Model-Aware Contrastive Learning: Towards Escaping the Dilemmas

Zizheng Huang [1] **Haoxing Chen** [1 2] **Ziqi Wen** [3] **Chao Zhang** [1] **Huaxiong Li** [1] **Bo Wang** [1] **Chunlin Chen** [1]

Abstract

Contrastive learning (CL) continuously achieves significant breakthroughs across multiple domains. However, the most common InfoNCE-based methods suffer from some dilemmas, such as *uniformity-tolerance dilemma* (UTD) and *gradient reduction*, both of which are related to a $\mathcal{P}_{ij}$ term. It has been identified that UTD can lead to unexpected performance degradation. We argue that the fixity of temperature is to blame for UTD. To tackle this challenge, we enrich the CL loss family by presenting a Model-Aware Contrastive Learning (MACL) strategy, whose temperature is adaptive to the magnitude of alignment that reflects the basic confidence of the instance discrimination task, then enables CL loss to adjust the penalty strength for hard negatives adaptively. Regarding another dilemma, the gradient reduction issue, we derive the limits of an involved gradient scaling factor, which allows us to explain from a unified perspective why some recent approaches are effective with fewer negative samples, and summarily present a gradient reweighting to escape this dilemma. Extensive remarkable empirical results in vision, sentence, and graph modality validate our approach's general improvement for representation learning and downstream tasks.

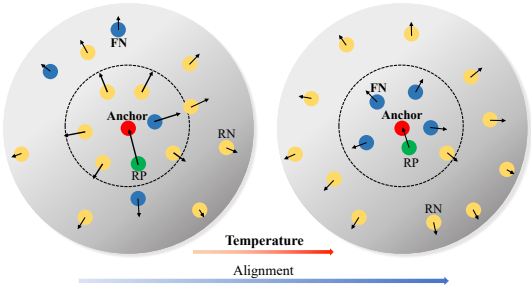


Figure 1. **Illustration of model-aware temperature strategy.** Points in red, green, yellow, and blue on the hypersphere denote anchor, the real positive sample (RP), real negative samples (RN), and false negatives (FNs), respectively. Since alignment magnitude can indicate discrimination confidence of the CL model, then the alignment-adaptive temperature dynamically controls penalty strength (arrow length) to negative samples to balance uniformity and tolerance for samples.

1. Introduction

Modern representation learning has been greatly facilitated by deep neural networks (Bengio et al., 2013; Dosovitskiy et al., 2020; He et al., 2016; Vaswani et al., 2017). Self-supervised learning (SSL) is one of the most popular paradigms in the unsupervised scenario, which can learn transferable representations without depending on manual labeling (Gidaris et al., 2018; He et al., 2022; Grill et al., 2020). Especially, SSL methods based on contrastive loss have highly boosted CV, NLP, graph, and multi-modal tasks (Chen et al., 2020b; He et al., 2020; You et al., 2021; Gao et al., 2021; Radford et al., 2021). These contrastive learning (CL) frameworks generally map raw data onto a hypersphere embedding space, whose embedding similarity can reflect the semantic relationship (Wu et al., 2018b; He et al., 2020). Among diverse contrastive losses, InfoNCE (Van den Oord et al., 2018; Tian et al., 2020a) is widely adopted in various CL algorithms (Chen et al., 2020a; 2021b; Dwibedi et al., 2021), which attempts to attract positive samples to the anchor while pushing all the negative samples away.

InfoNCE loss is essential to the success of CL (Tian, 2022; Wang & Isola, 2020) but still troubled by several dilemmas. An interesting hardness-aware property has been pointed out, which enables CL automatically concentrate on hard negative samples (HNs, those having high similarities with the anchor) (Wang & Liu, 2021; Tian, 2022). Particularly, the temperature parameter τ determines the weight distribution on negatives. But this also causes a Uniformity-Tolerance Dilemma (UTD) that plagues CL performance (Wang & Liu, 2021). Specifically, as for the common instance discrimination task in CL, models are trained by maximizing the similarities of the anchor with its augmenta-

[1]Nanjing University [2]Ant Group [3]China Telecom. Correspondence to: Huaxiong Li <huaxiongli@nju.edu.cn>, Haoxing Chen <hx.chen@hotmail.com>, Zizheng Huang <zizhenghuang@smail.nju.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

tions and minimizing that of all the other different instances (Wu et al., 2018b; Tian et al., 2020b). Such a strategy neglects the underlying semantic relationships, which can be explicitly subscribed by labels when in the supervised scenario. Those HNs might contain false negative samples (FNs) in this context. Owing to the hardness-aware property, a smaller τ is conducive to the uniformity of the embedding space (Wang & Isola, 2020), but goes against FNs due to excessive penalties on HNs. On the contrary, larger temperature parameters are beneficial for exploring underlying semantic correlations, while detrimental for learning separable informative features.

This work mainly focuses on two dilemmas in CL, both of which are related to a $\mathcal{P}_{ij}$ term. (1) *The uniformity-tolerance dilemma*, which is still an open problem in contrastive learning. We argue that a training-adaptive temperature is key to alleviating UTD. In the learning phase, alignment of positive paris (Wang & Isola, 2020) exactly can reflect the prior expectation of the instance discrimination task but also needs no extra computations in InfoNCE. Specifically, its alignment is underperforming for a poorly trained CL model. In this case, a smaller temperature parameter does help to improve the uniformity of the hypersphere embedding space (Wang & Isola, 2020). In contrast, the well-trained one is much better in terms of alignment, for which a larger temperature contributes to the tolerance for latent semantic relationships. Thus, we propose a model-aware temperature strategy based on alignment to solve the UTD problem. This strategy is illustrated in Figure 1. (2) *The gradient reduction dilemma of InfoNCE*. We identify the importance of negative sample size K and temperature τ for this gradient reduction problem. From a unified perspective, two propositions explain why some previous work (Yeh et al., 2022; Zhang et al., 2022; Chen et al., 2021a) are experimentally valid. As a result, we also provide a reweighting method for learning with small negative sizes. Owing to these explorations and Model-Aware Contrastive Learning (MACL) strategy, we reconstruct the contrastive loss to enable CL models to generate high-quality representations. Experiments and analysis on some benchmarks in different modalities demonstrate that the proposed MACL strategy does help improve the learned embeddings and escape dilemmas.

2. Related Work

Self-supervised learning has achieved significant success, which can provide semantically meaningful representations for downstream tasks (Bardes et al., 2022; Radford et al., 2021; Zbontar et al., 2021; He et al., 2017; Karpukhin et al., 2020). More recently, the instance discrimination task has achieved state-of-the-art, and even exhibited to be competitive performance to supervised methods (Chen et al., 2020a; 2021b; Gao et al., 2021; Dwibedi et al., 2021).

2.1. Contrastive Self-Supervised Learning

Contrastive instance discrimination originates from (Dosovitskiy et al., 2014; Wu et al., 2018b), whose core idea is to learn instance-invariant representations, i.e. each instance is viewed as a single class. The rational assumption behind is that maximizing similarities of the positive pairs and minimizing negative similarities can equip models with discrimination (Van den Oord et al., 2018). To construct the negative sampling appropriately, Wu et al. (2018b); Tian et al. (2020a) and Moco family (He et al., 2020; Chen et al., 2020c) adopt extra structures to store negative vectors of instances. Instead, without additional parts for storing negative samples, other methods explore negative sampling within a large mini-batch, e.g., SimCLR (Chen et al., 2020a), CLIP (Radford et al., 2021), and SimCSE (Gao et al., 2021). Some approaches successfully incorporate clusters or prototypes into CL (Caron et al., 2020; Huang et al., 2019; Dwibedi et al., 2021; Li et al., 2020). It is also possible to learn only relying on positive samples (Grill et al., 2020; Chen & He, 2021), but InfoNCE-based contrastive methods are still the mainstream for various modalities and tasks (Afham et al., 2022; Gao et al., 2021; Radford et al., 2021; Wang et al., 2021; Li et al., 2022).

2.2. Contrastive InfoNCE Loss

To understand the success of CL methods and enhance them, recent work has attempted to explore important properties of contrastive loss (Jing et al., 2022). InfoNCE is constructed by CPC (Van den Oord et al., 2018) and CMC (Tian et al., 2020a) to maximize the mutual information of features from same the instance. Besides, some work focuses on the positive and negative pairwise similarity in InfoNCE. For example, Wang & Isola (2020) attribute the effectiveness of InfoNCE to the asymptotical alignment and uniformity properties of features on hypersphere. Following this, Wang & Liu (2021) have proven that the temperature parameter plays an essential role in controlling the penalty strength on negative samples, which is related to the hardness-aware property and a uniformity-tolerance dilemma. This temperature effect is also mentioned in (Chen et al., 2021a). α-CL (Tian, 2022) formulates InfoNCE as a coordinate-wise optimization, in which the pairwise importance α determines the importance weights of samples.

