Improving Fine-tuning of Self-supervised Models with Contrastive Initialization

Haolin Pan[a,*], Yong Guo[a,*], Qinyi Deng[a,*], Haomin Yang[a], Yiqun Chen[b], Jian Chen[a,**]

[a]*South China University of Technology, China*
[b]*Guangdong University of Education, China*

Abstract

Self-supervised learning (SSL) has achieved remarkable performance in pre-training the models that can be further used in downstream tasks via fine-tuning. However, these self-supervised models may not capture meaningful semantic information since the images belonging to the same class are always regarded as negative pairs in the contrastive loss. Consequently, the images of the same class are often located far away from each other in learned feature space, which would inevitably hamper the fine-tuning process. To address this issue, we seek to provide a better initialization for the self-supervised models by enhancing the semantic information. To this end, we propose a Contrastive Initialization (COIN) method that breaks the standard fine-tuning pipeline by introducing an extra initialization stage before fine-tuning. Extensive experiments show that, with the enriched semantics, our COIN significantly outperforms existing methods *without introducing extra training cost* and sets new state-of-the-arts on multiple downstream tasks.

Keywords: self-supervised model, model fine-tuning, model initialization, semantic information, supervised contrastive loss

---

[*]These authors contributed equally to this work.
[**]Corresponding author
Email addresses: `sephl@mail.scut.edu.cn` (Haolin Pan), `guoyongcs@gmail.com` (Yong Guo), `sedengqy@mail.scut.edu.cn` (Qinyi Deng), `sehaomin@mail.scut.edu.cn` (Haomin Yang), `chenyiqun@gdei.edu.cn` (Yiqun Chen), `ellachen@scut.edu.cn` (Jian Chen)

1. Introduction

Recently, self-supervised learning (SSL) has achieved great success in the pre-training of deep models based on a large-scale unlabelled dataset [1, 2, 3, 4, 5]. Specifically, one can train the models by maximizing the feature similarity between two augmented views of the same instance while minimizing the similarity between two distinct instances. In practice, these self-supervised models have shown remarkable generalization ability across diverse downstream tasks when fine-tuning their parameters on the target datasets [6, 7, 8, 9, 10]. To be specific, existing methods often exploit the cross-entropy loss, optionally combined with a contrastive loss [11, 12, 13, 14], to fine-tune the pre-trained models.

Nevertheless, the fine-tuning performance is still very limited since the pre-trained model does not necessarily provide a strong/meaningful semantic relation among instances, which, however, is essential for learning a good classifier [15, 16, 17, 18]. As shown in Figure 1 (A), even on the pre-training dataset, e.g., ImageNet, SSL may learn a feature space where the instances belonging to the same class are far from each other [19, 20]. The main reason is that SSL only takes different augmented views of the same image as the positive pairs and simply treats all the other images as the negative ones. As a result, the images belonging to the same category are not necessarily located close to each other in the learned feature space, i.e., with very weak semantic relation. More critically, this phenomenon would be much more severe when we consider a target dataset that has a different distribution from the pre-training dataset. In practice, such a weak semantic relation among instances would inevitably hamper the fine-tuning process from learning a good classifier on the downstream tasks.

To address this issue, when fine-tuning, we seek to provide a better initialization for the self-supervised models by enhancing the semantic relation among instances. Intuitively, if we can encourage the model to capture better semantic information on the target dataset, it would be easier to learn a promising classifier during fine-tuning. To be specific, as shown by the example in Figure 1 (B), based on the meaningful semantic relation where the instances belonging to the

2

Figure 1: The feature spaces obtained by self-supervised models. Each mark represents an instance in the feature space, and the shape of the mark indicates the class of instances. Since the similarities of all positive pairs are large enough, and those of all negative pairs are small enough (*i.e.*, uniformly distributed in the feature space), the values of contrastive loss for both (A) and (B) are small enough to be considered a good feature space. But obviously, the semantic relation among instances in (B) is more meaningful than in (A) due to the separable features. The figure indicates that the goal of self-supervised methods has little positive effect on enriching the semantic information on the downstream tasks. Therefore, it is difficult for the self-supervised models to provide a good initialization for subsequent fine-tuning process.

same category are close to each other, we can simply use a linear classifier to discriminate the instances of different categories in the feature space.

Inspired by this, we propose to break the pipeline of fine-tuning self-supervised models by introducing an extra class-aware initialization stage before fine-tuning the classifier. In this paper, we develop a Contrastive Initialization (COIN) method that exploits a supervised contrastive loss to enrich the semantic information on the target dataset, by pulling together the instances of the same class and pushing away those from different classes. In this way, we are able to obtain easily separable features with better semantic information and then help learn a good classifier during fine-tuning. To verify this, we visualize the feature space learned by different methods in Figure 2. Compared with two popular fine-tuning methods, *i.e.*, CE-Tuning [5] and SCL [21], our COIN effectively pushes away the instances of different classes and yields a narrow ellipse area for each class. In fact, such narrow ellipse areas often come with better discriminative power, which

