A. APPENDIX

A.1. Additional experiment results

Accuracy of each class on CIFAR-100-LT. We visualize the accuracy of each class of both SCL and SBCL on CIFAR-100-LT with imbalance ratio 100 (Figure 5). From the results, we can see that SBCL improves performance on tail classes over SCL without the expense of the perforamnce of the head classes.

Selection of negative instances in SBCL. Our proposed loss in Eq. 4 consists of two supervised contrastive losses with subclass and class labels respectively. The first term regards instances in different subclasses as negative instead of these in different classes; In Table 7, we show that such design choice leads to better performance than using instances in different classes as negative, which illustrates the effectiveness of exploiting the rich semantic in head classses.

Table 7: Different selection of negative instances in SBCL on CIFAR-100-LT with different imbalance ratio. The 'Class label' row are the first term of loss function is constructed by class label and the 'Subclass label' row are subclass label.

Negative samples	Imbalance Ratio		
	100	50	10
Class label	43.7	47.6	56.8
Subclass label	**44.9**	**48.7**	**57.9**

Analysis of feature distribution. To analyze the representation learned by SBCL, we firstly define the euclidean distance between a given sample and other samples from the same/different classes as intra/inter-class distance. Concretely, the euclidean distance between a sample z_i and a set S is defined as $\mathbf{D}(z_i, S) = \frac{1}{|S|}\sum_{z_j \in S} \|z_i - z_j\|_2$. Then, the intra- and inter-class distance of sample z_i can be defined as $\mathbf{D}(z_i, P_i)$ and $\mathbf{D}(z_i, \mathcal{D}/P_i)$ separately; and the intra- and inter-subclass distance of sample z_i can be defined as $\mathbf{D}(z_i, M_i)$ and $\mathbf{D}(z_i, P_i/M_i)$ separately.

To leverage instance semantic coherence to balance the feature space, we expect instances of high semantic coherence to form a more concentrated cluster than other instances in the same class. So, we embed the subclass-balancing adaptive clustering strategy on SBCL to illustrate this on CIFAR-100-LT with imbalance ratio 100. In Table 8, we report the intra-subclass/inter-subclass distance on different splits. The results show that SBCL achieves to concentrate instances from the same subclass and pulls instances from different subclasses away on all splits.

SBCL aims at learning a compact representation space, in which representations from different classes are far from each other and the feature space spanned by representations of each class is invariant to the long-tailed distribution. The average intra/inter-class distance are summarized in Table 8 and the distances of different groups are reported separately. The results also show SBCL clears the boundary of feature distribution on the different class splits.

Table 8: Average intra/inter-subclass and intra/inter-class distance of features learned by SBCL.

Distance	Many	Medium	Few	All
Intra-subclass	0.68	0.76	0.87	0.70
Inter-subclass	1.02	0.99	0.89	1.01
Intra-class	1.00	0.94	0.88	0.99
Inter-class	1.39	1.38	1.37	1.39

Subclass-balancing adaptive clustering. We propose an noval K-means algorithm aimed at achieving balanced intra-cluster sample quantities.

In the initial step, we adopt a strategy to select cluster center points that are maximally distant from each other. This ensures an optimal distribution of initial cluster centers [3].

Next, in the assignment step, we calculate the similarity between each sample point and the selected cluster centers. The sample points are then assigned to the cluster centers in descending order of their similarity scores. However, we introduce a modification to this step by incorporating a constraint on the maximum number of samples assigned to a given cluster center. Once this threshold is reached, the cluster center is not eligible to receive any samples.

In the update step, we revise the cluster centers which are determined as the average values of the samples allocated to this cluster.

We continue to iterate through the assignment and update steps until we satisfy the termination condition, which is achieving a predetermined number of iterations (M). This iterative process facilitates a balanced distribution of samples within each cluster, leading to release the long-tailed phenomena.

Hyperparameter analysis on CIFAR-100-LT. Figure 6a and Table 9 show the distribute situation of sample number in subclasses obtained by different cluster algorithms on CIFAR-100-LT with imbalance ratio 100. For Kmean cluster algorithm, the imbalance phenomenon of subclasses is obvious. When using our proposed cluster algorithm, the imbalance ratio of sample number in subclasses deceases from 40 to 9.5. And the standard deviation of sample number on CIFAR-100-LT is relatively small, which denotes the number of samples in most subclasses keeps stable in a certain range.