Motivated by reducing the training batch size, DCL (Yeh et al., 2022) removes the positive similarity in the denominator of InfoNCE to eliminate a negative-positive-coupling effect. Furthermore, Zhang et al. (2022) extend the hardness-aware property anchor-wise and introduce an extra larger temperature for InfoNCE. There are also some efforts in explicitly modeling false/hard negative samples in training to improve CL (Shah et al., 2022; Kalantidis et al., 2020), e.g., HCL (Robinson et al., 2021) develops an importance

sampling strategy to recognize true and false negatives. Our work mainly focuses on alleviating uniformity-tolerance dilemma and exploring the gradient reduction problem.

3. Problem Definition

3.1. Contrastive Loss Function

Let $X = \{\boldsymbol{x}_i\}_{i=1}^N$ denote the unlabeled training dataset. Also given encoders f and g, instance $\boldsymbol{x}_i$ is mapped to a query feature$\boldsymbol{f}_i = f(\boldsymbol{x}_i)$ and a corresponding key feature $\boldsymbol{g}_i = g(\boldsymbol{x}_i)$ on hypersphere with augmentations, respectively. g maybe a weight-shared network of f or a momentum-updated encoder. Assume that the generated query (anchor) set and key set are denoted by $F = \{\boldsymbol{f}_i\}_{i=1}^N$ and $G = \{\boldsymbol{g}_i\}_{i=1}^{K+1}$, respectively, where N is batch size and K denotes the negative size. Then, the InfoNCE loss of the instance $\boldsymbol{x}_i$ can be formulated as:

$$\mathcal{L}_{\boldsymbol{x}_i} = -\log \frac{\exp\left(\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{g}_i/\tau\right)}{\exp\left(\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{g}_i/\tau\right) + \sum_{j=1}^K \exp\left(\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{g}_j/\tau\right)}, \tag{1}$$

where $\{\boldsymbol{f}_i, \boldsymbol{g}_i\}$ is the positive pair of the i-th instance, and $\boldsymbol{g}_j$ denotes the negative sample from a distinct instance. Temperature parameter is τ and $\tau > 0$. Negative pairs can also be incremental from the same-side encoder like NT-Xent (Chen et al., 2020a). The final total loss of an iteration is the mean value on the mini-batch: $\mathcal{L} = \sum_{i=1}^N \mathcal{L}_{\boldsymbol{x}_i}/N$.

3.2. Hardness-aware Property

Previous work identifies the important hardness-aware property via gradient analysis. For convenience, let $\mathcal{P}_{ij}$ indicate the similarity between x_i and x_j after scaled by temperature τ and Softmax operation:

$$\mathcal{P}_{ij} = \frac{\exp\left(\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{g}_j/\tau\right)}{\exp\left(\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{g}_i/\tau\right) + \sum_{r=1}^K \exp\left(\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{g}_r/\tau\right)}, \tag{2}$$

Then the gradient w.r.t the anchor $\boldsymbol{f}_i$ can be formulated as follows (more details are show in Appendix A.1):

$$\frac{\partial \mathcal{L}_{\boldsymbol{x}_i}}{\partial \boldsymbol{f}_i} = -\frac{\mathcal{W}_i}{\tau}\left(\boldsymbol{g}_i - \sum_{j=1}^K \hat{\mathcal{P}}_{ij}\cdot \boldsymbol{g}_j\right), \tag{3}$$

where $\mathcal{W}_i = \sum_{j=1}^K \mathcal{P}_{ij}$ can be seen as a gradient sacling factor, and there exists $\hat{\mathcal{P}}_{ij} = \mathcal{P}_{ij}/\sum_{r=1}^K \mathcal{P}_{ij}$. It is worth noting $\sum_{j=1}^K \hat{\mathcal{P}}_{ij} = 1$, in which $\hat{\mathcal{P}}_{ij}$ indicates an hardness-aware property. It implies that InfoNCE automatically puts larger penalty weights on the hard negatives (Wang & Liu, 2021), which are higher similar to the anchor sample.

3.3. Uniformity-Tolerance Dilemma

The weight on the negative sample x_j is formulated as:

$$\hat{\mathcal{P}}_{ij} = \frac{\exp\left(\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{g}_j/\tau\right)}{\sum_{r=1}^K \exp\left(\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{g}_r/\tau\right)}, \quad i \neq j, \tag{4}$$

which is controlled by the temperature parameter (Wang & Liu, 2021). (1) As τ decreases, the shape of $\hat{\mathcal{P}}_{ij}$ becomes sharper. Thus, a smaller temperature causes larger penalties on the high similarity region, which encourages the separation of embeddings but has less tolerance for FNs. (2) A larger temperature makes the shape of $\hat{\mathcal{P}}_{ij}$ flatter, then tends to give all negative samples equal magnitude of penalties. In this case, the optimization process is more tolerant to FNs while concentrating less on uniformity.

4. Model-Aware Temperature Strategy

The existence of the uniformity-tolerance dilemma leads to suboptimal embedding space and performance degradation of downstream tasks (Wang & Liu, 2021). Selecting an ideal temperature may be helpful, but it is not easy to get that balance. Instead, considering that the fixity of temperature prevents InfoNCE from focusing both on uniformity and potentially semantic relationships, we design an adaptive strategy for contrastive learning to mitigate the challenge.

4.1. Adaptive to Alignment

The uniformity-tolerance dilemma is rooted in the unsupervised instance discrimination task. Intuitively, the discrimination of a model will be gradually improved along with the training process, then the high similarity region is more likely to contain FNs. A dynamic temperature that changes according to iterations might deal with UTD better. *However, since the training iteration does not reflect the level of semantic confidence for a CL model, such temperature strategies are still rough and heuristic by now.* The more reasonable temperature adjustment strategy is needed to be investigated. What motivates us is the alignment property of the embedding space.

Alignment property is one of the most critical prior assumptions for instance discrimination (Wang & Isola, 2020; Wu et al., 2018b; Ye et al., 2019). It means that the representations from a positive pair should have high similarity. Since there are no labels available, it is impossible for SSL to explicitly construct semantic guidance. Instead, different views of the same instance are exploited for self-supervised learning. Alignment represents the awareness of view-invariance of a CL model, which is the base for exploring semantically consistent samples. Wang & Isola (2020) formulate the alignment loss as the expected distance

of positive pairs:

$$\mathcal{L}_{\text{align}} = \mathop{\mathbb{E}}_{\boldsymbol{x}_i \sim X} \left[\| f(\boldsymbol{x}_i) - g(\boldsymbol{x}_i) \|_2^2 \right]. \tag{5}$$

Another significant thing is that estimating the magnitude of alignment is not a computationally expensive operation. As shown in Eqn.(1), the calculation of sample similarities is a required step for CL loss, in which the part of positive pairs can be directly exploited for alignment. In this paper, we define the alignment magnitude $\mathcal{A}$ as the expected similarity of positive pairs. Hence, no additional structures and computations are needed. Here exists:

$$\begin{aligned} \mathcal{A} &= \mathop{\mathbb{E}}_{\boldsymbol{x}_i \sim X} \left[f(\boldsymbol{x}_i)^{\mathrm{T}} g(\boldsymbol{x}_i) \right] \\ &= 1 - \frac{1}{2} \mathop{\mathbb{E}}_{\boldsymbol{x}_i \sim X} \left[\| f(\boldsymbol{x}_i) - g(\boldsymbol{x}_i) \|_2^2 \right]. \end{aligned} \tag{6}$$

Thus, we have $\mathcal{A} = 1 - \mathcal{L}_{\text{align}}/2$ for alignment (detailed in Appendix A.2). $\mathcal{A} = 1$ implies perfect alignment.

4.2. Implementation Details

Then the proposed alignment-adaptive temperature strategy is formulated as:

$$\begin{aligned} \tau_a &= \tau_0 + \alpha \left(\mathop{\mathbb{E}}_{\boldsymbol{x}_i \sim X} \left[f(\boldsymbol{x}_i)^{\mathrm{T}} g(\boldsymbol{x}_i) \right] - \mathcal{A}_0 \right) \tau_0 \\ &= \left[1 + \alpha(\mathcal{A} - \mathcal{A}_0) \right] \tau_0, \end{aligned} \tag{7}$$

where α is a scaling factor and $\alpha \in [0, 1]$. $\mathcal{A}_0$ is a initial threshold for alignment magnitude. On the unit hypersphere, $\boldsymbol{f}_i^{\mathrm{T}} \boldsymbol{g}_i \in [-1, 1]$, then $\tau_a \in [(1 - \alpha - \alpha\mathcal{A}_0)\tau_0, (1 + \alpha - \alpha\mathcal{A}_0)\tau_0]$. *In particular, iff $\alpha = 0$, the temperature degenerates to the ordinary fixed case.* τ_a will be detached by stop gradient operation. The above form ensures the temperature changes in a proper range. In fact, lots of variants can be explored, but being alignment-adaptive is the most important point.