3

Figure 2: Visualization of the feature space on CIFAR-10, with each color representing a specific class. We compare the initial state and two popular fine-tuning methods, i.e., CE-Tuning with a cross-entropy (CE) loss and SCL [21], which combines CE loss with a supervised contrastive loss. Clearly, our COIN effectively pushes away the instances of different classes and yields narrow ellipse areas with fewer overlapping (*i.e.*, lower S_Dbw score), coming with better discriminative power. As a result, COIN greatly improves the fine-tuning performance.

can be evaluated by the S_Dbw score (*lower is better*) with the consideration of both the inter-class discrepancy and the intra-class compactness [22]. As shown in Figure 2, our COIN greatly reduces the S_Dbw score from 0.48 to 0.28 and yields a large accuracy improvement of $>0.5\%$ on CIFAR-10. More critically, we highlight that our initialization stage does not increase the total training cost, since we reduce the number of iterations for the following fine-tuning process to keep the total training iterations unchanged.

Our main contributions can be summarized as follows:

- We break the standard fine-tuning pipeline for self-supervised models by introducing an additional initialization stage before fine-tuning. To be specific, we first encourage the model to capture better semantic information on the target dataset. Then, we take the model with the enriched semantic information as a better initialization for the subsequent fine-tuning process.

- To enrich the semantic information, we propose a Contrastive Initialization (COIN) method that exploits a supervised contrastive loss to perform class-aware clustering. Specifically, we pull together the instances of the same class and push away those instances from different classes. In this way, we can obtain easily separable features with better semantic information on the target dataset, which significantly boost the fine-tuning performance.

4

- To avoid introducing extra training cost, we reduce the number of training iterations for the final fine-tuning process to keep the total training iterations unchanged. Extensive experiments show that the proposed method, COIN, significantly improves the fine-tuning process of self-supervised models and yields new state-of-the-arts on various benchmark datasets.

2. Related Work

2.1. Contrastive Learning

In the past few years, Many works applying contrastive learning to pre-train self-supervised models have attracted attention due to their impressive performances [23, 6, 24, 5, 25, 10]. The self-supervised contrastive learning models learn an instance-distinct-based feature representation to achieve state-of-the-art performance on the ImageNet [26] classification task. However, the superior performance of a self-supervised model in one scenario does not necessarily reflect its performance in others. Because the learned feature spaces closely match the distribution of ImageNet, which easily overfit some similar downstream tasks but hamper others [27, 28, 29]. Therefore, in this paper, we focus on better fine-tuning the self-supervised models on various downstream tasks.

2.2. Model Fine-tuning

In deep learning, fine-tuning a pre-trained model on a target dataset has become a standard training paradigm in various applications. Fine-tuning is one of the main pipelines for improving the transferability of self-supervised models. To be precise, fine-tuning can be regarded as a model transfer method, but it usually does not need to know the data distribution of the source domain. Most fine-tuning methods are designed for supervised pre-trained models instead of self-supervised models [30, 31]. Besides fine-tuning parameters, one can also fine-tune architectures to improve performance [32, 33]. In recent studies, how to achieve a better fine-tuning performance of self-supervised models has attracted more attention [34, 35], in which contrastive learning plays an important role [21, 13,

14]. Recently, Noisy-Tune [36] proposed to improve the fine-tuning performance by introducing noises into the parameters. Unlike the above methods, we seek to boost the fine-tuning process from a new perspective, i.e., providing a better initialization with strong semantic information.

2.3. Domain Adaption

The Purpose of domain adaption is to transfer a trained model from the source domain to the target domain across the distribution shift. Many domain adaptation studies focus on minimizing the discrepancy between the data distributions of the two domains [37, 10, 38, 39, 40, 41]. In addition, some studies focus on improving model transferability using source domain data so that the model can be more easily adapted to the target domain [42, 43, 44]. Some of them show that self-supervised models often benefit from semantic information and feature uniformity [19, 20, 45, 46]. Besides model adaptation, improving the generalization ability or out-of-distribution robustness [47, 48] can also handle the data from another domain to some extent. Unlike the above methods that train models on the data of the source domain, we focus on obtaining a better initialization on the target domain/dataset.

3. Fine-tuning with Contrastive Initialization

3.1. Motivation and Method Overview

Since self-supervised learning (SSL) takes different augmented views of an instance as positive pairs and simply treats all other instances as negative pairs, self-supervised models are often hard to capture meaningful semantic information. As shown in Figure 1 (A), in the feature space learned by SSL, the instances belonging to the same class may be far away from each other, i.e., with weak semantic relation among instances. In practice, such weak semantic information often hampers the fine-tuning process from learning a good classifier and results in suboptimal fine-tuning performance.

To address this issue, we seek to provide a better initialization for fine-tuning by enhancing the semantic relation among instances. Intuitively, as shown in

6

Figure 3: The overall fine-tuning pipeline with COIN. We introduce an additional Contrastive Initialization (COIN) stage that exploits a supervised contrastive loss to enrich the semantic information on the target dataset. Then, we can perform any existing fine-tuning method, *e.g.*, SCL [21] with an additional contrastive loss, to learn the classifier. Besides, we introduce a stage split ratio α to allocate different computational budgets for each stage, while keeping the training epochs N unchanged.

