Learning from Neighbors: Category Extrapolation for Long-Tail Learning

Supplementary Material

A. Appendix

In this supplementary material, we first provide more implementation details in Appendix B about training configurations (Appendix B.1) and auxiliary data collection (Appendix B.2). Then we conduct additional experiments in Appendix C including an experimental comparison to improved SOTA with DIONOv2 (Appendix C.1), and extended ablation studies (Appendix C.2) related to λ_s in the proposed neighbor-silencing loss and the number of samples in the auxiliary dataset, and feature visualization to validate the effectiveness of auxiliary categories (Appendix C.3), and analysis for long-tail in iNaturalist18 [39] (Appendix C.4). In Appendix D, we discuss our contributions (Appendix D.1), limitations (Appendix D.2), and future work (Appendix D.3).

B. Implementation Details

B.1. Training

We employ LiVT [44] as our baseline since it achieves the top performance under the training from scratch paradigm using ViT [10]. Specifically, when training from scratch, following LiVT [44], we conduct MAE [13] training on the downstream dataset because training directly on a long-tail dataset with randomly initialized parameters makes it difficult to converge. When using pre-training paradigms of CLIP and DINOv2, we directly initialize ViT from their weights. Furthermore, the models are trained with AdamW optimizer [24] with $\beta_s = \{0.9, 0.95\}$, with an effective batch size of 512 on 4 NVIDIA 3090 GPUs. The values for weight decay and layer decay are 0.05 and 0.75, respectively. We train all models with $\mathrm{RandAug}(9, 0.5)$ [5], $\mathrm{Mixup}(0.8)$ [47] and $\mathrm{Cutmix}(1.0)$ [46]. Following LiVT [44], the number of training epochs for ImageNet-LT, iNaturalist 18, and Place-LT is set to 100, 100, and 30, respectively. The number of epochs for warmup is set to 10, 10, and 5. The learning rate is set to 1e-3, 1e-5, and 3.5e-5 for training from scratch, CLIP, and DINOv2, respectively. We set a cosine learning rate schedule and the minimum learning rate is 1e-6. We set the maximum sampling number for each auxiliary category to 50 in each training epoch. The hyper-parameter λ_s is set to 0.1. For the ratio of neighbor category for head, medium, and tail classes, we set to $1 : \left\lceil \frac{N_h}{N_m} \right\rceil : \left\lceil \frac{N_h}{N_t} \right\rceil$, where N_h, N_m, and N_t denote the instance number of head, medium, and tail classes, respectively. $\lceil \cdot \rceil$ stands for ceiling, which rounds a number up to the nearest integer.

B.2. Data Collection

We leverage GPT-3.5/4 [28] to search names of visually similar categories for the downstream long-tail datasets. We design a structural prompt with in-context learning and the below shows one example of our interaction with GPT-4 [28].

> **Prompt:** Now I will give you one category name. Please create a list which contains 10 visually similar categories of the provided category.
> For example: If I give you a category name: Acacia cochliacantha. You should return: [Acacia cambagei, Acacia calamifolia, Acacia campylacantha, Acacia cardiophylla, Acacia colei, Acacia colletioides, Acacia compacta, Acacia corymbosa, Acacia crocophylla, Acacia cuthbertii]
> Now, I give you this category name: Abaeis Nicippe.
> You should return:
>
> **Response:** [Eurema ada, Eurema alitha, Eurema andersonii, Eurema beatrix, Eurema blanda, Eurema brigitta, Eurema candida, Eurema celebensis, Eurema desjardinsii, Eurema esakii]

Tab. 7 shows examples of searched category names for each query class on three benchmark datasets. The results show that LLM can provide satisfactory responses using our prompts. After removing duplicates, we obtain 8913, 2318, and 99192 class names for ImageNet-LT [22], Place-LT [49], and iNat18 [39] datasets, respectively. Then we search images for each queried name through the web (*e.g.*, Google/Duckduckgo Image Search Engine). After removing the dissimilar images, concretely, we collect 4.1M, 1.1M, and 3.6M images in 5012, 1895, and 20380 categories as auxiliary data. Fig. 7 shows the distribution of instance numbers for three datasets in each training epoch. It can be observed that 'Tail' is extended by auxiliary data for each dataset.

