Rebalanced Siamese Contrastive Mining for Long-Tailed Recognition

Zhisheng Zhong[1] Jiequan Cui[1] Eric Lo[1] Zeming Li[2] Jian Sun[2] Jiaya Jia[1,3]
The Chinese University of Hong Kong[1] MEGVII Technology[2] SmartMore[3]

Abstract

Deep neural networks perform poorly on heavily class-imbalanced datasets. Given the promising performance of contrastive learning, we propose **Re**balanced **S**iamese **Co**ntrastive **M**ining (ResCom) to tackle imbalanced recognition. Based on the mathematical analysis and simulation results, we claim that supervised contrastive learning suffers a dual class-imbalance problem at both the original batch and Siamese batch levels, which is more serious than long-tailed classification learning. In this paper, at the original batch level, we introduce a class-balanced supervised contrastive loss to assign adaptive weights for different classes. At the Siamese batch level, we present a class-balanced queue, which maintains the same number of keys for all classes. Furthermore, we note that the contrastive loss gradient with respect to the contrastive logits can be decoupled into the positives and negatives, and easy positives and easy negatives will make the contrastive gradient vanish. We propose supervised hard positive and negative pairs mining to pick up informative pairs for contrastive computation and improve representation learning. Finally, to approximately maximize the mutual information between the two views, we propose Siamese Balanced Softmax and joint it with the contrastive loss for one-stage training. ResCom **outperforms** the previous methods by **large margins** on multiple long-tailed recognition benchmarks. Our code will be made publicly available at: https://github.com/dvlab-research/ResCom.[1]

1 Introduction

With the emergence of powerful GPUs, and numerous available large-scale and high-quality datasets such as ImageNet [40], COCO [32], and Places [55], deep neural networks (DNNs) have shown great success to many visual discriminative tasks, including image recognition [19, 29], object detection [38], and semantic segmentation [9]. The above datasets are all carefully constructed and approximately balanced concerning the number of instances for each class or object. However, real-world problems typically exhibit a long-tailed distribution, where a few classes contain plenty of samples but the others are associated with only a few samples. Learning in such a real-world case is challenging as the low-frequency classes can be easily overwhelmed by high-frequency ones, which makes DNNs suffer from significant performance degradation.

Recently, contrastive learning has shown great promise in unsupervised representation learning [4, 7]. Supervised contrastive learning (SupCon) is an extension to contrastive learning by incorporating the label information to compose positive and negative pairs [28]. SupCon provides consistent boosts in top-1 accuracy compared with conventional cross-entropy loss. Moreover, it is also more

[1]Part of the work was done in MEGVII Research.

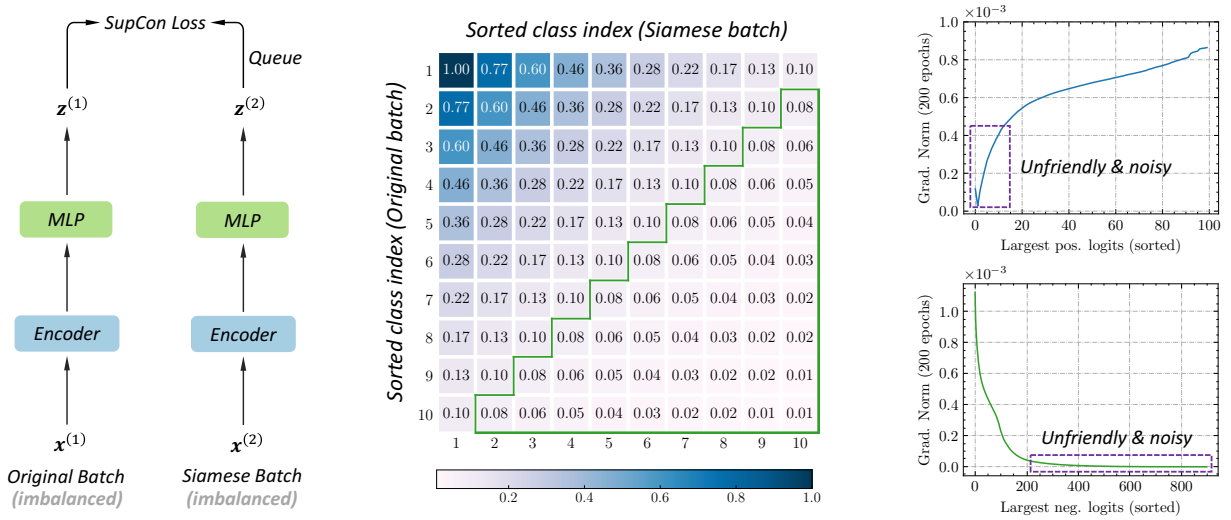


Figure 1: Left: illustration of the original batch, the Siamese batch, and the queue in the contrastive learning framework. *These three parts are all imbalanced* on long-tailed datasets. Center: frequency visualization of contrastive pairs for all class combinations normalized by the maximal frequency of contrastive pairs, *i.e.*, the total number of pairs from (Class-1, Class-1). The sampling simulation is run for 200 epochs on CIFAR-10-LT, imbalanced factor 10. The classes are sorted by descending values of the number of samples per class. The imbalance issue of supervised contrastive learning comes from *two levels*: the *original batch* and the *Siamese batch (queue)*. Right: the average gradient norm for the SupCon loss with respect to the positives and negatives (sorted by similarity in descending order). The model is trained on CIFAR-10-LT, imbalanced factor 10 for 200 epochs. The memory queue contains about 100 positives and 900 negatives. *The gradient of most easy positives and negatives could be vanished and noisy.*

robust to natural corruptions. However, to ensure performance, SupCon needs a large batch size for constructing enough contrastive pairs, which is extremely computation and memory costly. When using with memory based alternatives [18] (queue), SupCon can achieve better recognition performances and allows to use a small batch size for contrastive training, which significantly reduces compute and memory footprint.

Though supervised contrastive learning works well in a balanced setting, for imbalanced datasets, SupCon suffers a serious *dual class-imbalance issue* at both the *original batch-level* and the *Siamese batch-level* (Fig. 1 (left) for illustration). Here we provide a brief mathematical explanation (*a rigorous mathematical proof is shown in our supplementary meterial*): suppose the batch size is B, the frequency for Class-k of the training dataset is π_k. For a total batch, it approximately contains $B\pi_{\max}$ samples from the most frequent class for both the original and the Siamese batch, which constructs $B\pi_{\max} \cdot B\pi_{\max}$ positive pairs for the most frequent class. Similarly, SupCon includes $B\pi_{\min} \cdot B\pi_{\min}$ positive pairs of the least frequent class in a total batch. Thus, the imbalanced factor for the contrastive domain can be approximately computed by $\frac{B\pi_{\max} \cdot B\pi_{\max}}{B\pi_{\min} \cdot B\pi_{\min}} = \left(\frac{\pi_{\max}}{\pi_{\min}}\right)^2$, where $\frac{\pi_{\max}}{\pi_{\min}}$ is the imbalanced factor for classification learning. It means the degree of imbalance for supervised contrastive learning is *quadratic* to classification learning. Likewise, we can derive the imbalanced factor for the memory queue version of SupCon: suppose the queue size is Q and *queue is enqueued from the Siamese batch*, the imbalanced factor can be written as $\frac{B\pi_{\max} \cdot Q\pi_{\max}}{B\pi_{\min} \cdot Q\pi_{\min}} = \left(\frac{\pi_{\max}}{\pi_{\min}}\right)^2$, which meets the same conclusion. The simulation experiment shown in Fig. 1 (center) also verifies the above proposition: the related frequency of many contrastive pairs (in the green region) can be less than

0.1 (the reciprocal of classification imbalanced factor), and the minimal related frequency closes to 0.01. Thus, long-tailed contrastive learning suffers a *more difficult imbalance issue* and we should explore new ways to deal with.

In this work, we present **Re**balanced **S**iamese **Co**ntrastive **M**ining (**ResCom**) to tackle the dual contrastive imbalance issue: **(i)** We propose a class-balanced contrastive loss, which *assigns adaptive weights for different classes*. Under this case, the network will pay more attention to the tail classes samples to relieve the original batch-level imbalance. **(ii)** We maintain the dictionary as a class-balanced queue: *all classes contain the same number of keys*. When computing the SupCon loss, each class will construct the same number of positive and negative contrastive pairs. It can greatly alleviate the Siamese batch-level imbalance. Furthermore, we note that the contrastive loss gradient with respect to the contrastive logits can be *decoupled into the positive and negative parts*. Besides, as shown in Fig. 1 (right), *most easy positive and easy negative pairs make the contrastive gradient vanish and damage or disturb the learning process*. We propose **S**upervised hard positive and negative **P**airs **M**ining (**SPM**) to pick up these useful and informative positives and negatives, and enhance the representation learning. Finally, to approximately *maximize the mutual information of features from the two views* on long-tailed datasets, we propose **Siam**ese **B**alanced **S**oftmax (**SiamBS**), which can be *effectively and jointly trained with a contrastive loss in one-stage* for better recognition performances.

