AN IMPROVED BASELINE FOR MASKED CONTRASTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Contrastive learning has significantly advanced self-supervised visual representation learning, making linear probe accuracy close to its supervised counterpart on ImageNet. However, vision transformers pre-trained with contrastive learning typically underperform those pre-trained with masked image prediction, when evaluated on fine-tuning benchmarks, e.g., image classification, object detection, and segmentation. In this paper, we improve the fine-tuning transfer performance of prior state-of-the-art contrastive approaches, e.g., MoCo-v3 and BYOL, from the following empirical perspectives: (i) applying masking strategies to input views; (ii) studying and comparing the effectiveness of Batch Normalization and Layer Normalization in projection and prediction heads; (iii) investigating the effectiveness of data augmentation and finding lighter augmentation during pre-training improves fine-tuning performance. As a result, we come up with a better baseline for contrastive transformers that outperforms baseline MoCo-v3 by 0.6% on ImageNet fine-tuning, and 2.1 mAP on MS COCO detection and segmentation benchmark for ViT-B, rivaling that of masked image prediction. Furthermore, our approach is significantly more efficient than MoCo-v3 due to the use of masking. These results suggest that, contrary to recent trends, contrastive learning remains competitive with masked image prediction on standard vision tasks.

1 INTRODUCTION

Contrastive learning has made steady progress for un/self-supervised visual representation learning in the past couple of years (Oord et al., 2018; Wu et al., 2018; He et al., 2019; Chen et al., 2020a), but has recently been overshadowed by Masked Image Prediction (MIP) paradigm (Bao et al., 2021; He et al., 2022; Xie et al., 2022). MIP is a form of denosing autoencoder (Vincent et al., 2008), which was recently popularized by BERT (Devlin et al., 2018) in Natural Language Processing. Despite obtaining strong off-the-shelf features (Caron et al., 2021; Chen et al., 2021), contrastive learning methods have been underperforming MIP methods in tasks where the pre-trained weights are fine-tunable (Li et al., 2021), especially for vision transformers (Dosovitskiy et al., 2020).

However, contrastive learning methods still have their own advantages compared with MIP methods. While it is natural to perform contrastive learning across modalities, *e.g.*, image-text pairs (Radford et al., 2021), MIP has thus far shown the most success only on single modality pre-training (cross-modal masked prediction is in principle possible but to the best of our knowledge there are not yet competitive results in this direction). In addition, contrastive learning tends to learn more discriminative frozen features, sometimes making linear probe match the supervised learning counterpart (Tomasev et al., 2022). In contrast, MIP methods need to fine-tune the pre-trained weights to show their generalization power. This comparison favors contrastive learning when pre-trained models are large and it is computationally infeasible to update the weights.

Therefore, it is of great interest if the fine-tuning gap between contrastive and MIP methods can be bridged. This paper thus focuses on this question, and aims to establish a stronger contrastive baseline. We choose MoCo-v3 (Chen et al., 2021) as the base method, because of its balance in performance and simplicity. There are two major differences between contrastive learning and MIP-based transformers: (a) the MIP methods take as input *masked images*; (b) and their learning objective function is *local*, *e.g.*, predicting local pixels (He et al., 2022) or features (Wei et al., 2022a). While local objectives facilitate fine-grained representations and may benefit dense prediction tasks such

as object detection and semantic segmentation, they have two drawbacks. They may need domain specific design, *e.g.*, normalizing pixels in MAE (He et al., 2022) and HOG (Dalal & Triggs, 2005) features in MaskFeat (Wei et al., 2022a). In addition, how to associate local targets across modalities (e.g. image and text) is unclear. Therefore we stick to the *global* objective function in contrastive learning as it requires the least domain knowledge and is thus more general. We conjecture that *masking* brings most (if not all) of the power for representation learning, and so we study the effects of masking in a *global* contrastive learning framework.

We improve the MoCo-v3 baseline from three empirical perspectives. First, with the elegant design of dropping masked patches in the transformer encoder (He et al., 2022), we improve both the accuracy and efficiency of MoCo-v3, *i.e.*, we achieve higher accuracy for the same number of training epochs, and for each epoch we save around 20% FLOPs. Second, we study the use of Batch Normalization (Ioffe & Szegedy, 2015) layer in projection & prediction heads, and find it introduces instability to the learned representations during the training phase. This seems to correlate with the phenomenon of random failing in training as we observed. We thus replace it with Layer Normalization (Ba et al., 2016), which achieves more stable training and better convergence. Third, inspired by prior works (Chen et al., 2020a; Tian et al., 2020) which show that data augmentation plays a key role in contrastive learning, we investigate on how different augmentation strategies in pre-training phase affect the fine-tuning accuracy. With the masking strategy applied, we find that relatively lighter data augmentation comes with better fine-tuning performance.