Figure 1 (B), if we capture better semantic information, one can obtain an easily separable feature space that makes it easier to learn a good classifier during fine-tuning. Inspired by this, we break the standard fine-tuning pipeline of self-supervised models by introducing an extra class-aware initialization stage, resulting in a new fine-tuning pipeline that contains two stages. As shown in Figure 3 and Algorithm 1, we first adopt a Contrastive Initialization (COIN) stage that exploits a supervised contrastive loss to enrich the semantic information. Then, we fine-tune the model that contains richer semantic information with any existing fine-tuning approaches. Following [21], besides the cross-entropy loss, we additionally incorporate a contrastive loss since it often benefits the fine-tuning process. For fair comparisons with existing fine-tuning methods, we consider COIN as a part of the overall fine-tuning pipeline and keep the total training iterations unchanged. Given a budget of N training epochs in total, we introduce a hyperparameter, the stage split ratio α, to allocate αN epochs for COIN and $(1 - \alpha)N$ epochs for the subsequent fine-tuning process.

7

Algorithm 1 Fine-tuning with Contrastive Initialization (COIN).

Require: Pre-trained model f, projector g, classifier h, step size η, model parameters w, stage split ratio α, epochs N, weight of contrastive loss λ.

1: Initialize the classifier h and projector g;

2: *// Contrastive Initialization*

3: **for** $i = 1$ to αN **do**

4:    Sample a batch of training data x and extract features $z = f(x)$;

5:    Compute projected features $g(z)$ to obtain $\mathcal{L}_{\text{con}}^{\text{sup}}$;

6:    Update $\{w_f, w_g\}$ by descending the gradient:

   Update model f: $w_f \leftarrow w_f - \eta \nabla_{w_f} \mathcal{L}_{\text{con}}^{\text{sup}}$;

   Update projector g: $w_g \leftarrow w_g - \eta \nabla_{w_g} \mathcal{L}_{\text{con}}^{\text{sup}}$;

7: **end for**

8: *// Any Existing Fine-tuning Method, e.g., SCL [21]*

9: **for** $i = \alpha N + 1$ to N **do**

10:    Sample a batch of training data x and extract features $z = f(x)$;

11:    Compute the projected features $g(z)$ to obtain $\mathcal{L}_{\text{con}}^{\text{sup}}$;

12:    Compute logits $h(z)$ to obtain $\mathcal{L}_{\text{ce}}$;

13:    Update $\{w_f, w_g, w_h\}$ by descending the gradient:

   Update model f: $w_f \leftarrow w_f - \eta \left(\nabla_{w_f} \mathcal{L}_{\text{ce}} + \lambda \nabla_{w_f} \mathcal{L}_{\text{con}}^{\text{sup}} \right)$;

   Update projector g: $w_g \leftarrow w_g - \eta \left(\nabla_{w_g} \mathcal{L}_{\text{ce}} + \lambda \nabla_{w_g} \mathcal{L}_{\text{con}}^{\text{sup}} \right)$;

   Update classifier h: $w_h \leftarrow w_h - \eta \left(\nabla_{w_h} \mathcal{L}_{\text{ce}} + \lambda \nabla_{w_h} \mathcal{L}_{\text{con}}^{\text{sup}} \right)$;

14: **end for**

3.2. Contrastive Initialization (COIN)

To provide a better initialization for the subsequent fine-tuning process, we propose a Contrastive Initialization (COIN) stage that exploits a supervised contrastive loss to enrich the semantic information. Specifically, on the target dataset, we seek to pull together the instances of the same classes and push away those from different classes. With the help of resultant meaningful semantic relation among instances, our COIN often obtains easily separable features with strong discriminative power. Based on this, we directly take the enriched model

as a better initialization to boost the subsequent fine-tuning process.

Let x be the input images. We extract the features of x through a self-supervised pre-trained model f followed by a projector g via $v = g(f(x))$. To enhance the semantic information on the target dataset, for the i-th sample x_i, we take instances of the same class as the positive pairs and those from different classes as the negative pairs. The training loss of COIN becomes:

$$\mathcal{L}_{\text{con}}^{\text{sup}} = -\frac{1}{n} \sum_{i=1}^{n} \frac{1}{|P_i|} \sum_{v_j \in P_i} \log \frac{e^{\left(v_i^T v_j / \tau\right)}}{\sum_{v_k \in A_i} e^{\left(v_i^T v_k / \tau\right)}}, \tag{1}$$

where n denotes the number of instances in a mini-batch, v_i denotes the features of the i-th instance. P_i and A_i denote the set of positive pairs and all possible pairs w.r.t. x_i, respectively. To construct data pairs regarding v_i for contrastive learning, we build $A_i = \{v_k | k \leq n, k \neq i\}$ which includes the features of the other instances in this batch excluding v_i itself. τ is a temperature coefficient which is important for contrastive loss.

Combining COIN with the subsequent fine-tuning process. As an initialization method, our COIN can be easily combined with any existing fine-tuning method. In this paper, we combine COIN with a popular fine-tuning method, SCL [21], which jointly optimizes a CE loss and a supervised contrastive loss. The training loss of SCL [21] can be formulated as $\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda \mathcal{L}_{\text{con}}^{\text{sup}}$. When fine-tuning, we simply introduce a classifier h to compute the CE loss and set $\lambda = 0.1$, thus constructing SCL [21].