Ablation studies show ResCom can *improve the performances of the tail and medium classes while maintaining the performance of the head classes* compared with its counterparts. On multiple popular long-tailed recognition benchmark datasets, ResCom can *consistently surpass* previous methods by *large margins*, demonstrating the effectiveness of ResCom.

2 Related Work

Contrastive learning is a framework that learns similar and dissimilar representations from data that are organized into similar and dissimilar pairs, respectively. Recently, contrastive learning has shown great promise in unsupervised representation learning [5, 18, 36]. Chen *et al*. [5] proposed SimCLR, and is the first to match the performance of a supervised ResNet [19] with only a linear classifier trained on self-supervised representation on large-scale datasets. He *et al*. proposed MoCo [6, 18], which uses a momentum encoder to maintain consistent representations of negative pairs drawn from a memory bank. Without using negative pairs, Grill *et al*. proposed BYOL [17], which uses a momentum network to produce prediction targets as a means of stabilizing the bootstrap step. Supervised contrastive learning [28] is an extension to contrastive learning by incorporating the label information to compose positive and negative pairs, which can get better feature representation than cross-entropy.

Resampling and reweighting are the most intuitive methods to deal with long-tailed recognition. There are two groups of resampling strategies: over-sampling the tail classes [2, 42] and under-sampling the head classes [1, 24]. Reweighting [23, 50] is another prominent strategy. It assigns different weights for classes and even instances. The vanilla reweighting method gives class weights in reverse proportion to the number of samples of classes. However, with large-scale data, reweighting makes deep models difficult to optimize during training. Cui *et al*. [13] relieved the problem by using a effective number to calculate the class weight. Another line of work [31, 43] is to adaptively re-weight each instance.

Loss margin modification seeks to handle class imbalance by adjusting the decision margin.

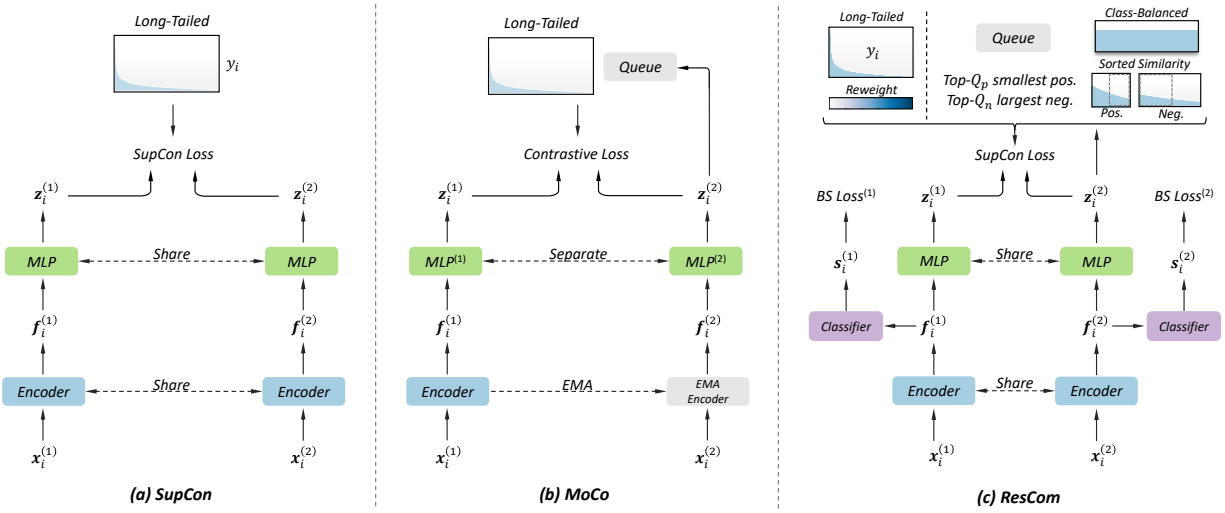


Figure 2: Conceptual comparison of three contrastive mechanisms for long-tailed recognition (empirical comparisons are listed in Sec. 4.2). (a) The SupCon encoders for computing the contrastive pairs (based on *the long-tailed distribution labels*) are updated end-to-end by back-propagation (the two encoders are shared). (b) MoCo encodes the new keys by a momentum-updated encoder, and maintains a *long-tailed distributed queue* of keys. (c) ResCom tackles the serious long-tailed contrastive learning issue by a class-balanced contrastive loss, a class-balanced queue, supervised hard positive and negative pairs mining, and Siamese Balanced Softmax for one-stage end-to-end training.

Cao *et al*. proposed LDAM [3] to integrate per-class margin into the cross-entropy loss. Balanced Softmax [37] proposed to use the label frequencies to adjust model predictions. Following the similar idea, Menon *et al*. proposed logit adjustment to post-hoc shift the model logits based on label frequencies [35]. LADE [22] proposed to use the label frequencies of test data to post-adjust the model outputs, so that the trained model can be calibrated for arbitrary test distribution.

Long-tailed contrastive recognition is also explored by many researchers. Yang *et al*. [52] proposed self-supervised pre-training, which is the first to use self-supervised learning (*e.g*., contrastive learning [18] or rotation prediction [16]) for model pre-training, followed by standard training on long-tailed data. Kang *et al*. [26] proposed a k-positive contrastive loss to learn a balanced feature space, which helps to alleviate class imbalance and improve model generalization. Following that, Hybrid network [48] introduced a prototypical contrastive learning strategy to enhance long-tailed learning. DRO-LT [41] extended the prototypical contrastive learning with distribution robust optimization, which makes the learned model more robust. PaCo [12] further innovated supervised contrastive learning by adding a set of parametric learnable class centers, which play the same role as a classifier if regarding the class centers as the classifier.

3 Rebalanced Siamese Contrastive Mining

3.1 Preliminaries

Supervised contrastive learning. Khosla *et al*. [28] extended the self-supervised contrastive loss with *label information* into a supervised contrastive loss. Within a multi-viewed batch (batch size is B), let $i \in \{1, ..., B\}$ be the index of an arbitrary sample. For a training sample $\{\boldsymbol{x}_i, y_i\}$,

two separate data augmentation operators are sampled from the same family of augmentations and applied to each data example $\boldsymbol{x}_i$ to obtain two correlated views $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$. $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$ share the same class label y_i. For simplification, we define $y_i = y_i^{(1)} = y_i^{(2)}$ here. In the following, an encoder network, *e.g.*, ResNet [19], is shared between two Siamese branches to learn the image features, $\boldsymbol{f}_i^{(1)}$ and $\boldsymbol{f}_i^{(2)}$ of two views. A MLP projection model and the l_2 normalization map the image features $\boldsymbol{f}_i^{(1)}$ and $\boldsymbol{f}_i^{(2)}$ into the embedding representations $\boldsymbol{z}_i^{(1)}$ and $\boldsymbol{z}_i^{(2)}$ for contrastive learning. Finally, an InfoNCE [36] style loss function is adopted on the embedding representations $\boldsymbol{z}$ and labels y.

Dynamic memory queue. The supervised contrastive loss usually *requires a large batch size*, *e.g.*, $B = 6,144$, for training to guarantee good performances, which requires a lot of computing resources and memory. He *et al*. [18] proposed a dynamic memory queue for mini-batch contrastive learning. [18] maintained a dictionary as a queue of embedding representation samples ($\boldsymbol{z}$) coming from several preceding mini-batches. The queue decouples the dictionary size from the mini-batch size. Here we present the queue vision of the supervised contrastive loss L_i^{SupCon} as:

$$L_i^{\text{SupCon}} = -\frac{1}{|\mathcal{P}_i|} \sum_{\boldsymbol{z}_p \in \mathcal{P}_i} \log \frac{\exp\left(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_p / \tau\right)}{\sum\limits_{\boldsymbol{z}_q \in \mathcal{Q}} \exp\left(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_q / \tau\right)}, \tag{1}$$

where $\mathcal{Q} = \{\boldsymbol{z}_i^{(2)}$ from several preceding mini-batches$\}$, $\mathcal{P}_i = \{\boldsymbol{z} \in \mathcal{Q} \wedge$ the class label of $\boldsymbol{z}$ is equal to $y_i\}$, $|\mathcal{P}_i|$ is its cardinality, the $\cdot$ symbol denotes the inner dot product, and τ is the temperature.

Disadvantages. If we directly apply the queue vision of supervised contrastive loss, Eq. (1), to long-tailed recognition, it will have the following problems: **(i)** *the original batch-level imbalance* and **(ii)** *the queue-level imbalance*.