C. Additional Experiments

C.1. Comparison to Improved SOTA with DINOv2

As shown in Tab. 8, we re-implement LiVT [44] on DINOv2 [26], which is the first work to apply ViT [10] to long-tail learning and leads the performance under the training from scratch paradigm. Our implementation differs only in that LVIT conducts MAE [13] training on the downstream dataset because training directly on a long-tail dataset with randomly initialized parameters is difficult to converge, whereas we initialize directly with the weight

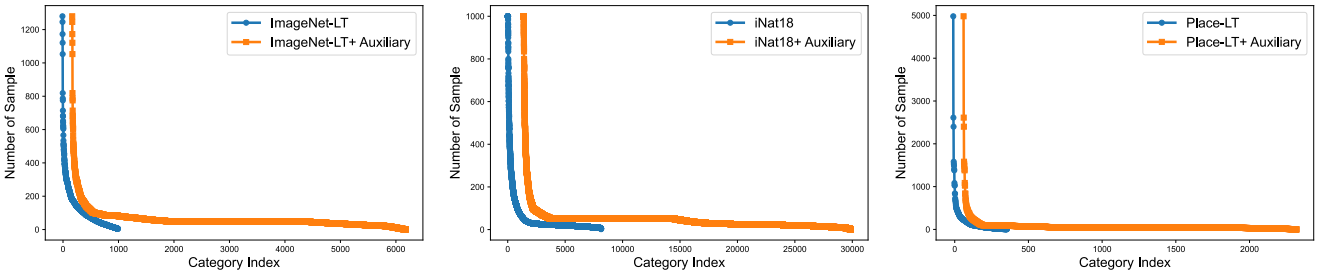


Figure 7. Distribution of samples of original datasets and corresponding datasets with auxiliary data. Please note that because two lines partially overlap, for a better display, the index of the augmented dataset is slightly shifted.

Table 7. Examples of query classes and respective auxiliary classes across three datasets.

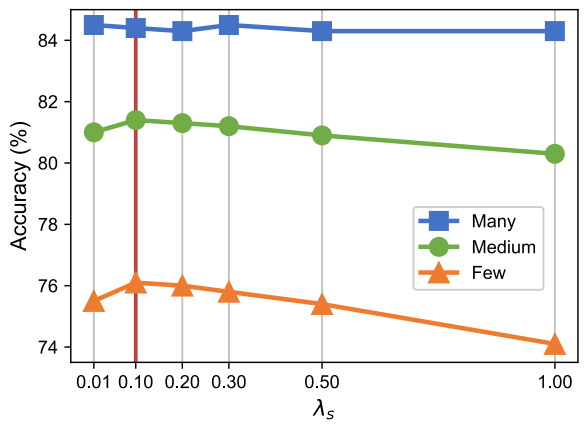
Query	Neighbor Categories
ImageNet-LT	
Wolf Spider	Grass Spider, Fishing Spider, Funnel Web Spider, Garden Spider, Dock Spider, huntsman spider
Irish Wolfhound	Greyhound, Pharaoh hound, Silken Windhound, Coonhound, Plott Hound, Bearded Collie
Basketball	Handball, Football, Badminton Shuttlecock, Softball, Cricket Ball, Billiard Ball, Bowling Ball
Kingsnake	Milk Snake, Corn Snake, Hognose Snake, Ribbon Snak, Black Racer, Speckled Kingsnake
iNaturalist 18	
Dryopteris Expansa	Dryopteris Austriaca, Dryopteris Carthusiana, Dryopteris Dilatata, Dryopteris Filix-mas
Polypodium Virginianum	Polypodium Amorphum, Polypodium Californicum, Polypodium Vulgare, Polypodium Scouleri
Adiantum Hispidulum	Adiantum Diaphanum, Adiantum Raddianum, Adiantum Reniforme, Adiantum Venustum
Spilosoma Lubricipeda	Arctia Caja, Arctia Villica, Callimorpha Dominula, Diaphora Mendica, Eilema Depressa
Place-LT	
Bus Interior	Airplane Interior, Tram Interior, Subway Interior, Van Interior, Taxi Interior, Limo Interior
Bamboo Forest	Tropical forest, Evergreen Forest, Pine Forest, Birch Forest, Cypress Forest, Mangrove Forest
Fastfood Restaurant	Seafood Restaurant, Vegetarian Restaurant, Pizza Restaurant, Mexican Restaurant, Steakhouse
Physics Laboratory	Materials Laboratory, Environmental Laboratory, Geology Laboratory, Engineering Laboratory