Eqn.(7) shows that the τ_a is an increasing function of $\mathcal{A}$, enabling the temperature to be adaptive to the alignment magnitude of the CL model during training. Specifically, a smaller temperature works when the model is lacking training by heavily penalizing those HNs. For the better-trained stage, the improved alignment indicates the CL model is more discriminative for samples. Naturally, larger temperature parameters can relax the penalty strength on the high similarity region, where is more possible to exist FNs now.

The proposed strategy is a *fine-grained adjusting approach*. As CL models are trained by sampling mini-batches, $\mathcal{A}$ can be estimated within a batch to promptly adjust the temperature. Thus, τ_a automatically adapts to the model of t-th optimization iteration. Compared with the one that simply increases by epochs, our adaptive strategy is more online. Thus, the proposed method is a Model-Aware Contrastive Learning (MACL) strategy.

5. Gradient Reduction Dilemma

With the above temperature strategy, the improved CL loss helps to escape UTD. However, the $\mathcal{P}_{ij}$ term also impedes efficient contrastive learning in another aspect. The problem is that CL models are typically trained with a large number K of negative samples to achieve better performance, which is computationally demanding, especially for large batch sizes. Some recent work tries to address this problem by modifying InfoNCE loss *but they each have their own opinions* (Yeh et al., 2022; Zhang et al., 2022; Chen et al., 2021a), whereas we prove they fall into a similar solution targeting the gradient reduction dilemma, but also summarily propose a simple reweighting method.

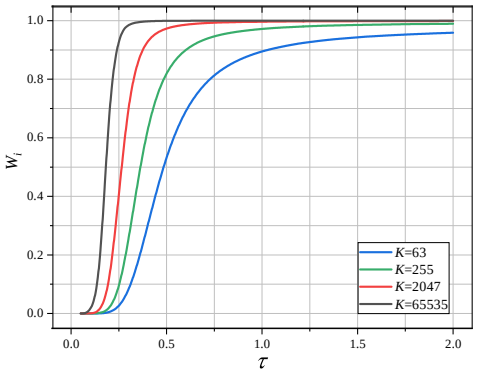


Figure 2. **Effect of the τ and K** on the gradient scaling factor $\mathcal{W}_i$.

5.1. Gradient Reduction Caused by the Sum Item

The gradient scaling factor $\mathcal{W}_i$ is a sum item of $\mathcal{P}_{ij}$ in Eqn.(3) and can also be described as:

$$\mathcal{W}_i = 1 - \frac{\exp\left(\boldsymbol{f}_i^{\mathrm{T}} \boldsymbol{g}_i / \tau\right)}{\exp\left(\boldsymbol{f}_i^{\mathrm{T}} \boldsymbol{g}_i / \tau\right) + \sum_{j=1}^{K} \exp\left(\boldsymbol{f}_i^{\mathrm{T}} \boldsymbol{g}_j / \tau\right)}. \tag{8}$$

This item has small values for those easy positive pairs, which will reduce the gradient in Eqn.(2) and has been mentioned in (Yeh et al., 2022). Therefore, the gradient reduction problem will hinder the model learning, especially for those deeper units in low-precision floating-point training with the chain rule. In addition, a smaller K leads to a significant gradient reduction as there is an insufficient accumulation of negative similarities. This is the intuitive rationale that state-of-the-art CL models are often trained with a large number of negative samples.

From another aspect, $\mathcal{W}_i$ is a monotonic function of τ. In particular, the shape of the sum item tends to become flat as temperature increases. We present an extreme example in Fig. 2, in which the similarities of the positive pair and negative pairs are set to 1 and -1, respectively. For these

analyses, we have the following propositions (please check Appendix A.3 for proof details):

Proposition 1 (Bound of gradient scaling factor w.r.t K)**.** *Given the anchor feature $\boldsymbol{f}_i$ and temperature τ, if $K \to +\infty$, then $\mathcal{W}_i$ approaches its upper bound 1. The limit is formulated as:*

$$\lim_{K \to +\infty} \mathcal{W}_i = 1. \tag{9}$$

Proposition 2 (Bound of gradient scaling factor w.r.t τ)**.** *Given $\boldsymbol{f}_i$ and key set G, $\mathcal{W}_i$ monotonically changes with respect to τ. The monotonicity is determined by the similarity distribution of samples. If $\tau \to +\infty$, then $\mathcal{W}_i$ approaches its bound $K/(K+1)$, formulated as:*

$$\lim_{\tau \to +\infty} \mathcal{W}_i = \frac{K}{1+K}. \tag{10}$$

5.2. Discussion about Previous Studies

These explorations show that the gradient reduction dilemma can be addressed by increasing the number of negative keys or adopting an extra large temperature for $\mathcal{W}_i$. More specifically, sampling more negative keys helps to promote the accumulation of the exponential similarities and then inhibit $\mathcal{W}_i$ too small. This is one of the reasons that most InfoNCE-based CL methods benefit from a large K, whether a big batch (Chen et al., 2020a; Dwibedi et al., 2021) or a large dictionary size (He et al., 2020; Tian et al., 2020a). In another case, adopting a larger separate temperature makes $\mathcal{W}_i$ approach its bound then also improves this issue, which is the key of (Zhang et al., 2022). Additionally, DCL (Yeh et al., 2022) removes the positive term from the denominator, then the corresponding gradient does not include $\mathcal{W}_i$ anymore. FlatNCE (Chen et al., 2021a) has exactly the same gradient expression with DCL, thus it is also feasible. We will recall their relations and provide some experimental evidence in Sec. 7.2.

5.3. Reweighting InfoNCE with Upper Bound

The above analysis essentially explains why some previous work is experimentally effective. We also design an approach for the gradient reduction issue when learning with small negative sizes, which is formulated as follows:

$$\mathcal{L}_{\boldsymbol{x}_i}^{M} = -\mathcal{V}_i \cdot \log \frac{\exp(\boldsymbol{f}_i^{\mathrm{T}} \boldsymbol{g}_i / \tau_a)}{\sum_{j=1}^{K+1} \exp(\boldsymbol{f}_i^{\mathrm{T}} \boldsymbol{g}_j / \tau_a)}, \tag{11}$$

where $\mathcal{V}_i = \mathrm{sg}\left[1/\mathcal{W}_i\right]$ and $\mathrm{sg}\left[\cdot\right]$ is the stop gradient operation to maintain the basic assumptions of InfoNCE, which is commonly used and finished by `detach` in code. In this case, the $\mathcal{W}_i$ item is rearranged with 1 for the small K cases Eqn.(3), assigned to its upper bound directly. A simple example pseudocode of Eqn.(11) is shown as Algorithm 1.

Algorithm 1 Pseudocode of MACL in a PyTorch-like style.

```
# pos: positive similarities, Nx1
# neg: negative similarities, NxK
# t_0: basic temperature
# a: scaling factor
# A_0: initial alignment threshold

def MACL(pos, neg, t_0, a, A_0):

    # model-aware temperature
    A = torch.mean(pos.detach())
    t = t_0 * (1 + a * (A - A_0))

    logits = torch.cat([pos, neg], dim=1)
    P = torch.softmax(logits/t, dim=1)[:, 0]

    # reweighting the loss
    V = 1 / (1 - P)
    loss = -V.detach() * torch.log(P)

    return loss.mean()
```

6. Empirical Study

In this section, we empirically evaluate the proposed strategy for enhancing CL performance in different cases. To demonstrate the general improvement, experiments are mainly implemented on the learning of images, but also include sentences and graph representations.

6.1. Experiments on Image Representation

We mainly experiment on the ImageNet ILSVRC-2012 (i.e., ImageNet-1K) (Deng et al., 2009) and use standard ResNet-50 (He et al., 2016) as image encoders. CIFAR-10 (Krizhevsky et al., 2009) and the subset ImageNet-100 (Tian et al., 2020a) are also considered. We choose SimCLR (Chen et al., 2020a) as the baseline but also perform some MoCo v2 (Chen et al., 2020c) evaluations. They use InfoNCE (or NT-Xent) as the basic schedule and are representative of mainstream frameworks, sampling negatives within mini-batches, from a momentum queue, respectively. We strictly follow their settings, augmentations, and linear evaluation protocol or reproduce under the same standard. As such, comparisons are solely on loss function impact. Details are laid out in Appendix B.1.

Effect of Negative Sizes First, we compare MACL against vanilla CL loss for negative sizes. Figure 3 recapitulates the results of SimCLR and MACL with batch sizes from 256 to 2048. From these linear evaluation scores, we can see that encoders trained with MACL significantly outperform the vanilla versions (NT-Xent) under all the negative sizes, and the accuracy under 256-batch size is higher than the 512 one of the counterpart. In fact, *our accuracy 66.5% under 1024-batch size is on par with the original 8192 one (66.5% vs 66.6%)*, which indicates the effectiveness for MACL strategy to escape the dilemmas.

Affected by the gradient reduction problem discussed in Sec.