Advantages over existing methods. Compared to the existing advanced methods, our COIN significantly boosts the subsequent fine-tuning performance on various benchmark datasets. In fact, with the help of the introduced class-aware initialization stage, COIN greatly enriches the semantic information, which comes with stronger discriminative power on downstream tasks and then provides a better initialization of self-supervised models for the subsequent fine-tuning. Furthermore, COIN avoids introducing extra training cost to the standard fine-tuning pipeline. Specifically, COIN keeps the total computational budgets unchanged and allocates the budgets to the initialization stage and the fine-tuning stage by a simple hyperparameter α.

9

4. Experiments

In this section, we evaluate the effectiveness of COIN on various benchmark datasets, and compare it with other advanced methods. We will elaborate on the settings and show the experimental results in the following.

4.1. Datasets and Metrics

We test on nine datasets including ImageNet-20 [13], CIFAR-10 [49], CIFAR-100 [49], Caltech-101 [50], Stanford Cars [51], FGVC Aircraft [52], Oxford 102 Flowers [53], Oxford-IIIT Pets [54], DTD [55], covering common image classification tasks including coarse-grained object classification, fine-grained image classification and texture classification. We do not directly test on ImageNet but on ImageNet-20 because the model we used is pre-trained on ImageNet. ImageNet-20 is a subset of ImageNet with 20 classes, including ImageNette and ImageWoof [56]. Oxford 102 Flowers is obtained from Kaggle. Caltech-101 is obtained from TensorFlow. Others are downloaded from their official websites.

We report the metrics including top-1 accuracy, S_Dbw score, and training time cost. [19] uses the aggregation degree of similar instances to evaluate the semantic relation among instances, while COIN considers learning higher intra-class compactness and larger inter-class discrepancy by pulling together the instances of the same class and pushing those apart from different classes. In this way, COIN can effectively enrich the semantic information on the target dataset. To quantify the quality of the semantic information, we use S_Dbw score [22] to measure the intra-class discrepancy by average scattering (denoted by $Scat$), and measure the inter-class compactness by inter-class density (denoted by $Dens\_bw$). Formally, it can be computed by: $S\_Dbw = Scat + Dens\_bw$. A lower S_Dbw score means richer semantic information, which comes with stronger discriminative power and boosts the fine-tuning performance.

4.2. Experimental Settings

Our experiments follow the nearly same settings as Core-Tuning [13] to fine-tune self-supervised models. We use ResNet-50 pre-trained by MoCo-v2 [24]

Figure 4: Comparisons of the curves of training accuracy among different fine-tuning methods on CIFAR-10. COIN only uses a supervised contrastive loss during the early 70 epochs, thus showing a lower accuracy, but effectively enriching the semantic information (S_Dbw score decreases from 0.88 to 0.36) in this period.

on ImageNet with 800 training epochs as the pre-trained model, which can be downloaded from the moco GitHub repository. We implement COIN in PyTorch[1]. As shown in Algorithm 1, we first update the parameters of pre-trained model f and projector g in our COIN. In the subsequent fine-tuning process, besides f and g, we also update the parameters of classifier h. For a fair comparison, all methods use consistent data preprocessing schemes on the same dataset.

The hyperparameters of our experiments are nearly the same as Core-Tuning [13] except for the newly introduced stage split ratio α. Specifically, we set the training epochs $N = 100$, the temperature coefficient $\tau = 0.3$, and the weight of supervised contrastive loss $\lambda = 0.1$. Besides, we set $\alpha = 0.7$ on CIFAR-10. Note that the value of α is specific to the dataset due to the different convergence rates when fine-tuning models on different datasets. Under the premise of ensuring the self-supervised models in all experiments can converge during the fine-tuning, we slightly adjust the value of α for each dataset to allocate as many computational budgets as possible to the initialization stage. We finally report the value of α at which the model finally achieves the best accuracy.

---

[1]The code of the proposed COIN is available at https://github.com/PorientHaolin/COIN.

Table 1: Comparisons of accuracy on different classification downstream tasks among various fine-tuning methods with the self-supervised ResNet-50 model pre-trained by MoCo-v2. COIN significantly boosts the fine-tuning performance and sets new state-of-the-arts on multiple datasets. * denotes our reproduced results. The last row shows the accuracy improvement of our COIN over SCL [21].

