

Based on the insights above, we further notice that pretraining is often carried out on a large pretrain dataset, e.g. ImageNet, while SSL finetuning is often on a smaller target dataset, e.g. CIFAR-10, CIFAR-100, PascalVOC. There is an inevitable distribution mismatch between pretrain and target datasets, commonly referred to as covariate shift [21]. The backbone network optimized for pretrain data distribution is subject to covariate shift and therefore will not be optimal for direct reuse on finetuning tasks. To bridge the data distribution gap, we introduce an additional target pretraining step before semi-supervised finetuning. Target pretraining is realized by doing contrastive training on both labeled and unlabeled data in the target dataset. With this additional step, model parameters will adapt to target data distribution and eventually improve finetuning performance. In particular, we find this target pretraining step is most effective when labeled data is sparse, reflecting that confirmation bias can be mitigated by a better initial weights.

We validate the effectiveness of target pretraining on both semi-supervised image classification and segmentation tasks. As augmentation is particularly important for segmentation tasks, we further introduce a differentiable geo-

metric augmentation module to enhance the representation learning ability for segmentation tasks. We summarize the contributions of this work as follows:

- We reveal that good initial weights through pretraining on large dataset is universally helpful to both finetuning based and task-specific semi-supervised learning approaches.
- Realizing the data distribution shift between pretrain and target datasets, we propose a target pretraining step to adapt pretrained weights to target dataset before semi-supervised finetuning.
- We demonstrate on both classification and semantic segmentation tasks that the proposed target pretraining strategy can substantially improve finetuning performance at low-label regime.

## 2. Related Work

### 2.1. Semi-Supervised Learning

The success of many computer vision tasks depends on training deep neural networks with large amount of labeled data which is expensive to acquire. Semi-supervised learning (SSL) aims to address learning with small labeled data and large unlabeled data [1, 2, 22, 20, 23, 3, 24, 25].

**Task-Specific SSL** is a major SSL paradigm which mainly produces pseudo-labels for unlabeled data to utilize them in training. In [1], pseudo-labels are produced as moving average of network predictions. To further exploit the ensemble of networks for stable and efficient pseudo-label prediction, Tarvainen et al. [2] proposed to utilize moving average of network parameters as teacher network which supervises a student network. More recently, data augmentation has been increasingly investigated under SSL. MixMatch [22] mixes up multiple images and their labels for stronger regularization for SSL. Xie et al. [20] argues that advanced data augmentation methods play a crucial role in semi-supervised learning. More recently, FixMatch [3] achieved significantly higher performance by simultaneously exploiting strong and weak augmentations for consistency matching. Alternative to self-training paradigm, exploiting graph constraints has been adopted for semi-supervised learning [26, 24]. Other than classification task, SSL has been adapted to semantic segmentation tasks [7, 27, 28] mainly adopting a Mean Teacher like

architecture. Specialized data augmentation strategy is mainly accountable for the improvement in segmentation tasks.

**Finetuning + SSL** is an emerging approach towards SSL [17, 6, 29] by combining transfer learning with semi-supervised finetuning. An initial attempt [17] in this direction pretrains the backbone network on ImageNet [5] via contrastive learning [4]. The network is further finetuned on limited labeled data, and then distilled on unlabeled data. Empirical study suggests the superiority of pretraining plus finetuning over task-specific SSL with small labeled data and large network capacity. Self-Tuning [6] further extended the framework by using ImageNet supervisely pretrained weights and finetune on target datasets by combining cross-entropy loss with contrastive loss. A significant improvement over task-specific SSL is observed on fine-grained classification tasks. In this work, observing the performance gap between finetuning + SSL and task-specific SSL, we suspect the initial weights are accountable and further reveal that improving initial weights by target pretraining are universally helpful to both paradigms.

## 2.2. Domain Adaptation

Transfer learning aims to reuse the data or model from source task/domain to target task/domain [30]. One of the key challenges in transfer learning is the domain shift between source and target datasets and domain adaptation aims to tackle this issue. When both source and target data are available during model training, UDA [31] introduced a discriminator to align the distribution between source and target domains. Nevertheless, the assumption of access to target domain data during model training restricts the application and recently source-free domain adaptation, a.k.a. test time training, emerges as a solution which no longer requires the access to labeled data during adaptation [32, 33, 34] at the cost of modifying the learning objective during training. To avoid altering training objective, which could be very expensive for large-scale pretraining tasks, Tent [35] adopted a self-training like approach to adapt only the affine transformation parameters for batch-norm layers by minimizing an entropy loss. Despite requiring no changes to training objective, Tent requires the prediction task to be consistent between source and target domains. In contrast to source-free domain adaptation and test time training, our target pretrain does not modify training objective in the source domain nor assume the prediction task to be identical between the source and target domains. A recent related work addressed finetuning on target domain in an unsupervised manner [36]. To avoid large deviations

from pretrained model parameters they simultaneously finetune on both target and source domain data. Therefore, the adaptation still requires access to source domain data.

### 2.3. Contrastive Learning

Contrastive learning [4, 10, 11, 12] is particularly effective in pretraining backbone network on large unlabeled dataset by enforcing the image-wise features subject to two augmentation to be consistent. As such the network is able to attend to the discriminative areas for feature extraction. To focus on dense prediction downstream tasks, e.g. segmentation, detection, etc., contrasting pixel-wise features was proposed [37, 38] to train the backbone network to generate local features invariant to augmentations. Contrastive learning has been demonstrated to be friendly to data-efficient learning [12] as the feature representation becomes more amenable for training classifiers after pretraining. In a concurrent work, given a pretrained model and small-scale unlabeled target data, unsupervised finetuning [36] is to adapt the representation pretrained from the source domain to the target domain so that better transfer performance can be obtained. Unsupervised finetuning differs from target pretraining in that revisiting source domain data is necessary to avoid destroying pretrained model while we simply apply weights regularization. Contrastive learning has been applied to improve finetuning on few-shot labeled data [39], however it is demonstrated that unlabeled data can still benefit the finetuning stage, i.e. semi-supervised learning. To improve model’s robustness to out-of-distribution testing data, [34] employed contrastive learning to adapt pretrained model to target domain distribution in an unsupervised manner. In this setup, source and target domains share the same label space. However, using contrastive learning to adapt pretrained weights to heterogeneous target dataset is yet to be fully explored, and in this work we employ contrastive learning as a target pretraining approach and demonstrate that semi-supervised finetuning can benefit substantially from target pretraining when labeled data is sparse.

## 3. Methodology

We first provide an overview of the proposed target pretraining strategy in Fig. 2. An initial model weights  $\mathcal{M}$  pretrained on large dataset, e.g. ImageNet, is assumed to be available. In addition, we have a target dataset consisting of labeled training samples  $\mathcal{D}_L = \{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1 \dots N_L}$  and unlabeled

samples  $\mathcal{D}_U = \{\mathbf{X}_j, \mathbf{Y}_j\}_{j=1 \dots N_U}$ . A key step is to adapt pretrained model weights by doing target pretraining on the target dataset. In the rest of this section, we first briefly review the contrastive learning approach. To further improve contrastive target pretraining for dense prediction tasks, we include a differentiable affine transformation [40] to improve the contrastive learning efficacy. Lastly, we provide insights into why contrastive training improves adaptation pretrained model to target domain data.