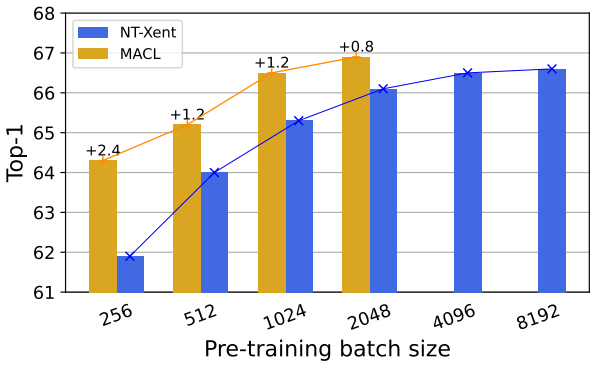


Figure 3. **Effect of batch sizes** (top-1 linear evaluation accuracies on ImageNet-1K with 200-epoch pre-training). Numbers on the top of bars are absolute gains of MACL under same settings.

5, SimCLR has a 4.2% drop from batch size of 2048 to 256. With MACL, the trained encoders are less sensitive to batch size as have a smaller corresponding 2.6% drop, and have higher improvement under smaller batch sizes. Besides, comparisons and discussion of queue size with MACL and InfoNCE on ImageNet-100 with MoCo v2 are reported in Appendix B.2.1. These results suggest the rationality of the gradient reduction dilemma analysis and reweighting approach for alleviating it.

Robustness to Training Length We conduct longer training with MACL, and the linear classification accuracies are shown in Table 1. There are some observations. First, MACL benefits from longer training length, which is consistent with vanilla contrastive loss. Moreover, MACL-based results are significantly better than ordinary ones. Our 200 and 400 epochs accuracies based on SimCLR are even comparable to the original ones with twice epochs (400 and 800), which demonstrates the learning efficiency brought by MACL. This also validates the advantage of MACL for dealing with the underlying dilemmas in InfoNCE.

Table 1. **Effect of training lengths** (top-1 linear evaluation accuracies on ImageNet1K with 256-batch size pre-training).

Epoch	200	400	800
SimCLR	61.9	64.7	66.6
w/ MACL	**64.3**(+2.4)	**66.3**(+1.6)	**68.1**(+1.5)

Transfer to Object Detection We evaluate representations learned by MACL on downstream detection task. We use VOC07+12 (Everingham et al., 2010) to finetune the encoders of SimCLR and MACL, then test models on VOC2007 test benchmark. Scores in Table 2 indicate that MACL strategy can provide better performance in terms of mean average precision metrics, demonstrating its effectiveness for learning transferable representations to detection.

Table 2. **Transfer to object detection** on VOC07+12 using Faster R-CNN with C4-backbone and 1$\times$ schedule. Encoders are trained with batch size of 256.

Pre-train	AP_{all}	AP_{50}	AP_{75}
SimCLR	49.7	79.4	53.6
w/ MACL	**50.1**(+0.4)	**79.7**(+0.3)	**53.7**(+0.1)

6.2. Experiments on Sentence Embedding

We adopt SimCSE (Gao et al., 2021) as the baseline in this part, which successfully facilities sentence embedding learning with contrastive learning framework using InfoNCE. The datasets and setups of training and evaluation follow the original literature and are detailed in Appendix B.4. Results under RoBERTa (Liu et al., 2019) backbone are reported in Table 3, and BERT (Kenton & Toutanova, 2019) scores are listed in Appendix Table B.4.

Performance on STS Tasks We conduct seven semantic textual similarity (STS) tasks to evaluate the capability of sentence embedding following (Gao et al., 2021). The results are measured by Spearman's correlation. For both models with RoBERT and BERT backbones, those trained with the MACL strategy achieve better performance on 6 of 7 STS tasks. Additionally, there are also noticeable gains w.r.t the average score. With the help of MACL, the learned embeddings are able to boost the clustering of semantically similar sentences.

Table 3. **STS and transfer tasks comparisons** of sentence embeddings with RoBERTa encoder.

STS task	STS12	STS13	STS14	STS15	STS16	STSB	SICKR
SimCSE	70.16	**81.77**	73.24	81.36	80.65	80.22	68.56
w/ MACL	**70.76** (+0.60)	81.43 (-0.34)	**74.29** (+1.05)	**82.92** (+1.56)	**81.86** (+1.21)	**81.17** (+0.95)	**70.70** (+2.14)
Transfer task	**MR**	**CR**	**SUBJ**	**MPQA**	**SST2**	**TREC**	**MRPC**
SimCSE	81.04	87.74	93.28	86.94	86.60	84.60	73.68
w/ MACL	**82.32** (+1.28)	**88.03** (+0.29)	**93.51** (+0.23)	**87.92** (+0.98)	**87.81** (+1.21)	**85.80** (+1.20)	**75.54** (+1.86)

Performance on Transfer Tasks We further investigate transfer tasks following (Gao et al., 2021) to verify the superiority of transferring to downstream settings. A logistic regression classifier is trained on top of the frozen pre-trained models. From the exhibited evaluation scores, it can be observed that the model trained with MACL achieves superior results on all the tasks and obtain 1.01% gain w.r.t the average score. In the BERT context, our MACL strategy outperforms on 5/7 tasks over the original SimCSE and also shows superiority in the average score. More experimental details are described in Appendix B.4. Results both on STS and transfer tasks fully suggest that the proposed MACL strategy provides higher quality representations, then gives considerable improvement for sentence embedding learning.

Table 4. **Downstream classification accuracies in graph representation learning** on different datasets.

Dataset	NCI1	PROTEINS	MUTAG
GraphCL	77.87±0.41	74.39±0.45	86.80±1.34
w/ MACL	**78.41±0.47**	**74.47±0.85**	**89.04±0.98**
Dataset	RDT-B	DD	IMDB-B
GraphCL	89.53±0.84	78.62±0.40	71.14±0.44
w/ MACL	**90.59±0.36**	**78.80±0.66**	**71.42±1.05**

6.3. Experiments on Graph Representation

To evaluate on graph representation learning, we choose GraphCL (You et al., 2020) as the baseline and compare MACL against ordinary CL loss on various benchmarks. The pre-training and evaluation settings are the default of GraphCL detailed in Appendix B.5.

Downstream Classification For the graph classification task, we conduct experiments on six commonly used benchmarks (Morris et al., 2020). They are denser or not-so-dense and cover areas of the social network, bioinformatics data, molecules data, etc. GNN-based encoders are the same in (Chen et al., 2019). Methods are trained with contrastive strategies, and the generated graph embeddings are fed into a downstream SVM classifier then reporting the mean and standard deviation values of five times following (You et al., 2020). As the scores shown in Table 4, our MACL strategy enables the framework to achieve better or comparable performance on these six different-scale (number of average nodes) datasets belonging to distinct fields.

Table 5. **Transfer learning comparisons of graph representation learning** on different datasets.

Dataset	Tox21	BBBP	ToxCast	SIDER
GraphCL	73.87±0.66	**69.68±0.67**	62.40±0.57	60.53±0.88
w/ MACL	**74.39±0.29**	67.98±0.97	**62.96±0.28**	**61.46±0.39**
Dataset	ClinTox	MUV	HIV	BACE
GraphCL	75.99±2.65	69.80±2.66	**78.47±1.22**	75.38±1.44
w/ MACL	**78.13±4.29**	**72.77±1.25**	77.56±1.12	**76.07±0.90**

Transfer to Chemistry Data Transfer learning comparisons are also considered. We experiment on molecular property prediction in chemistry following (You et al., 2020; Hu et al., 2020). Pre-trains and finetunes are in different datasets (Wu et al., 2018a). Models trained with MACL outperform original GraphCL on 6 of 8 datasets in Table 5. Both the downstream classification task and transfer learning results illustrate that MACL can boost representations with better generalizability and transferability, which further verifies the general improvement for vanilla CL loss.

6.4. Ablations

We present ablations of the proposed approach in this section to further understand its effectivity. Unless otherwise stated, settings are the same as those in Sec. 6.1.