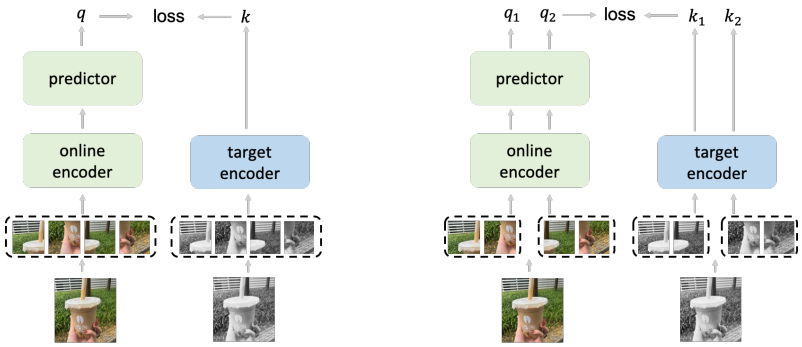
These changes result in a better baseline for contrastive transformers, which we call Masked MoCo (M-MoCo), (we also demonstrate the same changes, applied to BYOL (Grill et al., 2020), result in similarly strong numbers). Our goal is not to obtain the best numbers on benchmarks. Instead we hope this baseline can help inspire future research. Our contributions are:

1. A baseline that improves both accuracy and training speed over the well established method MoCo-v3. We achieve the same fine-tuning accuracy as MAE but only require 300 epochs of pre-training, compared to 1600 epochs for MAE.
2. A systematic study of how masking strategies proposed by MIP methods can be applied in conjunction with contrastive learning.
3. Significant improvement on downstream transferring tasks, *e.g.*, improving 2.1 and 1.9 mAP over MoCo-v3 for MS-COCO object detection and instance segmentation (Lin et al., 2014), respectively. This performance also may alleviate concerns raised in (Li et al., 2021) that contrastive learning cannot improve over random initialization.

2 Related Work

Masked Image Prediction. The Vision Transformer (ViT) architecture (Dosovitskiy et al., 2020) is fundamentally quite different from that of convolutional networks, using patch tokenization followed by multiple attention layers. This leads to significantly more flexible modeling capabilities. Works such as DINO (Caron et al., 2021) and MoCo-v3 (Chen et al., 2021) demonstrated that some self-supervised approaches developed on convolutional network backbones could perform well on ViTs after proper tuning to suit the new architecture. Masked image prediction (Bao et al., 2021; Chen et al., 2022; Xie et al., 2022), are directly inspired by pre-training methods used in NLP (Devlin et al., 2018), and as such, are well suited to the tokenized nature of ViT architectures. Masked Autoencoder (He et al., 2022) showed that classical masked autoencoding approaches (Vincent et al., 2008) could be used to pre-train ViTs. A key innovation in MAE was to drop masked tokens at the input to the encoder: this provides both a computational as well as a performance advantage.

Contrastive Learning. Contrastive learning (Wu et al., 2018; Oord et al., 2018; Tian et al., 2019; Bachman et al., 2019; Chen et al., 2020b; He et al., 2020) has achieved state of the art performance by enforcing invariance to augmentations. Negative samples (Robinson et al., 2021; Ge et al., 2021) are used to avoid trivial solutions by spreading the embedding out uniformly on the sphere (Wang & Isola, 2020). Contrastive pre-training task is therefore a very different methodology from masked image prediction. The above contrastive methods used only an online encoder along with a projection head to train the encoder. MoCo and variants (He et al., 2019; Chen et al., 2020c; He et al., 2020) added a momentum encoder, which is a moving average to the weights of the online encoder to perform contrasting between the online and momentum encoders. In general, the use of a momentum



(a) MoCo/BYOL (b) Masked MoCo/BYOL (ours)

Figure 1: Compared with traditional contrastive methods (a) that take full images as input, we adopt masking strategies from Masked Image Prediction (MIP) methods (b). This leads to better accuracy *and* efficiency. Note that (as with prior work) projection MLP is part of both online and target backbones.

encoder is a consistent way to provide a performance boost over methods that do not use it (Chen & He, 2021).

Other Pretext Tasks for Self-Supervised Representation Learning. Above, we have considered two major classes of pretext tasks for self-supervised representation learning: masked image predictions and contrastive learning. A disadvantage of SimClr (Chen et al., 2020b) and MoCo (He et al., 2019) is that they require the use of positives and negatives in the loss function, and sometimes many negatives are required in order to stabilize the loss function. Recent works have tried to do away with the use of negatives and instead use other methods of regularization such as the use of a prediction network (Grill et al., 2020), convariance-based regularizers (Bardes et al., 2021) and redundancy reduction (Zbontar et al., 2021). In this paper we build on top of both the MoCo-v3 (He et al., 2020) and BYOL (Grill et al., 2020) works.

3 METHOD

3.1 PRELIMINARY

The core idea of contrastive learning is simple. Given two random variables, discriminate between samples from the joint distribution and samples from the product of marginals. It is such a generic and flexible framework that a large variety of paired data can be modeled by it. This paper specifically focuses on visual representation learning.

MoCo-v3. We build on top of MoCo-v3, because of its good balance in simplicity and accuracy. We summarize MoCo-v3 as below. As shown in Fig. 1(a). Given an image, we apply random data augmentation to extract two crops. The first crop is encoded by an online encoder with output q; the second is encoded by a target encoder as k. The target encoder is usually an exponential moving average of the the online encoder f_q (He et al., 2020). Given the "query" q, contrastive learning is formulated as retrieving the corresponding "key" k out from a set of negatives k^- that come from other images in the same batch (and/or previous batches). Specially, the InfoNCE (Oord et al., 2018) contrastive loss function is:

$$\mathcal{L}_{\text{MoCo}}(q,k) = -\log \frac{\exp\left(q \cdot k/\tau\right)}{\exp\left(q \cdot k/\tau\right) + \sum_{k_-} \exp\left(q \cdot k^-/\tau\right)} \tag{1}$$

where $\tau > 0$ is a temperature hyperparameter that adjusts the peakiness of the distribution, and q and k are normalised to unit vectors before being used in Eq. 1.