### 3.1. Preliminaries on Contrastive Learning

We use  $\mathbf{Z} = f(\mathbf{X}; \Phi)$  to denote a backbone network, parameterized by  $\Phi$ , which takes input image  $\mathbf{X}$  and produces a feature embedding  $\mathbf{Z}$ . We further denote a projection head, parameterized by  $\Psi$ , as  $\hat{\mathbf{E}} = g(\mathbf{Z}; \Psi)$  which outputs a lower dimension embedding. Contrastive learning is formulated as minimizing a contrastive loss defined over two augmentations of the same input image. Formally, the InfoNCE loss [41] is adopted as contrastive loss as in Eq. (1).

$$\mathcal{L}_{InfoNCE} = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k \neq i} \exp(s_{i,k}/\tau)} \quad (1)$$

In this formulation, the positive pair  $(i, j)$  is two different augmentations of the same input image, and negative pairs  $(i, k)$  are constructed as different images and  $s_{i,j}$  defines a cosine similarity between feature encodings of pair  $(i, j)$ . It is identified in [4] that a large batchsize is essential to contrastive learning performance which is computationally expensive. Instead of maintaining both positive and negative pairs, Bootstrap your own latent (BYOL) [10] relies on similarity of features between a target network and an online network with target network being a parameter moving average of online network; the output of target network is used as a target for the online network to match against. Because of the slowly updating target network, trivial solutions are prevented without requiring large batchsize. BYOL optimizes the cosine distance, as in Eq. 2, between the representation subject to two augmentations as below, where  $\hat{\mathbf{E}}_1 = g(f(t_1(\mathbf{X})))$  and  $\hat{\mathbf{E}}_2 = g(\hat{f}(t_2(\mathbf{X})))$  are respectively the features after projection head with two augmentations  $t_1(\cdot)$  and  $t_2(\cdot)$  and  $g(\cdot)$  is a predictor. The model predicting the second view is an exponential moving average of the first view. An illustration of BYOL is presented in Fig. 3 where two random augmentations  $t_1(\mathbf{X})$  and  $t_2(\mathbf{X})$  are applied, and the byol loss is defined as the cosine

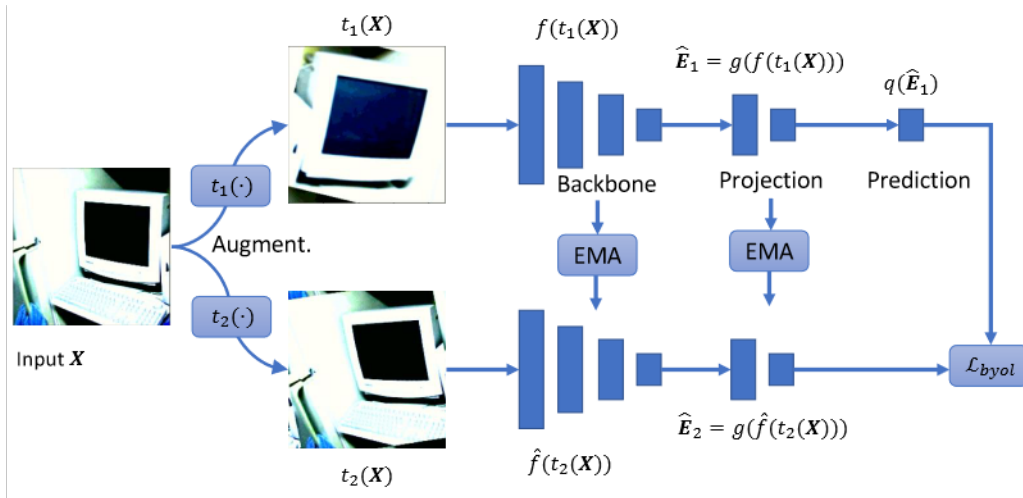


Figure 3: Illustration of contrastive training through BYOL. We apply two random augmentations to the input  $\mathbf{X}$ , followed by backbone and projection. The cosine distance loss is calculated between the view one feature after Predictor and view two feature.

distance between the predictions of two views.

$$\mathcal{L}_{byol} = -\frac{q(\hat{\mathbf{E}}_1)^\top \hat{\mathbf{E}}_2}{\|q(\hat{\mathbf{E}}_1)\| \cdot \|\hat{\mathbf{E}}_2\|} \quad (2)$$

**Dense Contrastive Pretraining:** BYOL is carried out at instance-level thus a global average pooling layer is appended at the end of backbone network to obtain a single feature vector for each image. For dense prediction tasks, e.g. segmentation, contrastive pretraining should be carried out with spatial awareness. PixPro [37] proposed a dense contrastive learning approach where global average pooling layer is removed, and pixel-wise features are used for contrasting instead of image-wise features adopted in BYOL. One of the most critical ingredients in contrastive training is data augmentation. An investigation into effective data augmentation was carried out in [4] and for dense prediction PixPro [37] also follows the augmentation policies developed for classification tasks. CNNs are known to be translation equivariant, i.e. feature encoding of a translated image can be obtained by translating the feature encoding of the original image. Therefore, cropping and weak resizing, which are adopted in PixPro as augmentation, are not strong enough to force the network to attend to locally semantic features.

To further improve the effectiveness of pixel-wise contrastive training, we propose to include stronger differentiable affine transformations [40] to achieve more diverse geometric poses in augmented images. An affine transformation involves a combination of *scaling*, *rotation*, *shearing* and *translation*. During augmentation, one affine transformation is constructed by randomly choosing each of the 4 operations with 50% chance and drawing parameters from a predefined distribution. For input image, we apply the affine transformation after all non-differentiable data augmentation operations, e.g. Gaussian blur, color jittering, etc. As with STN [40], we use bilinear interpolation to estimate the pixel intensity after transformation. We denote the transformation applied to image  $\mathbf{X}$  as  $t(\mathbf{X})$ . To enable computing pixel-wise contrastive loss, we further apply the inverse transformation to the output of both online and target networks as  $\mathbf{Z} = t^{-1}(f(t(\mathbf{X})))$ . It is easy to verify that the affine transformation and its inverse are differentiable w.r.t. its input. With the inverse transformation, predictions on two arbitrarily augmented images are directly comparable at pixel-level. An illustration of images augmented with and without affine transformation is shown in Fig. 4. Our proposed augmentation involves affine transformation which further improves generalization.

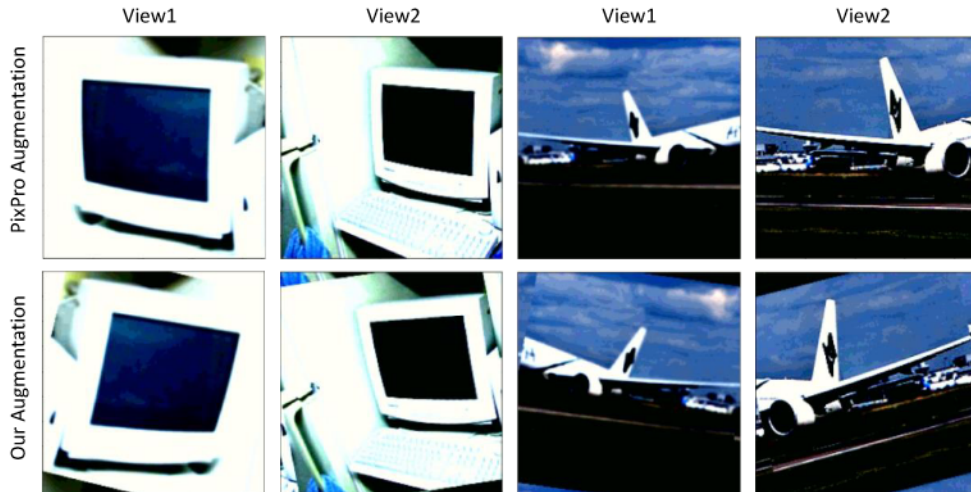


Figure 4: Examples comparing the augmentation adopted in PixPro [37] and our stronger augmentation which involves affine transformation.