Figure 6b shows the impact of batch size on SBCL/TSC. We find that larger batch sizes have a significant advantage

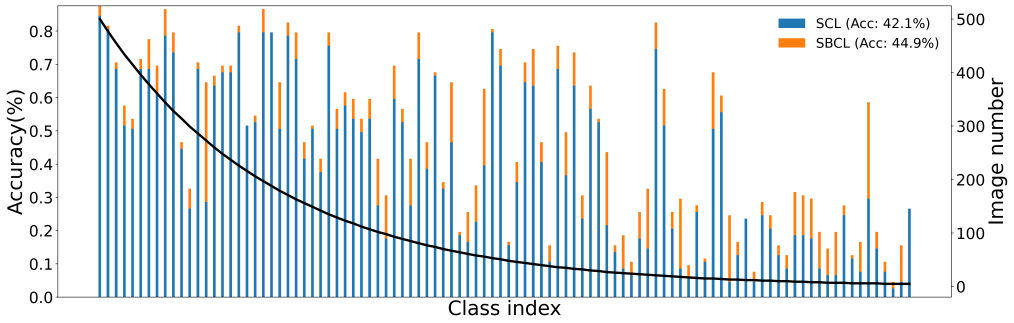

Figure 5: Accuracy of each classes on CIFAR100-LT. The black line is the class distribution, and the classes in the left part are head classes while those in the right part are tail classes.

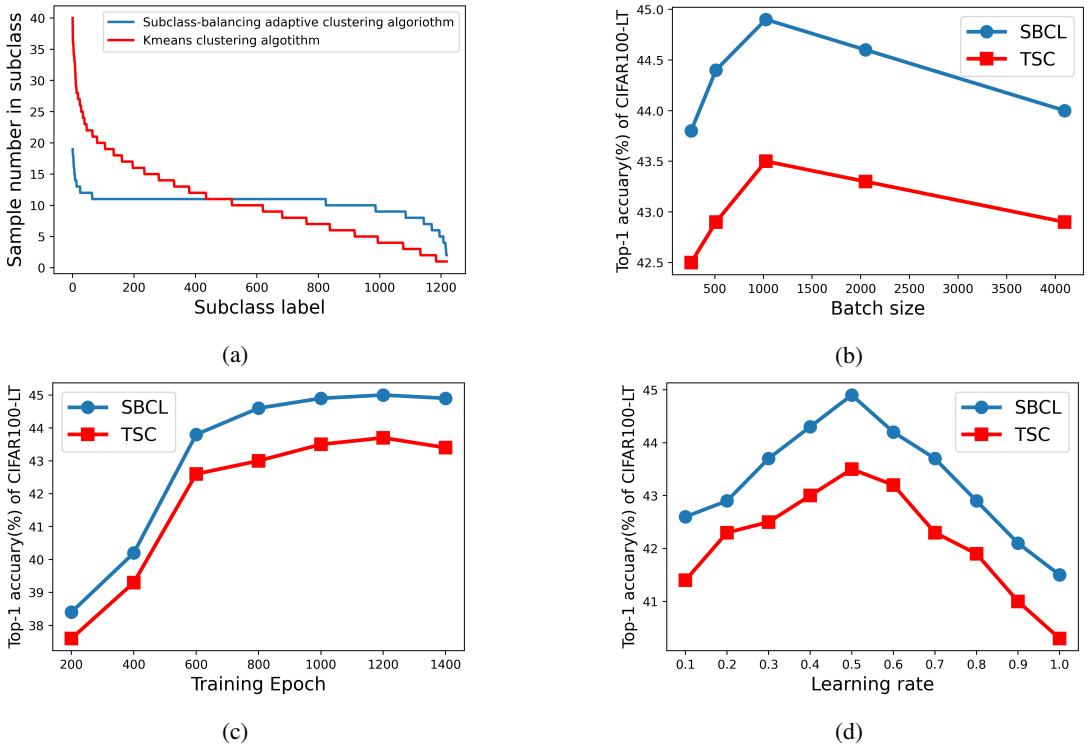

Figure 6: **Analysis of SBCL as a loss function of different hyperparameters on CIFAR-100-LT with imbalance ratio 100.** (a): Sample number in clusters with different cluster algorithm. (b): Top-1 accuracy of SBCL/TSC as a function of different batch size. (c): Top-1 accuracy of SBCL/TSC as a function of different pretraining epochs. (d): Top-1 accuracy of SBCL/TSC as a function of different learning rates.

over the smaller ones. This is because larger batch sizes provide more negative examples to facilitate convergence. However, the over-large batch size hurts the model performance. And SBCL and TSC are equally sensitive to batch size on CIFAR-100-LT.

