Supplement material: SuperDisco: Super-Class Discovery Improves Visual Recognition for the Long-Tail

Yingjun Du[1], Jiayi Shen[1], Xiantong Zhen[1,2*], Cees G. M. Snoek[1]
[1]AIM Lab, University of Amsterdam [2]Inception Institute of Artificial Intelligence

1. Effect of number of super-class levels on more datasets.

We experimented with ***ImageNet-LT*** [1] and ***Places-LT*** [1] with different number of super-class levels. The results are reported in Table 1 and Table 2, respectively. On the ***ImageNet-LT*** [1], we find that the performance of the super-class graphs with the different number of super-class levels is higher than the baseline. However, with more hierarchies (*i.e.*the last row), the performance on the few-shot classes is the highest, while (4, 8, 16, 32, 64) achieves the best performance on all classes. On the ***Places-LT*** [1], with more complex hierarchies *i.e.*(4, 8, 16, 32, 64, 128, 258) achieves the best performance on all classes and few-shot classes. We also conduct experiment on the ***iNaturalis*** [3] to analysis the effect of number of super-class levels in the Figure 1. We can find that with more hierarchies, the performance will consistently increase. 64 achieves the peak performance on the all classes and any-shot classes. For this experiment, we attribute this to our model's ability to explore relatively balanced super-class spaces, thus making the refined tail category features discriminative. We conclude that deeper and broader graphs are needed to discover the super-classes in the case of severe class imbalance.

	Many	Medium	Few	All
Baseline	57.1	45.2	29.3	47.7
(2, 4, 8)	58.6	47.1	31.1	49.8
(4, 8, 16)	59.8	48.3	33.2	50.1
(4, 8, 16, 32)	61.3	49.7	35.1	52.9
(8, 16, 32, 64)	**66.5**	49.8	36.1	55.1
(4, 8, 16, 32, 64)	66.4	**53.3**	37.1	**57.1**
(4, 8, 16, 32, 64, 128)	66.1	52.3	**37.9**	56.5

Table 1. **Effect of number of super-class levels on *ImageNet-LT***. Meta-SuperDisco achieves consistent performance gains with more complex hierarchies.

*Currently with United Imaging Healthcare, Co., Ltd., China.

	Many	Medium	Few	All
Baseline	40.6	39.1	28.6	37.6
(2, 4, 8)	43.1	39.1	29.9	37.5
(4, 8, 16)	44.2	39.9	30.3	38.1
(4, 8, 16, 32)	**45.9**	40.4	31.1	38.9
(4, 8, 16, 32, 64)	44.9	41.3	32.3	39.2
(4, 8, 16, 32, 64, 128)	44.3	**43.1**	34.5	39.9
(4, 8, 16, 32, 64, 128, 256)	45.3	42.8	**35.3**	**40.3**
(4, 8, 16, 32, 64, 128, 256, 512)	44.1	42.3	34.0	**39.1**

Table 2. **Effect of number of super-class levels on *Places-LT***. Meta-SuperDisco achieves consistent performance gains with more complex hierarchies.

2. Benefit of SuperDisco and Meta-SuperDisco

We also give the ablation to show the benefit of SuperDisco and Meta-SuperDisco on ***ImageNet-LT/Places-LT/iNaturalist*** in Table 3. The Meta-SuperDisco consistently surpasses the SuperDisco for all shots. The consistent improvements confirm that Meta-SuperDisco learns even more robust super-class graphs, leading to a discriminative representation of the tail data.

3. Computation cost

We report the computation cost and accuracy gain ablation in Table 4 for ImageNet-LT. Although our model requires more parameters and computational costs compared to the baseline, it brings a 7.2% improvement in accuracy. Compared to the state-of-the-art method by Park *et al.* [2], our model requires a considerably lower amount of additional parameters and computational cost while still delivering better results.

4. Evaluation protocol

We evaluate our model on the test sets for each dataset and report commonly used top-1 accuracy over all classes. For the CIFAR-100-LT dataset, we report the accuracy with different imbalance factors. For the ***ImageNet-LT***, ***Places-LT***, and ***iNaturalist***, we follow [1] and further report accuracy on three different splits of the set of classes: *Many-shot* (>100

	ImageNet-LT				*Places-LT*				*iNaturalist*			
	Many	Medium	Few	All	Many	Medium	Few	All	Many	Medium	Few	All
Baseline	58.4	49.3	34.8	52.7	42.1	39.2	30.9	36.3	68.3	69.2	67.1	68.5
SuperDisco	65.1	52.1	35.9	55.9	44.7	41.1	34.2	39.2	71.3	71.0	69.6	72.1
Meta-SuperDisco	66.1	53.3	37.1	57.1	45.3	42.8	35.3	40.3	72.3	72.9	71.3	73.6

Table 3. **Benefit of SuperDisco and Meta-SuperDisco.** SuperDisco achieves better performance compared to a baseline fine-tuning on all shots, while Meta-SuperDisco is even better for long-tailed recognition.

Table 4. **Computation cost and accuracy gain** for SuperDisco on ImageNet-LT compared to the baseline and state-of-the-art. SuperDisco provides a good trade off.