### 3.2. Target Pretraining for SSL

**Contrastive Target Pretraining:** In a regular transfer learning setting, we first assume a backbone network, parameterized by  $\Phi$ , is pretrained on a large dataset, e.g. ImageNet. After pretraining, the model  $\mathcal{M}$  comes with parameter weights  $\Phi_{pre}$ . Then,  $\Phi_{pre}$  is used to initialize the backbone network for the downstream/target task, e.g. finetuning on a smaller dataset or for semantic segmentation. As discussed, we believe directly re-using  $\Phi_{pre}$  as initial weights for finetuning is sub-optimal. Instead, we propose to apply contrastive target pretraining on the target dataset. Specifically, we adopt BYOL contrastive loss with a regularization as below,

$$\min_{\Phi, \Psi} - \sum_{\mathbf{X}_i \in \mathcal{D}_L \cup \mathcal{D}_U} \frac{q(\hat{\mathbf{E}}_{i1})^\top \hat{\mathbf{E}}_{i2}}{\|q(\hat{\mathbf{E}}_{i1})\| \cdot \|\hat{\mathbf{E}}_{i2}\|} + \lambda \|\Phi - \Phi_{pre}\|_2^2 \quad (3)$$

We add a L2 weight regularization to prevent target pretrained weights from deviating too much from the initial weights. For a gradient-based optimization,  $\Phi$  is initialized by  $\Phi_{pre}$ . We find that by simply training enough iterations on target dataset, better weights can be obtained for downstream semi-supervised finetuning. We denote the weights after  $T$ -epoch target pretraining as  $\Phi^T$ .

**Semi-Supervised Finetuning:** After target pretraining, the backbone weights  $\Phi^T$  serve as a better starting point for finetuning on target dataset. For simplicity, we denote the classifier as  $h(\mathbf{Z}; \Theta)$ . Semi-supervised finetuning is then defined as optimizing the semi-supervised loss as,

$$\begin{aligned} \min_{\Theta, \Phi} \frac{1}{|\mathcal{D}_L|} \sum_{\mathbf{X}_i, \mathbf{Y}_i \in \mathcal{D}_L} \mathcal{L}_{CE}(h(f(\mathbf{X}_i; \Phi); \Theta), \mathbf{Y}_i) \\ + \lambda \frac{1}{|\mathcal{D}_U|} \sum_{\mathbf{X}_j \in \mathcal{D}_U} \mathcal{L}_{PL}(h(\mathbf{Z}_{j1}; \Theta), h(\mathbf{Z}_{j2}; \Theta)) \end{aligned} \quad (4)$$

The pseudo labeling loss  $\mathcal{L}_{PL}$  can be any instance of off-the-shelf semi-supervised learners, e.g. FixMatch [3].

### 3.3. Further Insights into Target Pretraining

We discuss in this part the hypotheses of why this simple target pretraining could be effective. Some early works managed to explain deep neural network pretraining from providing a regularization effect on model parameters [42]. For example, pretraining will initialize the parameters in the basin

of attraction of a good local optimum. Therefore, the following supervised finetuning will be easier to generalize. Another recent empirical study [9] argued that when there is enough labeled data, training from scratch is comparable to finetuning from pretrained weights. Nevertheless, a breakdown point, i.e. without pre-training the performance becomes much worse, was observed when labeled data becomes very scarce [9]. This observation agrees with the empirical results in this work, indicating pretraining is necessary when labeled data is low.

In contrast to the previous attempts to explain, we would like to analyze from a covariate shift point of view. Without loss of generality, we describe a general supervised pretraining and finetuning example; the conclusion applies to semi-supervised finetuning too. For simplicity, we denote the unlabelled pretraining dataset as  $\mathcal{D}_{pre}$ , e.g. the ImageNet dataset, and the target finetuning dataset as  $\mathcal{D}_{ft} = \mathcal{D}_L \cup \mathcal{D}_U$ . The objective of supervised pretraining is often realized as minimizing the following cross entropy loss.

$$\Theta_{pre}^*, \Phi_{pre}^* = \arg \min_{\Theta, \Phi} \frac{1}{|\mathcal{D}_{pre}|} \sum_{\mathbf{X}_i, \mathbf{Y}_i \in \mathcal{D}_{pre}} \mathcal{L}_{CE}(h(f(\mathbf{X}_i; \Phi); \Theta), \mathbf{Y}_i) \quad (5)$$

The objective of finetuning can be written below.

$$\Theta_{ft}^*, \Phi_{ft}^* = \arg \min_{\Theta, \Phi} \frac{1}{|\mathcal{D}_{ft}|} \sum_{\mathbf{X}_i, \mathbf{Y}_i \in \mathcal{D}_{ft}} \mathcal{L}_{CE}(h(f(\mathbf{X}_i; \Phi); \Theta), \mathbf{Y}_i) \quad (6)$$

The data distribution between pretraining and finetuning stages are not identical thus leading to a covariate shift, i.e.  $p_{pre}(\mathbf{X}) \neq p_{ft}(\mathbf{X})$ . It is reasonable to believe the optimal backbone networks subject to pretraining and finetuning objectives respectively are not identical  $f(\mathbf{X}_i; \Phi_{pre}^*) \neq f(\mathbf{X}_i; \Phi_{ft}^*)$  because of the covariate shift. As a result, directly re-using  $f(\mathbf{X}_i; \Phi_{pre}^*)$  to initialize the supervised finetuning task is sub-optimal and we propose to further pretrain the backbone on the whole target dataset as in Eq. (7).

$$\Phi_{trg-pre}^* = \arg \min_{\Phi} \frac{1}{|\mathcal{D}_{ft}|} \sum_{\mathbf{X}_i \in \mathcal{D}_{ft}} \mathcal{L}_{byol}(\mathbf{X}_i; \Phi) \quad (7)$$

During target pretraining backbone parameters are initialized by the pre-trained parameters, i.e.  $\Phi_{trg-pre}^0 = \Phi_{pre}^*$ . With the pretrained backbone, we can further do semi-supervised finetuning by initializing with the target

pretrained weights, i.e.  $\Phi_{ft}^0 = \Phi_{trg-pre}^*$ . Compared with direct finetuning from  $\Phi_{pre}^*$ , target pretrained weights  $\Phi_{trg-pre}^*$  is first updated on the whole target dataset  $\mathcal{D}_{ft}$ . With covariate shift being eliminated,  $\Phi_{trg-pre}^*$  will move towards the optimal weights for target task. Eventually,  $\Phi_{trg-pre}^*$  as initial weights for supervised finetuning can ease the difficulty in optimization for finetuning and mitigate the confirmation bias issue. In the experiment section 4.4.1, we provide empirical evaluations on transferring target pretrained weights across different datasets to validate the hypothesis.

### 3.4. Pretraining and Semi-Supervised Finetuning Algorithm

In this section, we present the overall pretraining and semi-supervised finetuning algorithm in Alg. 1. Specifically, given labeled and unlabeled data  $\mathcal{D}_L$  and  $\mathcal{D}_U$  and pre-trained backbone weights  $\Phi_{pre}$ , we first apply BYOL target pretraining on  $\mathcal{D}_L \cup \mathcal{D}_U$  to adjust the backbone weights  $\Phi_t$  and projector head weights  $\Psi_t$  at step  $t$ . During the semi-supervised finetuning stage, we use the target pretrained backbone weights to initialize backbone network  $\Phi_0 = \Phi_T$  while the classifier network  $\Theta_0$  is randomly initialized. We then implement semi-supervised learning with the initialization.

## 4. Experiments

In this section, we introduce the datasets to evaluate, experiment settings and then provide benchmarking on semi-supervised finetuning with different initial weights. Finally, we validate the hypothesis of target pretraining as an adaptation through empirical evidence.

