# Revisiting the Critical Factors of Augmentation-Invariant Representation Learning

Junqiang Huang[†], Xiangwen Kong[†], and Xiangyu Zhang

MEGVII Technology, Beijing, China
{huangjunqiang,kongxiangwen,zhangxiangyu}@megvii.com

**Abstract.** We focus on better understanding the critical factors of augmentation-invariant representation learning. We revisit MoCo v2 and BYOL and try to prove the authenticity of the following assumption: different frameworks bring about representations of different characteristics even with the same pretext task. We establish the first benchmark for fair comparisons between MoCo v2 and BYOL, and observe: (i) sophisticated model configurations enable better adaptation to pre-training dataset; (ii) mismatched optimization strategies of pre-training and fine-tuning hinder model from achieving competitive transfer performances. Given the fair benchmark, we make further investigation and find asymmetry of network structure endows contrastive frameworks to work well under the linear evaluation protocol, while may hurt the transfer performances on long-tailed classification tasks. Moreover, negative samples do not make models more sensible to the choice of data augmentations, nor does the asymmetric network structure. We believe our findings provide useful information for future work.

## 1 Introduction

Recently, with the advancement of research on pretext tasks [12,11,16,29,30,40], self-supervised learning (SSL) presents extraordinary potential in computer vision, pushing the frontier of transfer learning. The effectiveness of self-supervised learned representations has been empirically verified. Compared to supervised pre-training counterparts, MoCo series [20,7,9] achieves comparable or even better performances on object detection, semantic segmentation, etc. Moreover, under the linear evaluation protocol on ImageNet [34] (an often used evaluation metric for SSL), BYOL [18] and SwAV [3] have largely shrunk the gap with supervised learning.

Among various pretext tasks, one of the most promising ways is to pull together the positive sample pairs (different augmented views of the same image), which enables the model to learn augmentation-invariant representations. The simplicity of this pretext task also brings about a notorious problem: without careful design, the model will collapse to a trivial solution that all images are

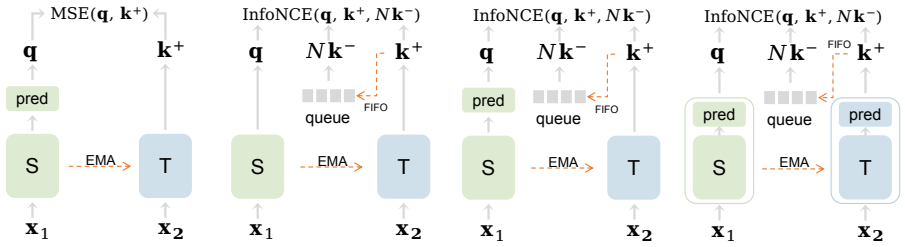
[†]Equal Contribution
Code: https://github.com/megvii-research/revisitAIRL

mapped to a constant vector, resulting in useless representations. To avoid this collapse, contrastive methods like MoCo impose regularization by pushing away the negative sample pairs (different images), while BYOL develops the asymmetric siamese network with a stop-gradient operation. Though sharing the same pretext task, MoCo v2 and BYOL show different results of linear classification and transfer learning. As reported in [18,8], BYOL has higher linear accuracy, while MoCo v2 presents better transferability. Given this observation, it is natural to assume different frameworks bring about representations of different characteristics.

To prove or disprove the above assumption, it is essential to build the benchmark for fair comparison between contrastive frameworks and BYOL. Since MoCo v2 shares many similarities with BYOL, which is convenient to perform controlled experiments, we choose it as the representative of contrastive methods. We aim to study the experimental impact of the following variables on augmentation-invariant representation learning: model configurations (i.e., network architecture, symmetry of training loss, etc.), combination of data augmentations, and optimization strategies. The evaluation criteria consist of linear classification accuracy and transfer performances of typical downstream tasks. Our efforts and contributions will be described next.

We challenge the opinion arising from previous experimental observations of [18,8] that the superiority of linear evaluation is unique to SSL frameworks without negative sample pairs (e.g., BYOL [18], SimSiam [8]). We ablate the differences in model configurations between MoCo v2 and BYOL, including network architecture, rule of momentum update, and symmetry of training loss. The differences are iteratively removed based on MoCo v2. Without searching pre-training hyper-parameters, the linear accuracy of MoCo v2 on ImageNet consistently benefits from the sophisticated model configurations (72.0% top-1 accuracy for 200-epoch pre-training). On top of this, we reformulate MoCo v2 into a more effective version as shown in Fig. 1c (MoCo v2+ for short). Moreover, when pre-training with more complex data augmentations, MoCo v2+ receives further improvement (72.4%). Our study suggests that the sophisticated design of model configurations affects a lot on the pretext task's performance.