Table 6. **Explorations of loss function**. Numbers are top-1 linear evaluation accuracies on ImageNet-1K with 200-epoch pre-training under 512-batch size. LR-s denotes the smaller learning rate case under the ordinary schedule, and LR-l is the larger case.

case	Adaptive	Reweighting	LR-s	LR-l
Baseline	✘	✘	64.0	65.6
(a)	✔	✘	64.9(+0.9)	67.5(+1.9)
(b)	✘	✔	65.0(+1.0)	67.8(+2.2)
(c)	✔	✔	65.2(+1.2)	68.1(+2.5)

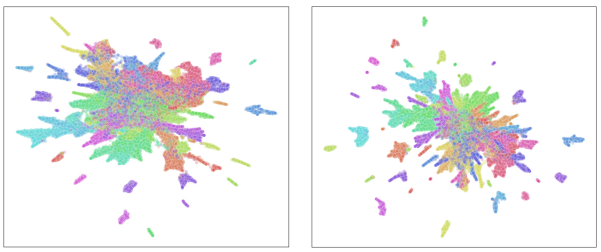
Loss Function Ablations To test the necessity of major components, we alter the loss function present in Eqn.(11) and validate encoders trained by variants. Linear evaluation scores are listed in Table 6 (the column of LR-s case). First, we can see that removing the adaptive temperature or reweighting operation leads to an accuracy drop compared to the full version. On the other hand, the model-aware adaptive method is designed to alleviate the performance degradation caused by the uniformity-tolerance dilemma. Utilizing this strategy in isolation yields a performance spike over the baseline. Since reweighting is designed to verify and improve the gradient reduction dilemma, only using this operation also achieves better performance. These observations support our motivation and designs. Chen et al. (2020a) show that SimCLR with a different learning rate schedule can improve the performance for models trained with small batch sizes and in smaller number of epochs. Interestingly, our MACL shows even higher improvement using a larger learning rate, which is present in the LR-l case in Table 6. More discussions are in Appendix B.1.1.

Table 7. **Ablation comparisons on ImageNet-100 with SimCLR framework** (linear evaluation accuracies with 200-epoch pre-training and batch size of 256).

Config	NT-Xent	DCL	MACL	
			w/ adaptive	w/o adaptive
Acc.	75.54 / 93.06	77.38 / 94.01	**78.28 / 94.25**	77.32 / 94.03

For another, we compare MACL with NT-Xent and DCL in Table 7. When α is set to 0, the temperature reduces to the fixed case, and only reweighting works. We can see that the top-1 score has a 1.78 gain over NT-Xent using reweighting in isolation and is on par with DCL, which supports the correctness of our judgment about gradient reduction dilemma. Besides, when equipped with the adaptive temperature, MACL obtains a further 0.96 improvement.

More Ablations We have already presented some ablations in the former experimental sections. From Figure 3



(a) NT-Xent (b) MACL

Figure 4. **UMAP visualization comparison** on ImageNet-100. ResNet-50 encoders are pre-trained 200 epochs under the batch size of 256 with NT-Xent and MACL, respectively. There are 100 colors indicating 100 semantic categories.

and Table 1, MACL exhibits significantly better robustness with respect to negative size and training length. We further present the UMAP visualization (McInnes et al., 2018) of features generated by encoders trained with our MACL and vanilla NT-Xent loss in Figure 4. Figure 4(b) exhibits better separability in the central area under the same training length, which indicates the learning efficiency and higher embedding quality brought by our approach.

Table 8. **Parameter analysis** for MACL strategy (linear evaluation accuracies of 200-epoch and 256-batch size pre-training on ImageNet-100 with SimCLR). The underlined configs are set to be fixed when the other are selected to be variables.

τ_0	0.05	0.1	0.5	1
Acc.	76.68 / 93.46	**78.14 / 94.16**	69.72 / 91.71	61.72 / 87.28
α	0	0.1	0.5	1
Acc.	77.32 / 94.03	77.18 / 94.06	**78.14 / 94.16**	77.74 / 94.14
$\mathcal{A}_0$	0	0.2	0.6	0.8
Acc.	78.14 / 94.16	**78.28 / 94.25**	78.1 / 94.14	77.54 / 93.95

Parameter analysis To better understand MACL as well as its parameters, we conduct experiments on and the scores are listed in Table 8. Since τ_0 is the datum point, it is essential to the performance of contrastive learning. Though the model is less sensitive to α and $\mathcal{A}_0$, they play an important role in adjusting the final temperature, yielding performance improvement with proper settings. The discussions of role of each parameter as well as the sensitivity analysis and ablations of NNCLR are present in Appendix B.2.2.

7. Discussion

7.1. Relations to Recent Temperature Schemes

Besides our alignment-adaptive strategy for addressing uniformity-tolerance dilemma, there are some interesting CL temperature schemes explored for different motivations. Zhang et al. (2021) aim to learn temperature as the uncertainty of embeddings for the out-of-distribution task. A dynamic multi-temperature method is proposed in (Khaertdinov et al., 2022) to scale instance-specific similarities in the Human Activity Recognition. The most recent (Kukleva et al., 2023) designs temperature as a cosine variation with epoch to improve CL performance on long-tail data. Additionally, as mentioned in Sec. 4.1, designing the temperature as a function of the iteration may potentially aid in escaping from UTD, however, such methods are incapable of providing real-time feedback on the training status.

Table 9. **Comparison of reweighting methods** (linear evaluation accuracies on CIFAR10, please check Appendix B.3 for setting details and corresponding kNN results).

Batch size	64	128	256	512	1024
NT-Xent	82.31	83.56	84.65	85.13	85.30
FlatNCE	86.30	86.28	86.11	86.02	85.84
DCL	86.28	86.04	86.29	86.33	85.61
Dual	86.32	86.40	85.86	86.23	86.05
MACL	**87.11**	**87.41**	**87.27**	**87.24**	**86.75**

7.2. Relations with Previous Reweighting Methods

As aforementioned, FlatNCE, DCL, and (Zhang et al., 2022) (Dual) essentially work against gradient reduction dilemma by approaching the bounds of the gradient scaling factor. Then we propose another feasible solution, reweighting the sum item with its upper bound directly. Furthermore, our MACL has an extra implicit alignment-adaptive reweighting for gradient of each step. For an under-optimization batch, the multiplier $1/\tau_a$ is bigger for Eqn.(3) as the lower alignment causes smaller τ_a, and vice versa. We test the performance of these methods. Results in Table 9 show that all the related methods outperform vanilla NT-Xent, especially under smaller batch sizes. FlatNCE, DCL, and Dual perform on par. Since MACL has an adaptive temperature which can alleviate UTD, it shows further superiority.

7.3. Contributions to α-CL

α-CL (Tian, 2022) formulates InfoNCE loss as coordinate-wise optimization, in which each element α_{ij} of the min player α is the pairwise importance of (i, j)-pair that is equal to $\mathcal{P}_{ij}$. Our adaptive temperature actually provides an iteration-dynamic feasible set for α, i.e., the landscape of constraint for α is different according to alignment magnitude. The entropy of α is a regularization for its min player, and will increase when the positive pairs are aligned better, since this entropy is a increasing function w.r.t τ (Wang & Liu, 2021). Furthermore, the constraint will reduce to a sample-agnostic case if the reweighting is applied.

7.4. Relations to Hard Negative Sampling

Hard negative sampling methods (Chuang et al., 2020) attempt to alleviate the drawback of instance discrimination via explicitly modeling false or hard negative samples. Such approaches have achieved promising results and are formulated by probability (Robinson et al., 2021), mixing(Kalantidis et al., 2020), aggregation (Huynh et al., 2022), or using SVM for the decision hyperplane of negatives (Shah et al., 2022). Instead, our MACL also pays attention to negatives but has adaptive penalty strengths on them, which is model-aware for FNs.

8. Conclusion

In this work, we analyze InfoNCE and provide strategies to escape the underlying dilemmas. To alleviate the uniformity-tolerance dilemma, an alignment-adaptive temperature is designed. Besides, we offer some insights into the importance of the negative sample size and the temperature by analyzing gradient reduction. A new contrastive loss is exploited based on these strategies. Experiment results in three modalities verify the superiority of our MACL strategy for improving contrastive learning.

Acknowledgements

We thank anonymous reviewers for their constructive comments. This work was partially supported by the National Natural Science Foundation of China (Nos. 62176116, 62276136, 62073160), and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 20KJA520006.

References

Afham, M., Dissanayake, I., Dissanayake, D., Dharmasiri, A., Thilakarathna, K., and Rodrigo, R. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *CVPR*, pp. 9902–9912, 2022.

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *ICLR*, 2022.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 33:9912–9924, 2020.

Chen, J., Gan, Z., Li, X., Guo, Q., Chen, L., Gao, S., Chung, T., Xu, Y., Zeng, B., Lu, W., et al. Simpler, faster, stronger: Breaking the log-k curse on contrastive learners with flatnce. *arXiv preprint arXiv:2107.01152*, 2021a.

Chen, T., Bian, S., and Sun, Y. Are powerful graph neural nets necessary? a dissection on graph classification. *arXiv preprint arXiv:1905.04579*, 2019.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, pp. 1597–1607, 2020a.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big self-supervised models are strong semi-supervised learners. *NeurIPS*, pp. 22243–22255, 2020b.

Chen, X. and He, K. Exploring simple siamese representation learning. In *CVPR*, pp. 15750–15758, 2021.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. In *ICCV*, pp. 9640–9649, 2021b.

Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., and Jegelka, S. Debiased contrastive learning. *NeurIPS*, 33: 8765–8775, 2020.

Contributors, M. MMSelfSup: Openmmlab self-supervised learning toolbox and benchmark. https://github.com/open-mmlab/mmselfsup, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, 2014.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *ICCV*, pp. 9588–9597, 2021.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–308, 2010.