Figure 6c shows the curve of the accuracy of SBCL/TSC vs. the number of training epochs. From the curve, we can see that the performance of SBCL and TSC both converge after 800 epochs. When the model is trained with SBCL over 600 epochs, its performance already exceeds TSC.

In Figure 6d, we display the performance of SBCL with different learning rates on CIFAR-100-LT with imbalance ratio 100. As shown in the figure, the learning rate has significant impact on the performance, and we set the learning rate as 0.5 for CIFAR-100-LT.

Combining TSC with dynamic temperature. According to Table 5, dynamic temperature effectively contributes to

Table 9: Distribution of sample number in subclass on CIFAR-100-LT with imbalance ratio 100.

Dataset	Max	Min	Average	Std	Imbalance ratio(Max/Min)
Kmean clustering algorithm	40	1	10.34	6.27	40
Subclass-balancing adaptive clustering algorithm	19	2	10.34	1.60	9.5

the improvement of accuracy. We also add the dynamic temperature to the second term of TSC [41] and the experiment results are shown in Table 10. However, the improvement of the dynamic temperature on TSC is less significant than that on our method, which is reasonable because we introduce dynamic temperature for the loss to distinguish between class and subclass, while TSC does not have subclass and therefore the dynamic temperature is less effective.

Time efficiency comparison of SBCL and TSC. In Table 11, SBCL consumes more time, because we use a clustering algorithm on head classes to train SBCL on CIFAR-100-LT with imbalance ratio 100. However, we update the clustering results every few training iterations, and ultimately achieve better results than TSC. Therefore, we believe that the additional small computational cost is worth the effort.

Warm-up on ImageNet-LT. Instead of using the SCL at the warm-up stage for CIFAR datasets, KCL is adopted for ImageNet-LT and iNaturalist 2018 datasets to warm up the feature extractor. As Table 12 shows, warm-up phase makes feature extractor improve accuracy on all splits of ImageNet-LT. This is because it prevents cluster assignment from feature random distribution at the beginning and avoids using the SCL to make the feature space dominated by the head class at the warm-up stage.

Advantages of cluster validity. Actually, previous studies [33, 41] have proven that randomly sampling balanced instances as positive pairs (such as KCL, TSC) is better than sampling all instances of the same class as positive pairs (such as SCL). However, this strategy may destruct instance semantic coherence. In Table 13, we replace the first team (regard subclasses as positive pairs) with the balanced positive sampling strategy (KCL) to prove this on ImageNet-LT. As the results show, subclass-balancing adaptive clustering strategy brings more improvement to SBCL than balanced positive sampling strategy.

COCO object detection and instance segmentation. In this section, following the experiment setting in [24], we use Mask R-CNN [25] to conduct the object detection and instance segmentation experiments on COCO dataset. The schedule is the default 2× in [24]. Table 14 shows the pretrained model trained by SBCL outperforms it learned with other contrastive learning for the downstream tasks.

Hyperparameter studies. Here, we study the effect of hyperparameters β and δ. Note that β controls the balance of two loss terms in Eq. 4 and δ determines the lower bound of the cluster size in Eq. 3. Specifically, on CIFAR-100-LT with imbalance ratio 100, we vary the values of β from $\{0.1, 0.2, 0.5, 0.8, 1.0, 2.0\}$ with $\delta = 10$ and the value of δ from $\{5, 10, 20, 30, 50, 100\}$ with $\beta = 0.2$. The results are summarized in Table 15. We observe that the smaller β values (between 0.1 and 0.5) can achieve relatively good performance, with the best being 0.2. This observation aligns with our intuition of emphasizing the subclass-level contrastive loss, because smaller β is equivalent to putting more weights on the first term of Eq. 4, which corresponds to the subclass-level contrastive. For δ, the values between 5 and 30 yield high accuracy, with the best being 10. We can see that large δ values ($\delta = 50, 100$) lead to significant drop in performance. We argue that this is because large δ value would result in subclasses that contain more instance than tail classes and therefore affect the subclass-balance, leading to suboptimal performance. In addition, smaller δ value ($\delta = 5$) also causes performance drop; the reason could be small cluster size may let similar instance being assigned to different clusters and therefore affect the learned representations. Therefore, we fix $\beta = 0.2$ and $\delta = 10$ for all experiments.

