DOES SELF-SUPERVISED CONTRASTIVE LEARNING REALLY NEED LARGE QUANTITIES OF NEGATIVE PAIRS?

Anonymous authors
Paper under double-blind review

ABSTRACT

It has been a conventional wisdom that enlarging the scale of negative samples is essential to obtain competitive results in self-supervised contrastive learning recently. In this paper, we challenge the viewpoint because the optimal hyper-parameters (e.g. margin and learning rate) might vary with the number of contrastive pairs, however, which are not properly evaluated in recent works. Inspired by the InfoMax principle, we point that the margin term in contrastive loss needs to be adaptively scaled according to the number of negative pairs, in order to keep steady mutual information bound and gradient magnitude; in addition, the learning rate should also grow linearly with the number of positive queries per batch. The analysis thus derives **Eq**uivalent Rules for **Co**ntrastive Learning (EqCo), bridging the performance gap among a wide range of negative sample sizes. EqCo can be applied to state-of-the-art contrastive learning frameworks such as MoCo and SimCLR, which is demonstrated to obtain high performances using only a few negative samples. For example, with the help of EqCo, our simplified MoCo (SiMo) achieves comparable accuracy with MoCo v2 on ImageNet (linear evaluation protocol) while only involves 16 negative pairs per query instead of 65536, suggesting that large quantities of negative samples might not be a critical factor in contrastive learning frameworks.

1 INTRODUCTION AND BACKGROUND

Self-supervised learning has recently received much attention in the field of visual representation learning (Hadsell et al. (2006); Dosovitskiy et al. (2014); Oord et al. (2018); Bachman et al. (2019); Hénaff et al. (2019); Wu et al. (2018); Tian et al. (2019); He et al. (2020); Misra & Maaten (2020); Grill et al. (2020); Cao et al. (2020)), as its potential to learn universal representations from unlabeled data. Among various of self-supervised methods, one of the most promising research paths is *contrastive learning* (Oord et al. (2018)), which has been demonstrated to achieve comparable or even better performances than supervised training for many downstream tasks such as image classification, object detection and semantic segmentation (Chen et al. (2020b); He et al. (2020); Chen et al. (2020a)). The core idea of contrastive learning is briefly summarized as follows: first, extracting a pair of embedding vectors $(\mathbf{q}(x), \mathbf{k}(x))$ (named *query* and *key* respectively) from the two augmented views of each instance x; then, learning to maximize the similarity of each *positive pair* $(\mathbf{q}(x), \mathbf{k}(x))$ while pushing the *negative pairs* $(\mathbf{q}(x), \mathbf{k}(x'))$ (i.e. query and key extracted from different instances accordingly) away from each other. To optimize, an *InfoNCE loss* (Oord et al. (2018); Wu et al. (2018)) is conventionally employed in the following formulation (slightly modified with an additional *margin* term):

$$\mathcal{L}_{NCE} = \mathop{\mathbb{E}}_{\mathbf{q}\sim\mathcal{D}, \mathbf{k}_0\sim\mathcal{D}'(\mathbf{q}), \mathbf{k}_i\sim\mathcal{D}'} \left[-\log \frac{e^{(\mathbf{q}^\top \mathbf{k}_0 - m)/\tau}}{e^{(\mathbf{q}^\top \mathbf{k}_0 - m)/\tau} + \sum_{i=1}^{K} e^{\mathbf{q}^\top \mathbf{k}_i/\tau}} \right], \tag{1}$$

where $\mathbf{q}$ and $\mathbf{k}_i$ $(i = 0, \ldots, K)$ stand for the query and keys sampled from the two (augmented) data distributions $\mathcal{D}$ and $\mathcal{D}'$ respectively. Specially, $\mathbf{k}_0$ is associated to the same instance as $\mathbf{q}$'s while other $\mathbf{k}_i$s not, hence we name $\mathbf{k}_0$ and $\mathbf{k}_i$ $(i > 0)$ *positive sample* and *negative samples* respectively in the remaining text, in which K is the number of negative samples (or pairs) for each query. The

temperature τ and the *margin* m are hyper-parameters. In most previous works, m is trivially set to zero (e.g. Oord et al. (2018); He et al. (2020); Chen et al. (2020a)) or some handcraft values (e.g. Cao et al. (2020)).

In contrastive learning research, it has been a conventional wisdom that enlarging the number of negative samples K boosts the performance (Hénaff et al. (2019); Tian et al. (2019); Bachman et al. (2019)). For example, in MoCo (He et al. (2020)) the ImageNet accuracy rises from 54.7% to 60.6% under linear classification protocol when K grows from 256 to 65536. Such observation further drives a line of studies how to effectively optimize under a number of negative pairs, such as *memory bank* methods (Wu et al. (2018); He et al. (2020)) and large batch training (Chen et al. (2020a)), either of which empirically reports superior performances when K becomes large. Analogously, in the field of *supervised metric learning* (Deng et al. (2019); Wang et al. (2018); Sun et al. (2020); Wang et al. (2020)), loss in the similar form as Eq. 1 is often applied on a lot of negative pairs for hard negative mining. Besides, there are also a few theoretical studies supporting the viewpoint. For instance, Oord et al. (2018) points that the mutual information between the positive pair tends to increase with the number of negative pairs K; Wang & Isola (2020) find that the negative pairs encourage features' uniformity on the hypersphere; Chuang et al. (2020) suggests that large K leads to more precise estimation of the debiased contrastive loss; etc.