Second, we try to uncover the mystery of BYOL's poor transferability. It seems that practitioners struggle to fully unleash the potential of BYOL even with heavy computation to search fine-tuning learning rates [8]. We tackle this issue by investigating the optimization strategy (e.g., optimizer, learning rate, etc.) of pre-training and fine-tuning. By delving into the original implementation of BYOL and its LARS optimizer [42], we find the distribution of LARS-trained representations is different from that of SGD-trained representations. The currently used fine-tuning optimization strategy best selected for SGD-trained features is not suitable for LARS-trained features. We therefore can conclude that the mismatched optimizer choices (LARS for pre-training and SGD for fine-tuning) cause the sub-optimal performances of BYOL. Obviously, using matched optimizers or searching optimization hyper-parameters for fine-tuning can circumvent this issue. In this paper, we also propose one simple yet effective technique NormRescale




(a) BYOL (b) MoCo v2 (c) MoCo v2+ (d) S-MoCo v2+

Fig. 1: This figure compares the structures of four SSL frameworks discussed in our paper. All of them are siamese network along with the stop-gradient operation and momentum update. For convenient reference, we name the encoder updated by gradients as student encoder, and the encoder with stop-gradient operation as the teacher encoder. They are represented by the capital letter, **S** and **T** respectively. Note that the backbones of both student and teacher encoder include a projector that is a non-linear 2-layer MLP (not shown in the picture). **pred** in the green box represents the predictor (also a non-linear 2-layer MLP). The only difference between MoCo v2+ and S-MoCo v2+ lies in the existence of teacher encoder's predictor

to solve this problem. NormRescale rescales the weight norm of LARS-trained model by the SGD-trained counterpart. NormRescale works well across many downstream tasks and significantly outperforms the baseline, which proves its capability to recover BYOL's transferability.

Thus far, a fair benchmark has been established. We can make a robust argument that it is not the frameworks but the training details that determine the characteristics of learned representations.

Thanks to the unified training details, we are able to quest for the experimental impact of the asymmetric network structure. Previous work [18,8,9] has verified the effectiveness of asymmetric network structure for linear classification. The influences on transfer learning are yet to be examined. To this goal, we symmetrize the network structure of MoCo v2+, which gives us the Symmetric MoCo v2+ (abbreviated as S-MoCo v2+, the structure can be seen in Fig. 1d). Based on the comparison among MoCo v2+, S-MoCo v2+, and BYOL, our findings are threefold: (i) asymmetric network structure leads to better adaptation on the pre-training datasets but does not mean higher transferability; (ii) the performances of long-tailed classification datasets are more outstanding for contrastive methods, and will be further improved by the symmetric network structure; (iii) contrary to the claim in [18,43], contrastive methods with or without symmetry of network structure are not more susceptible to data augmentations than BYOL.

Compared to the current literature, our findings are surprising and challenge existing understanding of self-supervised learning. The extensive experiments convey a main idea that *training details determine the characteristics of learned representations*. As long as we align the model configurations, combination of

data augmentations and optimization strategy of MoCo v2 and BYOL, they show similar performances in linear evaluation and transferring to other downstream tasks. We hope the fair benchmark and our observations will motivate future research.

## 2 Related Work

**Augmentation-invariant representation learning.** There have been a great deal of pretext task [12,11,16,29,31,30,40,37,2,20,28,6,18,3,39,41,43,14,4,35,23] proposed in self-supervised learning. Amongst them, augmentation-invariant representation learning shines brightly. The core idea of augmentation-invariant representation learning is to attract different augmented views of the same image as closely as possible. Many research branches are derived from the creative endeavor of the community. Contrastive methods [30,37,20,6,39,41,23] follow the idea proposed in [19] to pull together the positive sample pairs and push away the negative sample pairs. BYOL [18] and SimSiam [8] directly minimize the distance of positive sample pairs, along with an asymmetric siamese network. W-MSE [14] attracts the positive pairs based on the whitening features. SwAV [3] first performs online clustering and then classification according to the clustering label generated by its positive sample. DINO [4] optimizes the distribution distances of positive sample pairs along with the "centering" and "sharpening" operations. BarlowTwins [43] maximizes the correlation of positive sample pairs and decorrelates the features of different images. Research on augmentation-invariant representation learning has sprung up, which also illustrates the advantages of augmentation-invariant representation learning as a pretext task for self-supervised learning.

**Impact of training details.** Discussion about the impact of training details on representation quality is not new to the community. Previous work has explored which factors enable performance promotion for their algorithms. For example, in order to boost the accuracy of linear evaluation, MoCo series [20,7,9] and SimCLR [6] search for optimization hyper-parameters (e.g., learning rate, learning rate decay schedule, batch size, etc.), the combination of data augmentations, and the number of negative samples. BYOL [18,33] and SimSiam [8] ablates the coefficients of momentum update and the choice of batch normalization. Due to the lack of a fair benchmark, the successes of these methods seem to be binding together with their unique framework. We are not aware of whether future work can learn from their successes.

Other work like [44] makes a contribution to better understanding the transfer performance of instance discrimination. But the scope of their study is narrowed down to MoCo v2. SimSiam [8] pays more attention to what the optimization problem for frameworks without using negative samples is. [13] focuses on the fine-tuning results based on frozen pre-trained weights. Unlike them, we provide extensive experiments based on MoCo v2 and BYOL that are pre-trained given various training details. The standard evaluation protocol includes linear

evaluation on pre-training datasets and transfer performance of some typical downstream tasks. Our goal is to build the first fair benchmark to compare MoCo v2 and BYOL, two influential frameworks in augmentation-invariant representation learning.