BYOL (Grill et al., 2020). This approach drops the dispersion term in Eq.1 (*i.e.*, the denominator) and only considers maximizing the similarity of q and k that are from the same image. After q and k are normalized, the loss of BYOL is:

$$\mathcal{L}_{\text{BYOL}}(q,k) = ||q-k||_2^2 = 2 - 2 \cdot q \cdot k \tag{2}$$

To avoid collapse to a degenerate solution (*e.g.*, the network outputs a constant vector for any image), BYOL introduces an additional *predictor* to the online encoder, which makes the two encoders

Figure 2: Our mechanism to generate multiple masked views is shown above: we first shuffle the patches in an image and then partition the patch set into n equal length groups, $n = 4$ here.

asymmetric. While this is not strictly considered "contrastive learning", nevertheless we demonstrate in the paper that our strategies also work well for it.

MAE. MAE's are a form of Masked Image Prediction (MIP). Given an input image, the model only sees a sub-part of it and learns to recover the missing (masked) portions. While it is not straightforward to process a sub-part of an image with convolutional networks, the attention operator in Vision Transformers is more natural since it does not require the input to have specific spatial structure. Therefore in MAE, a random masking strategy is applied and the transformer encoder gracefully ignores the masked patches without bringing in large domain gaps between pre-training and fine-tuning (the main remaining gap is the sequence length). While MIP has achieved great success, it is mostly coupled with local objective functions, *e.g.*, predicting local features or pixels. On the other hand, how this masking strategy, combined with Vision Transformers (ViT), can benefit global objectives such as MoCo and BYOL is less studied. We investigate this question step by step in the following sections, and term our model as Masked MoCo or Masked BYOL.

Baseline setup. We conduct our exploration on ImageNet-100 (Deng et al., 2009) with the split from (Tian et al., 2019). Unless specified, our default setup follows MoCo-v3 as below:

- *Optimization*. We use the AdamW optimizer with momentum set as $\beta_1 = 0.9, \beta_2 = 0.95$. We use a learning rate of $blr \times$BatchSize/256, with the base rate $blr = 1.5e\text{-}4$. Weight decay (wd) is 0.1. We pre-train for 200 epochs with cosine learning rate decay schedule. Batch size is 1024.
- *Architecture*. We adopt ViT-S as our default backbone. We use the same Projector MLP and Predictor MLP as MoCo-v3. Both have BatchNorm (BN) layer applied to each fully-connected (fc) layer. For BYOL, we found it critical to remove the BN for the output fc of the predictor MLP. The projector and predictor have 3 and 2 layers, respectively. Both have a hidden dimension of 4096 and an output dimension of 256.

Evaluation. We evaluate the quality of the self-supervised learned representations by three standard approaches:

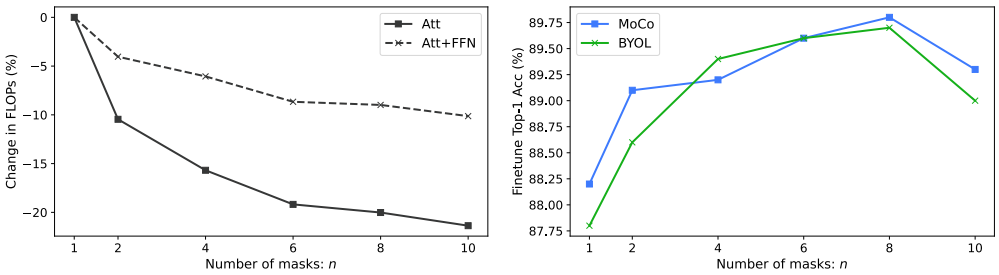
- *Fine-tuning*. We fine-tune the network with an appended classification head for 100-way classification. We follow the public repository of MAE[1]. We set blr as $5e\text{-}4$ with a wd of 0.05, and use layer-wise learning rate decay (Clark et al., 2020) of 0.65. We fine-tune for 100 epochs.
- *Linear probing*. We use LARS optimizer (You et al., 2019) with a blr of 1.0, weight decay = 0 (He et al., 2019), bsz = 4096. The linear head is trained for 100 epochs.

3.2 Masking on Online Encoder

The masking strategy is simple to implement. Given an image, we first convert it into a sequence of N patches (or patch embeddings), following prior works (Dosovitskiy et al., 2020). Then we randomly shuffle this sequence, and slice it into n chunks of equal length, as shown in Figure 2 (here $n = 4$). We append a `CLS` token to each chunk, and feed these chunks into the online transformer encoder as independent images. For the target encoder, we keep the full view (*i.e.*, the entire patch sequence) as the input. We then simply average the loss over the resulting multiple queries $\{q_i\}_{i=1}^n$ as:

$$\mathcal{L}_{\text{M-MoCo/BYOL}}(\{q_i\}, k) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{\text{MoCo/BYOL}}(q_i, k) \tag{3}$$

[1] https://github.com/facebookresearch/mae



(a) FLOPs (b) Fine-tuning Top-1

Figure 3: (a) As the partition parameter n is increased, the relative FLOP count of the Attention module steadily drops due to the quadratic complexity of the module; the overall relative saving for the full Transformer block (Att + FFN) is somewhat less, but still significant. (b) The absolute finetune top-1 accuracy steadily increases with n until about $n = 8$ for both MoCo-v3 and BYOL.