from DINOv2. LiVT leverages the Bal-BCE [44] loss by default. We also implement Bal-CE [44]) to train LiVT with DINOv2. Tab. 8 demonstrates that our method shows superior performance on "Medium" and "Few" splits across three standard benchmarks. For example, our method surpasses LiVT(Bal-BCE) 3.2% and 7.6% on "Medium" and "Few' in ImageNet-LT. Note that we set LiVT (Bal-CE) as the baseline method under three pre-training paradigms (training from scratch, CLIP, and DINOv2).

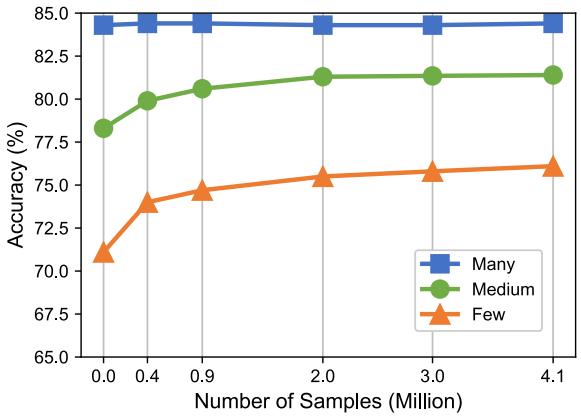
C.2. Extended Ablation Study

Effect of λ_s. As shown in Fig. 9c, we study the effect of λ_s in the proposed neighbor-silencing loss. The optional values are $\{0.01, 0.10, 0.20, 0.30, 0.50, 1.00\}$. It can be seen that as λ_s increases to 0.1, the performance improves. However, when λ_s increases to 1.0, the performance drops. This can be attributed that as λ_s gradually increases, the proposed neighbor-silencing loss will gradually downgrade to the standard cross-entropy loss. In this case, the downstream dataset and the auxiliary dataset are treated equally during the training optimization, and the inconsistency between the network's optimization objective and the testing process leads to a decline in performance.

Number of Auxiliary Samples. As shown in Fig. 8b, we study the effect of the number of samples in the auxiliary dataset. We find that as the number increases from 0 to 0.9 million, there is a dramatic improvement in the accuracy in the few and medium categories, and relatively satisfactory performance is achieved, where +3.7% and 2.3% improvement in the few and medium categories, respectively.



(a) Ablation study on λ_s in the proposed neighbor-silencing loss.



(b) Ablation study on the number of samples in the auxiliary dataset.

Figure 8. More ablation studies. Experiments are conducted on ImageNet-LT [22].

Methods	Backbone	Overall	Many	Medium	Few
Results on ImageNet-LT with DINOv2 pretraining					
LiVT(Bal-BCE) [44]	ViT-B	79.4	**84.9**	78.2	68.5
LiVT(Bal-CE) [44]	ViT-B	79.6	84.3	78.3	71.1
Ours	ViT-B	**81.9**	84.4	**81.4**	**76.1**
Results on iNat18 with DINOv2 pretraining					
LiVT(Bal-BCE) [44]	ViT-B	84.5	84.4	85.4	83.3
LiVT(Bal-CE) [44]	ViT-B	85.0	85.7	86.2	84.2
Ours	ViT-B	**87.0**	**86.4**	**87.4**	**86.7**
Results on Place-LT with DINOv2 pretraining					
LiVT(Bal-BCE) [44]	ViT-B	49.6	**52.4**	49.7	45.2
LiVT(Bal-CE) [44]	ViT-B	49.5	49.2	51.3	46.1
Ours	ViT-B	**50.8**	49.4	**52.4**	**49.2**

Table 8. Re-implementation of previous method with DINOv2. We report the performance on three standard benchmark datasets (*i.e.*, ImageNet-LT, iNaturalist 18, and Place-LT).