**Asymmetric network structure.** The notorious problem of augmentation-invariant representation learning is that without careful design, all input images are mapped to a constant vector. The solution of contrastive frameworks is simple and intuitive—repulsing the negative sample pairs. Likewise, feature decorrelation methods [14,4,24] separate the features according to the specific rules to avoid the collapse. BYOL [18] and SimSiam [8] rely on the asymmetric network structure and the stop-gradient operation. To study the optimization problem based on asymmetric network structure, SimSiam ablates many hyper-parameters of pre-training. Later work [38] concentrates on the theoretical influence of the asymmetric network structures.

It should be noted that the focus of this work is not on advancing the development of SSL by proposing a new algorithm. On the opposite, we aim to present a fair and comprehensive investigation based on existing algorithms to gain better understanding.

## 3 Experimental Setup

### 3.1 Framework

In this section, we briefly review two well-known frameworks of augmentation-invariant representation learning: MoCo v2 [7] and BYOL [18]. Both of them adopt the design of teacher-student siamese network with momentum update rule [36], where the teacher encoder is updated by the exponential moving average of the student encoder. This unity of network structure is convenient for us to perform controlled experiments. It is worth noting that other self-supervised learning frameworks pre-training with different pretext tasks are beyond the scope of our paper.

**MoCo v2.** By optimizing the contrastive loss [19], MoCo v2 learns to pull the features of positive sample pairs (different augmented views of the same image) together and to push the features of negative sample pairs (different images) away. Different from other contrastive frameworks [30,40,37,28,6], MoCo v2 designs a memory queue (first-in, first-out) to store features computed in previous training iterations. Meanwhile, the rule of momentum update helps maintain the feature consistency. In practice, a batch of input images will be independently transformed twice, resulting in a batch of positive sample pairs. The teacher-student siamese network then encodes them as features respectively. The mini-batch contrastive loss is described as follow:

$$L = -\frac{1}{N}\sum_{\mathbf{q}} \log\left(\frac{\exp(\mathbf{q}^{\mathsf{T}}\mathbf{k}^{+}/\tau)}{\exp(\mathbf{q}^{\mathsf{T}}\mathbf{k}^{+}/\tau) + \sum_{\mathbf{k}^{-}} \exp(\mathbf{q}^{\mathsf{T}}\mathbf{k}^{-}/\tau)}\right) \tag{1}$$

$\mathbf{q}$ and $\mathbf{k}^+$ stand for the student feature and teacher feature that are encoded from the positive sample pair by the siamese network respectively. $\mathbf{k}^-$ is the negative feature stored in the memory queue. $N$ is the batch size and $\tau$ is the temperature (for the following experiments of our paper, we use 0.2 by default). After back-propagating the contrastive loss, all the teacher features $\{\mathbf{k}^+\}$ are enqueued and the "oldest" of the memory queue features are dequeued.

**BYOL.** Similar to contrastive methods, BYOL learns to attract the positive sample pairs as close as possible in feature space without regularization of negative sample pairs. Previous work [18,8] have stated the asymmetric structure of siamese network and the stop-gradient operation (no gradient will flow to the teacher encoder) are critical to avoiding trivial solution in BYOL. The asymmetric structure refers to that the student branch of siamese network is followed by a predictor (a non-linear two-layer MLP), yielding the asymmetry between student and teacher branches. The mini-batch training loss of BYOL is symmetric:

$$L = \frac{1}{N}\left(\sum_{\mathbf{q_1}} \|\mathbf{q_1} - \mathbf{k_1}\|^2 + \sum_{\mathbf{q_2}} \|\mathbf{q_2} - \mathbf{k_2}\|^2\right) \tag{2}$$

The samples from a positive pair are mapped to $\mathbf{q_1}$ and $\mathbf{q_2}$ by the student encoder, and mapped to $\mathbf{k_1}$ and $\mathbf{k_2}$ by the teacher encoder. $\|\cdot\|$ is the Euclidean distance.

### 3.2 Pre-training and Evaluation

In this section, we provide the required information on pre-training and fine-tuning for our experiments. The backbone of siamese network is ResNet-50 [22], and the pre-training dataset is ImageNet [34]. The details about data augmentations can be found in Supplementary Materials.

**Pre-training.** To re-implement MoCo v2 efficiently, we make the following adjustments: increasing the training batch size to 1024, linearly scaling up the learning rate to 0.12 according to [17], and introducing a 10-epoch linear warm-up schedule before the decay of learning rate. Note that these modifications do not change the performance of MoCo v2. To reproduce BYOL, we faithfully follow the training settings in [18]. There are two combinations of data augmentations mentioned in BYOL. We use the symmetric one for our experiments. In the crossover study of Sec. 4.2, when training MoCo v2+ with LARS [42] optimizer, the training hyper-parameters are copied from the implementation of the original BYOL. Likewise, when training BYOL with SGD optimizer, we adopt the same hyper-parameters used in MoCo v2+.

**Linear evaluation.** The common practice of linear evaluation is to freeze the backbone and train a linear classifier based on the fixed representations. Here, we provide two settings for the training phase of linear evaluation. For models pre-trained with SGD optimizer, we use SGD optimizer to train for 100 epochs. The batch size is 256, and the initial learning rate is 30 which is decayed by a factor

of 10 at the 60 and 80-th epoch. For models pre-trained with LARS optimizer, we follow the hyper-parameters adopted in BYOL. We use SGD optimizer with Nesterov to train for 80 epochs. The batch size is 1024, and the initial learning rate is 0.8 and is decayed to 0 by the cosine schedule. Both training settings use a momentum of 0.9 and no weight decay. After training, we report the single-crop classification accuracy on ImageNet validation set.