As shown in Fig. 2(a), for the original batch-level imbalance, we consider the contrastive loss of a total batch: $L^{\text{SupCon}} = \frac{1}{B}\sum_{i=1}^{B} L_i^{\text{SupCon}}$. Each L_i^{SupCon} just focuses on positive pairs from Class-y_i. Due to the imbalanced class distribution, the total batch contrastive loss also is class-imbalanced, which makes the network pay more attention to positive pairs of the head classes rather than the tailed classes, which is inconsistent with what we expect.

As shown in Fig. 2(b), for the queue-level imbalance, the queue is enqueued by the preceding several mini-batches of embedding samples from View-2, the Siamese batch. Thus, the queue is also class-imbalanced. The queue contains more embedding samples from the head classes than the tail classes. We consider the contrastive loss of one query and revisit Eq. (1). Since there are more positives for the head classes, networks will get more feedback on the head classes pairs.

Overall, both the original batch-level and queue-level imbalance will lead long-tailed contrastive learning more difficult. Assuming that $N_{\max}$ and $N_{\min}$ are the numbers of training samples for the most and least frequent classes respectively, as analyzed in Sec. 1, the imbalanced factor for classification is $\frac{N_{\max}}{N_{\min}}$, while the imbalanced factor for contrastive learning becomes $(\frac{N_{\max}}{N_{\min}})^2$. To solve the above issues, we propose rebalanced Siamese contrastive mining.

3.2 Original Batch-level: Class-Balanced SupCon

To solve the batch-level imbalance issue, the most intuitive method is to directly use label frequencies of training samples for loss reweighting. Class-balanced cross-entropy loss [13] introduced the novel concept of *effective numbers* to approximate the expected sample number of different classes. The effective number is an exponential function of the training sample number. Following this concept,

we also introduce an effective number to reweight the supervised contrastive loss:

$$L_i^{\mathrm{CB}} = -\frac{w_{y_i}}{|\mathcal{P}(i)|} \sum_{\boldsymbol{z}_p \in \mathcal{P}(i)} \log \frac{\exp\left(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_p / \tau\right)}{\sum\limits_{\boldsymbol{z}_q \in \mathcal{Q}} \exp\left(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_q / \tau\right)}, \quad w_{y_i} = \frac{1-\beta}{1-\beta^{N_{y_i}}}, \tag{2}$$

where $\beta \in [0, 1)$ is a hyper-parameter. $\frac{1-\beta^{N_{y_i}}}{1-\beta}$ is the effective number of samples for Class-y_i. Under the class-balanced contrastive loss, tailed class samples will be *assigned larger weights* to relieve the original batch-level imbalance issue.

3.3 Queue-level: Class-Balanced Queue

To solve the queue-level imbalance issue, we propose a class-balanced queue for imbalanced contrastive learning. We suppose the number of classes is K. In our class-balanced queue model, we maintain K queues for each class separately. For each $\mathcal{Q}_k$, it contains Q contrastive logit samples only from the k-th class, *i.e.*, $|\mathcal{Q}_k| = Q, \forall k = 1, 2, ..., K,$ the class label of $\boldsymbol{z}$ equals $k, \forall \boldsymbol{z} \in \mathcal{Q}_k$. The contribution of each category to the class-balanced queue is the same. Thus, the total size of contrastive pairs equals KQ, and the balanced queue version of contrastive loss becomes:

$$L_i^{\mathrm{BQ}} = -\frac{w_{y_i}}{Q} \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \log \frac{\exp\left(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_p / \tau\right)}{\sum\limits_{k=1}^{K} \sum\limits_{\boldsymbol{z}_q \in \mathcal{Q}_k} \exp(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_q / \tau)}. \tag{3}$$

By adopting a class-balanced queue into contrastive learning, the queue-level imbalance issue can be solved very well. Regardless of the head classes or the tail classes sample, when calculating the contrastive loss of a query, it will have *the same number of positive pairs and negative pairs.*

3.4 Supervised Hard Positive and Negative Pairs Mining

Here we analyze the class-balanced queue version of contrastive loss from the gradient view. The gradient for L_i^{BQ} with respect to the embedding $\boldsymbol{z}_i^{(1)}$ has the following form:

$$\frac{\partial L_i^{\mathrm{BQ}}}{\partial \boldsymbol{z}_i^{(1)}} = \frac{w_{y_i}}{\tau} \Bigg[\underbrace{\sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \boldsymbol{z}_p \left(P_{ip} - \frac{1}{Q} \right)}_{\text{positive pairs}} + \underbrace{\sum_{k \neq y_i} \sum_{\boldsymbol{z}_n \in \mathcal{Q}_k} \boldsymbol{z}_n P_{in}}_{\text{negative pairs}} \Bigg], P_{ij} = \frac{\exp\left(\frac{\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_j}{\tau} \right)}{\sum\limits_{k=1}^{K} \sum\limits_{\boldsymbol{z}_q \in \mathcal{Q}_k} \exp\left(\frac{\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_q}{\tau} \right)}. \tag{4}$$

From Eq. (4), the total gradient can be *decoupled into the positive pairs part and the negative pairs part.* Many studies [5, 18, 21, 45] show that the number of positive pairs and negative pairs for each batch seriously affects the final recognition accuracy and show increased performance with an appropriate number of positives and negatives, wherein the ability to discriminate between signal and noise (negatives) is improved. On the other hand, [8, 25, 39] argue for the value of hard negatives in *unsupervised contrastive representation learning*: by mining harder negatives, one can get higher performance after training for fewer epochs. However, in supervised contrastive learning, as shown in Fig. 1 (right), *for most easy positive pairs,* $P_{ip} \to \frac{1}{Q}$*, and for most easy negative pairs,* $P_{in} \to 0$*. Both two cases will make the contrastive gradient* Eq. (4) *vanish, damage or disturb the whole learning process.*

Based on the above analysis, we propose Supervised hard positive and negative Pairs Mining (**SPM**). It makes supervised contrastive learning more flexible and effective. Concretely, in our SPM model, we introduce two hyper-parameters, Q_p ($Q_p \leq Q$), and Q_n to represent

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

```
# encoder: encoder network, FC: a single linear layer classifier (D x K)
# MLP: projection head (including l2 normalization)
# Qp, Qn: the number of the hard positive, negative pairs
# K: class number, t: temperature, lambda: loss weight
# queue: a class-balanced queue of keys (C x KQ)

for x in loader: # load a minibatch x with N samples
x1, x2 = aug(x), aug(x) # randomly augmented

# Siamese forward
f1, f2 = encoder(x1), encoder(x2)            # features: N x D
s1, s2 = FC(f1), FC(f2)             # classified logits: N x K
z1, z2 = MLP(f1), MLP(f2)            # keys and queries: N x C

# supervised hard positive and negative pairs mining
logits_pos = topk(-mm(z1, GetPos(queue, labels)), Qp) # small pos. logit: N x Qp
logits_neg = topk( mm(z1, GetNeg(queue, labels)), Qn) # large neg. logit: N x Qn

# contrastive logits: N x (Qp + Qn)
logits_cont = cat([logits_pos, logits_neg], dim=1)

# class-balanced supervised contrastive loss
mask = cat([ones(N, Qp), zeros(N, Qn)], dim=1)
l_cont = ClassBalancedSupConLoss(logits_cont / t, mask, labels)

# Siamese Balanced Softmax
l_cls = 0.5 * BalSfxLoss(s1, labels) + 0.5 * BalSfxLoss(s2, labels)

# total loss
loss = l_cls + lambda * l_cont

# SGD update: encoder, MLP, and the FC classifier
loss.backward(), optimizer.step()

# update the class-balanced queue
dequeue_enqueue(queue, z2, labels)
```

`mm`: matrix multiplication; `cat`: concatenation; `zeros`/`ones`: a tensor filled with 0/1; `topk`: returns the k largest elements of the given input.

the cardinality for the positive pairs and the negative pairs, respectively. According to the label y_i of $\boldsymbol{z}_i^{(1)}$, we reconstruct a new positive queue by mining the top-Q_p hard positive pairs: $\mathcal{Q}_p = \text{topk}(-\mathcal{Q}_{y_i} \cdot \boldsymbol{z}_i^{(1)}, Q_p)$. Similarly, we reconstruct a new negative queue by mining the top-Q_n hard negative pairs: $\mathcal{Q}_n = \text{topk}((\bigcup_{k \neq y_i}^{K} \mathcal{Q}_k) \cdot \boldsymbol{z}_i^{(1)}, Q_n)$. Both the hard positives and negatives are more likely to have larger gradients norm and contain more useful information. Thus, the SPM version of contrastive loss can be designed as:

$$L_i^{\text{SPM}} = -\frac{w_{y_i}}{Q_p} \sum_{\boldsymbol{z}_p \in \mathcal{Q}_p} \log \frac{\exp\left(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_p / \tau\right)}{\text{pos}(\boldsymbol{z}_i^{(1)}) + \text{neg}(\boldsymbol{z}_i^{(1)})}, \tag{5}$$

where $\text{pos}(\boldsymbol{z}_i^{(1)}) = \sum_{\boldsymbol{z}_p \in \mathcal{Q}_p} \exp(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_p / \tau)$, $\text{neg}(\boldsymbol{z}_i^{(1)}) = \sum_{\boldsymbol{z}_n \in \mathcal{Q}_n} \exp\left(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_n / \tau\right)$. If we set $Q_p = Q$ and $Q_n = (K-1)Q$, the SPM model will degrade to the class-balanced queue model, which means the SPM model is more flexible and the class-balanced queue is just a special case. We can specify the number of positive pairs and the number of negative pairs, and mine the hard positives and negatives in contrastive loss for better representation learning.