Despite the above empirical or theoretical evidences, however, we point that the reason why to use many negative pairs is still less convincing. **First**, unlike the metric learning mentioned above, in self-supervised learning the negative terms $\mathbf{k}_i$ in Eq. 1 include both "true negative" (whose underlying class label is different from the query's, similarly hereinafter) and "false negative" samples, since the actual ground truth label is not available. So, intuitively large K should not be always beneficial because the risk of false negative samples also increases (known as *class collision* problem). Arora et al. (2019) thus theoretically concludes that a large number of negative samples could not be necessarily help. **Second**, empirical studies like Chen et al. (2020a); He et al. (2020) show that if K exceeds some "sweet spot", the performance starts to saturate or even degrade, indicating the optimal choice of K could be finite. **Last** but not least, a recent work *BYOL* (Grill et al. (2020)) suggests that carefully designing the encoder and using a proper solver (i.e. *mean teacher* (Tarvainen & Valpola (2017))) instead of InfoNCE loss, state-of-the-art performance can still be obtained even without any explicit negative pairs. In conclusion, it is still an open question whether large quantities of negative samples are essential to contrastive learning.

After referring to the above two aspects, in this paper, we propose to rethink the effect of large quantities of negative pairs from a different view: *what if the optimal hyper-parameters (e.g.* τ *and* m *in Eq. 1) change with different* K*s*? We notice that in most of previous empirical studies (He et al. (2020); Chen et al. (2020a)), those hyper-parameters are usually kept constant under different number of negative samples. Hence if our conjecture is true, the performance gains observed from larger Ks may be a *false interpretation* – simply brought by suboptimal hyper-parameters' choices for small Ks, rather than much of an essential.

However, to examine the optimal hyper-parameters for a variety of Ks one by one is tedious and might not generalize well to other training settings. Instead, we investigate the *equivalent* relationship between hyper-parameters' choices under different numbers of negative pairs, hence we can transfer the optimal choice across multiple configurations. Following the "equivalant" methodology as well, the well-known *linear scaling rule* (Krizhevsky (2014); Goyal et al. (2017)) gives a good practice, which suggests to scale the learning rate proportional to the batch size if the loss satisfies with the linear averaged form: $L = \frac{1}{N}\sum_{i=1}^{N} f(x_i;\theta)$. But, we have to point that *linear scaling rule* is inferior to be directly applicable on InfoNCE loss (Eq. 1), which is partially because InfoNCE loss includes two batch sizes (number of *queries* and *keys* respectively) while linear scaling rule only involves one, in addition to the nonlinearity of the keys in InfoNCE loss. In the experiments of *SimCLR* (Chen et al. (2020a)), learning rates under different batch sizes are adjusted with linear scaling rule (or square root rule for LARS (You et al. (2017)) optimizer), but the accuracy gap is still very large (57.5%@batch=256 vs. 64+%@batch=8192).

Our study finally derives the **Eq**uivalent Rules for **Co**ntrastive learning (*EqCo*) as follows. Following the *InfoMax principle* (Linsker (1988)) and the derivation in *CPC* (Oord et al. (2018)), we find that if the *margin* m in Eq. 1 is adaptively adjusted in $m = \tau \ln \frac{\alpha}{K}$ (where α is a constant), the lower bound of the mutual information keeps steady under various numbers of negative samples. Moreover, from back-propagation perspective, we further prove that in such configuration the upper

bound of the gradient norm is also free of K's scale. The proposed equivalent rule implies that, by assigning $\alpha = K_0$, it can "mimic" the optimization behavior under K_0 negative samples even if the physical number of negatives $K \neq K_0$. Our empirical studies further verify the correctness, for example, in MoCo framework, by introducing EqCo the performance gap between $K = 256$ and $K = 65536$ (the best configuration reported in He et al. (2020)) is *almost eliminated* (from 6.1% decrease to 0.16%). For completeness, as the other part of *EqCo* we point that adjusting the learning rate according to the conventional *linear scaling rule* satisfies the equivalence for different number of *queries* per batch.

EqCo challenges the convention that self-supervised contrastive learning requires large quantities of negative pairs to obtain competitive performance, which makes it possible to design simpler algorithms. We thus propose *SiMo*, a simplified contrastive learning framework based on *MoCo v2* (Chen et al. (2020b)). SiMo is elegant, efficient, free of large batch training and memory bank; moreover, it is able to obtain superior performances over the state-of-the-arts even if the number negative pairs is very small (e.g. 16), without bells and whistles.

To conclude, the contributions of our paper are summarized as follows:

- We challenge the widely accepted cognition that large size of negative samples is critical for contrastive learning from a different view: it may be a false interpretation as the hyper-parameters are not aligned to the optimal under various configurations.
- We propose EqCo, an equivalent rule to transfer hyper-parameters between small and large numbers of negative samples, which proves to bridge the performance gap.
- We present SiMo, a simpler but stronger baseline for contrastive learning.

2 EqCo: Equivalent Rules for Contrastive Learning

In this section we introduce *EqCo*. We mainly consider the circumstance of optimizing the *InfoNCE loss* (Eq. 1) with *SGD*. For each batch of training, there are two meanings of the concept "batch size", i.e. the size of *negative samples/pairs* K per query, and the number of *queries* (or *positive pairs*) N per batch. Hence our equivalent rules accordingly consist of two parts, which will be introduced in the next subsections.