	Frozen PE	n	Finetune	Linear
MoCo-v3	✓	1	87.9	78.6
M-MoCo		1	88.2	78.7
		2	89.1	**79.4**
		4	89.2	79.2
		6	89.6	77.3
		8	**89.8**	75.7
		10	89.3	75.2

(a) Online masking w/ MoCo v3.

	Frozen PE	n	Finetune	Linear
BYOL	✓	1	87.3	77.5
M-BYOL		1	87.8	77.4
		2	88.6	**78.9**
		4	89.4	78.3
		6	89.6	77.3
		8	**89.7**	75.4
		10	89.0	74.1

(b) Online masking w/ BYOL.

Table 1: We perform masking for the **online** encoder, and keep the full view for the target encoder. As we increase the number of views n (each view sees $1/n$ of the input patches), the fine-tuning performance is improved until $n = 8$, while the linear accuracy peaks at $n = 2$. See (Chen et al., 2021) for frozen PatchEmbed. Numbers are for 200 epochs of pre-training.

When $n = 1$, this reduces to the standard MoCo or BYOL losses. Our encoder will see the same number of image patches as MoCo-v3 in each iteration. But now the ViT attention mechanism is restricted to operate within each chunk, which is $1/n$ of the full sequence. This reduces the computational complexity of cross-token inner product from $\mathcal{O}(N^2)$ to $\mathcal{O}(N^2/n)$. Practically, each transformer block consists of an attention (Att) module and a Feed-Forward Network (FFN) module, and a portion of the attention module computations are accelerated by our masking strategy. In Figure 3(a), we plot the relative change of FLOPs for the attention module and the complete transformer block (Att + FFN), as n increases.

We evaluate the pre-trained models and summarize the observations in Table 1. As a baseline, we re-implemented MoCo-v3 and BYOL, including the *Frozen Patch Embedding* (Frozen PE) trick (Chen et al., 2021). Fine-tuning them gives 87.9% and 87.3% top-1 accuracy, which are already higher than a supervised ViT-S trained from scratch, *i.e.*, 83.1%. Lifting the frozen PE restriction slightly improves the fine-tuning accuracy to 88.2% and 87.8%, respectively. We observe that: (1) all masking variants (n from 2 to 10) improve upon the fine-tuning accuracy of no masking (i.e. $n = 1$); (2) both M-MoCo and M-BYOL peak at $n = 8$, shown in Figure 3(b). Therefore we choose $n = 8$ as our default setup unless otherwise specified.

The masking strategy also improves the frozen representation *linear probing*. Different from fine-tuning, linear accuracy peaks at $n = 2$ and starts to decrease with larger n. This may be because the domain gap (mainly sequence length) increases, since larger n leads to shorter sequence during pre-training while the linear probing takes as input the full sequence. When $n > 4$, the linear accuracy drops below the baseline, indicating that optimizing linear accuracy requires different tuning.

3.3 Masking on Target Encoder

Similarly, we can apply the same masking strategy for the target encoder. For a controlled comparison with MoCo-v3, we first keep the full view (no slicing) for the online encoder, and only apply the masking strategy to the target encoder. Suppose we slice the target crop into m chunks, then we

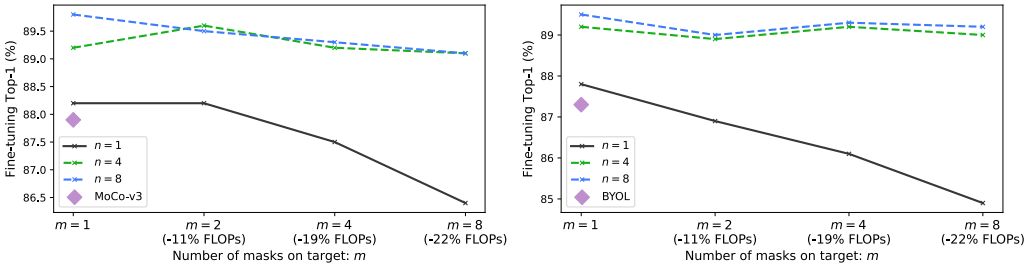
	Frozen PE	m	Finetune	Linear
MoCo v3	✓	1	87.9	78.6
M-MoCo		1	**88.2**	**78.7**
		2	88.2	74.1
		4	87.5	68.1
		8	86.4	64.6

(a) Target masking w/ MoCo v3.

	Frozen PE	m	Finetune	Linear
BYOL	✓	1	87.3	77.5
M-BYOL		1	**87.8**	**77.4**
		2	87.1	73.6
		4	86.1	67.3
		8	84.9	62.3

(b) Target masking w/ BYOL.

Table 2: We perform masking to the **target** encoder, and keep the full view for the online encoder. As we increase the number of target views m, both fine-tuning and linear accuracy monotonically decrease.