3.5 Siamese Balanced Softmax

The two image features $\boldsymbol{f}_i^{(1)}, \boldsymbol{f}_i^{(2)}$ are sent to a shared classifier to get two classified logits $\boldsymbol{s}_i^{(1)}$ and $\boldsymbol{s}_i^{(2)}$. Since $\boldsymbol{s}_i^{(1)}$ and $\boldsymbol{s}_i^{(2)}$ are come from the same input $\boldsymbol{x}_i$ and shared the same label y_i, we expect to *maximize the mutual information between the two view classified logits*: $I(\boldsymbol{s}_i^{(1)}, \boldsymbol{s}_i^{(2)})$. However, directly computing the mutual information on imbalanced datasets is not easy. Here we introduce a explicit

simplifying assumption to approximately measure it: $I(\boldsymbol{s}_i^{(1)}, \boldsymbol{s}_i^{(2)}) \approx c - H(\boldsymbol{s}_i^{(1)}, y_i) - H(\boldsymbol{s}_i^{(2)}, y_i)$, where H represents the Balanced Softmax version of cross-entropy [37], and c is a constant. Thus, maximizing the mutual information between the representations of two views is equivalent to minimizing the cross-entropy of them. Ignoring the constant term c, we get Siamese Balanced Softmax (**SiamBS**) to approximate the negative mutual information:

$$L_i^{\text{NMI}} = -\frac{1}{2}\sum_{v=1}^{2}\log\left(\frac{N_{y_i}\exp\left(\boldsymbol{s}_{i,y_i}^{(v)}\right)}{\sum_{k=1}^{K} N_k \exp\left(\boldsymbol{s}_{i,k}^{(v)}\right)}\right), \tag{6}$$

where N_k is the number of samples in Class-k, and v is the view index and can only take values from $\{1, 2\}$.

Overview. Fig. 2(c) shows the overview of the proposed ResCom for long-tailed recognition. The framework consists of two Siamese branches for contrastive representation learning and classification learning. We use the supervised hard positive and negative pairs mining version of contrastive loss, Eq. (5), as the regularization term for representation learning. Thus, the total loss of ResCom can be written as follows:

$$L_i^{\text{ResCom}} = L_i^{\text{NMI}} + \lambda L_i^{\text{SPM}}(\boldsymbol{z}_i^{(1)},\ y_i,\ \bigcup_{k=1}^{K}\mathcal{Q}_k,\ Q_p,\ Q_n), \tag{7}$$

where λ is a loss weight hyper-parameter. Unlike [18, 28, 52], the ResCom loss allows us to train the model in one stage without linear probing or fine-tuning. Algorithm 1 provides the pseudo-code of ResCom for detailed reference.

Evaluation. When using ResCom for evaluation, we only *preserve a single classification branch*. It means that just the encoder and the classifier are preserved, while the MLP is discarded. Therefore, the evaluation of ResCom is very efficient: It is consistent with conventional backbone like the plain ResNet [19], *without any additional computations*.

4 Experiments

4.1 Datasets and Setup

Our experimental setup including the implementation details mainly follows the previous studies [3, 12, 37, 54] for CIFAR-10-LT, CIFAR-100-LT, ImageNet-LT, Places-LT, and iNuturalist 2018. After long-tailed training, we evaluate the models on the corresponding balanced validation datasets, and report the commonly used top-1 accuracy over all classes, denoted as **All**. We also follow [27, 34] to split the categories into three subsets and report the average accuracy rates in these three subsets: **Many**-shot (>100 images), **Medium**-shot (20-100 images), and **Few**-shot (<20 images), which are also called the head, medium and tail categories, respectively. *More implementation and training details please refer our supplementary material.*

CIFAR-10 and CIFAR-100 both have 60,000 images — 50,000 for training and 10,000 for validation with 10 and 100 categories, respectively. For a fair comparison, we use the long-tailed version of CIFAR datasets with the same setting as those used in [13, 37, 56]. They control the degrees of data imbalance with an imbalance factor, $\frac{N_{\max}}{N_{\min}}$. Following [13], we conduct experiments with imbalance factors 100, 50, and 10.

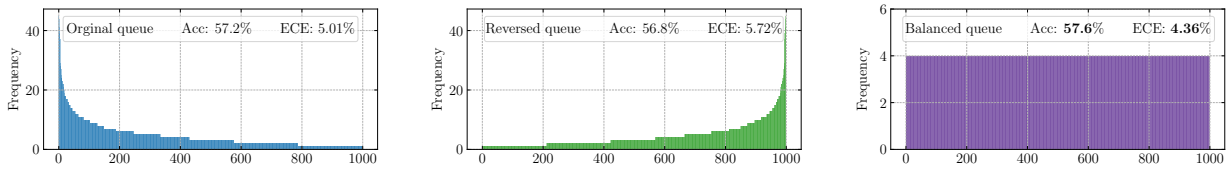


Figure 3: Class distribution visualization of different types of queues. The sizes of different queues are close to 4,000. From left to right: original queue, reversed queue, and class-balanced queue. The classes are sorted by descending values of the number of samples per class. Top-1 accuracy and ECE comparisons for different types of queues on ImageNet-LT with ResNet-50 are listed at the top.

ImageNet-LT was proposed by Liu *et al*. [34]. It is a long-tailed version of the large-scale object classification dataset, ImageNet [40], by sampling a subset following the Pareto distribution with a power value equaling to 6. It contains 115.8K images from 1,000 categories and its imbalance factor is 256, with class cardinality ranging from 5 to 1,280.

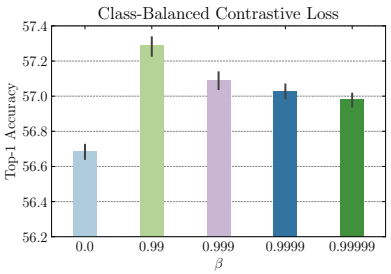
Places-LT is similar to ImageNet-LT. It was also proposed by Liu *et al*. [34], and is a long-tailed version of the large-scale scene classification dataset called Places [55]. It consists of 184.5K images from 365 categories and its imbalance factor is 996, with class cardinality ranging from 5 to 4,980.

iNaturalist 2018 [46] is a species classification dataset, which is on a large scale and suffers from extremely imbalanced label distribution. It is composed of 437.5K images from 8,142 categories and its imbalance factor is 500, with class cardinality ranging from 2 to 1,000. In addition to the extreme imbalance, iNaturalist 2018 also confronts the fine-grained problem [51].

4.2 Ablation Study

Ablation: class-balanced contrastive loss. In our class-balanced contrastive loss, there is one hyper-parameter β in Eq. (2), which controls the weight penalty of classes. It's worth noting that, $\beta = 0$ corresponds to no reweighting (each class is treated equally), and $\beta \to 1$ corresponds to reweighting by the inverse class frequency. Here we conduct experiments by varying the β from 0.0 to 0.99999 on ImageNet-LT with ResNet-50. We plot the performances upon β in Fig. 4 (left) for several possible variants. It shows that the top-1 accuracy can be further improved by **0.5%** compared with the conventional supervised contrastive loss ($\beta = 0$) when we pick $\beta = 0.99$ for the class-balanced contrastive loss. *Consistent improvements* are yielded when picking β for other possible values. However, when β becomes larger (the class weight distribution becomes sharper), the performances degrade compared with $\beta = 0.99$.