2.1 The Case of Negative Pairs

Our derivation is mainly inspired by the model of *Contrastive Predictive Coding (CPC)* (Oord et al. (2018)), in which *InfoNCE loss* (Eq. 1) is interpreted as a mutual information estimator. We further extend the method so that it is applicable to InfoNCE loss with a *margin* term (Eq. 1), which is not considered in Oord et al. (2018).

Following the concept in Oord et al. (2018), given an *query* embedding $\mathbf{q}$ as the context and suppose $K+1$ random *key* embeddings $\mathbf{x} = \{\mathbf{x}_i\}_{i=0,\dots,K}$, where there exists exact one entry (e.g. $\mathbf{x}_i$) sampled from the conditional distribution $\mathrm{P}(\mathbf{x}_i|\mathbf{q})$ while others (e.g. $\mathbf{x}_j$) sampled from the "proposal" distribution $\mathrm{P}(\mathbf{x}_j)$ independently. According to which entry corresponds to the conditional distribution, we therefore defines $K+1$ *candidate* distributions for $\mathbf{x}$ (denoted by $\{H_i\}_{i=0,\dots,K}$), where the probability density of $\mathbf{x}$ under H_i is $\mathrm{P}_{H_i}(\mathbf{x}) = \mathrm{P}(\mathbf{x}_i|\mathbf{q}) \prod_{j \neq i} \mathrm{P}(\mathbf{x}_j)$. So, given the observed data $X = \{\mathbf{k}_0, \dots, \mathbf{k}_K\}$ of $\mathbf{x}$, the probability where $\mathbf{x}$ is sampled from H_0 rather than other candidates is thus derived with Bayes theorem:

$$
\begin{aligned}
\Pr[\mathbf{x} \sim H_0 | \mathbf{q}, X] &= \frac{\mathrm{P}^+ \mathrm{P}_{H_0}(X)}{\mathrm{P}^+ \mathrm{P}_{H_0}(X) + \mathrm{P}^- \sum_{i=1}^{K} \mathrm{P}_{H_i}(X)} \\
&= \frac{\frac{\mathrm{P}^+}{\mathrm{P}^-} \frac{\mathrm{P}(\mathbf{k}_0|\mathbf{q})}{\mathrm{P}(\mathbf{k}_0)}}{\frac{\mathrm{P}^+}{\mathrm{P}^-} \frac{\mathrm{P}(\mathbf{k}_0|\mathbf{q})}{\mathrm{P}(\mathbf{k}_0)} + \sum_{i=1}^{K} \frac{\mathrm{P}(\mathbf{k}_i|\mathbf{q})}{\mathrm{P}(\mathbf{k}_i)}},
\end{aligned}
\tag{2}
$$

where we denote P^+ and P^- as the *prior* probabilities of H_0 and $H_i (i > 0)$ respectively. We point that Eq. 2 introduces a *generalized form* to that in Oord et al. (2018) by taking the priors into account. Referring to the notations in Eq. 1, we suppose that H_0 is the ground truth distribution of $\mathbf{x}$ (since $\mathbf{k}_0$

is the only positive sample). By modeling the density ratio $\mathrm{P}(\mathbf{k}_i|\mathbf{q})/\mathrm{P}(\mathbf{k}_i) \propto e^{\mathbf{q}^\top \mathbf{k}_i}(i = 0, \ldots, K)$ and letting $\mathrm{P}^+/\mathrm{P}^- = e^{-m/\tau}$, the negative log-likelihood $\mathcal{L}_{opt} \triangleq \mathbb{E}_{\mathbf{q},X} \; -\log \Pr[x \sim H_0|\mathbf{q}, X]$ is the optimal value of $\mathcal{L}_{NCE}$.

Similar to the methodology of Oord et al. (2018), we explore the lower bound of $\mathcal{L}_{opt}$:

$$
\begin{aligned}
\mathcal{L}_{opt} &= \mathop{\mathbb{E}}_{\mathbf{q}\sim\mathcal{D},\mathbf{k}_0\sim\mathcal{D}'(\mathbf{q}),\mathbf{k}_i\sim\mathcal{D}'} \log\left(1 + e^{m/\tau}\frac{\mathrm{P}(\mathbf{k}_0)}{\mathrm{P}(\mathbf{k}_0|\mathbf{q})}\sum_{i=1}^{K}\frac{\mathrm{P}(\mathbf{k}_i|\mathbf{q})}{\mathrm{P}(\mathbf{k}_i)}\right) \\
&\approx \mathop{\mathbb{E}}_{\mathbf{q}\sim\mathcal{D},\mathbf{k}_0\sim\mathcal{D}'(\mathbf{q})} \log\left(1 + Ke^{m/\tau}\frac{\mathrm{P}(\mathbf{k}_0)}{\mathrm{P}(\mathbf{k}_0|\mathbf{q})}\left(\mathop{\mathbb{E}}_{\mathbf{k}_i\sim\mathcal{D}'}\frac{\mathrm{P}(\mathbf{k}_i|\mathbf{q})}{\mathrm{P}(\mathbf{k}_i)}\right)\right) \\
&= \mathop{\mathbb{E}}_{\mathbf{q}\sim\mathcal{D},\mathbf{k}_0\sim\mathcal{D}'(\mathbf{q})} \log\left(1 + Ke^{m/\tau}\frac{\mathrm{P}(\mathbf{k}_0)}{\mathrm{P}(\mathbf{k}_0|\mathbf{q})}\right) \\
&\geq \log K + m/\tau - \mathcal{I}(\mathbf{k}_0, \mathbf{q})
\end{aligned}
\tag{3}
$$