### 4.1. Datasets

We demonstrate the effectiveness of the proposed contrastive target pretraining strategy on six datasets. For classification tasks, we evaluate on CIFAR-10 [5], CIFAR-100 [5] and SVHN [43]. We further evaluate on additional semantic segmentation datasets, Pascal VOC 2012 [44], CityScapes [45] and ISIC2017 [46]. **CIFAR-10** is a image classification dataset with 50,000 training images and 10,000 testing images evenly categorized into 10 classes. For semi-supervised learning setting, we evaluate at 40, 250 and 1000 labeled data. **CIFAR-100** extended from CIFAR-10 with 10 times more categories and the same amount of training/testing samples. Therefore, the categories are more fine-grained than CIFAR-10. For semi-supervised learning setting, we evaluate at 400 and 2500 labeled data. **SVHN** is a street view house

---

**Algorithm 1:** Target Pretraining for Semi-Supervised Learning

---

**input :** Labeled data  $\mathcal{D}_L = \{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1 \dots N_L}$ , unlabeled data

$\mathcal{D}_U = \{\mathbf{X}_j, \mathbf{Y}_j\}_{j=1 \dots N_U}$  and pre-trained backbone weights

$\Phi_{pre}$

# Target Pretraining Stage ( $T_{pre}$  steps):

Initialize  $\Phi_0 = \Phi_{pre}$ ,  $\Psi_0$  by Kaiming Init.

**for**  $t = 1 \dots T_{pre}$  **do**

    Sample a minibatch  $\mathcal{B}^t \in \mathcal{D}_L \cup \mathcal{D}_U$

    Calculate  $\mathcal{L} = - \sum_{\mathbf{x}_i \in \mathcal{B}^t} \frac{q(\hat{\mathbf{E}}_{i1})^\top \hat{\mathbf{E}}_{i2}}{\|q(\hat{\mathbf{E}}_{i1})\| \cdot \|\hat{\mathbf{E}}_{i2}\|} + \lambda \|\Phi_t - \Phi_{pre}\|_2^2$  by Eq. 2

    Gradient descent update  $\Phi_t, \Psi_t$

# Semi-Supervised Finetuning Stage ( $T_{ft}$  steps):

Initialize weights  $\Phi_0 = \Phi_T$ ,  $\Theta_0$  by Kaiming Init.

**for**  $t = 1 \dots T_{ft}$  **do**

    Sample a mimibatch  $\mathcal{B}_L^t \in \mathcal{D}_L$ ,  $\mathcal{B}_U^t \in \mathcal{D}_U$

    Calculate  $\mathcal{L} = \frac{1}{|\mathcal{B}_L^t|} \sum_{\mathbf{x}_i, \mathbf{Y}_i \in \mathcal{B}_L^t} \mathcal{L}_{CE} + \lambda \frac{1}{|\mathcal{B}_U^t|} \sum_{\mathbf{x}_i \in \mathcal{B}_U^t} \mathcal{L}_{PL}$  by Eq. 4

    Gradient descent update  $\Phi_t, \Theta_t$

---

numbers recognition dataset with all numbers cropped and registered. There are 73257 digits for training, 26032 digits for testing respectively. For semi-supervised learning setting, we evaluate at 40 labeled data. **Pascal VOC 2012** has been widely adopted for semantic segmentation tasks. The dataset consists of 1464 training images and 1449 validation images. Following the practice in [7, 37] we augment the training set with additional training images, resulting in 10582 training images in total. For semi-supervised learning, we evaluate at 1%, 2% and 5% of the labeled data. **CityScapes** was collected from cars driving in urban environment with aim to segment objects commonly seen on the street. It has 2975 training and 500 validation samples, with 19 semantic classes. We evaluate at 100 and 372 labeled data. **ISIC2017** is a medical image segmentation dataset detecting skin lesions. It has 2000 training images and features a significant difference from the source ImageNet dataset. We evaluate SSL at 10, 20 and 50 labeled data.

#### 4.2. Experiment Settings

**Backbone Network:** We adopt Resnet18 and Resnet50 [47] for classification and segmentation tasks respectively, as the weights pretrained on ImageNet is publicly available.

**Data Augmentation:** Since backbone network is pretrained on ImageNet with input size  $224 \times 224$ , for target pretrain and finetuning, we first apply random cropping and then resize cropped region to  $224 \times 224$ . We further apply the same augmentations in BYOL [10] for target pretraining on classification tasks and with additional affine transformation for segmentation tasks.

**Contrastive Target Pretraining:** We adopt BYOL [10] and PixPro [37] for contrastive target pretraining for classification and segmentation respectively. We reduce the learning rate 10 times from the default ImageNet experiments in respective methods and further pretrain 300 epochs. For BYOL, we use the following hyperparameters, batchsize 256, base learning rate 0.3, momentum 0.9 and total epochs 300. For PixPro, we adopt the following hyperparameters, batchsize 8, base learning rate 0.1, momentum 0.9 and total epochs 200. The LARS optimizer is adopted for both pretraining. Weight regularization strength is fixed as  $\lambda = 1e - 2$ .

**Semi-Supervised Finetuning:** In classification task we use FixMatch [3] as the semi-supervised learner, we stick to the original hyperparameters for CIFAR-10, CIFAR-100 and SVHN reported in [3]. The following ones are adopted, the confidence threshold  $\tau = 0.95$ , the unlabeled loss weight  $\lambda_u = 1$ , the unlabeled to labeled data ratio in a minibatch  $\mu = 7$ , the batchsize  $B = 64$  and the learning rate  $1e - 1$ . SGD optimizer is adopted with momentum  $\beta = 0.9$ . For segmentation tasks, we adopt CutMix [7] as the semi-supervised learner. We adopt the following hyperparameters, the initial learning rate  $1e - 5$ , batchsize 20/8 for PascalVOC and CityScapes respectively, iterations per epoch 1000 and the number of total training epochs 40. The Adam optimizer is adopted for both datasets. We randomly crop out a  $321 \times 321$  and  $256 \times 512$  region for training on PascalVOC and CityScapes respectively. For CityScapes, we downsample the image to  $512 \times 512$  to allow more efficient training.

#### 4.3. Contrastive Target Pretraining

In this section, we investigate the effectiveness of contrastive target pretraining on classification and segmentation tasks.

#### 4.3.1. Classification Tasks.

We first evaluate fully supervised baseline, Mean Teacher (MT) [2] and virtual adversarial training (VAT) [19] as reference methods. We further evaluate two state-of-the-art SSL approaches, Self-Tuning [6] and FixMatch [3] with different initial weights. As shown in Tab. 1 and Tab. 2, we use “Pre-train” to refer to weights pretrained on large pretraining dataset, “Trg. Pre.” indicates contrastive target pretraining and “SSL” indicates the semi-supervised learner. We make the following observations from this table.

Table 1: Evaluation of the impact of target pretraining on CIFAR-10 dataset. “IN Sup” indicates supervise pretrained on ImageNet. We report the average and standard deviation over 5 runs. All numbers are in (%).