Ablation: class-balanced queue. In this part, we verify the performances of the class-balanced queue. We define three types of queues here, original queue, reversed queue, and class-balanced queue. There is no constraint on the original queue. Thus, the class distribution of the original queue follows the long-tailed distribution. For the reversed queue, we constrain the class number for each class. The class number of the reversed queue follows the reversed long-tailed class distribution, which means the queue will include more instances for the tail classes than for head classes. As mentioned in Sec. 3.3, we construct the class-balanced queue with an equal class number for all classes. We draw the class distribution visualization of the above three types of queues in Fig. 3. For a fair comparison, the queue sizes for different types are close to 4,000. We train these three variants with the class-balanced contrastive loss on ImageNet-LT with ResNet-50. The detailed performance results are listed at the top of Fig. 3. From it, the class-balanced queue can *achieve the best recognition accuracy*. Moreover, we also measure the expected calibration error (ECE) for



KQ	Q_p	Q_n	Acc.
1000×4	1	500	**59.2%**
1000×4	2	2000	58.6%
1000×4	4	3996	58.3%
1000×8	2	500	**59.0%**
1000×8	2	2000	58.7%
1000×8	4	2000	58.4%
1000×8	8	7992	58.2%

Supervised Hard Positive and Negative Pairs Mining

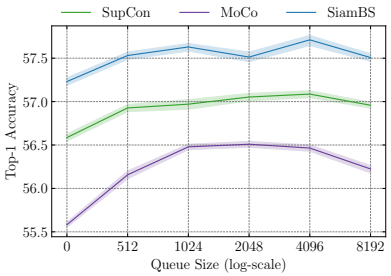


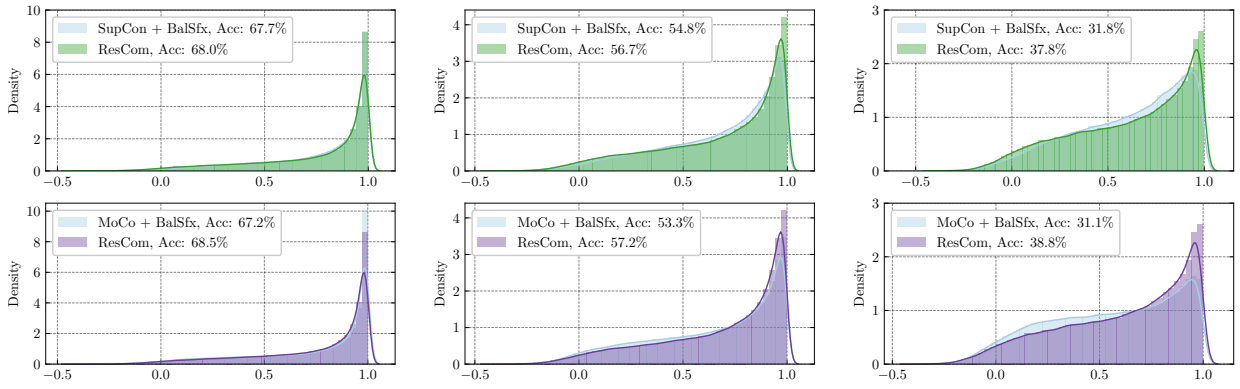
Figure 4: Left: ablation experiment of the class-balanced contrastive loss for different β. Center: ablation study of the SPM model for different Q, Q_p, and Q_n. Right: ablation of three joint contrastive and classification learning mechanisms: SupCon + Balanced Softmax, MoCo + Balanced Softmax, and SupCon + SiamBS.

these variants to measure the confidence calibration performances on long-tailed recognition [54]. The results also show that the class-balanced queue also *increases the calibration robustness* for long-tailed learning. Perhaps due to the fact that there are too few positive contrastive pairs, the reversed queue has not achieved very good results, and its performance is even worse than that of the original queue.

Ablation: supervised hard positive and negative pairs mining. In our SPM model, there are two hyper-parameters Q_p and Q_n to control the mining process of the positive contrastive pairs and negative contrastive pairs. Here we conduct experiments to test the effectiveness of SPM. We train ResCom with different Q, Q_p, and Q_n on ImageNet-LT with ResNeXt-50. We list the top-1 accuracy results upon Q, Q_p, and Q_n in Fig. 4 (center) for some possible variants. As we mentioned in the Sec 3.4, if $Q_p = Q$ and $Q_n = K(Q-1)$, the SPM model will degrade to the plain class-balanced queue model. According to the results (Row1, Row-3 and Row-4, Row-7), the SPM model can further improve the recognition performance by about 0.8% compared with the plain class-balanced queue model. We also observe that smaller Q_p and Q_n tend to achieve better results: We learn that, not all positives and negatives offer significant contributions to the final performances and most positives and negatives do not help a lot towards the whole learning process. Based on mining an appropriate part (usually small) of hard positives and negatives, the model can learn better representative features and further improve the performance.

Ablation: Siamese Balanced Softmax. In the final framework of our ResCom, we involve Siamese Balanced Softmax (SiamBS) in our total loss. To verify the power of the SiamBS part, we build two baseline models here. In the first model, we adopt the Balanced Softmax loss for single view (on $\boldsymbol{s}^{(1)}$) classification learning and SupCon for additional contrastive regularization. In the second model, we adopt the Balanced Softmax loss for single view classification learning and MoCo style contrastive learning for the regularization. We build our model with SiamBS for double views classification learning and SupCon for the contrastive regularization. We draw the results in Fig. 4 (right). From it, our SiamBS yields consistent improvements **0.6%** on ImageNet-LT with the ResNet-50 backbone for all possible queue sizes over the SupCon and MoCo based models. The above results firmly manifest the effectiveness of SiamBS.

Entirety. The above experiments show the effectiveness of the proposed three components, *i.e.*, class-balanced contrastive loss, class-balanced queue, and Siamese Balanced Softmax in ResCom. In this part, we combine these three components together to verify the final performances of ResCom. We set two baseline models: joint training with contrastive loss and Balanced Softmax loss under



Head classes Medium classes Tail classes

Figure 5: Similarity distribution of the positives on the ImageNet-LT validation dataset with ResNet-50. We compare the results of three methods, SupCon + Balanced Softmax, MoCo + Balanced Softmax, and ResCom. From left to right: head classes, medium classes, and tail classes. ResCom maintains the accuracy of the head classes while improving the accuracy of the medium and tail classes (more positive contrastive pairs get higher similarities, *best viewed in color*).

the SupCon [28] framework and the MoCo [18] framework. the SupCon framework can achieve 56.5% top-1 accuracy on ImageNet-LT, and the MoCo framework can achieve 55.7% top-1 accuracy. Under the same setting, ResCom can achieve surprising **58.7%** top-1 accuracy, which is more than **2%** higher than the results of two baseline models. To understand how ResCom rebalances the performance for the head, medium, and the tailed classes, we draw the similarity distribution of the positive contrastive pairs on the ImageNet-LT validation dataset. As shown in Fig. 5, the proposed ResCom maintains the accuracy of the head classes and improves the performances of the medium and tail classes at the same time. Using ResCom, the similarity distributions of the medium classes and the tailed classes will have *more positive pairs close to 1*, which means ResCom *rebalances the similarity distributions* and is more friendly to tailed classes.

4.3 Comparison with State-of-the-arts

To verify the effectiveness, we compare the proposed method ResCom against four groups of state-of-the-art methods on common-used long-tailed datasets: *Logits modification* methods, including LDAM [3], Causal Norm [44], Balanced Softmax [37], and LADE [22]. *Architecture modification* methods, like BBN [56], ELF [15], ResLT [11], GistNet [33], RIDE [49], and DiVE [20]. *Multi-stage* methods such as Decouple [27], DisAlign [53], MiSLAS [54], SSD [30]. *Contrastive learning* methods, *e.g.*, Hybrid-PSC [48], RSG [47], DRO-LT [41], and PaCo [12].

Comparison on CIFAR-LT. The experimental results on CIFAR-10-LT, CIFAR-100-LT are listed in Table 1 and 2, repectively. We mainly compare with the SOTA methods Balanced Softmax [37] and PaCo [12] under the same training setting where Cutout [14] and AutoAugment [10] are used in training. As shown in Table 1 and 2, ResCom consistently outperforms them across all common-used imbalance factors under such a strong setting. Concretely, ResCom surpasses the previous best by **2.8%**, **2.3%** and **0.7%** on CIFAR-10-LT, **1.8%**, **2.0%** and **1.9%** on CIFAR-100-LT, under imbalance factor 100, 50, and 10, respectively, which testifies the powerful effectiveness of ResCom.

Comparison on ImageNet-LT. Table 3 and 4 show extensive experimental results for comparison with recent state-of-the-art methods with the ResNet-50 and ResNeXt-50 backbone. ResCom also

Table 1: Top-1 accuracy (%) comparison on **CIFAR-10-LT** with **ResNet-32** for different imbalance factors.