(a) M-MoCo (b) M-BYOL

Figure 4: We investigate how the *target* masking parameter m changes the fine-tuning performance. When online masking is on ($n > 1$), increasing the number of target masks m does not induce significant performance drop but can save training FLOPs for the target encoder (in addition to online encoder FLOP savings for $n > 1$).

obtain m key features $\{k_j\}_{j=1}^m$. There are two choices for the loss: (1) associate q with each k_j separately and average the loss; (2) associate q with the spherical mean of $\{k_j\}_{j=1}^m$. In our pilot study we found the latter one generally works better, so we present the results with it. Specifically,

$$\mathcal{L}_{\text{M-MoCo}}(q, \{k_j\}) = \mathcal{L}_{\text{M-MoCo}}(q, \hat{k}), \text{ where } \hat{k} = \frac{\sum_{i=1}^m k_i}{\|\sum_{i=1}^m k_i\|_2} \tag{4}$$

As shown in Table 2, both the fine-tuning and linear probing accuracy monotonically drop as we increase the number of masks m, *e.g.*, as we increase m from 1 to 8, fine-tuning drops 1.8% and linear drops 14.1% for M-MoCo, and 2.9% and 15.1% for M-BYOL. This is opposite to the result on the online encoder where masking helps. We conjecture that this is because the task becomes easier as m increases, since now the model is predicting a partial view by seeing the full view, rather than vice versa.

Finally, we investigate applying masking to both the online and target encoders. This is simply done by substituting k with $\hat{k}$ into Equation 3. For each online masking parameter n, we vary the target masking parameter m. We show the resulting plots in Figure 4. We find that, different from $n = 1$, turning on the online masking strategy ($n = 4$ or $n = 8$) enables the possibility of applying target masking without losing accuracy (or only marginal change). For example, when $n = 4$ and $n = 8$, increasing m from 1 to 8 only leads to 0.2% and 0.3% drop in accuracy for M-BYOL. This saves FLOPs for the target encoder by nearly 20%, and may potentially be helpful when distilling from large pre-trained foundation models (Wei et al., 2022b; Peng et al., 2022; Hou et al., 2022). For $m > 1$ and $n > 1$, FLOP savings accumulate on both encoders.

3.4 Replacing BatchNorm with LayerNorm

Batch normalization (BN) has played an important role in contrastive/siamese representation learning. It can enhance the dispersion between different images inside a batch (Chen et al., 2020a; Cai et al., 2021), or as an implicit repulsion loss (Chen & He, 2021). However, it can also create shortcuts for training if not carefully handled with small batch size (He et al., 2019), due to the gathering of cross-sample statistics.

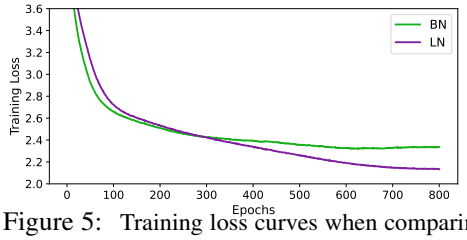


Figure 5: Training loss curves when comparing either BatchNorm (BN) or LayerNorm (LN) in the projection and prediction heads. LN leads to lower training loss.

Proj & Pred Heads	Fine-tuning	Linear
BatchNorm	91.5	76.5
LayerNorm	91.8	76.7
LayerNorm + CA	**91.9**	**78.7**

Table 3: Comparing fine-tuning and linear accuracy with different head designs. CA stands for additional cross-attention blocks (Touvron et al., 2021).

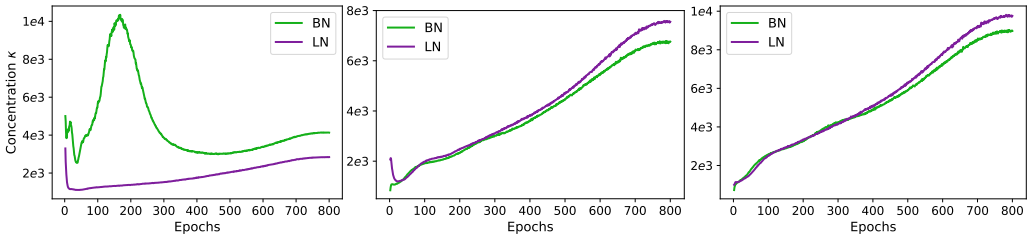


Figure 6: We monitor the concentration degree of features from the same image during training (see text). We observe that BatchNorm (BN) brings instability to the representations, which will be hidden after the projection layer. Instead LayerNorm (LN) is more stable.

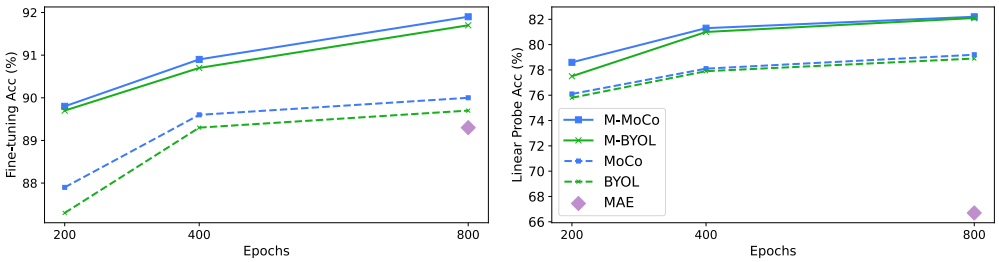
MoCo-v3 (Chen et al., 2021) uses BN, so M-MoCo naturally inherits this design in previous sections. We found that when we increase the training epochs, *e.g.*, from 200 to 800, sometimes training fails. The loss increases significantly at a random iteration and is not able to recover in the remainder of the trianing. We conjecture this is related to BN layers in the projection & prediction heads. Inspired by prior works (Richemond et al., 2020; Wu et al., 2022), we replace BN with Layer Normalization (LN), which is widely used in transformers. Figure 5 shows the training loss with BN and LN heads. With BN the loss converges faster at the beginning but plateaus or even increases at the end. Switching to LN leads to a smoother training curve and improved convergence. As shown in Table 3, the switch from BN to LN improves the fine-tuning accuracy by 0.3% and linear accuracy by 0.6%.