We start by reviewing the contrastive loss and its properties. Briefly, $\boldsymbol{x}_q$ and $\boldsymbol{x}_{k_0}$ are two augmented views originated from a given image $\boldsymbol{x}$. With predefined encoders, $\boldsymbol{x}_q$ and $\boldsymbol{x}_{k_0}$ are mapped to feature vectors $\boldsymbol{q}$ and $\boldsymbol{k}_0$ respectively. In addition, we randomly sample K other images as negatives and map them to feature vectors set $\{k_i, i = 1, ..., K\}$. Note that all these feature vectors are normalized by their L2-norm, so the inner product of two vectors equals to the cosine similarity between them. Following InfoNCE [cpc], the loss function is given by:

$$
\mathcal{L} = -log\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau) + \sum_{i=1}^{K} exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)} \tag{4}
$$

Here τ is a temperature hyper-parameter. According to CPC [cpc], Equation (4) is related to the mutual information between $\boldsymbol{q}$ and $\boldsymbol{k}_0$:

$$
\mathcal{I}(\boldsymbol{q}, \boldsymbol{k}_0) \geq log(K+1) - \mathbb{E}_{(\boldsymbol{q},\boldsymbol{k}_0)\sim\mathbb{P}}[\mathcal{L}] \tag{5}
$$

As we can see in Inequality (5), $log(K+1) - \mathbb{E}_{(\boldsymbol{q},\boldsymbol{k}_0)\sim\mathbb{P}}[\mathcal{L}]$ is the lower bound of the mutual information. And this inequality theoretically explains the phenomenon that the performance of contrastive learning based on Equation (4) largely depends on K. From this perspective, one can easily draw a conclusion that the larger the K is, the better the performance will be. Nonetheless, the experiments reported in MoCo [moco] and SimCLR [simclr] does not support this conclusion well. There are performance stagnancy or degradation as K is larger than some certain "sweet spot". A plausible explanation for this is that logarithmic function saturates at large K (e.g., 65536), and the loss will also increase as we enlarge K, making it hard for optimization.

Need some experiments to exemplify this opinion! To better understand contrastive learning, we formulate it as a pair-based metric learning problem with noisy labels which refer to the inconsistent labels of semantically similar images. To our knowledge, enlarging the size of mini-batch essentially boosts the performance of pair-based metric learning for there are more opportunities of mining hard negative samples. Analogously, it is natural to enlarge K in contrastive learning. But on account of the noisy labels, hard negative samples contain a part of "false negative" samples, which actually share similar semantic information with $\boldsymbol{x}_q$. By optimizing contrastive loss, these similar pairs are forced to isolate from each other, hurting the consistency of learned representation. Weigh the pros and cons, we doubt the necessity of large K in contrastive learning.

The other problem of the existing contrastive loss lies in the process of optimization. Let us consider the state-of-the-art contrastive learning framework, MoCo v1/v2 [moco v1/v2]. All $\boldsymbol{k}_i$ (including $\boldsymbol{k}_0$) are detached from computational graph, so the gradients with respect to the weights of neural network are only back-propagated by $\boldsymbol{q}$. The gradient with respect to $\boldsymbol{q}$ is written as:

$$
\frac{d\mathcal{L}}{d\boldsymbol{q}} = -\frac{1}{\tau}(1 - \frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{Z})\boldsymbol{k}_0 + \frac{1}{\tau}\sum_{i=1}^{K}\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)}{Z}\boldsymbol{k}_i \tag{6}
$$

where Z is the normalization factor $exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)+\sum_{i=1}^{K} exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)$. Observing Equation (6), it is easy to check that the gradient with respect to $\boldsymbol{q}$ is dependent on K. However, previous studies on contrastive learning ignore this, and apply the same optimization hyper-parameters (e.g., learning rate, weight decay, momentum and so on.) under all circumstances. To maintain performances across a much broader range of K, some hyper-paramter adjustment need to be done.

Optimization hyper-parameters tuning is not new in deep learning. Linear scaling rule [linear rule] is designed for adapting to different sizes of mini-batch. Unfortunately, it may fail to solve this problem due to the non-linear correlation between the gradients and K. Consider a complex example that not only K but also the size of mini-batch varies (In those contrastive learning frameworks [simclr] that use samples of current mini-batch as negatives, it happens when K changes). It may cause confusion when applying the linear scaling rule. Therefore, we need another scaling rule that is disentangled from the linear scaline rule.

2.2 Equal Contrastive Loss

As shown in the analysis above, the problem that K influences the lower bound of mutual information and the optimization behaviour stems from a prior: probability that $\boldsymbol{k}_0$ is chosen as the positive among the feature set $\mathbb{K} = \{\boldsymbol{k}_i, i = 0, 1, ..., K\}$ varies with K. To reduce the dependence of contrastive learning on K, we introduce a prior that the probability of describing $\boldsymbol{k}_i, i \neq 0$ is sampled from conditional distribution $p(\boldsymbol{k}|\boldsymbol{q})$ and other features $\boldsymbol{k}_j, j \neq i$ are sampled from the marginal distribution $p(\boldsymbol{k})$ is α/K times the probability of describing $\boldsymbol{k}_0$ is sampled from conditional distribution $p(\boldsymbol{k}|\boldsymbol{q})$ and other features $\boldsymbol{k}_j, j \neq 0$ are sampled from the marginal distribution $p(\boldsymbol{k})$. α is a predefined constant. Then the posterior probability that given query q, and a feature set $\mathbb{K}$, $\boldsymbol{k}_0$ is drawn from the conditional distribution $p(\boldsymbol{k}|\boldsymbol{q})$ (i.e., the distribution of positive) rather than the marginal distribution $p(\boldsymbol{k})$ (i.e., the distribution of negatives) is derived as follows:

$$\begin{aligned} p(pos = \boldsymbol{k}_0|\mathbb{K}, \boldsymbol{q}) &= \frac{p(\boldsymbol{k}_0|\boldsymbol{q})\Pi_{i\neq 0}p(\boldsymbol{k}_i)}{p(\boldsymbol{k}_0|\boldsymbol{q})\Pi_{i\neq 0}p(\boldsymbol{k}_i) + \frac{\alpha}{K}\sum_{i=1}^{K} p(\boldsymbol{k}_i|\boldsymbol{q})\Pi_{j\neq i}p(\boldsymbol{k}_j)} \\ &= \frac{\frac{p(\boldsymbol{k}_0|\boldsymbol{q})}{p(\boldsymbol{k}_0)}}{\frac{p(\boldsymbol{k}_0|\boldsymbol{q})}{p(\boldsymbol{k}_0)} + \frac{\alpha}{K}\sum_{i=1}^{K}\frac{p(\boldsymbol{k}_i|\boldsymbol{q})}{p(\boldsymbol{k}_i)}} \end{aligned} \tag{7}$$

Then the Equal Constrastive Loss (EqCo) is defined as:

$$\mathcal{L}^* = -log\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau) + \frac{\alpha}{K}\sum_{i=1}^{K} exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)} \tag{8}$$

It is easy to establish the relation between EqCo and mutual information:

$$\mathcal{I}(\boldsymbol{q}, \boldsymbol{k}_0) \geq log(\alpha) - \mathbb{E}_{(\boldsymbol{q},\boldsymbol{k}_0)\sim\mathbb{P}}[\mathcal{L}^*] \tag{9}$$

As we can see in Inequality (9), the mutual information is instead bounded by a predefined constant α, making it efficient for increasing the mutual information. Of course, EqCo does not provide a better lower bound of mutual information. Optimization difficulty also arises when α becomes large. But EqCo liberates contrastive learning from the curse of requiring tens of hundreds of negative samples. More importantly, we show below that the optimization behaviour is very consistent for different K. The gradient of $\mathcal{L}^*$ with respect to $\boldsymbol{q}$ is given by:

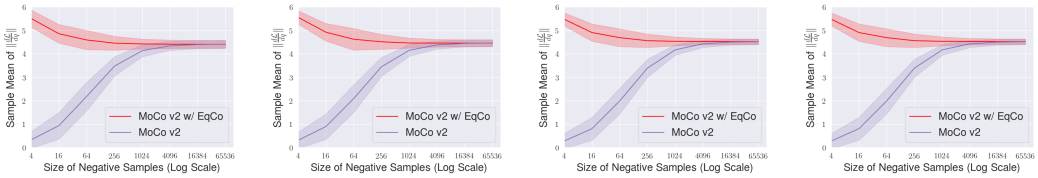
$$\frac{d\mathcal{L}^*}{d\boldsymbol{q}} = -\frac{1}{\tau}(1 - \frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{Z^*})\boldsymbol{k}_0 + \frac{1}{\tau}\frac{\alpha}{K}\sum_{i=1}^{K}\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)}{Z^*}\boldsymbol{k}_i \tag{10}$$

where Z^* is the normalization factor $exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau) + \alpha/K\sum_{i=1}^{K} exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)$. Suppose $\{\boldsymbol{k}_i, i = 1, ..., K\}$ are K i.i.d (independent and identically distributed) random variables. Given feature $\boldsymbol{q}$, the

expectation of $exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)$ exists since the similarity $\boldsymbol{q}^T\boldsymbol{k}_i$ is bounded in $[-1, 1]$. From Jensen's Inequality and Triangle Inequality, the upper bound of the norm of gradient is:

$$\mathbb{E}_{\boldsymbol{k}_i}\|\frac{d\mathcal{L}}{d\boldsymbol{q}}\| \leq \frac{2}{\tau}(1 - \frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau) + \alpha\mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]}) \tag{11}$$

The proof is detailed in Appendix. The Inequality (11) shows that the upper bound of this expectation is relevant to α but not to K. Interestingly, as shown in Figure **??**, we find that there is little deviation between this lower bound and the sample mean when K is large enough to offset the sampling bias (e.g., 16).



(a) @50 epochs (b) @100 epochs (c) @150 epochs (d) @200 epochs

Figure 1: The sample mean of $\|\frac{d\mathcal{L}}{d\boldsymbol{q}}\|$. We train a normal

2.3 SIMPLER MOCO

Simpler MoCo (SiMo) is a simpler but more effective contrastive learning framework, which is modified based on MoCo [moco v1/v2]. To elaborate on SiMo, let us quickly review the three key parts of MoCo v2. First, there are two encoders to map images to features. One is query encoder, which encodes the query image $\boldsymbol{x}_q$ as feature $\boldsymbol{q}$ and is updated by gradients. The other is key encoder, which encodes the key image $\boldsymbol{x}_{k_0}$ as feature $\boldsymbol{k}_0$ and is updated by the following rule:

$$\theta_k = m\theta_k + (1 - m)\theta_q \tag{12}$$

where θ_q and θ_k are the parameters of query encoder and key encoder respectively, and $m \in [0, 1]$ is the momentum coefficient that controls how fast the key encoder move to query encoder. Second, to enlarge the size of negative samples while keeping the consistency of negative features, a queue-like meomry bank that stores the features seen in previous iterations is proposed in MoCo v1 [moco v1]. Last but not least, shuffling BN that shuffles the sample order of the batch of key images is designed to prevent information leakage by BN statistics. As for SiMo, momentum update is preserved, but the memory bank is deprecated. Moreover, we use sync BN to prevent information leakage instead of shuffling BN. The reasons for the structure of SiMo are as follows.