Pretrain	Trg. Pre.	SSL	#40	#250	#1000
-	-	MT [2]	27.37±2.92	47.40±1.72	65.61±1.18
-	-	VAT [19]	31.90±2.55	57.09±1.39	75.75±1.09
-	-	-	18.45±4.31	29.82±2.71	45.05±1.75
-	BYOL	-	49.58±1.58	66.43±1.21	75.36±1.07
IN Sup	-	-	49.74±1.61	77.59±0.96	85.60±0.91
IN Sup	BYOL	-	<b>65.32±1.19</b>	<b>82.19±0.94</b>	<b>87.86±0.89</b>
-	-	Self-Tuning [6]	27.95±2.86	45.87±1.68	61.17±1.29
-	BYOL	Self-Tuning	49.78±1.70	73.88±1.11	82.29±1.00
IN Sup	-	Self-Tuning	48.90±1.67	82.60±0.97	90.02±0.93
IN Sup	BYOL	Self-Tuning	<b>62.62±1.27</b>	<b>84.87±0.96</b>	<b>90.98±0.92</b>
-	-	FixMatch [3]	60.83±1.36	91.48±0.85	93.04±0.89
-	BYOL	FixMatch	75.60±1.11	91.75±0.85	94.37±0.91
IN Sup	-	FixMatch	75.33±1.07	94.46±0.86	<b>95.92±0.81</b>
IN Sup	BYOL	FixMatch	<b>90.21±0.88</b>	<b>95.10±0.85</b>	95.51±0.84

i) Target pretraining with BYOL is very effective regardless of the initial weights being ImageNet supervised pretrained or from scratch. For example, if naive supervised finetuning is adopted, target pretraining improves 30% and 15% respectively from random initialized weights and ImageNet weights at CIFAR-10 with 40 labeled samples. The improvement is maintained with SelfTuning and FixMatch as semi-supervised learner on all datasets and labeling budgets.

ii) As expected, the gap diminishes when more labeled data becomes available and SOTA semi-supervised learner is adopted for finetuning. For example, on CIFAR-10 with 250 and 1000 labeled samples with FixMatch finetuning, the improvement is negligible. This is due to that more labeled data can mitigate the confirmation bias during semi-supervised learning.

Table 2: Evaluation of the impact of target pretraining on CIFAR-100 and SVHN datasets. “IN Sup” indicates supervise pretrained on ImageNet. We report the average and standard deviation over 5 runs. All numbers are in (%).

Pretrain	Trg. Pre.	SSL	CIFAR-100		SVHN
			#400	#2500	#40
-	-	MT [2]	10.36±7.79	37.49±2.13	27.73±2.93
-	-	VAT [19]	12.85±6.20	32.32±2.52	15.21±5.21
-	-	-	08.67±9.24	21.69±3.69	10.47±7.69
-	BYOL	-	15.18±5.29	35.20±2.36	18.25±4.39
IN Sup	-	-	30.85±2.64	57.36±1.40	16.92±4.65
IN Sup	BYOL	-	<b>34.63±2.30</b>	<b>58.65±1.36</b>	<b>35.82±2.23</b>
-	-	Self-Tuning [6]	09.17±8.77	23.72±3.39	13.49±5.87
-	BYOL	Self-Tuning	19.05±4.17	41.40±1.82	22.23±3.52
IN Sup	-	Self-Tuning	35.56±2.24	62.86±1.19	45.84±1.74
IN Sup	BYOL	Self-Tuning	<b>38.36±2.12</b>	<b>67.31±1.21</b>	<b>46.95±1.72</b>
-	-	FixMatch [3]	29.85±2.69	59.12±1.31	32.76±2.45
-	BYOL	FixMatch	40.41±2.00	63.34±1.26	36.94±2.18
IN Sup	-	FixMatch	44.34±1.79	71.81±1.08	79.89±1.01
IN Sup	BYOL	FixMatch	<b>46.23±1.66</b>	<b>72.19±1.10</b>	<b>92.97±0.90</b>

iii) Self-Tuning (ST) [6] proposed to combine cross-entropy loss with a contrastive loss during finetuning. ST was originally claimed to be superior to FixMatch at low-label regime [6]. However, through our extensive comparison, we conclude that the advantage of ST is mainly due to better initial weights. For example, with the best initial weights (IN Sup+BYOL), ST is still far behind FixMatch on all datasets. This implies that developing more effective pseudo-labeling algorithm for SSL is still one of the most effective ways to exploit unlabeled data. Nevertheless, good initial weights are equally important, combining finetuning with semi-supervised learning should receive more attention for more data-efficient SSL.

#### 4.3.2. Segmentation Tasks.

We further present semi-supervised finetuning results on semantic segmentation task in Tab. 3. We compare three state-of-the-art semi-supervised segmentation approaches, CutMix [7], ClassMix [27] and CLMB [28]. We make the following observations from the segmentation experiments.

i) Contrastive target pretraining demonstrates consistent effectiveness on semantic segmentation tasks. The improvement is particularly large for CutMix at low-label regime, e.g. Pascal VOC 1% and CityScapes 100 labeled data. This again implies the bottleneck of from-scratch semi-supervised fine-

tuning is prone to confirmation bias when labeled data is small. As a result, better initial weights are required.

ii) It is also surprising to see with the target pretraining step, the final results of CutMix can sometimes match or slightly outperform its counterpart with ImageNet pretrained weights. It is worth noting that no labels are required for PixPro to pretrain ImageNet, thus a substantial amount of labeling efforts is saved under this setting.

Table 3: Evaluation of Semi-supervised finetuning with different model initial weights for semantic segmentation. IN PixPro indicates weights pretrained with PixPro on ImageNet. We report the average and standard deviation over 5 runs. The results for CutMix, ClassMix and CLMB are reported in respective papers.

Pretrain	Trg. Pre.	SSL	Pascal VOC			CityScapes	
			1%	2%	5%	#100	#372
IN Sup	-	CutMix [7]	53.79	64.81	66.48	57.89	65.64
IN Sup	-	ClassMix [27]	40.64	52.84	59.82	-	-
IN Sup	-	CLMB [28]	-	63.40	69.10	64.90	70.00
IN PixPro	-	-	38.20±2.07	49.95±1.56	60.35±1.34	52.99±1.52	63.05±1.27
IN PixPro	PixPro	-	41.35±1.96	52.97±1.53	63.11±1.32	53.26±1.53	63.26±1.28
IN PixPro	-	CutMix	49.58±1.62	<b>61.63±1.32</b>	67.61±1.20	58.30±1.35	<b>68.21±1.24</b>
IN PixPro	PixPro	CutMix	<b>52.20±1.57</b>	61.25±1.38	<b>68.59±1.17</b>	<b>59.20±1.37</b>	68.02±1.15

#### 4.3.3. Transferring to Medical Image Segmentation

We further evaluate target pretraining on target datasets with a large domain gap by performing an experiment transferring a model pretrained on ImageNet to a skin lesion segmentation dataset, ISIC2017 [48]. We follow the settings proposed in CutMix and consider labeling budgets of 10, 20 and 50 labeled samples. Results are shown in Tab. 4. For finetuning on a more specialised dataset, we also observe consistent improvement when target pretraining is introduced. The gap is obviously more significant at the lower labeling regime, e.g. 10 labeled images. These results again validate the effectiveness of target pretraining at low label regime.

#### 4.3.4. Runtime Complexity Analysis

The proposed target pretraining introduces additional computation overhead due to the pretraining step. Therefore, we further investigate a comparison by allowing model with and without target pretraining to be trained with the same overall wall-clock time (target pretraining time + semi-supervised finetuning time). For this purpose, we increase the training epochs of without target pretraining for all datasets such that the average wall-clock time is

Table 4: Target pretraining (Trg. Pre.) on skin lesion segmentation dataset (ISIC2017). We report the average and standard deviation over 5 runs. Result in () was reported in [7].