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*, pp. 6894–6910, 2021.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, pp. 21271–21284, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *ICCV*, pp. 2961–2969, 2017.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pp. 9729–9738, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *CVPR*, pp. 16000–16009, 2022.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for pre-training graph neural networks. In *ICLR*, 2020.

Huang, J., Dong, Q., Gong, S., and Zhu, X. Unsupervised deep learning by neighbourhood discovery. In *ICML*, pp. 2849–2858, 2019.

Huynh, T., Kornblith, S., Walter, M. R., Maire, M., and Khademi, M. Boosting contrastive self-supervised learning with false negative cancellation. In *WACV*, pp. 2785–2795, 2022.

Jing, L., Vincent, P., LeCun, Y., and Tian, Y. Understanding dimensional collapse in contrastive self-supervised learning. *ICLR*, 2022.

Kalantidis, Y., Sariyildiz, M. B., Pion, N., Weinzaepfel, P., and Larlus, D. Hard negative mixing for contrastive learning. *NeurIPS*, pp. 21798–21809, 2020.

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In *EMNLP*, pp. 6769–6781, 2020.

Kenton, J. D. M.-W. C. and Toutanova, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pp. 4171–4186, 2019.

Khaertdinov, B., Asteriadis, S., and Ghaleb, E. Dynamic temperature scaling in contrastive self-supervised learning for sensor-based human activity recognition. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, pp. 1–8, 2022.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kukleva, A., Böhle, M., Schiele, B., Kuehne, H., and Rupprecht, C. Temperature schedules for self-supervised contrastive methods on long-tail data. In *ICLR*, 2023.

Li, J., Zhou, P., Xiong, C., and Hoi, S. C. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

Li, S., Wang, X., Zhang, A., Wu, Y., He, X., and Chua, T.-S. Let invariant rationale discovery inspire graph contrastive learning. In *ICML*, pp. 13052–13065, 2022.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763, 2021.

Robinson, J., Chuang, C.-Y., Sra, S., and Jegelka, S. Contrastive learning with hard negative samples. *ICLR*, 2021.

Shah, A., Sra, S., Chellappa, R., and Cherian, A. Max-margin contrastive learning. In *AAAI*, pp. 8220–8230, 2022.

Tian, Y. Deep contrastive learning is provably (almost) principal component analysis. *arXiv preprint arXiv:2201.12680*, 2022.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. In *ECCV*, pp. 776–794, 2020a.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning? *NeurIPS*, 33:6827–6839, 2020b.

Van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Wang, F. and Liu, H. Understanding the behaviour of contrastive loss. In *CVPR*, pp. 2495–2504, 2021.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pp. 9929–9939, 2020.

Wang, X., Zhang, R., Shen, C., Kong, T., and Li, L. Dense contrastive learning for self-supervised visual pretraining. In *CVPR*, pp. 3024–3033, 2021.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018a.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pp. 3733–3742, 2018b.

Ye, M., Zhang, X., Yuen, P. C., and Chang, S.-F. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, pp. 6210–6219, 2019.

Yeh, C.-H., Hong, C.-Y., Hsu, Y.-C., Liu, T.-L., Chen, Y., and LeCun, Y. Decoupled contrastive learning. In *ECCV*, pp. 668–684, 2022.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. In *NeurIPS*, pp. 5812–5823, 2020.

You, Y., Chen, T., Shen, Y., and Wang, Z. Graph contrastive learning automated. In *ICML*, pp. 12121–12132, 2021.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, pp. 12310–12320, 2021.

Zhang, C., Zhang, K., Pham, T. X., Niu, A., Qiao, Z., Yoo, C. D., and Kweon, I. S. Dual temperature helps contrastive learning without many negative samples: Towards understanding and simplifying moco. In *CVPR*, pp. 14441–14450, 2022.

Zhang, O., Wu, M., Bayrooti, J., and Goodman, N. Temperature as uncertainty in contrastive learning. *arXiv preprint arXiv:2110.04403*, 2021.

A. Proofs and Additional Analysis

A.1. Gradient of InfoNCE

Given sampled mini-batch of instances with K negative samples, the InfoNCE loss of instance $\boldsymbol{x}_i$ is expressed as:

$$\mathcal{L}_{\boldsymbol{x}_i} = -\log \frac{\exp\left(\boldsymbol{f}_i^{\mathrm{T}} \boldsymbol{g}_i/\tau\right)}{\exp\left(\boldsymbol{f}_i^{\mathrm{T}} \boldsymbol{g}_i/\tau\right) + \sum_{j=1}^{K} \exp\left(\boldsymbol{f}_i^{\mathrm{T}} \boldsymbol{g}_j/\tau\right)}.$$

For simplicity, let $E_k = \exp\left(\boldsymbol{f}_i^{\mathrm{T}} \boldsymbol{g}_k/\tau\right)$, and $\mathcal{L}_{\boldsymbol{x}_i}$ is reformulated as:

$$\mathcal{L}_{\boldsymbol{x}_i} = -\log \frac{E_i}{E_i + \sum_{j=1}^{K} E_j}.$$

Then the gradient with respect to $\boldsymbol{f}_i$ is:

$$\frac{\partial \mathcal{L}_{\boldsymbol{x}_i}}{\partial \boldsymbol{f}_i} = \frac{-1}{\tau} \frac{\sum_{r=1}^{K} E_r}{E_i + \sum_{r=1}^{K} E_r} \cdot \left(\boldsymbol{g}_i - \sum_{j=1}^{K} \frac{E_j}{\sum_{k=1}^{K} E_k} \cdot \boldsymbol{g}_j \right).$$

Let $\mathcal{P}_{ij}$ denote

$$\mathcal{P}_{ij} = \frac{E_i}{E_i + \sum_{r=1}^{K} E_r},$$

$\mathcal{W}_i = \sum_{j=1}^{K} \mathcal{P}_{ij}$, and $\hat{\mathcal{P}}_{ij} = \mathcal{P}_{ij} / \sum_{r=1}^{K} \mathcal{P}_{ij}$, where $\sum_{j=1}^{K} \hat{\mathcal{P}}_{ij} = 1$. Therefore, the above gradient can be reformulated as:

$$\frac{\partial \mathcal{L}_{\boldsymbol{x}_i}}{\partial \boldsymbol{f}_i} = -\frac{\mathcal{W}_i}{\tau} \left(\boldsymbol{g}_i - \sum_{j=1}^{K} \hat{\mathcal{P}}_{ij} \cdot \boldsymbol{g}_j \right). \tag{12}$$

Since the MoCo type algorithms detach the features in key set via a stop gradient operation, thus we discuss the loss function according to Eqn.(12). For SimCLR type methods, we can also derive the corresponding gradient with respect to $\boldsymbol{g}_i$:

$$\frac{\partial \mathcal{L}_{\boldsymbol{x}_i}}{\partial \boldsymbol{g}_i} = -\frac{\mathcal{W}_i}{\tau} \cdot \boldsymbol{f}_i, \tag{13}$$

and the gradient with respect to $\boldsymbol{g}_j$:

$$\frac{\partial \mathcal{L}_{\boldsymbol{x}_i}}{\partial \boldsymbol{g}_j} = \frac{\mathcal{W}_i}{\tau} \hat{\mathcal{P}}_{ij} \cdot \boldsymbol{f}_i. \tag{14}$$

A.2. Proof of Equation (6)

Proof of $\mathcal{A}$. Since representations $\boldsymbol{f}_i = f(\boldsymbol{x}_i)$ and $\boldsymbol{g}_i = g(\boldsymbol{x}_i)$ lie on a unit hypersphere (ℓ_2 normalized after the last layer of encoders), i.e., $f, g : \mathbb{R}^d \to \mathcal{S}^{m-1}$, where d and m denote the dimension of data space and hypersphere feature space. For $f(\boldsymbol{x}_i), g(\boldsymbol{x}_i) \in \mathcal{S}^{m-1}$, there exists: $f(\boldsymbol{x}_i)^{\mathrm{T}} f(\boldsymbol{x}_i) = g(\boldsymbol{x}_i)^{\mathrm{T}} g(\boldsymbol{x}_i) = 1$, thus

$$\|f(\boldsymbol{x}_i) - g(\boldsymbol{x}_i)\|_2^2 = 2 - 2 f(\boldsymbol{x}_i)^{\mathrm{T}} g(\boldsymbol{x}_i),$$

then, the relation of alignment $\mathcal{A}$ and alignment loss $\mathcal{L}_{\text{align}}$ is derived as:

$$\begin{aligned}
\mathcal{A} &= \mathop{\mathbb{E}}_{\boldsymbol{x}_i \sim X} \left[f(\boldsymbol{x}_i)^{\mathrm{T}} g(\boldsymbol{x}_i) \right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}_i \sim X} \left[1 - \frac{2 - 2 f(\boldsymbol{x}_i)^{\mathrm{T}} g(\boldsymbol{x}_i)}{2} \right] \\
&= 1 - \frac{1}{2} \mathop{\mathbb{E}}_{\boldsymbol{x}_i \sim X} \left[\|f(\boldsymbol{x}_i) - g(\boldsymbol{x}_i)\|_2^2 \right] \\
&= 1 - \frac{1}{2} \mathcal{L}_{\text{align}}.
\end{aligned}$$

□

A.3. Proof of Propositions

We now recall Proposition 1 and 2.