Momentum update. As mentioned in MoCo v1 [moco v1], momentum update is designed to address the issue that the consistency of features $\{\boldsymbol{k}_i, i = 0, 1, ..., K\}$ is reduced by the rapid change of key encoder. It is worth noting that the representation learned in the case of $m = 0$ fails to transfer to linear evaluation. Owing to the queue-like memory bank, $\boldsymbol{k}_0$ is more "fresher" than $\boldsymbol{k}_i$. We believe that it is the discordance between the distribution of $\boldsymbol{k}_0$ and the distribution of $\boldsymbol{k}_i$ that causes this failure (The full explanation is given in Appendix.). Though such discordance is vanished in SiMo because of the deprecation of memory bank, we find it conducive to preserve momentum update after our comparative experiments.

Memory bank. Since large size of negatives is no longer needed in contrastive learning, it is natural to cancel the memory bank proposed in MoCo v1. In SiMo, we only use the samples of the current mini-batch as negatives. Specifically, in the mini-batch of key images, one is the corresponding positive; the others are the negative.

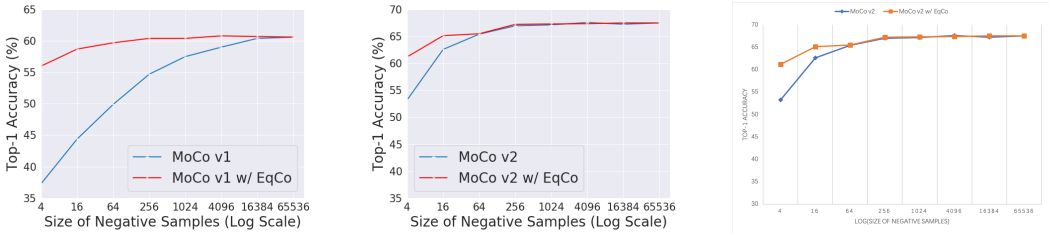
Sync BN. In MoCo v1/v2, shuffling BN is proposed to remove the obvious dissimilarities of the BN (Ioffe & Szegedy, 2015) statistics between the positive (from current mini-batch) and the nega-

tives (from memory bank), such that the model can make predictions based on the semantic information of images rather than the BN statistics. In contrary, the positive and negatives are from the same mini-batch in SiMo. As shown in Table x, when computing the BN statistics with large batch size, the performance is much better than its couterpart with small batch size. Therefore, using sync BN is a better choice for SiMo.

2.4 EXPERIMENTS FOR EQCO

To evaluate EqCo, we apply it to three representative contrastive learning frameworks: MoCo v1, MoCo v2, and SimCLR, and report the results of classification under linear evaluation protocol.

Implementation details. For all experiments, the backbone of encoder is ResNet-50 (He et al., 2016), and we train it for 200 epochs with mini-batch size of 256 in 8 GPUs. Most of the training hyper-parameters are the same as reported in their own papers. For the case that K is smaller than the size of mini-batch, K instances from the current mini-batch are randomly sampled as the negatives (SimCLR) or to be enqueued (MoCo v1/v2). In SimCLR, we adopt SGD as our optimizer, instead of LARS. For experiments with EqCo, We set $\alpha = 65536$ in MoCo v1/v2, while $\alpha = 256$ in SimCLR. The top-1 accuracy when varying K is plotted below.



(a) MoCo v1 (b) MoCo v2 (c) SimCLR

Figure 2: ImageNet top-1 accuracy under linear evaluation protocol. We evaluate the effectiveness of EqCo on three contrastive learning frameworks: MoCo v1, MoCo v2, and SimCLR.

As shown in Fig.2(a) and 2(b), the "sweet spots" (i.e., the saturation point of performance) of MoCo v1 and MoCo v2 are different (4096 for MoCo v1, and 256 for MoCo v2). We speculate that it is the larger temperature of MoCo v2 that causes this difference. According to InforNCE (Eq.1), when the temperature is small, the effect of hard mining is amplified, and the representations learned with larger K is more competitive. If we change the temperature of MoCo v1 to 0.2, the same thing happens (Appendix). Besides, when K is below the "sweet spot", MoCo v1/v2 with EqCo significantly outperforms its counterpart without EqCo, and maintains steady over a large range of K.

2.5 EXPERIMENTS FOR SIMO

We evaluate SiMo on ImageNet under linear evaluation protocol, and compare top-1 accuracy to several contrastive learning methods that rely on large quantity of negative samples.

Architecture. We adopt ResNet-50 as our backbone of SiMo and follow the MLP architecture proposed in SimCLR, instead of MoCo v2. Specifically, there is a batch normalization layer behind each fully-connected layer in MLP.