Visualization of generated clusters. In Figure 7, we show the clustering results of ImageNet-LT training images generated by subclass-balancing adaptive clustering algorithm. From the results, we can see that the algorithm is able to find the subclasses with similar patterns, helping the model learn semantic coherent representations. For example, the two subclasses in the bottom-left are telephone with/without human.

A.2. Additional information

Benchmark datasets statistical information and Implementation Details. We summarize the statistical information of the three benchmark datasets in Table 16. Following [33, 34, 41], we apply SBCL on the long-tailed recognition by using a two-stage training strategy: (i) train the representation with SBCL; (ii) learn a linear classifier on top of the fixed representation. The training process is the same as TSC [41]. Thus, we use TSC default hyperparameters and implementation details for the representation learning. For CIFAR-100-LT dataset, all experiments are performed on 2

Table 10: Combination of TSC and SBCL with dynamic temperature.

Dynamic temperature	CIFAR-100-LT					
	TSC			SBCL		
Imbalance Ratio	100	50	10	100	50	10
	43.5	47.6	58.7	43.8	47.8	57.0
✓	43.9(+0.4)	48.0(+0.4)	59.2(+0.5)	44.9(+1.1)	48.7(+0.9)	57.9(+0.9)

Table 11: Computing cost (GPU hours) on CIFAR-100-LT dataset with imbalance ratio 100.

Method	TSC	SBCL
GPU hours	2	2.4

Table 12: SBCL with and without warm-up stage on ImageNet-LT.

Method	Many	Medium	Few	All
SBCL without warm-up	62.9	49.6	29.3	52.0
SBCL with warm-up	**63.8**	**51.3**	**31.2**	**53.4**

Table 13: Subclass-balancing adaptive clustering strategy improves more than balanced positive sampling strategy on ImageNet-LT.

Method	Many	Medium	Few	All
FCL	61.4	47.0	28.2	49.8
KCL	62.4	49.0	29.5	51.5
TSC	63.5	49.7	30.4	52.4
SBCL (KCL)	63.3	49.5	30.6	52.2
SBCL	**63.8**	**51.3**	**31.2**	**53.4**

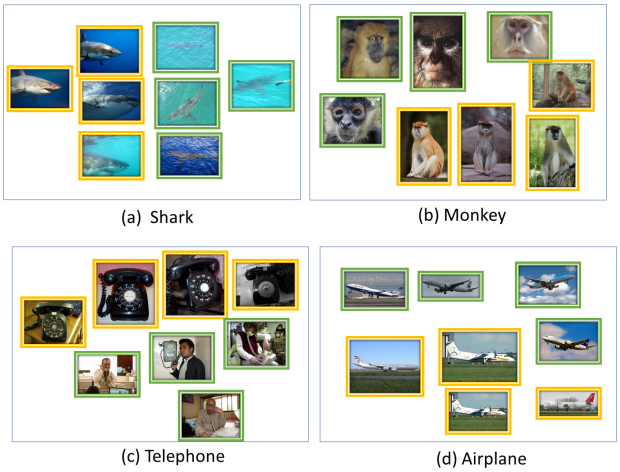

Figure 7: Visualization of subclasses generated by SBCL. Images with green and orange boarder are randomly drawn from different subclasses within the same classes. We can see that SBCL could produce semantically coherent subclasses.

NVIDIA RTX 3090 GPUs. For ImageNet-LT and iNaturalist 2018 datasets, we perform the experiments on 8 NVIDIA RTX 3090 GPUs. The detailed hyperparameters of TSC and SBCL are given in Table 17.