#Labeled	Pretrain	Trg. Pre.	mIoU
10	IN Sup	-	73.65±1.25
	IN Sup	BYOL	<b>77.60±1.34</b>
20	IN Sup	-	82.90±1.45
	IN Sup	BYOL	<b>82.96±1.21</b>
50	IN Sup	-	86.00±1.08 (74.57)
	IN Sup	BYOL	<b>86.04±1.10</b>

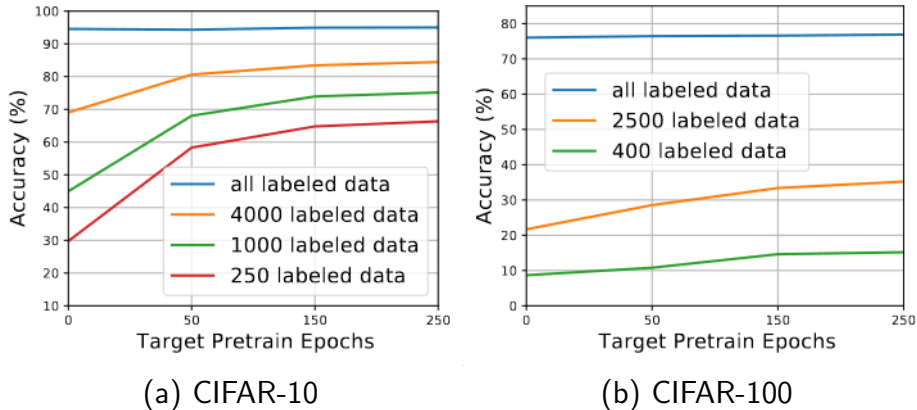


Figure 5: The impact of target pretraining on CIFAR-10 and CIFAR-100 datasets. We plot the testing accuracy after using the pre-trained weights for finetuning against target pretraining epochs.

comparable to with target pretraining. Specifically, we train 300 epochs for CIFAR-10/100 and SVHN without target pretraining, and 60 epochs for PascalVOC, CityScapes and ISIC. All wall-clock times are recorded by running experiments on DGX-1 with a single V100 GPU. We report the wall-clock time (W.C.T.) as hours in Tab. 5 and Tab. 6. The results with target pretraining are still significantly ahead of without target pretraining even with the same amount of overall training time. The additional training steps do not improve the performance for without target pretraining. This is also aligned with the observations made from Fig. 7 where the validation accuracy v.s. training epoch plots showcase the advantage of target pretraining.

Table 5: Overall Wall-Clock Time (W.C.T.) evaluation on with and without target pre-training for classification tasks. All results are averaged over 5 runs.

Pretrain	Trg. Pre.	SSL	CIFAR-10 #40		CIFAR-100 #400		SVHN # 40	
			Accuracy (%)	W.C.T.	Accuracy (%)	W.C.T.	Accuracy (%)	W.C.T.
IN Sup	-	FixMatch	75.33±1.07	90h±5h	44.34±1.79	88h±5h	79.89±1.01	129h±6h
IN Sup	-	FixMatch	75.52±1.09	138h±7h	44.28±1.75	136h±7h	80.16±1.03	199h±8h
IN Sup	BYOL	FixMatch	<b>90.21±0.88</b>	140h±8h	<b>46.23±1.66</b>	138h±8h	<b>92.97±0.90</b>	201h±8h

Table 6: Overall Wall-Clock Time (W.C.T.) evaluation on with and without target pre-training for segmentation tasks. All results are averaged over 5 runs.

Pretrain	Trg. Pre.	SSL	PascalVOC 1%		CityScapes #100		ISIC #10	
			mIoU (%)	W.C.T.	mIoU (%)	W.C.T.	mIoU (%)	W.C.T.
IN PixPro/Sup	-	CutMix	49.58±1.62	75h±4h	58.30±1.35	95h±5h	73.65±1.25	65h±2h
IN PixPro/Sup	-	CutMix	49.62±1.65	118h±5h	58.29±1.35	146h±7h	74.10±1.29	90h±3h
IN PixPro/Sup	PixPro/BYOL	CutMix	<b>52.20±1.57</b>	120h±5h	<b>59.20±1.37</b>	148h±8h	<b>77.60±1.34</b>	91h±3h

#### 4.4. Ablation Study

In this section, we carry out ablation study to verify the importance of weight regularization and affine transformation as augmentation. We carry out ablation on PascalVOC segmentation task. As shown in Tab. 7, we observe both weight regularization and affine transformation as augmentation improve the generalization of further pretrained model weights.

Table 7: Ablation study of weight regularization and differentiable affine transformation as augmentation.

Weight Reg.	Affine Aug.	PascalVOC		
		1%	2%	5%
-	-	38.20	49.95	60.35
✓	-	40.32	52.00	62.39
✓	✓	<b>41.35</b>	<b>52.97</b>	<b>63.11</b>

We perform additional evaluations on alternative contrastive learning and semi-supervised learning methods on CIFAR-10 with 40 labeled samples. We perform target pretraining with MoCo v2 [49] in addition to BYOL on CIFAR-10 with ImageNet pretrained weights and SSL finetuning with UDA [31] and MeanTeacher in addition to FixMatch. From Tab. 8, we observe consistent improvements with alternative contrastive learning methods and we observe the advantage of target pretraining with additional SSL methods as well.

Table 8: Additional contrastive learning and semi-supervised finetuning methods.

Pretrain	Trg. Pre.	Semi-Supervised Finetune			
		-	FixMatch [3]	UDA [31]	MT [2]
IN Sup	-	49.69	75.28	62.94	48.98
IN Sup	BYOL [10]	65.13	90.06	78.43	61.55
IN Sup	MoCo v2 [49]	71.13	90.37	87.04	69.02

#### 4.4.1. Target Pretraining as Adaptation

In this experiment, we validate the hypothesis of target pretraining as an adaptation of weights pretrained on generic dataset to specific target dataset. We keep all the target pretraining protocols the same with Sect. 4.3, an ImageNet pretrained weights are taken as initial weights, target pretraining is carried out on three classification datasets respectively. A cross-dataset finetuning experiment is carried out by transferring the target pretrained weights across CIFAR-10, CIFAR-100 and SVHN, with results presented in Tab. 9. We make the following observations. First, transferring weights target pretrained on CIFAR-10/100 to finetuning on SVHN yields moderate improvement from without target pretraining (16.92%  $\rightarrow$  19.71%/20.06%), while they are still behind target pretrained on SVHN itself. Due to the large distribution gap between CIFAR-10/100 (natural object images) and SVHN (digit images), this result suggests target pretraining is essentially adapting weights to target data distribution. Moreover, transferring SVHN target pretrained weights to CIFAR10/100 produces much inferior results than even without target pretraining and transferring between CIFAR-10 and CIFAR-100 gives a relatively smaller drop from within dataset pretraining. Both of the above two observations indicate target pretraining must be carried out on the distribution of target dataset which again validates the adaptation hypothesis.

Table 9: Transferring target pretrained weights across datasets. Accuracy (%) is reported.

Pretrain	Trg. Pre.	Finetune		
		CIFAR-10	CIFAR-100	SVHN
IN Sup	-	49.69	30.85	16.92
IN Sup	CIFAR-10	65.13	29.51	19.71
IN Sup	CIFAR-100	49.46	34.63	20.06
IN Sup	SVHN	24.91	12.12	35.82

#### 4.4.2. Target Pretraining Improves Class Discovery

We further investigate how target pretraining helps adapt to better initial weights by evaluating unsupervised class discovery (clustering). As an alternative approach towards unsupervised feature learning, clustering has been employed to generate pseudo-labels for representation learning [50, 51]. Therefore, if better clustering results are identified it means the features are potentially more useful for finetuning. Specifically, we first denote the cluster indices as  $\mathbf{L}_c \in \{0, 1\}^{N \times K}$  and the ground-truth classification label as  $\mathbf{L} \in \{0, 1\}^{N \times K}$ , where  $N$  is the number of samples and  $K$  is the number of classes/clusters. For both  $\mathbf{L}_c$  and  $\mathbf{L}$ , there is only a single element with 1 at each row. Since clustering is invariant to the permutation of cluster index the assignment of cluster index to class label is unknown. To calculate clustering accuracy, we first find the following permutation of cluster indices,

$$\begin{aligned} \mathbf{P} &\in \{0, 1\}^{K \times K} \\ \text{s.t. } \forall i, j \in \{0, \dots, K\}, \quad &\sum \mathbf{p}_i = 1, \quad \sum \mathbf{p}_j = 1 \end{aligned} \quad (8)$$