Training. We train SiMo by SGD optimizer with a weight decay of 0.0001 and a momentum of 0.9 for 200 epochs. Scaling to the batch size, the initial learning rate is defined as: $0.03 \times \text{BatchSize}/256$, and decays with cosine decay schedule without restarts. We use a batch size of 256 on 8 GPUs, so the maximum size of negative samples is 255 for each query. For the momentum update, the momentum coefficient is set to be 0.999. The temperature hyper-parameter is 0.2.

Data augmentation. SiMo uses the same data augmentation strategy as in MoCo v2. A patch is randomly cropped from the image, and resized to 224×224. Then, we perform random horizontal flip, color distortions, Gaussian blur on it. More details can be found in Appendix xx.

Table 1: ImageNet accuracy of different contrastive learning methods under the linear evaluation protocol based on ResNet-50.

Method	Epochs	Top-1 (%)	Top-5 (%)
InstDisc	200	54.0	-
LocalAgg	200	60.2	-
CPC v2	200	63.8	85.3
CMC	240	66.2	87.0
SimCLR	200	66.6	87.3
MoCo v2	200	67.5	-
SiMo	200	67.9	88.3
SiMo w/ EqCo ($\alpha = 65536$)	200	**68.5**	**88.7**
Longer training			
PIRL	800	63.6	-
SimCLR	1000	69.3	89.0
MoCo v2	800	71.1	-
SiMo	800	71.8	90.9
SiMo w/ EqCo ($\alpha = 65536$)	800	**72.1**	**91.0**

We report the top-1 and top-5 accuracy of different contrastive learning methods under the linear evaluation protocol based on ResNet-50 in Tab.1. We observe that SiMo achieves 67.9% top-1 accuracy for 200-epoch training, showing the quality of learned representations without large quantity of negative samples on contrastive learning. Morever, with EqCo, the performance of SiMo is further improved, outperforming previous state-of-the-art method, MoCo v2, by 1%. With longer training, SiMo consistently surpasses SimCLR and MoCo v2. More results of SiMo are detailed in Appendix xx.

2.6 Transfer to object detection

Table 2: Object detection fine-tuned on COCO val2017. 1x schedule

pre-train	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
random init						
supervised	39.0	59.1	42.6	22.4	42.2	50.6
MoCo v2	39.1	59.2	42.5	23.3	42.1	50.8
SiMo	39.0	59.2	42.3	22.9	41.8	50.5

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pp. 15535–15545, 2019.

Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. *arXiv preprint arXiv:2006.14618*, 2020.

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

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

Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pp. 766–774, 2014.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020.

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

Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6398–6407, 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pp. 1195–1204, 2017.

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

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274, 2018.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *arXiv preprint arXiv:2005.10242*, 2020.

Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6388–6397, 2020.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

A APPENDIX

A.1 THE MONOTONICITY OF LOWER BOUND

As we describe above, the InfoNCE is related to the lower bound of mutual information:

$$\begin{aligned}
\mathcal{I}(\boldsymbol{q}, \boldsymbol{k}_0) &\geq log(K+1) - \mathbb{E}_{(\boldsymbol{q},\boldsymbol{k_0})\sim\mathbb{P}}[\mathcal{L}] \\
&\approx log(K+1) + log\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau) + \sum_{i=1}^{K} exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)} \\
&= log(K+1) - log(1 + exp(-\boldsymbol{q}^T\boldsymbol{k}_0/\tau)\sum_{i=1}^{K} exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)) \\
&\approx log(K+1) - log(1 + exp(-\boldsymbol{q}^T\boldsymbol{k}_0/\tau)K\mathbb{E}_{\boldsymbol{k}_i} exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau))
\end{aligned}$$

Let $f(K) = log(K+1) - log(1 + exp(-\boldsymbol{q}^T\boldsymbol{k}_0/\tau)K\mathbb{E}_{\boldsymbol{k}_i} exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau))$, and the derivative of f is easy to derived:

$$\begin{aligned}
f'(K) &= \frac{1}{1+K} - \frac{exp(-\boldsymbol{q}^T\boldsymbol{k}_0/\tau)\mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]}{1 + exp(-\boldsymbol{q}^T\boldsymbol{k}_0/\tau)K\mathbb{E}_{\boldsymbol{k}_i} exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)} \\
&= \frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau) - \mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]}{(1+K)(exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau) + K\mathbb{E}_{\boldsymbol{k}_i} exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau))}
\end{aligned}$$

In contrastive learning, it is natural to assume that the cosine similarity of positive pair is significantly bigger than the similarity of negative pair. Therefore, the derivative of f is always positive, implying that the lower bound of mutual information increases monotonically with K.

A.2 THE LOWER BOUND RELATED TO EQCO

Following CPC (Oord et al., 2018), we model the density ratio as follows:

$$exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau) \propto \frac{p(\boldsymbol{k}_0|\boldsymbol{q})}{p(\boldsymbol{k}_0)}$$

The lower bound of mutual information:

$$\mathcal{I}(\boldsymbol{q}, \boldsymbol{k}_0) \geq log(\alpha) - \mathbb{E}_{(\boldsymbol{q},\boldsymbol{k_0})\sim\mathbb{P}}[\mathcal{L}^*]$$

Proof.