Method	ImageNet-20	CIFAR-10	CIFAR-100	DTD	Cars	Pets	Flowers	Caltech-101	Aircraft	Avg.
CE-Tuning	88.27	97.26	82.22	69.04	89.91	91.01	99.14	92.32	88.30	88.61
SL-CE-Tuning [13]	91.01	94.23	83.40	74.40	89.77	92.17	98.78	93.39	87.03	89.35
L2SP [57]	88.49	95.14	81.43	72.18	89.00	89.43	98.66	91.98	86.55	88.10
M&M [58]	88.53	95.02	80.58	72.43	88.90	89.60	98.57	92.91	87.45	88.22
DELTA [59]	88.35	94.76	80.39	72.23	88.73	89.54	98.65	92.19	87.05	87.99
BSS [60]	88.34	94.84	80.40	72.22	88.50	89.50	98.57	91.95	87.18	87.94
RIFLE [34]	89.06	94.71	80.36	72.45	89.72	90.05	98.70	91.94	87.60	88.29
Bi-Tuning [14]	89.06	95.12	81.42	73.53	89.41	89.90	98.57	92.83	87.39	88.58
SCL [21]	89.29	95.33	81.49	72.73	89.37	89.71	98.65	92.84	87.44	88.54
SCL* [21]	89.60	97.31	82.82	73.51	**90.32**	90.91	98.90	92.74	88.33	89.38
Core-Tuning [13]	92.73	97.31	84.13	75.37	90.17	92.36	99.18	**93.46**	89.48	90.47
Core-Tuning* [13]	93.84	97.40	83.25	74.04	90.16	92.04	98.78	92.62	88.86	90.11
Noisy-Tune [36]	94.39	97.59	83.50	71.97	90.16	91.99	99.14	70.58	88.83	87.57
COIN (ours)	**94.60**	**97.88**	**85.39**	**75.74**	**90.32**	**93.59**	**99.27**	93.00	**89.77**	**91.06**
	(+5.00)	(+0.57)	(+2.57)	(+2.23)	(+0.00)	(+2.68)	(+0.37)	(+0.26)	(+1.44)	(+1.68)

4.3. Comparison with State-of-the-arts

In this section, we compare COIN with various advanced fine-tuning methods including CE-Tuning, SL-CE-Tuning [13], L2SP [57], M&M [58], DELTA [59], BSS [60], RIFLE [34], Bi-Tuning [14], Noisy-Tune [36], SCL [21], and Core-Tuning [13]. Among them, Noisy-Tune is an initialization method that perturbs parameters before fine-tuning. In like manner of COIN, we also combine Noisy-Tune with SCL [21] to achieve a fair comparison with our COIN.

As shown in Figure 4, we simply compare COIN with several representative fine-tuning methods on CIFAR-10. The plotted training curves show how the class-aware initialization stage beneficially affects the subsequent fine-tuning process. We do not consider CE loss, but only use a supervised contrastive loss during the early 70 epochs. Therefore, the accuracy of COIN is lower than other methods in this period. However, COIN significantly enriches the semantic information on the target dataset (*i.e.*, decreasing the S_Dbw score from 0.88 to 0.36) before fine-tuning. With the help of enriched semantic information, it is easy to learn a more accurate classifier in the following fine-tuning process.

Table 2: Comparisons of S_Dbw score and Time Cost of different fine-tuning methods on multiple classification downstream tasks. S_Dbw score evaluates the quality of the semantic information captured by models on the target datasets. A lower S_Dbw score indicates richer semantic information. Time Cost represents the training time for 100 epochs. These results suggest that COIN can always obtain the richest semantic information among these methods without introducing additional training cost.

Method	ImageNet-20		CIFAR-10		CIFAR-100		Caltech-101		DTD	
	S_Dbw	Time Cost (h)	S_Dbw	Time Cost (h)	S_Dbw	Time Cost (h)	S_Dbw	Time Cost (h)	S_Dbw	Time Cost (h)
CE-Tuning	0.75	2.23	0.48	**6.75**	0.73	**7.26**	0.32	**0.54**	0.70	0.55
SCL [21]	0.68	2.19	0.47	7.31	0.65	7.80	0.33	0.55	0.61	**0.51**
Core-Tuning [13]	0.55	3.72	0.38	14.93	0.64	15.64	0.32	0.61	0.62	0.76
COIN (ours)	**0.47**	**2.10**	**0.28**	7.79	**0.46**	7.79	**0.31**	0.54	**0.56**	0.53

Method	Aircraft		Cars		Pets		Flowers		Avg.	
	S_Dbw	Time Cost (h)	S_Dbw	Time Cost (h)	S_Dbw	Time Cost (h)	S_Dbw	Time Cost (h)	S_Dbw	Time Cost (h)
CE-Tuning	0.43	1.54	0.49	**0.99**	0.50	0.40	0.35	0.89	0.53	**2.35**
SCL [21]	0.47	1.55	0.49	1.15	0.50	0.45	0.33	**0.84**	0.50	2.48
Core-Tuning [13]	0.45	1.76	0.51	2.05	0.50	0.81	0.34	1.58	0.48	4.65
COIN (ours)	**0.40**	**1.39**	**0.45**	1.15	**0.34**	**0.38**	**0.30**	0.88	**0.40**	2.50

We report more results on the accuracy of all considered methods on multiple benchmark datasets in Table 1. Under the same settings, COIN outperforms other methods on nearly all considered datasets except Caltech-101 and Cars, but shows comparable performance compared with the best method on these two datasets. To better illustrate this, we focus on comparing COIN with several representative methods. Among them, CE-Tuning is a standard fine-tuning method, while SCL [21] combines CE loss with supervised contrastive loss, which often benefits the fine-tuning process. Core-Tuning and Noisy-Tune are one of the most advanced fine-tuning methods and initialization methods, respectively. For the results in Table 1 and Table 2, COIN significantly improves SCL [21] by simply introducing an extra class-aware initialization stage, and always comes with the most enriched semantic information (lowest S_Dbw score). Unlike Core-Tuning, COIN always yields state-of-the-art performance without introducing large training cost. Besides, COIN will not increase the difficulties in fine-tuning like Noisy-Tune which shows limited fine-tuning performance on DTD and Caltech-101 datasets. Extensive results strongly demonstrate the effectiveness of COIN for fine-tuning self-supervised models.