**PASCAL VOC object detection.** We transfer the pre-trained models on PASCAL VOC [15] for object detection. We strictly follow the training details in [20], which uses a Faster R-CNN [32] detector with a backbone of ResNet50-C4. It takes 9k iterations to fine-tune on `trainval2007` set and 24k iterations to fine-tune on `trainval07+12` set. We report the results evaluated on `test2007` set.

**COCO object detection and instance segmentation.** We fine-tune the pre-trained models on COCO [27] for object detection and instance segmentation. We adopt Mask R-CNN [21] as the detector with two kinds of backbone, ResNet50-C4 and ResNet50-FPN. For a fair comparison, the training settings are exactly the same used in [20]. Following the 1× optimization setting, it takes 90k iterations to fine-tune on `train2017` set. Finally, we report the results evaluated on `val2017` set.

**CityScapes semantic segmentation.** We train on CityScapes [10] to evaluate the performance on semantic segmentation. For easy re-implementation, we use DeepLab-v3 architecture [5]. The backbone is ResNet50 with a stride of 8. The crop size is $512 \times 1024$ for training, and $1024 \times 2048$ for testing. It takes 40k iterations to fine-tune on `train_fine` set, and finally we report the results evaluated on `val` set.

## 4 Experiments and Analyses

### 4.1 What Matters in Linear Evaluations?

In the light of previous work, SSL methods without negative sample pairs (e.g., BYOL [18], SimSiam [8], DINO [4]) have higher accuracy in linear evaluation, compared to contrastive methods. What on earth hinders contrastive methods like MoCo v2 from better adapting to the pre-training dataset to achieve better performance in linear evaluation, the negative sample pairs or other previously ignored factors?

In this subsection, we seek to answer the question by exploring how to elevate MoCo v2 to achieve higher accuracy in linear evaluation. Given this goal, it is desirable to improve linear accuracy under modifications to model configurations. With reference to BYOL, we make the following adjustment on MoCo v2. First, we replace the ShufflingBN with synchronized BN (SyncBN). However, this direct replacement does not bring the expected performance improvement. Hence we

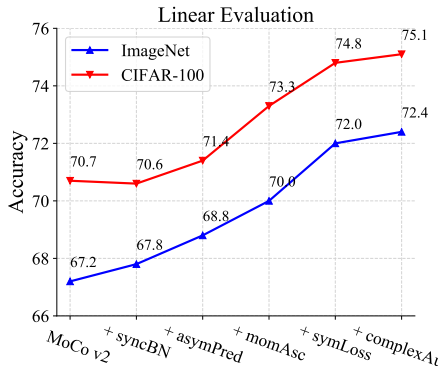



Fig. 2: Top-1 accuracy of linear evaluation on ImageNet and CIFAR-100. The x-axis represents the modifications of model configurations. We use MoCo v2 as our baseline. All models are trained for 200 epochs. The trend of these two curves indicates that the linear accuracy consistently benefits from the sophisticated model configurations

Table 1: The results of transfer learning on detection and segmentation tasks. All models are trained for 200 epochs. The best results are marked as bold

| | VOC07 | VOC07+12 | COCO | | | | CityScapes |
|---|---|---|---|---|---|---|---|
| | $AP_{50}$ | $AP_{50}$ | $AP^{C4}_{box}$ | $AP^{C4}_{seg}$ | $AP^{FPN}_{box}$ | $AP^{FPN}_{seg}$ | mIoU |
| MoCo v2 | 76.5 | 82.2 | 38.8 | 34.0 | 39.5 | 35.8 | 77.4 |
| + SyncBN | 76.7 | 82.0 | 38.6 | 33.7 | 39.7 | 35.8 | 76.9 |
| + Asymmetric Predictor | 76.7 | 82.1 | 39.0 | 34.1 | 39.8 | 35.9 | 77.3 |
| + Momentum Ascending | 77.0 | 82.3 | 39.0 | 34.3 | 39.7 | 35.8 | 77.4 |
| + Symmetric Loss (MoCo v2+) | 77.1 | **82.7** | **39.4** | **34.5** | 40.3 | 36.5 | **77.6** |
| + More Complex Augmentations | **77.3** | **82.7** | 39.2 | 34.4 | **40.4** | **36.7** | **77.6** |
| BYOL | 71.7 | 79.1 | 35.3 | 31.1 | 40.8 | 36.9 | 76.4 |

insert a BN to the hidden layer of the projector (a non-linear two-layer MLP after the backbone) as BYOL does. Second, we add a predictor (an MLP similar to projector) at the end of student encoder, yielding asymmetry between student encoder and teacher encoder. Third, the coefficient of momentum update no longer stays still, but increases from 0.99 to 1 according to a cosine schedule. Forth, we symmetrize the contrastive loss, as has been done in [6,18,3,8]. For convenient reference, we name this enhanced framework as MoCo v2+, which is an extension of MoCo v2. Last, we train MoCo v2+ with more complex data augmentations (introducing solarization to the combination)[1].