Monitoring training via a concentration parameter. For each image, we have multiple masks (chunks), yielding multiple output representations. We normalize these to unit vectors and treat them as samples from a von Mises–Fisher distribution (Gaussian-like on unit sphere). Then we estimate the concentration parameter κ via Banerjee et al. (2005). The greater the value of κ, the higher the concentration of the distribution. We consider three representations: (1) average pooling the output from the last transformer block (excluding `CLS`) and call it the "rep" layer; (2) the projection output; (3) the prediction output. As shown in Figure 6(a), the concentration degree of the "rep" layer is unstable with BN. The plot is averaged by epoch but in practice we observe significant instability across steps. Such instability is suppressed after projector and predictor, as seen in Figure 6(b, c). This makes it hard to detect this instability by observing the loss. Since there is no BN layer between the input and the "rep" layer, we conjecture that BN incurs such instability during back-propagation. We also observe that this instability becomes more severe with larger batch (κ can further increase by 5-10x), similar to observations in (Chen et al., 2021). In contrast, training with LN is more stable.

Extra cross-attention block. While using BN, we observe that blocks further from the BN layers suffer less from this instability. Therefore we insert two cross-attention blocks between ViT backbone and the projector, following (Wu et al., 2022) (but differently we keep `CLS` and thus make the backbone intact). This mitigates the instability of the "rep" layer. We also combine LN with this strategy to further stabilize the training, and marginally improve the accuracy (see Table 3).

3.5 Convergence Speed and Performance on 800 Epoch Pre-training

As our main goal is to improve fine-tuning performance, we choose the masking strategy that favors fine-tuning performance as our defaults, i.e., $n = 8$ and $m = 1$. We plot curves of accuracy v.s. epochs in Figure 7. For fine-tuning, M-MoCo and M-BYOL converge much faster, e.g., 200-epoch



(a) fine-tuning *v.s.* epochs (b) linear prob *v.s.* epochs

Figure 7: ImageNet-100 performance v.s. number of pre-training epochs. We see that M-MoCo and M-BYOL steadily increase in performance. We use $n = 8, m = 1$ for pre-training.

	Strong Aug.	Medium Aug.	Light Aug.
ViT-S	79.5	79.7	**79.9**

(a) ViT-S w/ different augmentations.

	Strong Aug.	Medium Aug.	Light Aug.
ViT-B	81.4	81.6	**81.7**

(b) ViT-B w/ different augmentations.

Table 4: We apply different augmentation strategies to ViT-S and ViT-B on ImageNet. ViT-S: pre-train and fine-tune 100 epochs; ViT-B: pre-train and fine-tune 60 epochs.

M-MoCo/BYOL is on par with 800-epoch MoCo/BYOL. With 800 epochs training, M-MoCo and M-BYOL reaches 91.9% and 91.7%, respectively, beating the unmasked counterparts by 1.9% and 2.0%. This improvement comes at the price of accuracy drop in linear probe. As shown in Table 1, fine-tuning and linear probe like different masking parameters. When comparing with MAE, M-MoCo increases the fine-tuning accuracy by 2.6%, and linear accuracy by 11.0%.

4 ImageNet and Transfer Learning

We pre-train our models on full ImageNet, and transfer them to ImageNet fine-tuning, COCO object detection (Lin et al., 2014), and ADE20K semantic segmentation (Zhou et al., 2017).

4.1 Data Augmentation

We study the effects of three different augmentation strategies for pre-training stage that may affect the final fine-tuning performance (the fine-tuning augmentation is kept unchanged):

- *Strong Augmentation*. This is the most widely used augmentation for contrastive learning (Chen et al., 2020a; Grill et al., 2020; Chen et al., 2021). For a given image, we independently extract two crops, and apply color augmentation (e.g., color jitter and gray scale) and Gaussian blur to them. Lastly we solarize one of the crop.
- *Medium Augmentation*. We extract a single 256x256 crop, and then get two 224x224 crops along the diagonal with a random distance between 0 and 32 pixels, partly inspired by (Huang et al., 2022) . Afterwards, we apply color augmentation to them.
- *Light Augmentation*. We only extract a shared single crop (Wu et al., 2022), and then apply color augmentation twice to get two versions.

We ablate these augmentations using M-MoCo with ViT-S (100 epochs pre-training and 100 epochs fine-tuning) and ViT-B (60 epochs pre-training and 60 epochs fine-tuning). As shown in Table 4, the light augmentation gets the best performance, outperforming strong augmentation by 0.4% for ViT-S and 0.3% for ViT-B. This differs from contrastive learning with ConvNets where a shared crop leads to shortcut during learning. This strongly suggests that random masking is a strong augmentation for contrastive learning.

4.2 Results on ImageNet-1K

Pre-training. We follow most of the settings on ImageNet-100, except that we increase the batch size to 4096. We train for 300/800 epochs with a warmup of 20/40 epochs. We use ViT-B/16 as backbone.

