[CVPR2022]

Supplementary Material

A. Setup of Figure 2 in the main manuscript

We train the model on CIFAR100 with MoCo v2 for 200 epochs on a single GPU. we use SGD optimizer with momentum 0.9 and weight decay 5e-4, and the temperature is set to 0.1. We use a linear warmup learning rate then decay learning rate following cosine decay schedule without restarts. Here, we adopt two independent dictionaries, $\boldsymbol{D}_{vector}$ and $\boldsymbol{D}_{scalar}$ to store negative sample keys for vector and scalar components, respectively. We fix one of them to have the dictionary size of 65536, while changing the dictionary size of the other one. We also report the results of a single dictionary with various dictionary size in Fig. 1. As expected, the performance decreases significantly when the dictionary size is small.

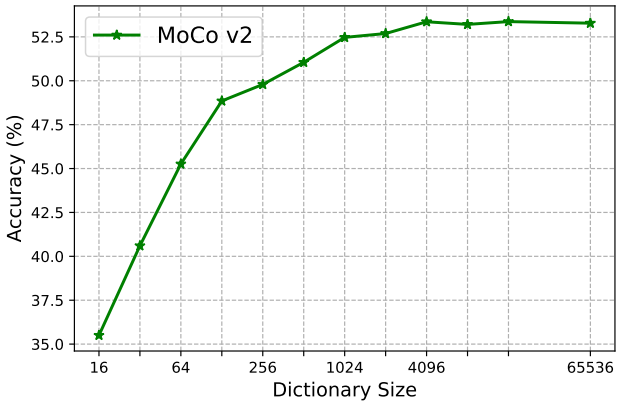

Figure 1. Influence of dictionary size in MoCo v2.

B. The scalar component is less sensitive to the quality of the keys

We also report the results for sampling a certain number (K_{scalar}) of keys from $\boldsymbol{D}_{scalar}$ while using the full $\boldsymbol{D}_{vector}$. We set K_{scalar} to 4096, since our results in Figure 2 of the main manuscript show that the scalar component requires a sufficiently large dictionary for competitive performance. The results in Table 1 show that there is only a small performance gap among the three sampling strategies. Notably, the model still converges well with a reasonable performance even when the earliest keys are sampled, while the model does not converge for K_{vector} in the same setup. The results show that the scalar component is less sensitive to the key quality.

Sampling strategies	Earliest	Random	Newest
Top-1 Accuracy(%)	52.13	52.75	53.32

Table 1. Comparison of various sampling strategies on CIFAR100.

C. The pseudo code for the relationship of dual temperature

The core difference between the InfoNCE with dual temperature in Eq 9 of the main manuscript and that in [1] lies in whether dual temperature is applied. Moreover, the loss in [1] uses negative samples from both encoders, while InfoNCE with dual temperature uses only half negative sample. For example, when q_i is the anchor, it only uses negative samples from the encoder k side, which simplifies the code implementation. The pseudo code is shown in Algorithm 1. Adopting negative samples from both sides is confirmed to yield equivalent performance.

D. MoCo v2 is more sensitive to temperature variation

Note that MoCo v2 by default adopts a single temperature, *i.e.* $\tau_\beta = \tau_\alpha$. When the temperature is very small, the inter-anchor hardness-aware sensitivity gets higher, leading to lower performance, while our SimMoCo and SimCo have no such concerns because τ_β is large. When the temperature is very large, the dependence of MoCo v2 on the old keys gets higher, *i.e.* lower PN consistency. The PN consistency for our SimMoCo and SimCo is always optimal because the negative keys are generated by the same encoder as the positive keys. Thus, our SimMoCo and SimCo have no such consistency concerns as MoCo v2. Overall, we observe that our proposed SimMoCo and SimCo consistently outperform the baseline MoCo v2.

E. InfoNCE in SSL *vs.* CE in SL.

The CE loss in supervised learning (SL) is shown as

$$\mathcal{L}_{CE} = -\log \frac{\exp(\boldsymbol{o}_{gt}/\tau)}{\sum_{c=1}^{C} \exp(\boldsymbol{o}_c/\tau)}, \tag{1}$$

where $\boldsymbol{o}$ indicates the network output which is a logit vector of length C (total number of classes) and gt indicates the index for the ground-truth (GT) class. Note that the sum is over the GT class and $(C-1)$ non-GT classes. With one hot vector defined as $\boldsymbol{y}$, there exists the following equivalence: $\boldsymbol{o}_{gt} = \boldsymbol{o} \cdot \boldsymbol{y}_{gt}$ and $\boldsymbol{o}_c = \boldsymbol{o} \cdot \boldsymbol{y}_c$.