The results of linear evaluation on ImageNet are in Fig. 2. Surprisingly, the linear accuracy consistently benefits from the modifications even without searching

[1]The detailed information about the combination of data augmentations can be found in Supplementary Materials

hyper-parameters. When training with more complex augmentations, MoCo v2+ finally catches up to BYOL in terms of linear accuracy (72.4% top-1 accuracy). Among these changes, the symmetrization of contrastive loss brings the most obvious improvement (2.0% accuracy increment). To validate the effectiveness of representations with high linear accuracy on ImageNet, we train a linear classifier on CIFAR-100 [26]. The accuracy curve is in Fig. 2. Similarly, we observe distinct promotions for linear evaluation on CIFAR-100 compared to baseline.

Tab. 1 presents the transfer performances on downstream tasks. The promotions (about 0.5% improvement) in transfer learning are not as obvious as in linear evaluation. One possible explanation for this contradiction is that better adaptation to pre-training dataset is more helpful to those datasets whose distribution are similar to the pre-training dataset. As we can see in Fig. 2, the trends of two curves in Fig. 2 are accordant. In a nutshell, *sophisticated design of model configurations affects a lot on the pretext task's performance.*

### 4.2 How to Improve Transfer Performances?

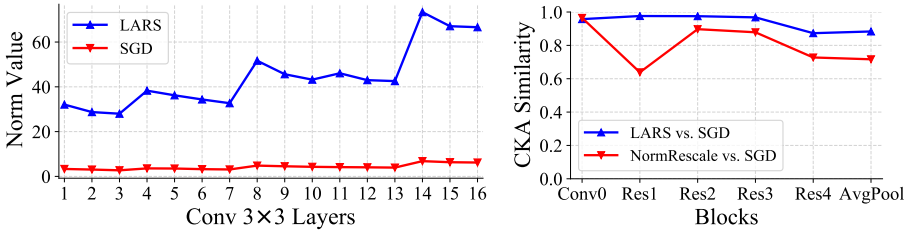
Despite BYOL being one of the significant frameworks in SSL, its capability on typical downstream tasks like object detection on VOC [15] and COCO [27] have not received enough attention. From one of only a few studies concerning this problem, we find MoCo v2 outperforms BYOL on VOC and COCO detection and instance segmentation [8]. Tab. 1 also reflects this issue. We notice these comparisons are based on misaligned optimization strategies. Specifically, BYOL utilizes LARS optimizer [42] to train with large batch size, while MoCo v2+ uses SGD optimizer. In this case, it remains an open question whether BYOL has innately poor transferability on those challenging downstream tasks given the same optimization strategy.

In this subsection, we investigate how to improve transfer performances of BYOL from the perspective of optimization. To understand how optimization strategy influences the transferability of learned representations, we provide a crossover study of SGD and LARS optimizers for pre-training. The results of downstream tasks are in Tab. 2. Both frameworks are less competitive on most downstream tasks when pre-trained with LARS. It seems that the poor results may originate from the pre-training optimization strategy. There is one exception, though, that LARS-trained models show comparable or even better results for the downstream tasks adopting ResNet50-FPN as backbone. We, therefore, infer that the LARS optimizer does not compromise the quality of learned representations.

By examining the implementation details of BYOL, we find that, unlike SGD, the LARS optimizer does not impose L2-regularization on the parameters of batch normalization layers. As training goes on, the weight norm becomes larger. We can see the clear contrast in Fig. 3a that the weight norms of the LARS-trained model are significantly larger. In other words, the distribution of learned representations is different from those trained by SGD. Fine-tuning LARS-trained models with the hyper-parameters best suited for SGD-trained models naturally yields sub-optimal performances. Thus, we can conclude that

Table 2: The results of crossover study involving SGD and LARS optimizer. All models are trained for 200 epochs. The best results are marked bold

| | Optimizer | ImageNet | VOC07 | VOC07+12 | COCO | | | | CityScapes |
|---|---|---|---|---|---|---|---|---|---|
| | | Acc | $AP_{50}$ | $AP_{50}$ | $AP^{C4}_{box}$ | $AP^{C4}_{seg}$ | $AP^{FPN}_{box}$ | $AP^{FPN}_{seg}$ | mIoU |
| MoCo v2+ | SGD | 72.0 | **77.1** | **82.7** | **39.4** | **34.5** | 40.3 | 36.5 | **77.7** |
| | LARS | **72.5** | 62.9 | 74.4 | 32.1 | 28.9 | 40.0 | 36.2 | 73.2 |
| BYOL | SGD | 72.1 | 76.2 | 82.4 | 38.8 | 33.9 | 39.9 | 36.1 | 77.5 |
| | LARS | 72.4 | 71.7 | 79.1 | 35.3 | 31.1 | **40.8** | **36.9** | 75.2 |




(a) Weight norm (b) CKA similarity

Fig. 3: (**a**): Weight norms of all conv3 × 3 layers from LARS-trained and SGD-trained models. (**b**): CKA similarities of LARS-trained and SGD-trained representations across all stages of ResNet-50. Best viewed in color

*mismatched optimizer used in pre-training and fine-tuning is the reason for performance degeneration in BYOL, but not the framework itself.*

Tab. 2 points out a solution to circumvent this issue—using SGD optimizer for pre-training. This solution, however, is not universally effective, since it does not apply to large batch size training where LARS is more popularly used. To alleviate this issue, [8] searches learning rates for fine-tuning LARS-trained models, inevitably inducing heavy computation. Next, we describe two findings that lead us to a flexible approach.