Fine-tuning results. The fine-tuning recipe is the same as MAE repo (see Section 3.1). As shown in Table 5, our M-MoCo rivals the performance of Masked Image Prediction methods, e.g., reaching 83.8% as SimMIM. M-MoCo also achieves best performance among contrastive learning methods, outperforming the baseline MoCo v3 by 0.6%. We note that MSN (Assran et al., 2022), trained with 600 epochs, also uses masked images as input but underperforms our M-MoCo with 300 epochs by 0.2%. We include a more complete comparison with other approaches in the appendix.

4.3 Transfer Learning

pre-train	AP^{box}	AP^{mask}
none (random init.)	48.1	42.6
IN-1k, supervised	47.6 (-0.5)	42.4 (-0.2)
MAE†	51.2 (+3.1)	45.5 (+2.9)
MoCo*	48.4 (+0.3)	42.9 (+0.3)
M-MoCo*	50.5 (+2.4)	44.8 (+2.2)

Table 6: Comparison of transfer learning on COCO benchmark. * our own run with 50 epochs fine-tuning; † 100 epochs fine-tuning.

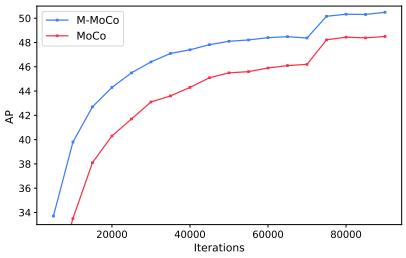


Figure 8: We compare M-MoCo and MoCo v3 by ploting their validation mAP during training.

Object detection and segmentation on COCO. We adopt the Mask-RCNN (He et al., 2017) framework for benchmarking this task, specifically ViTDet (Li et al., 2022). We fine-tune MoCo-v3 and M-MoCo for 50 epochs on COCO train2017 and report box AP and mask AP on the validation set. We directly use the optimization parameter provided by (Li et al., 2021). Consistent with (Li et al., 2021), we observe that MoCo-v3 performs at par with random initialization, while M-MoCo significantly outperforms it by 2.4 box mAP and 2.2 mask mAP, shown in Table 6. This resolves concerns, raised in (He et al., 2022) that contrastive pre-training can not help fine-tuning on COCO detection. M-MoCo still lags behind MAE by 0.7% box mAP and 0.6% mask mAP. We hope this gap can be narrowed by matching training epochs (now M-MoCo 50 v.s. MAE 100) and hyper-parameter searching. Comparison between M-MoCo and MoCo-v3 is shown in Figure 8.

Method	Objective	Epochs	Accuracy
Masked Image Prediction (MIP):			
BEiT	Local	800	83.2
SimMIM	Local	800	**83.8**
MAE	Local	1600	83.6
Contrastive or Siamese:			
DINO	Global	800	82.8
MoCo v3	Global	300	83.2
MSN	Global	600	83.4
M-MoCo (ours)	Global	300	83.6
M-MoCo (ours)	Global	800	**83.8**

Table 5: Compare with other methods on ImageNet fine-tuning benchmark with ViT-B.

Semantic Segmentation on ADE20k. Following prior work (Bao et al., 2021; He et al., 2022), we use UperNet (Xiao et al., 2018), and initialise ViT-B backbone from pre-training while other modules by random initialization. The training recipe in (He et al., 2022) is not publicly available so we end up with a shorter training recipe. We train for 80k iterations with a batch size of 16. Our MoCo-v3 run gets 46.6 mIoU (while in (He et al., 2022) MoCo-v3 achieves 47.3). Using our sub-optimal but head-to-head comparison, our M-MoCo improves upon MoCo-v3 by 1.0, reaching 47.6.

4.4 Limitations

Most of our experiments on ImageNet, MS COCO, and ADE20k datasets are one-shot training without hyper-parameter tuning, because of limitation on compute resources. This may lead to sub-optimal results. We have tried our best to keep an apples-to-apples comparison with MoCo-v3.

5 Conclusion

In this paper, we adopt the masking strategy from Masked Autoencoder to contrastive learning, and obtain a stronger baseline for fine-tuning and transfer learning. We hope this can help bridge the fine-tuning gap between contrastive learning and masked image prediction, and inspire further research.

REFERENCES

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. *arXiv preprint arXiv:2204.07141*, 2022. 9, 13

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv:1906.00910*, 2019. 2

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 2005. 7

Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 1, 2, 9, 13

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 3

Zhaowei Cai, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Zhuowen Tu, and Stefano Soatto. Exponential moving average normalization for self-supervised and semi-supervised learning. In *CVPR*, 2021. 6

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, 2021. 1, 2, 13

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv:2002.05709*, 2020a. 1, 2, 6, 8

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020b. 2, 3

Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. In *preprint arXiv:2202.03026*, 2022. 2, 13

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021. 3, 6

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv:2003.04297*, 2020c. 2

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 2, 5, 7, 8, 13

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020. 4

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pp. 886–893. Ieee, 2005. 2

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 2

Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu Peco. Perceptual codebook for bert pre-training of vision transformers. *arXiv preprint arXiv:2111.12710*, 2021. 13

Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Bootstrapped masked autoencoders for vision bert pretraining. *arXiv preprint arXiv:2207.07116*, 2022. 13

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 4

Yuxin Fang, Li Dong, Hangbo Bao, Xinggang Wang, and Furu Wei. Corrupted image modeling for self-supervised visual pre-training. *arXiv preprint arXiv:2202.03382*, 2022. 13

Peng Gao, Teli Ma, Hongsheng Li, Jifeng Dai, and Yu Qiao. Convmae: Masked convolution meets masked autoencoders. *arXiv preprint arXiv:2205.03892*, 2022. 13