Table 3: Comparisons of the accuracy of COIN with the different temperature coefficient τ on CIFAR-10. For these considered candidate values of τ, as the τ changes, the resultant accuracy does not change with apparent regularity.

τ	0.07	0.1	0.3	0.5	0.7	Avg.
Acc. (%)	97.60	97.55	**97.85**	97.80	96.69	97.70

Table 4: Comparisons of the accuracy of COIN with the different values of stage split ratio α on CIFAR-10. For a specific dataset, a moderate value of α (*e.g.*, $\alpha = 0.7$ on CIFAR-10) brings a significant improvement in resultant performance. A too-large value causes the fine-tuning process to lack sufficient time to make the model fully converge and consequently leads to limited fine-tuning performance.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg.
Acc. (%)	97.38	97.54	97.47	97.57	97.73	97.85	**97.88**	97.69	96.39	97.50

5. More Discussions

5.1. Imapct of Hyperparameters

In this section, we explore the impact of several important hyperparameters on the resultant fine-tuning performance of COIN.

Temperature coefficient τ. τ is an important hyperparameter of contrastive loss. To determine the value of τ , we consider a candidate set $\tau \in \{0.07, 0.1, 0.3, 0.5, 0.7\}$ like a previous work [19]. Table 3 shows that $\tau = 0.3$ works best under the setting of $\alpha = 0.6$. We empirically set $\tau = 0.3$ in other experiments because no apparent regularity is shown in these results.

Stage split ratio α. To fairly compare with advanced fine-tuning methods, we use α to allocate different training budgets to the initialization stage and the fine-tuning stage, keeping the number of total training iterations unchanged. Specifically, if the entire fine-tuning pipeline lasts N epochs, the initialization stage will last the first αN epochs. As shown in Table 4, COIN achieves the best fine-tuning performance on CIFAR-10 when $\alpha = 0.7$. When we set a large value of α, fine-tuning stage lacks sufficient time to fully converge the self-supervised models, resulting in a limited fine-tuning performance. Since the model requires

14

Table 5: Comparisons of the accuracy of several representative fine-tuning methods with different total training epochs on ImageNet-20. Since the models have almost converged within 100 training epochs, the best accuracy of each method does not improve significantly after increasing the training epochs.

Epochs	CE-Tuning	SCL [21]	Core-Tuning [13]	COIN (ours)
100 epochs	88.27	89.60	93.84	**94.60**
150 epochs	88.30	89.60	93.93	**94.80**

Table 6: Comparisons of the accuracy of COIN with different pre-trained models and fine-tuning settings on several benchmark datasets. COIN achieves higher accuracy when implemented on ViT pre-trained by MAE [61].

Method	ImageNet-20	CIFAR-10	CIFAR-100	DTD	Cars	Pets	Flowers	Avg.
MoCo-v2-based COIN [24]	94.60	97.88	85.39	75.74	90.32	93.59	99.27	90.97
MAE-based COIN [61]	**95.90**	**98.27**	**85.62**	**76.12**	**90.80**	**94.61**	**99.63**	**91.56**

different training iterations to be fully converged when fine-tuning on different datasets, we empirically choose a suitable value of α for each dataset.

Training epochs N**.** We follow [13] to fine-tune the self-supervised models with 100 training epochs. In terms of the results shown in Table 5, after increasing the epochs to 150, all models do not significantly improve because they almost converge within 100 epochs. COIN still maintains the performance gap over the considered fine-tuning methods.

5.2. Applying COIN to Vision Transformer

The previous experimental results fully demonstrate the effectiveness of our COIN on ResNet-50. In this section, we simply apply our COIN to a more advanced architecture and then fine-tune on those datasets whose input size is 224×224, using different fine-tuning settings. Specifically, we use Vision Transformer (ViT) pre-trained by AutoEncoder (MAE) [61] as the pre-trained model , and follow the fine-tuning settings obtained from the Github repository of MAE which is implemented on PyTorch. As shown in Table 6, with a stronger pre-trained model, our COIN obtains better results and sets new state-of-the-arts across different datasets.

6. Conclusion

This paper studies how to better fine-tune self-supervised models on various downstream tasks. We observe that the contrastive loss of SSL treats the instances belonging to the same class strictly as negative pairs, resulting in the limited capability of the self-supervised models to capture meaningful semantic information. We argue that the resultant weak semantic relation among instances indeed hampers the subsequent fine-tuning process. In response to this challenge, we thus propose a Contrastive Initialization (COIN) method to provide a better initialization of self-supervised models by simply introducing an extra class-aware stage before fine-tuning. As a result, COIN breaks the standard fine-tuning pipeline, leading to a new one. Specifically, COIN exploits a supervised contrastive loss to enrich the semantic information, which comes with stronger discriminative power and significantly improves the fine-tuning performance. We compare COIN with the advanced fine-tuning methods and initialization methods according to multiple evaluation metrics. The promising experimental results on various benchmark datasets show that COIN significantly benefits the subsequent fine-tuning process and consequently yields state-of-the-art performance without introducing additional training cost.