First, we utilize the CKA similarity [25] to measure how similar the representations learned by LARS and SGD are. The blue line of Fig. 3b indicates these representations are sufficiently similar although they follow different distributions. Second, as described in [20], the features for the region proposal are normalized by the newly initialized BN in ResNet50-FPN, while not in ResNet50-C4. We argue the rescale operation in newly initialized BN helps LARS-trained models to adapt to optimization of fine-tuning driven by SGD. Motivated by the analyses above, we present a simple yet effective technique, NormRescale, to address this issue. Assume we have a well-trained model that is pre-trained by SGD[2]. For any weight of the LARS-trained model, we rescale its norm as follows:

[2]In default, we choose the 200-epoch SGD-trained BYOL.

Table 3: The downstream performances of BYOL under various implementations. "NR" stands for NormRescale. All models are trained for 200 epochs. The best results are marked bold

| | VOC07 | | | VOC07+12 | | | COCO detection | | | COCO instance seg. | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $AP_{50}$ | AP | $AP_{75}$ | $AP_{50}$ | AP | $AP_{75}$ | $AP_{50}$ | AP | $AP_{75}$ | $AP^M_{50}$ | $AP^M$ | $AP^M_{75}$ |
| BYOL-SGD | 76.2 | **48.1** | **52.9** | **82.4** | 56.5 | **63.6** | 58.5 | 38.8 | 42.1 | 55.2 | 34.0 | 36.2 |
| BYOL-LARS | 71.7 | 38.8 | 37.0 | 79.1 | 48.7 | 51.7 | 56.2 | 35.3 | 37.5 | 52.3 | 31.1 | 32.2 |
| BYOL in [8] | **77.1** | 47.0 | 49.9 | 81.4 | 55.3 | 61.1 | 57.8 | 37.9 | 40.9 | 54.3 | 33.2 | 35.0 |
| BYOL-NR | 76.6 | **48.1** | 51.6 | 82.1 | **56.7** | 62.9 | **59.3** | **39.3** | **42.6** | **56.0** | **34.5** | **36.7** |

$$\mathbf{w}^* = \|\mathbf{w}_S\| \cdot \frac{\mathbf{w}_L}{\|\mathbf{w}_L\|}, \tag{3}$$

where $\mathbf{w}_L$ is the weight vector of LARS-trained model, and $\mathbf{w}_S$ is the corresponding weight vector of SGD-trained model. $\|\cdot\|$ stands for 2-norm. We skip the procedure of hyper-parameters searching and fine-tune the processed weight $\mathbf{w}^*$ on downstream tasks.

In Tab. 3, we compare the transfer performances of BYOL under different implementations. The results of NormRescale are about the same as that of BYOL-SGD and significantly better than that of the vanilla implementation (BYOL-LARS). Moreover, it also shows superior performances on most metrics for detection and segmentation against the reproduction of [8]. The comparisons confirm NormRescale can effectively recover the transferability of the LARS-trained model. We also plot the CKA similarities between the representations of BYOL-SGD and NormRescale in Fig. 3b (red line). It can be seen that NormRescale retains the characteristics of LARS-trained representations. Apart from using BYOL-SGD as the anchor weight, we also explore other anchor choices for NormRescale. The detailed comparison can be found in Supplementary Materials.

These results suggest that optimization strategy is the key to the transferability of BYOL. For efficient comparison, we adopt SGD as our default optimizer for all the below experiments. Without confusing references, we continue to use BYOL to stand for SGD-trained BYOL.

Thus far, we have presented the first fair benchmark to compare two important frameworks of SSL, namely MoCo and BYOL. Our extensive experiments show that the performances of linear evaluation and transfer learning are similar in MoCo v2+ and BYOL given the aligned training details, leading to an authentic argument that *the training details determine the characteristics of learned representations.*

### 4.3 What Is the Impact of Asymmetric Network Structure?

The asymmetric network structure first proposed in BYOL plays an important role in avoiding model collapse for augmentation-invariant representation learning.

Table 4: Results on ImageNet and downstream tasks of BYOL, MoCo v2+, S-MoCo v2+. All models are trained for 200 epochs. The best results are marked bold

| | Asymmetry | ImageNet | VOC07 | VOC07+12 | COCO | | | | CityScapes |
|---|---|---|---|---|---|---|---|---|---|
| | | Acc | $AP_{50}$ | $AP_{50}$ | $AP^{C4}_{box}$ | $AP^{C4}_{seg}$ | $AP^{FPN}_{box}$ | $AP^{FPN}_{seg}$ | mIoU |
| BYOL | ✓ | 72.1 | 77.2 | **82.7** | **39.3** | **34.5** | **40.6** | 36.6 | 77.1 |
| MoCo v2+ | ✓ | **72.4** | **77.3** | **82.7** | 39.2 | 34.4 | 40.4 | 36.7 | **77.6** |
| S-MoCo v2+ | | 71.2 | 77.1 | 82.4 | 39.1 | 34.2 | **40.6** | **36.7** | 77.2 |

There are follow-up studies on the asymmetric structure that are mainly about the theoretical understanding [38,8] and the effectiveness for linear classification [8,9]. Here, we explore the experimental impact of asymmetric structure in transfer learning and its comparison to symmetric one. We symmetrize the network structure of MoCo v2+ by adding an extra predictor to the teacher encoder. We call it Symmetric MoCo v2+ (abbreviated as S-MoCo v2+). We refer to Fig. 1d for visual description. In this subsection, the experiments are mainly about the following three parts.