	Added computational cost		
Models	FLOPs (M)	Parameters (M)	Accuracy
ResNet-32	0	0	45.3
Baseline	0.04	0.001	49.9
Park et al [2]	0.41	0.35	56.2
SuperDisco	0.15	0.03	56.4
Meta-SuperDisco	0.28	0.08	57.1

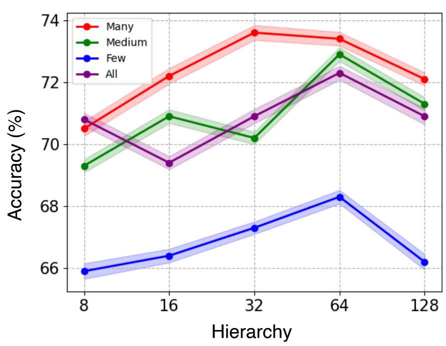

Figure 1. **Effect of number of super-class levels** on iNaturalis-LT.

images), *Medium-shot* (20-100 images) and *Few-shot* (<20 images). We report the average top-1 classification accuracy across all test images.

5. Algorithm

We give the detailed algorithms of SuperDisco and Meta-SuperDisco in Alg. 1 and Alg. 2, respectively.

References

[1] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pages 2537–2546, 2019. 1

[2] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6887–6896, 2022. 1, 2

[3] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, pages 8769–8778, 2018. 1

Algorithm 1 SuperDisco

Require: Training data: $\{x_k, y_k\}$; Number of super-class levels: l; Number of vertices in the l-th super-class level: C^l; Feature extractor: $f_\theta(\cdot)$; Graph function: $g_\phi(\cdot)$; Classifier function: $h_\psi(\cdot)$; Learning rate: α.

Randomly initialize all learnable parameters $\Phi = \{\theta, \phi, \psi\}$
while not done **do**
  Sample a batch of samples $\{x_i, y_i\}$
  Compute the original feature: $\mathbf{z} = f_\theta(x)$
  Construct the super-class graph $\mathcal{C}^l$ by computing the super-class vertex $\mathbf{H}^l_{\mathcal{C}}$ and weights $\mathbf{A}^l_{\mathcal{C}}$ based on the Eq. (1)
  Construct the graph $\mathcal{R}$ and compute the weight $A^l_{\mathcal{R}}$ based on the Eq. (2)
  for m in the number of layers of GNN **do**
    Apply GNN on the graph $\mathcal{R}$ by message passing and obtain the representations $\mathbf{H}^{(m+1)}_{\mathcal{R}}$ based on the Eq. (3)
  end for
  Get the refined feature $\mathbf{z^l} = \mathbf{H}^{(m+1)}_{\mathcal{R}}[0]$
  Compute the final prediction $\tilde{y} = h(\mathbf{z}^l)$
  Update $\Phi = \Phi - \alpha \nabla_\Phi \sum_{i=1}^{I} \mathcal{L}_{\mathrm{CE}}(\tilde{y}_i, y_i)$
end while

Algorithm 2 Meta-SuperDisco

Require: Training data: $\{x_k, y_k\}$; Balanced data: $\mathcal{M}$; Number of super-class levels: l; Number of vertices in the l-th super-class level: C^l; Feature extractor: $f_\theta(\cdot)$; Graph function: $g_\phi(\cdot)$; Classifier function: $h_\psi(\cdot)$; Learning rate: α.

Randomly initialize all learnable parameters $\Phi = \{\theta, \phi, \psi\}$
while not done **do**
  Sample a batch of samples $\{x_i, y_i\}$
  Compute the original feature: $\mathbf{z} = f_\theta(x)$
  Construct the super-class graph $\mathcal{C}^l$ by computing the super-class vertex $\mathbf{H}^l_{\mathcal{C}}$ and weights $\mathbf{A}^l_{\mathcal{C}}$ based on the Eq. (1)
  Construct the prototype graph $\mathcal{P}$ by computing the prototype vertex $\mathbf{C}_{\mathcal{P}}$ and weights $\mathbf{A}_{\mathcal{P}}$ based on the Eq. (4)
  Construct the graph $\mathcal{R}$ and compute the weight $A^l_{\mathcal{R}}$ based on the Eq. (2)
  Construct the super graph $\mathcal{S}$ and compute the vertices $\mathbf{C}^l_{\mathcal{P}}$ and weight $\mathbf{H}^l_{\mathcal{C}^l}$ based on the Eq. (5)
  for m in the number of layers of GNN **do**
    Apply GNN on the graph $\mathcal{S}$ by message passing and obtain the representations $\mathbf{M}^{(m+1)}$ based on the Eq. (6)
  end for
  Get the refined feature $\mathbf{z^l} = \mathbf{M}^{(m+1)}[0]$
  Compute the final prediction $\tilde{y} = h(\mathbf{z}^l)$
  Update $\Phi = \Phi - \alpha \nabla_\Phi \sum_{i=1}^{I} \mathcal{L}_{\mathrm{CE}}(\tilde{y}_i, y_i)$
end while