Imbalanced Factor	100	50	10
LDAM	77.1	81.1	88.2
ELF+LDAM	78.1	82.4	88.0
RSG	79.5	82.8	-
BBN	79.9	82.2	88.4
ResLT	80.5	83.5	89.1
Causal Norm	80.6	83.6	88.5
Hybrid-PSC	81.4	85.4	91.1
MiSLAS	82.1	85.7	90.0
Balanced Softmax	81.5	84.9	91.3
ResCom	**84.9** (+2.8)	**88.0** (+2.3)	**92.0** (+0.7)

Table 2: Top-1 accuracy (%) comparison on **CIFAR-100-LT** with **ResNet-32** for different imbalance factors.

Imbalanced Factor	100	50	10
BBN	42.6	47.0	59.1
Causal Norm	44.1	50.3	59.6
ResLT	45.3	50.0	60.8
LADE	45.4	50.5	61.7
Hybrid-PSC	45.0	49.0	62.4
RIDE(**3 experts**)	48.0	51.7	61.8
MiSLAS	47.0	52.3	63.2
Balanced Softmax	50.8	54.2	63.0
PaCo	52.0	56.0	64.2
ResCom	**53.8** (+1.8)	**58.0** (+2.0)	**66.1** (+1.9)

Table 3: Top-1 accuracy (%) comparison with state-of-the-art methods on **ImageNet-LT** with **ResNet-50**.

Method	Many	Med.	Few	**All**
cRT	62.5	47.4	29.5	50.3
LWS	61.8	48.6	33.5	51.2
ELF	64.3	47.9	31.4	52.0
MiSLAS	61.7	51.3	35.8	52.7
DisAlign	61.3	52.2	31.4	52.9
DRO-LT	64.0	49.8	33.1	53.7
RIDE(**4 experts**)	66.2	52.3	36.5	55.4
PaCo	65.0	55.7	38.2	57.0
ResCom	**68.5**	**57.2**	**38.8**	**58.7** (+1.7)

Table 4: Top-1 accuracy (%) comparison with state-of-the-art methods on **ImageNet-LT** with **ResNeXt-50**.

Method	Many	Med.	Few	**All**
LWS	60.5	47.2	31.2	50.1
Causal Norm	65.2	47.7	29.8	52.0
Balanced Softmax	63.6	48.4	32.9	52.1
LADE	65.1	48.9	33.4	53.0
DiVE	64.1	50.5	31.5	53.1
SSD	66.8	53.1	35.4	56.0
RIDE(**4 experts**)	68.2	53.8	36.0	56.8
PaCo	67.5	**56.9**	36.7	58.2
ResCom	**68.9**	56.7	**40.3**	**59.2** (+1.0)

surpasses all previous methods by large margins: **1.7%** for ResNet-50, and **1.0%** for ResNeXt-50. One more thing we would like to mention is RIDE-based ensemble methods have higher inference latency than standard network based methods like ResCom. The inference time comparison results are listed in Table 5. From it, RIDE [49] (or DiVE [20]) with four experts takes **18.5ms** for inference with a batch of 64 images on the Nvidia GeForce 2080Ti GPU. While ResCom is a single model and just takes **8.3ms** under the same situation for ResNet-50.

Comparison on Places-LT. The experimental results on Places-LT are summarized in Table 6. Our ResCom is flexible to reload the existing pre-trained models because the main network architecture is the same as those of [27, 34]. However, due to the architecture change of RIDE [49] and its variants, it is not applicable to load the publicly pre-trained models. Under a fair training setting by finely tuning 30 epochs without additional augmentation, ResCom also outperforms all previous methods.

Comparison on iNaturalist 2018. Table 7 lists experimental results on iNaturalist 2018. Under a fair training setting, ResCom consistently surpasses recent state-of-the-art methods *on all accuracy measurements* (many, medium, few, and all). ResCom improves top-1 accuracy by a large margin **1.8%**.

Table 5: Top-1 accuracy (%) and inference time (ms, with a batch of 64 images) results of ResCom and RIDE [49] on **ImageNet-LT**.

Backbone	Method	**Infr. time**	Many	Med.	Few	**All**
ResNet-50	RIDE (3 experts)	15.3 (+64%)	66.2	51.7	34.9	54.9
	RIDE (4 experts)	18.5 (+122%)	66.2	52.3	36.5	55.4
	ResCom	**8.3**	**68.5**	**57.2**	**38.8**	**58.7** (+3.3)
ResNeXt-50	RIDE (3 experts)	26.0 (+100%)	67.6	53.5	35.9	56.4
	RIDE (4 experts)	33.2 (+155%)	68.2	53.8	36.0	56.8
	ResCom	**13.0**	**68.9**	**56.7**	**40.3**	**59.2** (+2.4)

Table 6: Top-1 accuracy (%) comparison on **Places-LT**, starting from an ImageNet pre-trained **ResNet-152**.

Method	Many	Med.	Few	**All**
LWS	40.6	39.1	28.6	37.6
τ-norm	37.8	40.7	31.8	37.9
Balanced Softmax	42.0	39.3	30.5	38.6
LADE	42.8	39.0	31.2	38.8
RSG	41.9	41.4	32.0	39.3
DisAlign	40.4	42.4	30.1	39.3
GistNet	42.5	40.8	32.1	39.6
ResLT	39.8	43.6	31.4	39.8
MiSLAS	39.6	43.3	**36.1**	40.4
PaCo	37.5	**47.2**	33.9	41.2
ResCom	**43.0**	43.4	35.3	**41.7** (+0.5)

Table 7: Top-1 accuracy (%) comparison with state-of-the-art methods on **iNaturalist 2018** with **ResNet-50**.

Method	Many	Med.	Few	**All**
ELF	72.7	70.4	68.3	69.8
LADE	-	-	-	70.0
RSG	-	-	-	70.3
DisAlign	-	-	-	70.6
GistNet	-	-	-	70.8
SSD	-	-	-	71.5
MiSLAS	70.4	72.4	73.2	71.6
RIDE(**4 experts**)	70.9	72.4	73.1	72.6
PaCo	70.3	73.2	73.6	73.2
DiVE(**4 experts**)	-	-	-	73.4
ResCom	**71.6**	**75.0**	**75.6**	**75.2** (+1.8)

5 Conclusion

In this paper, we observe and mathematically analyze that SupCon suffers a *dual class-imbalance* problem at both the *original batch-level* and the *Siamese batch-level*, which is more difficult than it in long-tailed classification learning. We present Rebalanced Siamese Contrastive Mining (ResCom) to address it: we introduce a class-balanced supervised contrastive loss for the original batch-level, a class-balanced queue for the Siamese batch-level. Moreover, we noted that easy positives and negatives make the contrastive gradient vanish and may disturb the representation learning. We propose supervised hard positive and negative pairs mining, and Siamese Balanced Softmax to dig for more useful information. Based on the experiments, all proposed modules have been verified to be helpful for long-tailed recognition and ResCom *effectively rebalances the similarity distributions* and *improves the performances of both the medium and tail classes*, which is consistent with our original goal. Additionally, ResCom *surpasses* the previous competitors by *large margins* on multiple popular long-tailed benchmark datasets.

References

[1] Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks **106**, 249–259 (2018) 3

[2] Byrd, J., Lipton, Z.: What is the effect of importance weighting in deep learning? In: ICML. pp. 872–881 (2019) 3

[3] Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. In: NuerIPS. pp. 1567–1578 (2019) 4, 8, 11

[4] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: NuerIPS (2020) 1, 19

[5] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607 (2020) 3, 6, 19

[6] Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020) 3

[7] Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR. pp. 15750–15758 (2021) 1, 19

[8] Chuang, C.Y., Robinson, J., Lin, Y.C., Torralba, A., Jegelka, S.: Debiased contrastive learning. In: NuerIPS. pp. 8765–8775 (2020) 6

[9] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: CVPR. pp. 3213–3223 (2016) 1

[10] Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: AutoAugment: Learning augmentation policies from data. In: CVPR. pp. 113–123 (2019) 11

[11] Cui, J., Liu, S., Tian, Z., Zhong, Z., Jia, J.: ResLT: Residual learning for long-tailed recognition. arXiv preprint arXiv:2101.10633 (2021) 11

[12] Cui, J., Zhong, Z., Liu, S., Yu, B., Jia, J.: Parametric contrastive learning. In: ICCV. pp. 715–724 (2021) 4, 8, 11, 19

[13] Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: CVPR. pp. 9268–9277 (2019) 3, 5, 8

[14] DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017) 11

[15] Duggal, R., Freitas, S., Dhamnani, S., Chau, D.H., Sun, J.: ELF: An early-exiting framework for long-tailed classification. arXiv preprint arXiv:2006.11979 (2020) 11

[16] Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018) 4

[17] Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., et al.: Bootstrap your own latent: A new approach to self-supervised learning. In: NuerIPS (2020) 3, 19

[18] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR. pp. 9729–9738 (2020) 2, 3, 4, 5, 6, 8, 11, 19