Algorithm 1 Pytorch-like Pseudocode: Dual Temperature Loss

```
def simco_loss(query, key, intra_temperature,
    inter_temperature):
    """
    N: batch size
    D: the dimension of representation vector

    Args:
        query (torch.Tensor): NxD Tensor containing
            projected features from view 1.
        key (torch.Tensor): NxD Tensor containing
            projected features from view 2.
        intra_temperature (float): temperature factor
            for the intra component.
        inter_temperature (float): temperature factor
            for the inter component.

    Returns:
        torch.Tensor: SimCo loss.
    """
    # normalize query and key
    query = F.normalize(query, dim=-1)
    key = F.normalize(key, dim=-1)

    # calculate logits
    logits = query @ key.T

    # intra awareness
    logits_intra = logits / intra_temperature
    prob_intra = F.softmax(logits_intra, dim=1)

    # inter awareness
    logits_inter = logits / inter_temperature
    prob_inter = F.softmax(logits_inter, dim=1)

    # inter awareness changing factor
    mask = torch.ones(prob_inter.size()).
        fill_diagonal_(0)
    weight_alpha = (prob_intra * mask).sum(-1)
    weight_beta = (prob_inter * mask).sum(-1)

    inter_intra = weight_beta / weight_alpha

    # loss calculation
    log_softmax = F.log_softmax(logits, dim=-1)
    log_softmax_diag = log_softmax.diag()

    loss = -inter_intra.detach() * log_softmax_diag
    return loss.mean()
```

Based on the above equivalence, compared with Eq 1 in the main manuscript, we show that CE loss is a special case of InfoNCE by perceiving the GT one-hot vector as the positive key and other non-GT one-hot vectors as negative keys. With such a high resemblance between the two losses, however, unlike InfoNCE in SSL, this inter-anchor hardness-aware property is widely known to be important for competitive performance. In other words, alleviating the inter-anchor hardness-ware property does not help CE loss to improve the performance.

Here, we attempt to provide an intuitive explanation. Imagine that we do not have prior knowledge on the hardness of anchor sample, straightforwardly, the loss should be designed to treat every anchor sample equally. Given such prior knowledge, it is intuitive that the loss should put more weight on the hard anchor samples, such as CE does. Regarding this prior, the main difference between InfoNCE and CE is that the prior knowledge in CE is very reliable because the keys (both GT and non-GT) are fixed yet correct. However, this prior is less reliable in the InfoNCE loss because the keys are random. For example, the positive key with the same image of another random augmentation, and the negative keys are encoded from the random images. By changing the positive and negative keys randomly, we get two sets of r_+^i (see Eq 8 in the main manuscript) and calculate their similarity. The results in Figure 2 show that the similarity is low when the dictionary size is small, indicating this inter-anchor weight is not reliable. Intuitively, if this prior is unreliable, this inter-anchor hardness-aware property is misleading and thus it might be better to decrease this hardness-aware property, *i.e*. treating every anchor sample equally as in our investigation.

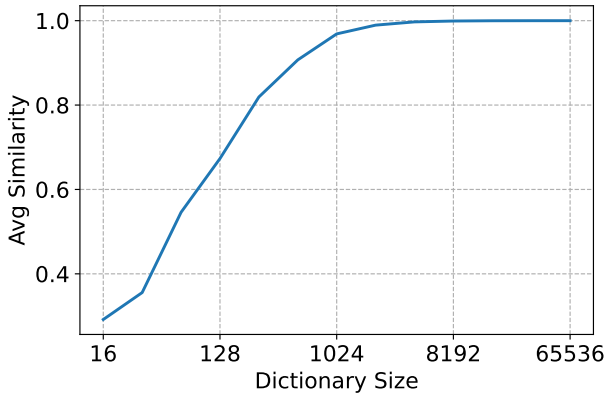

Figure 2. Cosine similarity between different r_+^i through changing the positive and negative keys randomly. Low similarity indicates that the inter-anchor hardness-aware weight is not reliable because a reliable prior should not deviate too much through changing the positive and negative keys.

Method	Symmetric			Asymmetric		
	0.4	0.6	0.8	0.4	0.6	0.8
CE	57.59	39.36	20.39	57.89	38.62	19.29
CE (DT)	**63.95**	**56.21**	**22.51**	**63.07**	**59.53**	**21.8**

Table 2. Test accuracy (%) of standard CE and CE (DT) on CIFAR10 with symmetric label noise ($\eta \in \{0.4, 0.6, 0.8\}$) and asymmetric label noise ($\eta \in \{0.4, 0.6, 0.8\}$).

With the above interpretation, the inter-anchor hardness-aware weight might also be detrimental to CE loss if the prior gets less reliable. A straightforward way to make the prior less reliable is to corrupt the data with noisy labels. We follow the setup in prior works [2] that study noisy labels. Specifically, the noise can be corrupted in a symmetric or asymmetric manner. The results with different noise ratios are shown in Table 2. We observe that CE with dual temper-

ature to remove the inter-anchor hardness-aware property outperforms the standard CE loss by a visible margin. Note that this experiment is conducted to prove our interpretation instead of pushing the SOTA performance in the setup of noisy labels.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1

[2] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, 2020. 2