**Standard evaluation tasks.** We first provide a direct comparison amongst MoCo v2+, S-MoCo v2+, and BYOL on the typical datasets. The results of linear evaluation and transfer learning are listed in Tab. 4. As shown in the third column (ImageNet Acc), S-MoCo v2 is inferior in linear evaluation, indicating that models with asymmetric structure may better fit pretext tasks. The performances of transfer learning, in contrast, are similarly good on many downstream tasks. The biggest gap between the best and the worst is within 0.4. We conclude that the transferability for these regular downstream tasks may be neutral to the symmetry of network structure.

**Long-tailed classification task.** We next study the effects on two long-tailed classification tasks (CIFAR-10-LT and CIFAR-100-LT [1]). The effectiveness of pre-trained models is measured in two aspects: linear evaluation and fine-tuning. For solid comparisons, we provide models pre-trained with different hyper-parameters (e.g., learning rate, training epochs, etc.). As plotted in Fig. 4, the horizontal coordinate for each point represents the pre-trained model's linear accuracy on ImageNet and the vertical coordinate stands for linear or fine-tuning accuracy on long-tailed datasets. Our findings can be summarized as follows:

(i) Model with higher linear accuracy on ImageNet shows better performance on CIFAR-10/100-LT under the linear evaluation protocol. But the situation is different for fine-tuning. We do not see a clear trend between linear accuracy on ImageNet and fine-tuning accuracy.

(ii) In all four sub-figures, we can observe a clear ranking result that S-MoCo v2+ has the best effect, followed by MoCo v2+, and finally BYOL. The superiority of contrastive methods in linear evaluation and fine-tuning implies that the regularization imposed by pushing away negative sample pairs which renders

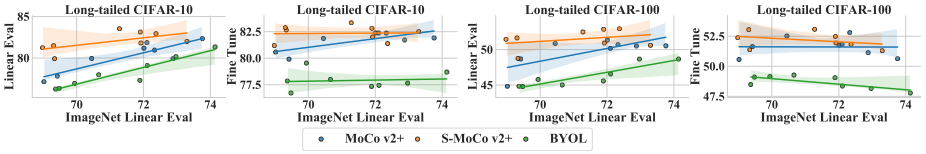



Fig. 4: Linear evaluation and fine-tuning results of BYOL, MoCo v2+ and S-MoCo v2+. The regression lines describe the correlation between linear accuracy on ImageNet and linear or fine-tuning accuracy on long-tailed classification datasets, with confidence intervals in shaded areas. Best viewed in color

a more uniform representation space is conducive to long-tailed classification. Besides, we point out the strength of symmetric network structure, as it provides the best performances of S-MoCo v2+ (see Fig. 4).

**Data augmentations.** We make an attempt to investigate the sensitivity of contrastive methods and BYOL to data augmentations, which has been discussed in [18,43]. The conclusions about this problem are consistent in their work—contrastive methods are more sensitive to the variation of data augmentations. Following our above analyses, we are sceptical about the validity of this conclusion where a fair benchmark is absent. To get a clear picture of it, we ablate data augmentations based on MoCo v2+, S-MoCo v2+, and BYOL. The baseline combination of data augmentations includes random cropping and resizing to $224 \times 224$, horizontal flipping, color jittering, gray scale converting, Gaussian blur, and solarization. The specific parameters of augmentations can be found in Supplementary Materials. Likewise, we iteratively remove the data augmentations involving color transformations. The order goes solarization, Gaussian blur, gray scale converting, and color jittering. After removing all color transformations, the combination is the same as used in supervised training. The results are depicted in Fig. 5.

Interestingly, the obvious accuracy gap between contrastive methods and BYOL reported in [18,43] vanishes; instead, we observe similar results of linear accuracy on ImageNet for contrastive methods and BYOL. The comparison of contrastive methods (MoCo v2+ vs. S-MoCo v2+) demonstrates that it is not the asymmetric head that causes these similarities. Therefore, we can present a convincing and empirically verified conclusion that a sufficiently complex combination of data augmentation is equally important for contrastive methods and BYOL. In turn, the consistency of effects between MoCo v2+, S-MoCo v2+ and BYOL suggests that ignoring training details can give misperceptions in SSL.