[19] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 1, 3, 5, 8

[20] He, Y.Y., Wu, J., Wei, X.S.: Distilling virtual examples for long-tailed recognition. In: ICCV (2021) 11, 12, 20

[21] Henaff, O.: Data-efficient image recognition with contrastive predictive coding. In: ICML. pp. 4182–4192 (2020) 6

[22] Hong, Y., Han, S., Choi, K., Seo, S., Kim, B., Chang, B.: Disentangling label distribution for long-tailed visual recognition. In: CVPR. pp. 6626–6636 (2021) 4, 11, 19

[23] Huang, C., Li, Y., Loy, C.C., Tang, X.: Learning deep representation for imbalanced classification. In: CVPR. pp. 5375–5384 (2016) 3

[24] Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent data analysis **6**(5), 429–449 (2002) 3

[25] Kalantidis, Y., Sariyildiz, M.B., Pion, N., Weinzaepfel, P., Larlus, D.: Hard negative mixing for contrastive learning. In: NuerIPS. pp. 21798–21809 (2020) 6

[26] Kang, B., Li, Y., Xie, S., Yuan, Z., Feng, J.: Exploring balanced feature spaces for representation learning. In: ICLR (2020) 4

[27] Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: ICLR (2020) 8, 11, 12, 19

[28] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: NuerIPS (2020) 1, 3, 4, 8, 11, 19

[29] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NuerIPS. pp. 1097–1105 (2012) 1

[30] Li, T., Wang, L., Wu, G.: Self supervision to distillation for long-tailed visual recognition. In: ICCV (2021) 11

[31] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2980–2988 (2017) 3

[32] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV. pp. 740–755 (2014) 1

[33] Liu, B., Li, H., Kang, H., Hua, G., Vasconcelos, N.: GistNet: a geometric structure transfer network for long-tailed recognition. In: ICCV (2021) 11

[34] Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: CVPR. pp. 2537–2546 (2019) 8, 9, 12

[35] Menon, A.K., Jayasumana, S., Rawat, A.S., Jain, H., Veit, A., Kumar, S.: Long-tail learning via logit adjustment. In: ICLR (2020) 4

[36] Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. In: NuerIPS (2018) 3, 5

[37] Ren, J., Yu, C., sheng, s., Ma, X., Zhao, H., Yi, S., Li, h.: Balanced meta-softmax for long-tailed visual recognition. In: NuerIPS. pp. 4175–4186 (2020) 4, 8, 11, 19

[38] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NuerIPS. pp. 91–99 (2015) 1

[39] Robinson, J., Chuang, C.Y., Sra, S., Jegelka, S.: Contrastive learning with hard negative samples. In: ICLR (2021) 6

[40] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: ImageNet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015) 1, 9

[41] Samuel, D., Chechik, G.: Distributional robustness loss for long-tail learning. In: ICCV (2021) 4, 11

[42] Shen, L., Lin, Z., Huang, Q.: Relay backpropagation for effective learning of deep convolutional neural networks. In: ECCV. pp. 467–482 (2016) 3

[43] Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., Yan, J.: Equalization loss for long-tailed object recognition. In: CVPR. pp. 11662–11671 (2020) 3

[44] Tang, K., Huang, J., Zhang, H.: Long-tailed classification by keeping the good and removing the bad momentum causal effect. In: NuerIPS (2020) 11

[45] Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: ECCV. pp. 776–794 (2020) 6

[46] Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The iNaturalist species classification and detection dataset. In: CVPR. pp. 8769–8778 (2018) 9

[47] Wang, J., Lukasiewicz, T., Hu, X., Cai, J., Xu, Z.: RSG: A simple but effective module for learning imbalanced datasets. In: CVPR. pp. 3784–3793 (2021) 11

[48] Wang, P., Han, K., Wei, X.S., Zhang, L., Wang, L.: Contrastive learning based hybrid networks for long-tailed image classification. In: CVPR. pp. 943–952 (2021) 4, 11

[49] Wang, X., Lian, L., Miao, Z., Liu, Z., Yu, S.X.: Long-tailed recognition by routing diverse distribution-aware experts. In: ICLR (2021) 11, 12, 13, 19, 20

[50] Wang, Y.X., Ramanan, D., Hebert, M.: Learning to model the tail. In: NuerIPS. pp. 7029–7039 (2017) 3

[51] Wei, X.S., Wang, P., Liu, L., Shen, C., Wu, J.: Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples. IEEE TIP **28**(12), 6116–6125 (2019) 9

[52] Yang, Y., Xu, Z.: Rethinking the value of labels for improving class-imbalanced learning. In: NuerIPS. vol. 33 (2020) 4, 8

[53] Zhang, S., Li, Z., Yan, S., He, X., Sun, J.: Distribution alignment: A unified framework for long-tail visual recognition. In: CVPR. pp. 2361–2370 (2021) 11, 19

[54] Zhong, Z., Cui, J., Liu, S., Jia, J.: Improving calibration for long-tailed recognition. In: CVPR (2021) 8, 10, 11, 19

[55] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE TPAMI **40**(6), 1452–1464 (2017) 1, 9

[56] Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: CVPR. pp. 9719–9728 (2020) 8, 11

Supplementary Material

A Proof of Dual Class-imbalance in the SupCon Loss

Definition 1 *Given a dataset $\mathcal{D}$ with totally N samples and K categories, where each category contains $n_1, n_2, \ldots, n_K$ samples, respectively (w.l.o.g, $N_1 \geq N_2 \geq \ldots \geq N_K$). Let $\mathcal{B}_t = \{b_{t,i}\}_{i=1}^{|B|}$ denote the t-th batch set obtained by random sampling, where $b_{t,i}$ is an instance in the batch $\mathcal{B}_t$, $1 \leq t \leq T$, T represents the total number of batches in one epoch. We also define $B_{t,k}$ as the number of samples which belongs to the Class-k in batch $\mathcal{B}_t$. For memory queue, $\mathcal{Q}_t = \{q_{t,i}\}_{i=1}^{|Q|}$ denote the memory bank composed of $|Q|$ samples, where $b_{t,i}$ is an instance in the queue $\mathcal{Q}_t$. We also define $Q_{t,k}$ as the number of samples which belongs to the Class-k in $\mathcal{Q}_t$. Thus, $P_{t,k} = B_{t,k}Q_{t,k}$ denotes the number of sample pairs $(b_{t,i}, q_{t,j})$ in which $b_{t,i}$, $q_{t,j}$ belong to the same Class-k (a positive pair). The contrastive imbalance ratio β is defined as:*

$$\beta = \frac{\max\limits_k \left[\mathbb{E}_{Q_{t,k}}\left(\sum\limits_{t=1}^{T} P_{t,k}\right)\right]}{\min\limits_k \left[\mathbb{E}_{Q_{t,k}}\left(\sum\limits_{t=1}^{T} P_{t,k}\right)\right]}.$$

Proposition 1 *Assume that each batch $\mathcal{B}_t$ in the training process is randomly sampled from the dataset $\mathcal{D}$ (without replacement), and the memory bank $\mathcal{Q}_t$ is enqueued by the previous batches. Thus, the random variable $Q_{t,k}$ follows a hypergeometric distribution $H(N, |Q|, N_k)$, and its expected value is $\frac{N_k|Q|}{N}$:*

$$\begin{aligned}
\mathbb{E}(Q_{t,k}) &= \sum_{n=0}^{|Q|} np(n) = \sum_{n=0}^{|Q|} \frac{n\binom{N_k}{n}\binom{N-N_k}{|Q|-n}}{\binom{N}{|Q|}} \\
&= \sum_{n=0}^{|Q|} \binom{N-N_k}{|Q|-n} \frac{nN_k!}{n!(N_k-n)!} \frac{|Q|!(N-|Q|)!}{N!} \\
&= \sum_{n=0}^{|Q|} \binom{N-N_k}{|Q|-n} \frac{N_k(N_k-1)!}{(n-1)!(N_k-n)!} \frac{|Q|(|Q|-1)!(N-|Q|)!}{N(N-1)!} \\
&= \frac{N_k|Q|}{N} \sum_{n=1}^{|Q|} \frac{\binom{N-N_k}{|Q|-n}\binom{N_k-1}{n-1}}{\binom{N-1}{|Q|-1}} \\
&= \frac{N_k|Q|}{N} \frac{1}{\binom{N-1}{|Q|-1}} \sum_{n=1}^{|Q|} \binom{N-N_k}{|Q|-n}\binom{N_k-1}{n-1} \\
&= \frac{N_k|Q|}{N} \frac{1}{\binom{N-1}{|Q|-1}} \sum_{n=0}^{|Q|-1} \binom{N-N_k}{|Q|-n-1}\binom{N_k-1}{n} \\
&= \frac{N_k|Q|}{N} \frac{\binom{N-1}{|Q|-1}}{\binom{N-1}{|Q|-1}} = \frac{N_k|Q|}{N}.
\end{aligned}$$