The following problem is formulated to find the best permutation that maximizes the accuracy, where  $\circ$  is an elementwise product.

$$\begin{aligned} &\max_{\mathbf{P}} \frac{1}{N} \|\text{vec}(\mathbf{L}_c \mathbf{P} \circ \mathbf{L})\|_1 \\ &\rightarrow \max_{\mathbf{P}} \frac{1}{N} \text{vec}(\mathbf{L}_c \mathbf{P})^\top \text{vec}(\mathbf{L}) \\ &\rightarrow \max_{\mathbf{P}} \frac{1}{N} \text{tr}(\mathbf{P}^\top \mathbf{L}_c^\top \mathbf{L}) \end{aligned} \quad (9)$$

The above problem can be solved by Hungarian algorithm where the cost matrix is  $-\mathbf{L}_c^\top \mathbf{L}$ . We eventually report the performance as the classification described above. With results presented in Tab. 10, we observe consistently better clustering results for target pretrained features suggesting the effectiveness of target pretraining as good initial weights for representation learning.

#### 4.4.3. TSNE Visualization.

We further visualize feature representations after target pretraining on target dataset to draw qualitative insight into its effect. Specifically, we collect 500 randomly selected testing samples from CIFAR-10 with features as

Table 10: Evaluation of target pretraining by clustering accuracy (%). Results on training and testing data are delimited by /.

Pretrain	Trg. Pre.	Cifar-10	Cifar-100	SVHN
IN Sup	-	46.31/45.66	6.40/6.74	12.96/12.64
IN Sup	BYOL	47.17/48.02	7.91/7.44	15.96/16.23

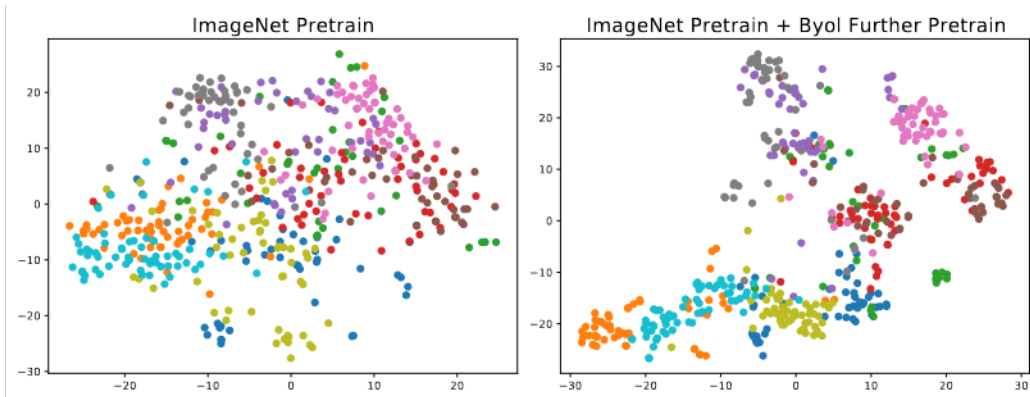
the output of last layer in backbone after global average pooling. We compared two set of features, the first is collected from ResNet18 with ImageNet pretrained weights and another is from ImageNet pretrained weights plus target pretraining. We concatenate both sets of features and project them into 2D via TSNE [52] for visualization. As shown in Fig. 6, we see a noticeable clustering structure after target pretraining on target data, demonstrating the effectiveness of target pretraining.

#### 4.4.4. Impact of Target Pretraining Steps.

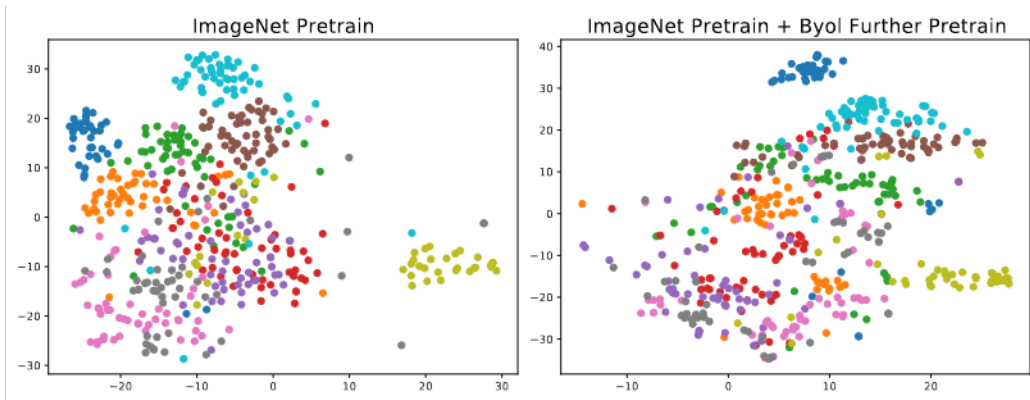
We study the impact of target pretraining w.r.t. the number of labeled data and steps of target pretraining. Specifically, we evaluate supervised finetuning performance on CIFAR-10 with 250, 1000, 4000 and all labeled data. The initial weights for finetuning is evaluated from 0 to 250 epochs. The results of testing accuracy are illustrated in Fig. 5. First, target pretraining is effective after at least 200 epochs of training, suggesting enough training iterations is required for a good adaptation to target dataset. Second, the improvement brought by target pretraining is consistently better at lower-labeling regime. This aligns with many previous investigations into the impact of pretraining [9].

#### 4.4.5. Comparing Convergence

As discussed, having good initialization allows faster finetuning with better results after convergence. In this section, we compare the convergence between with and without contrastive target pretraining on CIFAR-10 #40 labeled data, CIFAR-100 #400 labeled data and SVHN #40 labeled data. Specifically, we evaluate both supervised finetuning and semi-supervised finetuning with different initial weights and visualize the results in Fig. 7. In Fig. 7 (left), we first show the test accuracy curves on three classification datasets over 200 training epochs. Initial weights are randomly initialized before target pretraining. For all three datasets, it is very clear that with target pretraining (dashed line), the converged accuracy is substantially higher.



(a) TSNE visualization on CIFAR-10



(b) TSNE visualization on CIFAR-100

Figure 6: TSNE visualization of 500 randomly selected test samples from (a) CIFAR-10 and (b) CIFAR-100 before (left) / after (right) target pretraining.

In Fig. 7 (middle), with ImageNet supervised pretrained weights, target pretraining (dashed line) demonstrates superior performance with significant margin on CIFAR-10 and SVHN. Finally, in Fig. 7 (right), we compare the convergence of FixMatch semi-supervised finetuning with different initial weights. There is a very clear margin for CIFAR-10 and SVHN when target pretraining is included while the margin is smaller for CIFAR-100 probably because more labeled data (#400 labeled) is available in CIFAR-100 experiment. We also notice that all methods have converged within 200 epochs, thus making this comparison fair. All these observations again validate that good initial weights play an important role in SSL finetuning at low-label

regime and target pretraining is an effective way to produce better initial weights.