Proposition 1 (Bound of gradient scaling factor w.r.t. K)**.** *Given the anchor feature $\boldsymbol{f}_i$, and temperature τ, if $K \to +\infty$, then $\mathcal{W}_i$ approaches its upper bound 1. The limit is formulated as:*

$$\lim_{K\to+\infty} \mathcal{W}_i = 1.$$

Proposition 2 (Bound of gradient scaling factor w.r.t. τ)**.** *Given $\boldsymbol{f}_i$ and key set G, $\mathcal{W}_i$ monotonically changes with respect to τ. The monotonicity is determined by the similarity distribution of samples. If $\tau \to +\infty$, then $\mathcal{W}_i$ approaches its bound $K/(K+1)$, formulated as:*

$$\lim_{\tau\to+\infty} \mathcal{W}_i = \frac{K}{1+K}.$$

For simplicity, let $E_k = \exp\left(\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{g}_k/\tau\right)$, $s_k = \boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{g}_k$, $E_{max} = \max(E_1, \cdots, E_k, \cdots, E_K), k \neq i$, and $E_{min} = \min(E_1, \cdots, E_k, \cdots, E_K), k \neq i$.

Proof of Proposition 1. Here

$$\mathcal{W}_i = 1 - \frac{E_i}{E_i + \sum_{j=1}^{K} E_j}, \tag{15}$$

and the following inequality

$$1 - \frac{E_i}{E_i + K \cdot E_{min}} \leq \mathcal{W}_i \leq 1 - \frac{E_i}{E_i + K \cdot E_{max}}. \tag{16}$$

Since we have the limit of the left part

$$\begin{aligned}
&\lim_{K\to+\infty}(1 - \frac{E_i}{E_i + K \cdot E_{min}}) \\
=&\lim_{K\to+\infty}(1 - \frac{E_i/K}{E_i/K + E_{min}}) \\
=&1,
\end{aligned}$$

as well as the one of the right part

$$\lim_{K\to+\infty}(1 - \frac{E_i}{E_i + K \cdot E_{max}}) = 1,$$

thus the limit of $\mathcal{W}_i$ is

$$\lim_{K\to+\infty} \mathcal{W}_i = 1.$$

Notice that $E_k > 0$ strictly, then for a given K, $\mathcal{W}_i < 1$. Thus, $\mathcal{W}_i$ has its upper bound of 1 w.r.t. K. □

Proof of Proposition 2. For the temperature τ, we have

$$\begin{aligned}
\lim_{\tau\to+\infty} \mathcal{W}_i &= \frac{\lim_{\tau\to+\infty} \sum_{r=1}^{K} E_r}{\lim_{\tau\to+\infty} E_i + \lim_{\tau\to+\infty} \sum_{j=1}^{K} E_j} \\
&= \frac{\sum_{r=1}^{K} \lim_{\tau\to+\infty} E_r}{\lim_{\tau\to+\infty} E_i + \sum_{j=1}^{K} \lim_{\tau\to+\infty} E_j}.
\end{aligned} \tag{17}$$

Since the similarity value on hypersphere is bounded, i.e., $s_k = \boldsymbol{f}_i \cdot g_k \in [-1, 1]$, so

$$\lim_{\tau\to+\infty} E_k = 1. \tag{18}$$

Hence, from Eqn. (17) and (18)

$$\lim_{\tau\to+\infty} \mathcal{W}_i = \frac{K}{1+K}.$$

The gradient of $\mathcal{W}_i$ with respect to τ is derived as:

$$\frac{\partial \mathcal{W}_i}{\partial \tau} = \frac{1}{\tau^2} \cdot \frac{E_i}{(E_i + \sum_{r=1}^{K} E_j)^2} \cdot \sum_{j=1}^{K} (s_i - s_r) \cdot E_r. \tag{19}$$

As $E_k > 0$, then we have

$$\frac{\partial \mathcal{W}_i}{\partial \tau} \propto \sum_{j=1}^{K} (s_i - s_r) \cdot E_r. \tag{20}$$

For a batch of very poor embeddings, $\partial \mathcal{W}_i / \partial \tau \leq 0$, then $\mathcal{W}_i$ is a monotonic decreasing function with respect to τ. In contrast, for a batch of good embeddings, $\mathcal{W}_i$ monotonically increases as τ increases. So the similarity distribution of samples determine the monotonicity.

Naturally, Proposition 2 is a direct consequence of above conclusions. □

B. Implementation Details and Further Discussions

B.1. Experiments on ImageNet-1K

For MACL implementation on SimCLR framework, we follow the original augmentations (random crop, resize, random flip, color distortions, and Gaussian blur). The projection head is a 2-layer MLP projecting the representation to a 128-dimensional latent space. Models optimizations are completed by LARS with a base learning rate of 0.3 (0.3×BatchSize/256) and weight decay of 1e-6. We also use the cosine decay learning rate schedule with 10 epochs warmup. Parameters $\{\tau_0, \alpha, \mathcal{A}_0\}$ are set to $\{0.1, 0.5, 0\}$. *For MACL implementation on MoCo V2 framework*, we experiment on ImageNet-100, and the settings are listed in Appendix B.2.1. Codes of models are implemented on mmselfsup (Contributors, 2021) with several Tesla A100 80G GPUs.

B.1.1. Loss Function Ablation

Chen et al. (2020a) find the square root learning rate scaling is more desirable with LARS optimizer, i.e., $\text{LearningRate} = 0.075 \times \sqrt{\text{BatchSize}}$. Actually, for smaller batch sizes, such a scaling schedule provides a larger learning rate over the linear one, i.e., $\text{LearningRate} = 0.3 \times \text{BatchSize}/256$. For instance, the learning rate of 256-batch size is 1.2 under the square schedule while 0.3 under the linear schedule. Regarding ablations for MACL, we experiment with 512-batch size using SimCLR framework and linear learning rate scaling. We also present the much larger learning rate ablation results in Table 6, in which we set it to 2.4. There are some observations. First, similar to the baseline, variants of our MACL achieve significantly better performance under a larger learning rate. LR-l provides an even higher gain than that on the baseline. Besides, the ablations under LR-l also suggest the contributions made by different parts of the proposed loss function. Furthermore, trained for 200 epochs with 512-batch size, only using adaptive temperature or reweighting, our strategy can obtain better accuracies compared to the 512-batch size, 800-epoch or 1024-batch size, 400-epoch of the baseline.

B.2. Experiments on ImageNet-100

ImageNet-100 is a subset of ImageNet-1K, in which the images belong to 100 classes. The adopted encoders are ResNet-50 (He et al., 2016).

B.2.1. Queue Size Experiment

For MoCo v2 (Chen et al., 2020c), we follow their settings (including augmentations and architecture) on ImageNet-1K except for the learning rate of pre-training is 0.3 and a 10 epochs warmup is added. In linear evaluation, we use the batch size of 256 and an SGD optimizer with a learning rate of 10, and momentum of 0.9 without weight decay regularization. Epochs for pre-training and evaluation is 200 and 100, respectively. We set $\{\tau_0, \alpha, \mathcal{A}_0\}$ to $\{0.15, 0.5, 0.2\}$ for MACL experiments and the temperature is 0.2 for original MoCo v2 following their ImageNet-1K setup. The queue size experiment mentioned in Sec. 6.1 is reported in Table B.1. Instead of sampling negative samples within a mini-batch, MoCo family exploits a queue structure to store instance representations. From these results, we can see that MoCov2 has better stability in terms of negative size compared to SimCLR. Actually, *MoCo is less likely to be troubled with easy positive pairs* since the utilized momentum encoder is updated slowly (momentum value is 0.999). And the synchronous update framework with

weight-shared networks such as SimCLR is more likely to encode the same instance similarly, then is more sensitive to the gradient reduction dilemma. Even so, models have better performance with MACL strategy.

Table B.1. **Effect of queue sizes** (top-1/top-5 linear evaluation accuracies on ImageNet-100 with 200-epoch pre-training).