Then, we can get:

$$\mathbb{E}_{Q_{t,k}}\left(\sum_{t=1}^{T} P_{t,k}\right) = \sum_{t=1}^{T} B_{t,k}\mathbb{E}\left(Q_{t,k}\right) = N_k\mathbb{E}\left(Q_{t,k}\right) = \frac{N_k^2|Q|}{N}.$$

Thus, according to **Definition 1**, *the contrastive imbalance ratio* β *of the dataset (in one epoch) is:*

$$\beta = \frac{\max\limits_k\left[\mathbb{E}_{Q_{t,k}}\left(\sum\limits_{t=1}^{T} P_{t,k}\right)\right]}{\min\limits_k\left[\mathbb{E}_{Q_{t,k}}\left(\sum\limits_{t=1}^{T} P_{t,k}\right)\right]} = \frac{\max_k(\frac{N_k^2|Q|}{N})}{\min_k(\frac{N_k^2|Q|}{N})} = \frac{N_1^2}{N_K^2} = \frac{N_{\max}^2}{N_{\min}^2}.$$

B Implementation Details

As Chen *et al*. [5] concluded, contrastive learning usually benefits from longer training times compared with traditional supervised learning with the cross-entropy loss, which is also validated by previous work, *e.g*., MoCo [18], BYOL [17], SWAV [4], and SimSiam [7]. They trained their models in 800 epochs for convergence. Supervised contrastive learning [28] also trains 350 epochs for feature representation learning and another 350 epochs for classification learning. Thus, we run ResCo with 400 epochs on CIFAR-10-LT, CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018 for better converge of contrastive learning. For Places-LT, we follow previous work [12, 27] by loading the pre-trained model from the full ImageNet dataset and fine-tuning it for 30 epochs on Places-LT to *prevent over-fitting*. All models are trained using the SGD optimizer with momentum $\mu = 0.9$. For simplicity, we set the loss weight hyper-parameter $\lambda = 0.5$ and the contrastive loss temperature $\tau = 0.2$.

CIFAR-10-LT & CIFAR-100-LT

We use ResNet-32 as the backbone and strictly follow the setting of [12, 37] for fair comparison. We train ResCo on one GPU with batch size 128. The learning rate decays by a multi-step scheduler from 0.1 to 0.001.

ImageNet-LT

We used ResNet-50 and ResNeXt-50 as our backbones and strictly follow the setting of [12, 49] for fair comparison. We train ResCo on 8 GPUs with batch size 256. The learning rate decays by a cosine scheduler from 0.06 to 0 for 400 epochs.

Places-LT

Following previous setting [12, 22, 27, 53, 54], we choose ResNet-152 as the backbone network, which is pre-trained from the full ImageNet dataset (provided by PyTorch). We fine-tune it for 30 epochs. Similar to ImageNet-LT, the learning rate decays by a cosine scheduler from 0.02 to 0 with batch size 256 on 8 GPUs.

iNaturalist 2018

Following the previous setting [12, 49], we conduct experiments with ResNet-50. Similar to ImageNet-LT, the learning rate decays by a cosine scheduler from 0.08 to 0 with batch size 256 on 8 GPUs for 400 epochs.

C More Comparison Results about Inference Time

As we mentioned in Sec. 3.4 and Sec. 4.3, RIDE-style ensemble methods [20, 49] have more computations cost for both training and evaluation, while ResCom is very efficient. We show the comprehensive results with RIDE in Table 8. From these tables, ResCom can *achieve better recognition results* and *a faster inference process* simultaneously.

Table 8: Detailed top-1 accuracy (%) and inference time (ms) results of ResCom and RIDE [49] on **iNaturalist 2018**. Inference time is measured by evaluating with a batch of 64 images on the Nvidia GeForce 2080Ti GPU.

Backbone	Method	**Infr. time**	Many	Med.	Few	**All**
ResNet-50	RIDE (2 experts)	12.0 (+44%)	70.2	71.3	71.7	71.4
	RIDE (3 experts)	15.3 (+64%)	70.2	72.2	72.7	72.2
	RIDE (4 experts)	18.5 (+122%)	70.9	72.4	73.1	72.6
	ResCom	**8.3**	**71.6**	**75.0**	**75.6**	**75.2** (+2.6)

D Gradient Derivation of Eq.(3)

Here we derive the gradient of the class-balanced queue version of supervised contrastive loss Eq.(3):

$$
\begin{aligned}
\frac{\partial L_i^{\mathrm{BQ}}}{\partial \boldsymbol{z}_i^{(1)}} &= \frac{-w_{y_i}}{Q} \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \frac{\partial}{\partial \boldsymbol{z}_i^{(1)}} \left\{ \frac{\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_p}{\tau} - \log \sum_{k=1}^{K} \sum_{\boldsymbol{z}_q \in \mathcal{Q}_k} \exp\left(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_q / \tau\right) \right\}, \\
&= \frac{-w_{y_i}}{\tau Q} \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \left\{ \boldsymbol{z}_p - \frac{\sum_{k=1}^{K} \sum_{\boldsymbol{z}_q \in \mathcal{Q}_k} \boldsymbol{z}_q \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_q / \tau\right)}{\sum_{k=1}^{K} \sum_{\boldsymbol{z}_q \in \mathcal{Q}_k} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_q / \tau\right)} \right\}, \\
&= \frac{-w_{y_i}}{\tau Q} \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \left\{ \boldsymbol{z}_p - \sum_{\boldsymbol{z}_{p'} \in \mathcal{Q}_{y_i}} \boldsymbol{z}_{p'} P_{ip'} - \sum_{k \neq y_i} \sum_{\boldsymbol{z}_n \in \mathcal{Q}_k} \boldsymbol{z}_n P_{in} \right\}, \\
&= \frac{-w_{y_i}}{\tau Q} \left\{ \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \boldsymbol{z}_p - \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \sum_{\boldsymbol{z}_{p'} \in \mathcal{Q}_{y_i}} \boldsymbol{z}_{p'} P_{ip'} - \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \sum_{k \neq y_i} \sum_{\boldsymbol{z}_n \in \mathcal{Q}_k} \boldsymbol{z}_n P_{in} \right\}, \\
&= \frac{-w_{y_i}}{\tau Q} \left\{ \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \boldsymbol{z}_p - \sum_{\boldsymbol{z}_{p'} \in \mathcal{Q}_{y_i}} \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \boldsymbol{z}_{p'} P_{ip'} - \sum_{k \neq y_i} \sum_{\boldsymbol{z}_n \in \mathcal{Q}_k} \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \boldsymbol{z}_n P_{in} \right\}, \\
&= \frac{-w_{y_i}}{\tau Q} \left\{ \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \boldsymbol{z}_p - \sum_{\boldsymbol{z}_{p'} \in \mathcal{Q}_{y_i}} Q\, \boldsymbol{z}_{p'} P_{ip'} - \sum_{k \neq y_i} \sum_{\boldsymbol{z}_n \in \mathcal{Q}_k} Q\, \boldsymbol{z}_n P_{in} \right\}, \\
&= \frac{-w_{y_i}}{\tau Q} \left\{ \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \boldsymbol{z}_p - \sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} Q\, \boldsymbol{z}_p P_{ip} - \sum_{k \neq y_i} \sum_{\boldsymbol{z}_n \in \mathcal{Q}_k} Q\, \boldsymbol{z}_n P_{in} \right\}, \\
&= \frac{w_{y_i}}{\tau} \Bigg\{ \underbrace{\sum_{\boldsymbol{z}_p \in \mathcal{Q}_{y_i}} \boldsymbol{z}_p \left(P_{ip} - \frac{1}{Q}\right)}_{\text{positive pairs}} + \underbrace{\sum_{k \neq y_i} \sum_{\boldsymbol{z}_n \in \mathcal{Q}_k} \boldsymbol{z}_n P_{in}}_{\text{negative pairs}} \Bigg\},
\end{aligned}
$$

where we define:

$$
P_{ij} = \frac{\exp\left(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_j / \tau\right)}{\sum_{k=1}^{K} \sum_{\boldsymbol{z}_q \in \mathcal{Q}_k} \exp(\boldsymbol{z}_i^{(1)} \cdot \boldsymbol{z}_q / \tau)}.
$$

As we mentioned in Sec. 3.3, the gradient of the class-balanced queue version of supervised contrastive loss can be decoupled into the positive pairs part and the negative pairs part. In addition, for most easy positive pairs, $P_{ip} \to \frac{1}{Q}$, and for most easy negative pairs, $P_{in} \to 0$. Both two cases will make the above contrastive gradient vanish, damage or disturb the whole learning process.