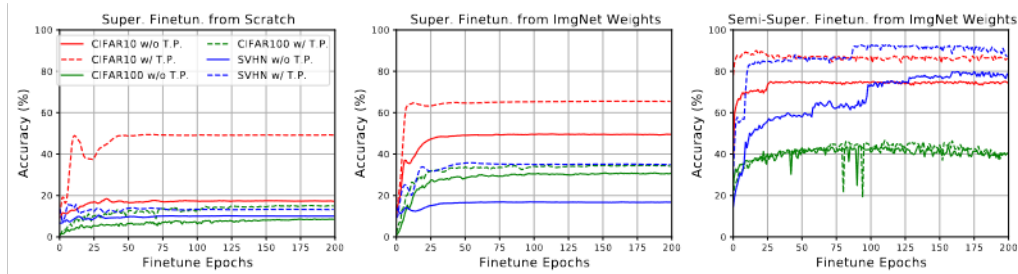


Figure 7: Comparing finetuning with different initial weights, “w/o T.P.” and “w/ T.P.” indicate with and without target pretraining respectively. Left and middle plots illustrate supervised finetuning from weights with target pretraining from random weights and ImageNet weights respectively. Right plot illustrates semi-supervised finetuning.

## 5. Conclusion

Motivated by the recent studies into combining transfer learning with semi-supervised learning, we first revealed that good initial weights are actually accountable for the substantial improvement on semi-supervised finetuning. We demonstrate that with the same initial weights, semi-supervised learning based on pseudo-labeling is still a better option. We further discover that due to the covariate shift between pretrain and target datasets, direct finetuning from weights pretrained on separate large dataset is not optimal. A contrastive target pretraining step is proposed to adapt the weights to target dataset. We demonstrate on multiple classification and segmentation datasets that semi-supervised finetuning can benefit substantially from the adapted weights at low-label regime. This study encourages people to rethink the relation between transfer learning and semi-supervised learning and contrastive learning could be a good way to synergistically combine both. **Acknowledgement:** This research is supported by the Agency for Science, Technology and Research (A\*STAR) under its AME Programmatic Funds (Grant No. A20H6b0151).

## References

- [1] L. Samuli, A. Timo, Temporal ensembling for semi-supervised learning, in: International Conference on Learning Representations, 2017.

- [2] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: *Advances in Neural Information Processing Systems*, 2017.
- [3] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, C. Raffel, Fixmatch: Simplifying semi-supervised learning with consistency and confidence, in: *Advances in Neural Information Processing Systems*, 2020.
- [4] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: *International Conference on Machine Learning*, 2020.
- [5] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, tech. rep. (2009).
- [6] X. Wang, J. Gao, M. Long, J. Wang, Self-tuning for data-efficient deep learning, in: *International Conference on Machine Learning*, 2021.
- [7] G. French, T. Aila, S. Laine, M. Mackiewicz, G. Finlayson, Semi-supervised semantic segmentation needs strong, high-dimensional perturbations, in: *British Machine Vision Conference*, 2019.
- [8] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, K. McGuinness, Pseudo-labeling and confirmation bias in deep semi-supervised learning, in: *International Joint Conference on Neural Networks*, 2020.
- [9] K. He, R. Girshick, P. Dollár, Rethinking imagenet pre-training, in: *IEEE/CVF International Conference on Computer Vision*, 2019.
- [10] J.-B. Grill, F. Strub, F. Alché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al., Bootstrap your own latent: A new approach to self-supervised learning, in: *Advances in Neural Information Processing Systems*, 2020.
- [11] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [12] O. Henaff, Data-efficient image recognition with contrastive predictive coding, in: *International Conference on Machine Learning*, 2020.

- [13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, *International Journal of Computer Vision* (2015).
- [14] L. Shao, F. Zhu, X. Li, Transfer learning for visual categorization: A survey, *IEEE Transactions on Neural Networks and Learning Systems* (2014).
- [15] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: *European Conference on Computer Vision*, 2018.
- [16] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: *Advances in Neural Information Processing Systems*, 2015.
- [17] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, G. Hinton, Big self-supervised models are strong semi-supervised learners, in: *Advances in Neural Information Processing Systems*, 2020.
- [18] K. You, Z. Kou, M. Long, J. Wang, Co-tuning for transfer learning, 2020.
- [19] T. Miyato, S.-i. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).
- [20] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, Q. V. Le, Unsupervised data augmentation for consistency training, in: *Advances in Neural Information Processing Systems*, 2020.
- [21] M. Wang, W. Deng, Deep visual domain adaptation: A survey, *Neurocomputing* (2018).
- [22] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C. Raffel, Mixmatch: A holistic approach to semi-supervised learning, in: *Advances in Neural Information Processing Systems*, 2019.

- [23] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, C. Raffel, Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring, in: International Conference on Learning Representations, 2020.
- [24] S. Shi, F. Nie, R. Wang, X. Li, Semi-supervised learning based on intra-view heterogeneity and inter-view compatibility for image classification, *Neurocomputing* (2022).
- [25] J. Chen, M. Yang, J. Ling, Attention-based label consistency for semi-supervised deep learning based image classification, *Neurocomputing* (2021).
- [26] Y. Chong, Y. Ding, Q. Yan, S. Pan, Graph-based semi-supervised learning: A review, *Neurocomputing* (2020).
- [27] V. Olsson, W. Tranheden, J. Pinto, L. Svensson, Classmix: Segmentation-based data augmentation for semi-supervised learning, in: *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.
- [28] I. Alonso, A. Sabater, D. Ferstl, L. Montesano, A. C. Murillo, Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank, in: *IEEE/CVF International Conference on Computer Vision*, 2021.
- [29] J.-C. Su, Z. Cheng, S. Maji, A realistic evaluation of semi-supervised learning for fine-grained classification, in: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
- [30] S. J. Pan, Q. Yang, A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering* (2009).
- [31] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: *International Conference on Machine Learning*, 2015.
- [32] J. N. Kundu, N. Venkat, R. V. Babu, et al., Universal source-free domain adaptation, in: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

- [33] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, M. Hardt, Test-time training with self-supervision for generalization under distribution shifts, in: International Conference on Machine Learning, 2020.
- [34] Y. Liu, P. Kothari, B. van Delft, B. Bellot-Gurlet, T. Mordan, A. Alahi, Ttt++: When does self-supervised test-time training fail or thrive?, in: Advances in Neural Information Processing Systems, 2021.
- [35] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, T. Darrell, Tent: Fully test-time adaptation by entropy minimization, in: International Conference on Learning Representations, 2021.
- [36] S. Li, D. Chen, Y. Chen, L. Yuan, L. Zhang, Q. Chu, B. Liu, N. Yu, Unsupervised finetuning, arXiv preprint arXiv:2110.09510 (2021).
- [37] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, H. Hu, Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
- [38] X. Wang, R. Zhang, C. Shen, T. Kong, L. Li, Dense contrastive learning for self-supervised visual pre-training, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
- [39] S. Li, D. Chen, Y. Chen, L. Yuan, L. Zhang, Q. Chu, B. Liu, N. Yu, Improve unsupervised pretraining for few-label transfer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021.
- [40] M. Jaderberg, K. Simonyan, A. Zisserman, et al., Spatial transformer networks, Advances in Neural Information Processing Systems (2015).
- [41] A. v. d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748 (2018).
- [42] D. Erhan, A. Courville, Y. Bengio, P. Vincent, Why does unsupervised pre-training help deep learning?, in: International Conference on Artificial Intelligence and Statistics, 2010.
- [43] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.

- [44] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [45] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- [46] C. Xue, Q. Dou, X. Shi, H. Chen, P.-A. Heng, Robust learning at noisy labeled medical images: Applied to skin lesion classification, in: IEEE International Symposium on Biomedical Imaging, 2019.
- [47] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- [48] N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, et al., Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic), in: IEEE International Symposium on Biomedical Imaging, 2018.
- [49] X. Chen, H. Fan, R. Girshick, K. He, Improved baselines with momentum contrastive learning, arXiv preprint arXiv:2003.04297 (2020).
- [50] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: European Conference on Computer Vision, 2018.
- [51] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, in: Advances in Neural Information Processing Systems, 2020.
- [52] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of Machine Learning Research (2008).