In addition to the linear accuracy on ImageNet, we also report the transfer performances across various downstream tasks. As clearly shown in Fig. 5, similar phenomena can be found in the results of different downstream tasks that transferability is positively correlated to the complexity of data augmentation combinations. Through careful observation, we find that most models meet

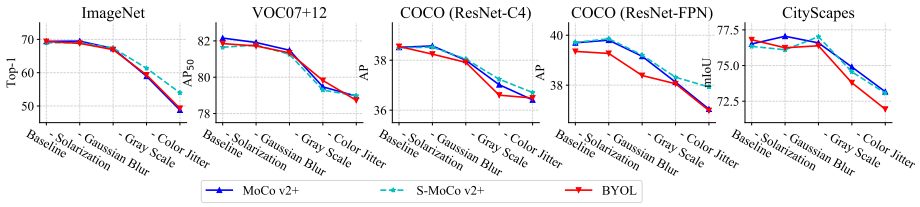



Fig. 5: The results of linear evaluation and downstream tasks under different combinations of data augmentations. Best viewed in color

significant performance degeneration if gray scale converting is cancelled. Strictly speaking, it is biased to believe gray scale converting is so important that SSL methods would face degradation without it. A more likely explanation is that the combination of data augmentations lacks complexity when cancelling out gray scale converting, inducing less competitive representations.

## 5 Conclusion

In summary, the extensive experiments throughout the paper revolve around the idea that training details determine the characteristics of learned representations in augmentation-invariant representation learning. In the process of verifying the idea, we observe the following:

(i) Sophisticated design of model configurations helps representations better adapt to the pre-training dataset, which in turn improves the linear accuracy on datasets with similar distribution to pre-training dataset.

(ii) What truly prevents BYOL from achieving competitive performances on typical downstream tasks is the mismatched optimization strategy for pre-training and fine-tuning. Using matched optimizers can remedy the performances drop. We also propose one simple yet effective technique to do the same, and it can apply to the situation where using mismatched optimizers is inevitable.

(iii) Asymmetric network structure leads to higher linear accuracy on pre-training dataset, while symmetric one has more competitive results on long-tailed classification tasks. Based on the fair comparisons among MoCo v2+, S-MoCo v2+ and BYOL, we confirm that contrastive methods and BYOL are equally sensitive to data augmentations.

We hope the fair benchmark and our observations will shed light on the understanding of MoCo v2 and BYOL, and help motivate future research to push forward the frontier of SSL.

**Acknowledgements.** This research was supported by National Key R&D Program of China (No. 2017YFA0700800) and Beijing Academy of Artificial Intelligence (BAAI).

## References


1. Cao, K., Wei, C., Gaidon, A., Aréchiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. In: NeurIPS. pp. 1565–1576 (2019)
2. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European conference on computer vision (ECCV). pp. 132–149 (2018)
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9650–9660 (2021)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 1597–1607. PMLR (2020)
7. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
8. Chen, X., He, K.: Exploring simple siamese representation learning. arXiv preprint arXiv:2011.10566 (2020)
9. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057 (2021)
10. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 3213–3223. IEEE Computer Society (2016). https://doi.org/10.1109/CVPR.2016.350
11. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. pp. 1422–1430. IEEE Computer Society (2015). https://doi.org/10.1109/ICCV.2015.167
12. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc. (2014)
13. Ericsson, L., Gouk, H., Hospedales, T.M.: How well do self-supervised models transfer? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5414–5423 (2021)
14. Ermolov, A., Siarohin, A., Sangineto, E., Sebe, N.: Whitening for self-supervised representation learning. arXiv preprint arXiv:2007.06346 (2020)
15. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision **111**(1), 98–136 (Jan 2015)

16. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018)
17. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677 (2017)
18. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)
19. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Proc. of CVPR. IEEE (2006)
20. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
21. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 2980–2988. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.322
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–778. IEEE Computer Society (2016). https://doi.org/10.1109/CVPR.2016.90
23. Hu, Q., Wang, X., Hu, W., Qi, G.J.: Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1074–1083 (2021)
24. Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., Zhao, H.: On feature decorrelation in self-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9598–9608 (2021)
25. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International Conference on Machine Learning. pp. 3519–3529. PMLR (2019)
26. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto (2009)
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. Springer (2014)
28. Misra, I., van der Maaten, L.: Self-supervised learning of pretext-invariant representations. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 6706–6716. IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.00674
29. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. Springer (2016)
30. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
31. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)

32. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada. pp. 91–99 (2015)
33. Richemond, P.H., Grill, J.B., Altché, F., Tallec, C., Strub, F., Brock, A., Smith, S., De, S., Pascanu, R., Piot, B., et al.: Byol works even without batch statistics. arXiv preprint arXiv:2010.10241 (2020)
34. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
35. Tao, C., Wang, H., Zhu, X., Dong, J., Song, S., Huang, G., Dai, J.: Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14431–14440 (2022)
36. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 1195–1204 (2017)
37. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. arXiv preprint arXiv:1906.05849 (2019)
38. Tian, Y., Chen, X., Ganguli, S.: Understanding self-supervised learning dynamics without contrastive pairs. In: International Conference on Machine Learning. pp. 10268–10278. PMLR (2021)
39. Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense contrastive learning for self-supervised visual pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3024–3033 (2021)
40. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 3733–3742. IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00393
41. Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, S., Hu, H.: Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. arXiv preprint arXiv:2011.10043 (2020)
42. You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017)
43. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. arXiv preprint arXiv:2103.03230 (2021)
44. Zhao, N., Wu, Z., Lau, R.W., Lin, S.: What makes instance discrimination good for transfer learning? arXiv preprint arXiv:2006.06606 (2020)