References

[1] H. Bao, L. Dong, S. Piao, F. Wei, Beit: Bert pre-training of image transformers, in: ICLR, 2022.

[2] V. Biscione, J. S. Bowers, Learning online visual invariances for novel objects via supervised and self-supervised training, Neural Networks 150 (2022) 222–236. `doi:https://doi.org/10.1016/j.neunet.2022.02.017`.

[3] X. Chen, K. He, Exploring simple siamese representation learning, in: CVPR, 2021, pp. 15750–15758.

[4] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al., Bootstrap

your own latent-a new approach to self-supervised learning, NIPS 33 (2020) 21271–21284.

[5] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: CVPR, 2020, pp. 9729–9738.

[6] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, G. Hinton, Big self-supervised models are strong semi-supervised learners, arXiv:2006.10029 (2020).

[7] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: CVPR, 2014, pp. 580–587.

[8] D. Hu, Q. Lu, L. Hong, H. Hu, Y. Zhang, Z. Li, A. Shen, J. Feng, How well self-supervised pre-training performs with streaming data?, ArXiv:2104.12081 (2021).

[9] W. Huang, M. Yi, X. Zhao, Towards the generalization of contrastive self-supervised learning, ArXiv:2111.00743 (2021).

[10] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, P. Isola, What makes for good views for contrastive learning?, arXiv:2005.10243 (2020).

[11] X. Tang, A. Nair, B. Wang, B. Wang, J. Desai, A. Wade, H. Li, A. Celikyilmaz, Y. Mehdad, D. Radev, Confit: Toward faithful dialogue summarization with linguistically-informed contrastive fine-tuning, ArXiv:2112.08713 (2021).

[12] J. Zhang, T. Bui, S. Yoon, X. Chen, Z. Liu, C. Xia, Q. H. Tran, W. Chang, P. Yu, Few-shot intent detection via contrastive pre-training and fine-tuning, ArXiv:2109.06349 (2021).

[13] Y. Zhang, B. Hooi, D. Hu, J. Liang, J. Feng, Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning, arXiv:2102.06605 (2021).

[14] J. Zhong, X. Wang, Z. Kou, J. Wang, M. Long, Bi-tuning of pre-trained representations, arXiv:2011.06182 (2020).

[15] S. Ge, S. Mishra, C.-L. Li, H. Wang, D. Jacobs, Robust contrastive learning using negative samples with diminished semantics, NIPS 34 (2021) 27356–27368.

[16] T. Huynh, S. Kornblith, M. R. Walter, M. Maire, M. Khademi, Boosting contrastive self-supervised learning with false negative cancellation, in: WACV, 2022, pp. 2785–2795.

[17] J. Liu, M. Yang, M. Zhou, S. Feng, P. Fournier-Viger, Enhancing hyperbolic graph embeddings via contrastive learning, ArXiv:2201.08554 (2022).

[18] J. Robinson, L. Sun, K. Yu, K. Batmanghelich, S. Jegelka, S. Sra, Can contrastive learning avoid shortcut solutions?, NIPS 34 (2021) 4974–4986.

[19] F. Wang, H. Liu, Understanding the behaviour of contrastive loss, in: CVPR, 2021, pp. 2495–2504.

[20] T. Wang, P. Isola, Understanding contrastive representation learning through alignment and uniformity on the hypersphere, in: ICML, PMLR, 2020, pp. 9929–9939.

[21] B. Gunel, J. Du, A. Conneau, V. Stoyanov, Supervised contrastive learning for pre-trained language model fine-tuning, arXiv:2011.01403 (2020).

[22] M. Halkidi, M. Vazirgiannis, Clustering validity assessment: finding the optimal partitioning of a data set, ICDM (2001) 187–194.

[23] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: ICML, PMLR, 2020, pp. 1597–1607.

[24] X. Chen, H. Fan, R. Girshick, K. He, Improved baselines with momentum contrastive learning, arXiv:2003.04297 (2020).

[25] Q. Liu, B. Wang, Neural extraction of multiscale essential structure for network dismantling, Neural Networks 154 (2022) 99–108. `doi:https://doi.org/10.1016/j.neunet.2022.07.015`.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: CVPR, IEEE, 2009, pp. 248–255.

[27] T. Chen, C. Luo, L. Li, Intriguing properties of contrastive losses, NIPS 34 (2021) 11834–11845.

[28] K. Kotar, G. Ilharco, L. Schmidt, K. Ehsani, R. Mottaghi, Contrasting contrastive self-supervised representation learning pipelines, in: ICCV, 2021, pp. 9949–9959.

[29] A. Newell, J. Deng, How useful is self-supervised pretraining for visual tasks?, in: CVPR, 2020, pp. 7345–7354.

[30] K. Zhang, N. Robinson, S.-W. Lee, C. Guan, Adaptive transfer learning for eeg motor imagery classification with deep convolutional neural network, Neural Networks 136 (2021) 1–10. `doi:https://doi.org/10.1016/j.neunet.2020.12.013`.