Queue size	256	512	4096	65536
MoCo v2	76.80 / 94.34	76.89 / 94.24	77.02 / 94.31	76.36 / 93.92
w/ MACL	**77.10 / 94.36** (+0.30) / (+0.02)	**77.24 / 94.39** (+0.35) / (+0.15)	**77.62 / 94.45** (+0.60) / (+0.14)	**77.46 / 94.16** (+1.10) / (+0.24)

B.2.2. Parameter and Ablation Analysis

Regarding the scores listed in Table 8, their settings are the same as that on ImageNet-1K. Similar to the trend of the vanilla NT-Xent loss in (Chen et al., 2020a), too large or small temperatures will lead to improper scaling for positive and negative similarities in Softmax, then plagues the CL. Thus, searching a proper τ_0 is necessary for the dynamic adaptation and we refer to the value of the fixed ones of original methods for our settings, e.g., 0.1 for SimCLR (Chen et al., 2020a). α can determine the change range of temperature, and we find that 0.5 provides a higher gain within this group of alternatives. $\mathcal{A}_0$ is the initial alignment threshold related to the change direction of τ_a. Too large $\mathcal{A}_0$ will lead to extremely small temperature in the early training period as alignment magnitude $\mathcal{A}$ is low. Overall, the final temperature in MACL is adaptive to alignment magnitude and scaled by these three factors. Since τ_0 is the datum point, models are more sensitive to its setting. Choosing appropriate parameters enable CL models to deal with uniformity-tolerance dilemma better.

We further conduct comparisons with NNCLR (Dwibedi et al., 2021) on ImageNet-100 and present them in Table B.2. It is worth noting that the InfoNCE objective construction in NNCLR is different from that in SimCLR and MoCo family. NNCLR obtains the positive key from a support set using nearest-neighbours to increase the richness of latent representation and go beyond single instance positives. As such, the positive pair of representations might belong to distinct instances. We set τ_0 and τ to 0.1 and use LARS optimizer following NNCLR literature, learning rate is 0.8, and cosine decay schedule with 10 epochs warmup. We find that under different parameters our MACL can generally outperform the original version and has the biggest 2.22 / 1.28 percent gain of top-1/top-5 accuracy. The performance demonstrates that our MACL is also applicable for such a support set framework to facilitate contrastive learning.

Table B.2. **Ablation comparisons on ImageNet-100 with NNCLR framework** (top-1/top-5 linear evaluation accuracies with 100-epoch pre-training, temperature 0.1, and 512-batch size).

α	0.5			1		NNCLR
$\mathcal{A}_0$	0	0.2	0.6	0	0.6	
Acc.	67.12 / 89.92	**67.72 / 90.02**	66.76 / 89.45	65.90 / 88.99	66.56 / 89.16	65.50 / 88.74

B.3. Experiments on CIFAR10

Encoders are CIFAR version ResNet-18 (He et al., 2016), in which the kernel size of the first 7×7 convolution is replaced with a 3×3 one, and the first max pooling module is removed. Unless otherwise stated, the temperature is 0.1 for all the losses and α=0.5, $\mathcal{A}_0$=0 for MACL. We make the loss symmetric in implementation and use four types of augmentations for pretraining: random cropping and resizing, random color jittering, random horizontal flip, and random grayscale conversion. The LARS optimizer in SimCLR (Chen et al., 2020a) is replaced by Adam with a base learning rate of 1e-3 and weight decay is 1e-6. For batch sizes that are larger than 256, the learning is scaled by 1e-3×Batchsize/256. We train the encoders for 200 epochs. For linear evaluation, the trained CL models are evaluated by fine-tuning a linear classifier for 100 epochs with 128-batch size on top of frozen backbones. We utilize an SGD optimizer by setting the learning rate to 0.02, momentum to 0.9, and weight decay to 0.

B.4. Sentence Embedding Experiments

Pre-training is completed on the 1-million randomly sampled sentences from English Wikipedia, which is the same as SimCSE. Following (Gao et al., 2021), learning starts from pre-trained checkpoints of the base version RoBERTa(cased)

Table B.3. **Comparison of reweighting methods** (top-1 linear evaluation / kNN accuracies on CIFAR10, $k = 200$).

Batch size	64	128	256	512	1024
NT-Xent	82.31 / 78.80	83.56 / 79.78	84.65 / 81.46	85.13 / 81.91	85.30 / 82.27
FlatNCE	86.30 / 84.50	86.28 / 84.47	86.11 / 84.08	86.02 / 83.99	85.84 / 83.54
DCL	86.28 / 84.59	86.04 / 84.64	86.29 / 83.86	86.33 / 84.02	85.61 / 83.07
Dual	86.32 / 84.40	86.40 / 84.69	85.86 / 83.87	86.23 / 83.75	86.05 / 83.64
MACL	**87.11 / 84.96**	**87.41 / 84.85**	**87.27 / 85.32**	**87.24 / 85.18**	**86.75 / 84.71**

(Liu et al., 2019) and BERT(uncased) (Kenton & Toutanova, 2019). We set $\{\tau_0, \alpha, \mathcal{A}_0\}$ to $\{0.05, 2, 0.8\}$. Following (Gao et al., 2021), algorithms are performed based on Huggingface's transformers package[1] and evaluated with SentEval toolkit[2]. The exploited Wikipedia sentence dataset is the one released by SimCSE authors. Models are trained for 1 epoch. For SimCSE, only dropout is exploited as augmentation, models have a good initial alignment for positive pairs (Gao et al., 2021). The batch size is set as 64, and learning rate for BERT version is 3e-5 and 1e-5 for the RoBERTa one. We try a stronger dropout in the experiment and found that the rate of 0.2 can generate better scores when cooperating with MACL, but is not suitable for vanilla InfoNCE. Note that as the original literature shows that the results are not sensitive to batch size, so we did not apply reweighting in this part.

Table B.4. **STS tasks comparisons of sentence embeddings** (the adopted metric is Spearman's correlation with "all" setting).

STS task	STS12	STS13	STS14	STS15	STS16	STSB	SICKR	Avg.
SimCSE-BERT	**68.40**	82.41	74.38	80.91	78.56	76.85	72.23	76.25
w/ MACL	67.16 (-1.24)	**82.78** (+0.36)	**74.41** (+0.03)	**82.52** (+1.61)	**79.07** (+0.51)	**77.69** (+0.84)	**73.00** (+0.77)	**76.66** (+0.41)
Transfer task	MR	CR	SUBJ	MPQA	SST2	TREC	MRPC	Avg.
SimCSE-BERT	81.18	**86.46**	94.45	88.88	85.50	**89.80**	74.43	85.81
w/ MACL	**81.80** (+0.62)	86.12 (-0.34)	**94.66** (+0.22)	**89.12** (+0.24)	**86.38** (+0.88)	88.60 (-1.20)	**76.46** (+2.03)	**86.16** (+0.34)

Same as the authors reminded, we also notice that the results are slightly different when implemented on different machines and CUDA versions (all package versions are the same as the author provided). But our MACL indeed can boost the performance on different machines. We try to experiment on Nvidia RTX 3090 with CUDA11.6, RTX 1080ti with CUDA11.4, and Tesla T4 with CUDA11.2 on Google colab[3] and finally report the results on Tesla T4. In fact, if compared against the reproduced results, our approach has an even more significant improvement. For example, the comparison on Tesla T4 is shown in Table B.5. We can see that the average score on STS tasks has a 1.57 and 0.89 improvement when using MACL strategy with RoBERTa and BERT, respectively.

Table B.5. **Reproduction** of sentence embedding performance on STS tasks.

STS task	STS12	STS13	STS14	STS15	STS16	STSB	SICKR	Avg.
SimCSE-RoBERTa	70.16	**81.77**	73.24	81.36	80.65	80.22	68.56	76.57
SimCSE-RoBERTa (repro)	67.88	81.55	72.44	81.31	80.73	80.38	67.83	76.02
w/ MACL	**70.76**	81.43	**74.29**	**82.92**	**81.86**	**81.17**	**70.70**	**77.59**
SimCSE-BERT	**68.40**	82.41	74.38	80.91	78.56	76.85	72.23	76.25
SimCSE-BERT (repro)	68.26	81.60	72.98	81.47	77.91	76.90	71.30	75.77
w/ MACL	67.16	**82.78**	**74.41**	**82.52**	**79.07**	**77.69**	**73.00**	**76.66**

[1] https://github.com/huggingface/transformers,version 4.2.1.
[2] https://github.com/facebookresearch/SentEval
[3] https://colab.research.google.com

B.5. Graph Representation Experiments

All of the augmentations and hyper-parameters except for those about loss function are taken from the baseline directly (You et al., 2020). τ_0 is set to 0.2 in unsupervised classification and 0.1 in transfer learning. $\{\alpha, \mathcal{A}_0\}$ are set to $\{0.5, 0\}$. The contrastive loss utilized in GraphCL (You et al., 2020) actually is DCL (Yeh et al., 2022), in which the positive similarity is removed from the denomination of InfoNCE. The transfer learning section is molecular property prediction in chemistry following (You et al., 2020). The adopted GNN-based encoders are from (Hu et al., 2020). Experiments are performed ten times and finally report the mean and standard deviation of ROC-AUC scores (%). From Table 5, we can see that MACL has the largest 2.97 percent improvement on MUV dataset and outperforms GraphCL on 6/8 dataset.