For the classify learning, training the linear classifier strategy is the same with TSC [41]; so, we use TSC default hyperparameters and implementation details for the classifier learning. For the detect model learning, we follow MoCo [24] to adopot the same setting, hyperparameters and evolution metrics with R50-C4 backbone. For Pascal VOC dataset, we train Faster R-CNN [51] on VOC07+12 and evaluate on the test set of VOC07. For COCO dataset, we train Mask R-CNN [25] on train2017 set and evaluate on val2017 set.

Limitations. SBCL has some limitations. First, clustering the head class in SBCL takes a long time on the training phase, especially for ImageNet-LT and iNaturalist 2018. Second, SBCL requires knowing the number of samples in each class to decide the cluster number; so, it is not applicable to problems where the number of samples is unknown.

Social impacts. This work aims to propose a novel representation learning to help people resolve the bias in the real world data recognition, which might has positive social impact. We do not foresee any form of negative social impact induced by our work.

Privacy information in data. All datesets we used in the experiment are public. The datasets only include the pictures, which most are animals and plants. No private information is included.

Baseline information. We report the accuracy of KCL and TSC on different benchmark datasets from [41]. For

Table 14: **Object detection and instance segmentation results on COCO dataset.** The representation model is trained on ImageNet and ImageNet-LT. We report results in bounding-box AP (AP^{bb}) and mask AP (AP^{mk}).

Method		ImageNet			ImageNet-LT		
		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
AP^{bb}	random init.	35.6	54.6	38.2	35.6	54.6	38.2
	CE	40.1	59.8	43.3	38.1	57.4	41.2
	CL [24]	40.4	60.1	44.1	39.7	59.4	42.7
	KCL [33]	40.8	60.6	44.0	39.4	59.1	42.6
	SBCL	**41.1**	**60.8**	**44.2**	**40.0**	**59.6**	**43.0**
AP^{mk}	random init.	31.4	51.5	33.5	31.4	51.5	33.5
	CE	34.9	56.6	37.0	33.3	54.2	35.4
	CL [24]	35.1	56.9	37.6	34.7	56.1	37.1
	KCL [33]	35.5	57.4	37.8	34.4	55.8	36.4
	SBCL	**35.7**	**57.5**	**37.9**	**35.0**	**56.3**	**37.3**

Table 15: Hyperparameter study of β and δ on CIFAR-100-LT with imbalance ratio 100.

β	0.1	0.2	0.5	0.8	1.0	2.0
ACC(%)	44.6	**44.9**	44.5	44.1	43.9	42.1
δ	5	10	20	30	50	100
ACC(%)	44.3	**44.9**	44.6	44.3	42.9	42.3

Table 16: Statistics of datasets. The imbalance ratio $\rho = n_1/n_C$.

Dataset	classes	training data	test data	imbalance ratio
CIFAR-100-LT	100	50,000	10,000	$\{100, 50, 10\}$
ImageNet-LT	1,000	115,846	50,000	256
iNaturalist 2018	8,142	437,513	24,426	500

SwAV[1] [8], PCL[2] [40] and BYOL[3] [21], we use their official open-source implementations.

[1]SwAV offical implementation: https://github.com/facebookresearch/swav.

[2]PCL offical implementation: https://github.com/salesforce/PCL.

[3]BYOL offical implementation: https://github.com/deepmind/deepmind-research/tree/master/byol.

Table 17: Hyperparameters used by different loss functions for benchmark datasets. The detailed hyperparameters of iNaturalist 2018 are the same as the ImageNet-LT.

Hyperparameters	ImageNet-LT		CIFAR100-LT	
	TSC	SBCL	TSC	SBCL
module	MoCo	MoCo	SimCLR	SimCLR
warm-up epoch	200	200	0	10
epoch	400	400	1000	1000
batch size	256	256	1024	1024
learning rate	0.1	0.1	0.5	0.5
learning rate schedule	cosine	cosine	cosine	cosine
memory size	65536	65536	-	-
encoder momentum	0.999	0.999	-	-
feature dimension	128	128	128	128
softmax temperature	0.07	0.07	0.1	0.1
k-positive number	6	-	4	-
hyperparameter of β	0.2	0.2	0.2	0.2
hyperparameter of δ	-	20	-	10
hyperparameter of α	-	10	-	10