Songwei Ge, Shlok Kumar Mishra, Haohan Wang, Chun-Liang Li, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. In *NeurIPS*, 2021. 2

Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2918–2928, 2021. 14

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2, 3, 8

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 9

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv:1911.05722*, 2019. 1, 2, 3, 4, 6

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 3

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 2, 9, 13

Zejiang Hou, Fei Sun, Yen-Kuang Chen, Yuan Xie, and Sun-Yuan Kung. Milan: Masked image pretraining on language assisted representation. *arXiv preprint arXiv:2208.06049*, 2022. 6, 13

Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *arXiv preprint arXiv:2207.13532*, 2022. 8, 13

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015. 2

Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021. 1, 2, 9, 14

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022. 9

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 8

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. 1, 2, 3

Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 6, 13

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021. 1, 13

Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020. 7

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *ICLR*, 2021. 2

Chenxin Tao, Xizhou Zhu, Gao Huang, Yu Qiao, Xiaogang Wang, and Jifeng Dai. Siamese image modeling for self-supervised vision representation learning. *arXiv preprint arXiv:2206.01204*, 2022. 13

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv:1906.05849*, 2019. 2, 4

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 2020. 2

Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? *arXiv preprint arXiv:2201.05119*, 2022. 1

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *CVPR*, 2021. 7

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008. 1, 2

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *IMCL*, 2020. 2

Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, 2022a. 1, 2, 13

Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Contrastive learning rivals masked image modeling in fine-tuning via feature distillation. *arXiv preprint arXiv:2205.14141*, 2022b. 6

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 1, 2

Zhirong Wu, Zihang Lai, Xiao Sun, and Stephen Lin. Extreme masking for learning instance and distributed visual representations. *arXiv preprint arXiv:2206.04667*, 2022. 7, 8, 13

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018. 9

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022. 1, 2, 13

Method	Objective	Epochs	Accuracy
Masked Image Prediction (MIP):			
BEiT (Bao et al., 2021)	Local	800	83.2
SimMIM (Xie et al., 2022)	Local	800	83.8
MAE (He et al., 2022)	Local	1600	83.6
CIM (Fang et al., 2022)	Local	300	83.1
PeCo (Dong et al., 2021)	Local	800	**84.5**
BootMAE (Dong et al., 2022)	Local	800	84.2
CAE (Chen et al., 2022)	Local	1600	83.8
ConvMAE (Gao et al., 2022)†	Local	1600	**85.0**
Masked Feature Prediction (MFP, a variant of MIP):			
MaskFeat (Wei et al., 2022a)	Local	1600	**84.0**
SIM (Tao et al., 2022)	Local	1600	83.8
Hybrid approach (integrating MIP with contrastive learning)			:
iBOT (Zhou et al., 2021)	Local & Global	1600	84.0
CMAE (Huang et al., 2022)	Local & Global	1600	**84.7**
Contrastive or Siamese:			
DINO (Caron et al., 2021)	Global	800	82.8
MoCo v3 (Chen et al., 2021)	Global	300	83.2
MSNZ (Assran et al., 2022)	Global	600	83.4
ExtreMA (Wu et al., 2022)	Global	300	83.7
M-MoCo (ours)	Global	800	**83.8**

Table 7: Compare with state-of-the-art self-supervised methods on ImageNet fine-tuning benchmark with ViT-B. There are other approaches, such as concurrent work BEiT v2 (Peng et al., 2022) and MILAN (Hou et al., 2022), that obtain better fine-tuning performance by distilling from CLIP models (Radford et al., 2021). For a head-to-head comparisons, we leave the out of the table. † a convolution-transformer hybrid architecture is used.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019. 4

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021. 3

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 8

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 13

A Appendix

A.1 Comprehensive Comparison with State-of-the-art Methods

While our goal is to establish an improved contrastive baseline, rather than to beat the best numbers, we compare our approach with other state of the art in Table 7.

A.2 Training details on ImageNet

The training recipe and hyper-parameter details are inlcuded in Table 8.

config	Pre-training	Fine-tuning
Augmentation	Light Augmentation (See Section 4.1)	RandAug, mixup, cutmix
Opitmizer	AdamW	AdamW
base learning rate	1.5e-4	5e-4
weight decay	0.1	0.05
optimizer momentum	0.9, 0.95	0.9, 0.999
layer-wise lr decay	none	0.65
batch size	4096	1024
learning rate schedule	cosine	cosine
warmup epochs	20(300), 40(800)	5
training epochs	300/800	100
drop path	0	0.1

Table 8: Hyper-parameters used in ImageNet-1k pre-training and fine-tuning.

A.3 THE COCO DETECTION

config	Detection Fine-tuning
Augmentation	LSJ (Ghiasi et al., 2021)
Opitmizer	AdamW
learning rate	1.6e-4
weight decay	0.1
optimizer momentum	0.9, 0.999
layer-wise lr decay	0.7
batch size	64
learning rate schedule	MultiStepDecay
warmup epochs	0.25
training epochs	50
drop path	0.1

Table 9: Hyper-parameters used by fine-tuning on COCO. We obtain such parameter from (Li et al., 2021) without tuning.

The fine-tuning hyper-parameters are inherited from (Li et al., 2021), and are listed in Table 9. Note that because of limited resources we did not tune it, but observe that such parameter works well with both MoCo-v3 and our M-MoCo.