$$\begin{aligned}
\mathbb{E}_{(\boldsymbol{q},\boldsymbol{k_0})\sim\mathbb{P}}[\mathcal{L}^*] &= \mathbb{E}_{(\boldsymbol{q},\boldsymbol{k_0})\sim\mathbb{P}}[-log\frac{\frac{p(\boldsymbol{k}_0|\boldsymbol{q})}{p(\boldsymbol{k}_0)}}{\frac{p(\boldsymbol{k}_0|\boldsymbol{q})}{p(\boldsymbol{k}_0)} + \frac{\alpha}{K}\sum_{i=1}^{K}\frac{p(\boldsymbol{k}_i|\boldsymbol{q})}{p(\boldsymbol{k}_i)}}] \\
&= \mathbb{E}_{(\boldsymbol{q},\boldsymbol{k_0})\sim\mathbb{P}}[log(1 + \alpha\frac{p(\boldsymbol{k}_0)}{p(\boldsymbol{k}_0|\boldsymbol{q})}\frac{1}{K}\sum_{i=1}^{K}\frac{p(\boldsymbol{k}_i|\boldsymbol{q})}{p(\boldsymbol{k}_i)})] \\
&\approx \mathbb{E}_{(\boldsymbol{q},\boldsymbol{k_0})\sim\mathbb{P}}[log(1 + \alpha\frac{p(\boldsymbol{k}_0)}{p(\boldsymbol{k}_0|\boldsymbol{q})}\mathbb{E}_{\boldsymbol{k}_i}\frac{p(\boldsymbol{k}_i|\boldsymbol{q})}{\boldsymbol{k}_i})] \\
&= \mathbb{E}_{(\boldsymbol{q},\boldsymbol{k_0})\sim\mathbb{P}}[log(1 + \alpha\frac{p(\boldsymbol{k}_0)}{p(\boldsymbol{k}_0|\boldsymbol{q})})] \\
&\geq \mathbb{E}_{(\boldsymbol{q},\boldsymbol{k_0})\sim\mathbb{P}}[log(\alpha\frac{p(\boldsymbol{k}_0)}{p(\boldsymbol{k}_0|\boldsymbol{q})})] \\
&\geq log(\alpha) - \mathcal{I}(\boldsymbol{q}, \boldsymbol{k}_0)
\end{aligned}$$

□

A.3 THE CONSISTENCY OF GRADIENTS

The upper bound of expectation:

$$\mathbb{E}_{\boldsymbol{k}_i}\|\frac{d\mathcal{L}}{d\boldsymbol{q}}\| \leq \frac{2}{\tau}(1-\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)+\alpha\mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]})$$

Proof. The features $\boldsymbol{q}$ and $\boldsymbol{k}_i, i=0,1,...,K$ are normalized by their L2-norm, then the following inequality is easy to get from Triangle Inequality:

$$\begin{aligned}\|\frac{d\mathcal{L}}{d\boldsymbol{q}}\| &\leq |\frac{1}{\tau}(1-\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{Z^*})|\|\boldsymbol{k}_0\| + \frac{\alpha}{\tau K}\sum_{i=1}^{K}|\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)}{Z^*}|\|\boldsymbol{k}_i\| \\ &= \frac{1}{\tau}(1-\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{Z^*}) + \frac{\alpha}{\tau K}\sum_{i=1}^{K}\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)}{Z^*}\end{aligned}$$

□

Further, according to Jensen's Inequality, we show that the expectation of the norm has the following upper bound:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{k}_i}\|\frac{d\mathcal{L}}{d\boldsymbol{q}}\| &= \frac{1}{\tau}(1-\mathbb{E}_{\boldsymbol{k}_i}[\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)+\frac{\alpha}{K}\sum_{i=1}^{K}exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)}]) + \frac{\alpha}{\tau K}\sum_{i=1}^{K}\mathbb{E}_{\boldsymbol{k}_i}[\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)+\frac{\alpha}{K}\sum_{i=1}^{K}exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)}] \\ &\leq \frac{1}{\tau}(1-\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)+\frac{\alpha}{K}\sum_{i=1}^{K}\mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]}) + \frac{\alpha}{\tau K}\sum_{i=1}^{K}\frac{\mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)+\frac{\alpha}{K}\sum_{i=1}^{K}\mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]} \\ &= \frac{1}{\tau}(1-\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)+\alpha\mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]}) + \frac{\alpha}{\tau K}\sum_{i=1}^{K}\frac{\mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)+\alpha\mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]} \\ &= \frac{2}{\tau}(1-\frac{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)}{exp(\boldsymbol{q}^T\boldsymbol{k}_0/\tau)+\alpha\mathbb{E}_{\boldsymbol{k}_i}[exp(\boldsymbol{q}^T\boldsymbol{k}_i/\tau)]})\end{aligned}$$

B MORE EXPERIMENTS ON SIMO

B.1 ABLATION ON MOMENTUM UPDATE

MoCo v1/v2 (He et al., 2020; Chen et al., 2020b) uses momentum update to control the the feature of negative samples are from previous batches. However, we only use the instances of current mini-batch as negative samples,

B.2 SYNC BN OR SHUFFLING BN?

In MoCo v1/v2 and SimCLR (Chen et al., 2020a), the query and positive sample are from the same sub-batch, so the batch statistics of them are very similar. Comparing with negative samples from different sub-batches (or queue-like memory bank), positive pair "stands out" for the obvious distinctions between positive pair and negative pairs. With an easy way to predict right from wrong, the model pays less attention on the semantic information of images, resulting in sub-optimal performance. To prevent information leakage like this, MoCo v1/v2 uses shuffling BN, while SimCLR uses sync BN.

For SiMo, both sync BN and shuffling BN are able to address the problem. To find out which one is better for SiMo, we conduct the following experiments.

B.3 SIMO WITH EQCO