[31] S. S. Basha, S. K. Vinakota, V. Pulabaigari, S. Mukherjee, S. R. Dubey, Autotune: Automatically tuning convolutional neural networks for improved transfer learning, Neural Networks 133 (2021) 112–122. `doi:https://doi.org/10.1016/j.neunet.2020.10.009`.

[32] Y. Guo, Y. Zheng, M. Tan, Q. Chen, J. Chen, P. Zhao, J. Huang, Nat: Neural architecture transformer for accurate and compact architectures, Advances in Neural Information Processing Systems 32 (2019).

[33] Y. Guo, Y. Zheng, M. Tan, Q. Chen, Z. Li, J. Chen, P. Zhao, J. Huang, Towards accurate and compact architectures via neural architecture transformer, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[34] X. Li, H. Xiong, H. An, C.-Z. Xu, D. Dou, Rifle: Backpropagation in depth for deep transfer learning through re-initializing the fully-connected layer, in: ICML, PMLR, 2020, pp. 6010–6019.

[35] K. You, Z. Kou, M. Long, J. Wang, Co-tuning for transfer learning, NIPS 33 (2020).

[36] C. Wu, F. Wu, T. Qi, Y. Huang, X. Xie, Noisytune: A little noise can help you finetune pretrained language models better, arXiv:2202.12024 (2022).

[37] J. Hou, X. Ding, J. D. Deng, S. Cranefield, Deep adversarial transition learning using cross-grafted generative stacks, Neural Networks 149 (2022) 172–183. `doi:https://doi.org/10.1016/j.neunet.2022.02.011`.

[38] Y. Guo, J. Chen, J. Wang, Q. Chen, J. Cao, Z. Deng, Y. Xu, M. Tan, Closed-loop matters: Dual regression networks for single image super-resolution, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 5407–5416.

[39] Z. Xie, Z. Wen, Y. Wang, Q. Wu, M. Tan, Towards effective deep transfer via attentive feature alignment, Neural Networks 138 (2021) 98–109. `doi: https://doi.org/10.1016/j.neunet.2021.01.022`.

[40] M. Jing, J. Li, K. Lu, L. Zhu, Y. Yang, Learning explicitly transferable representations for domain adaptation, Neural Networks 130 (2020) 39–48. `doi:https://doi.org/10.1016/j.neunet.2020.06.016`.

[41] Y. Guo, J. Wang, Q. Chen, J. Cao, Z. Deng, Y. Xu, J. Chen, M. Tan, Towards lightweight super-resolution with dual regression learning, arXiv:2207.07929 (2022).

[42] X. Chen, S. Wang, M. Long, J. Wang, Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation, in: ICML, PMLR, 2019, pp. 1081–1090.

[43] R. Dangovski, L. Jing, C. Loh, S. Han, A. Srivastava, B. Cheung, P. Agrawal, M. Soljačić, Equivariant contrastive learning, arXiv:2111.00899 (2021).

[44] T. Xiao, X. Wang, A. A. Efros, T. Darrell, What should not be contrastive in contrastive learning, arXiv:2008.05659 (2020).

[45] X. Wang, J. Gao, M. Long, J. Wang, Self-tuning for data-efficient deep learning, in: ICML, PMLR, 2021, pp. 10738–10748.

[46] X. Yan, I. Misra, A. Gupta, D. Ghadiyaram, D. Mahajan, Clusterfit: Improving generalization of visual representations, in: CVPR, 2020, pp. 6509–6518.

[47] Y. Guo, D. Stutz, B. Schiele, Improving robustness by enhancing weak subnets, in: European conference on computer vision, Springer, 2022.

[48] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, M. Bethge, Improving robustness against common corruptions by covariate shift adaptation, arXiv:2006.16971 (2020).

[49] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).

[50] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, in: CVPR Workshop, IEEE, 2004, pp. 178–178.

[51] J. Krause, J. Deng, M. Stark, L. Fei-Fei, Collecting a large-scale dataset of fine-grained cars (2013).

[52] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, arXiv:1306.5151 (2013).

[53] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: ICCV, IEEE, 2008, pp. 722–729.

[54] O. M. Parkhi, A. Vedaldi, A. Zisserman, C. Jawahar, Cats and dogs, in: CVPR, IEEE, 2012, pp. 3498–3505.

[55] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: CVPR, 2014, pp. 3606–3613.

[56] J. Howard, Imagenette and imagewoof, `https://github.com/fastai/imagenette`, github repository with links to dataset (2019).

[57] L. Xuhong, Y. Grandvalet, F. Davoine, Explicit inductive bias for transfer learning with convolutional networks, in: ICML, PMLR, 2018, pp. 2825–2834.

[58] X. Zhan, Z. Liu, P. Luo, X. Tang, C. Loy, Mix-and-match tuning for self-supervised semantic segmentation, in: AAAI, Vol. 32, 2018.

[59] X. Li, H. Xiong, H. Wang, Y. Rao, L. Liu, Z. Chen, J. Huan, Delta: Deep learning transfer using feature map with attention for convolutional networks, arXiv:1901.09229 (2019).

[60] X. Chen, S. Wang, B. Fu, M. Long, J. Wang, Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning, NIPS (2019).

[61] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, arXiv:2111.06377 (2021).