From 0.9 million to 4.1 million, the performance gradually increases. This indicates the data efficiency of our method.

C.3. Feature Visualization

In Fig. 9, we provide more examples to demonstrate the effectiveness of auxiliary fine-grained categories on the feature separation for the head and tail classes. We conduct the experiments on ImageNet-LT [22] and train the models from random initialization. The left column shows the feature extracted by the model without auxiliary data, and the right is with the auxiliary fine-grained categories. The results show that training with auxiliary fine-grained categories benefits better feature separation between original head and tail classes.

C.4. Long-Tail in iNaturalist18

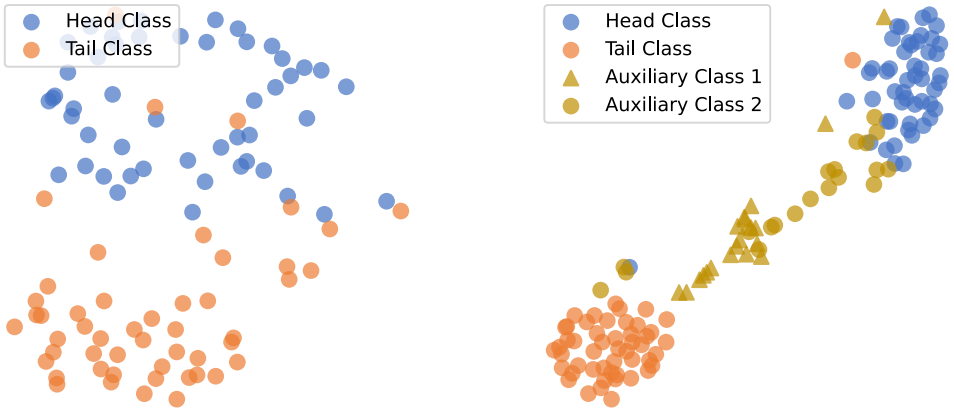
In Sec. 3.2, we validate the effect of granularity on the performance balance. Except for the granularity, we find that another difference between iNat18 and ImageNet-LT is that the number of tail categories in iNat18 is significantly larger than the number of head categories. To validate the effect of the proportion of tail categories, we sample 500 classes from the dataset pool, comprising 60 superclasses, with an imbalance ratio of 0.01. We conduct two sets of experiments: in the first set, we add extra categories to head classes (each category with more than 100 samples); in the second set, the extra categories are added to tail (each category with less than 20 samples). In both sets, the extra categories are fine-grained categories related to the original tail categories. As shown in Fig. 10, the results show that the long-tail benefits the performance balance, while the long-tail will exaggerate the imbalanced performance. This also validates our motivation of extending tail categories with fine-grained categories to balance the feature learning.

D. Discussions

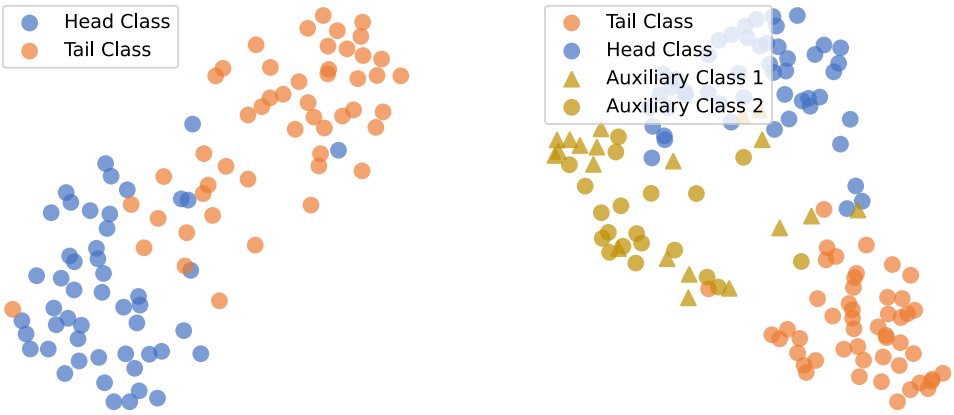
D.1. Contributions

We summarize and discuss our main contributions as follows:

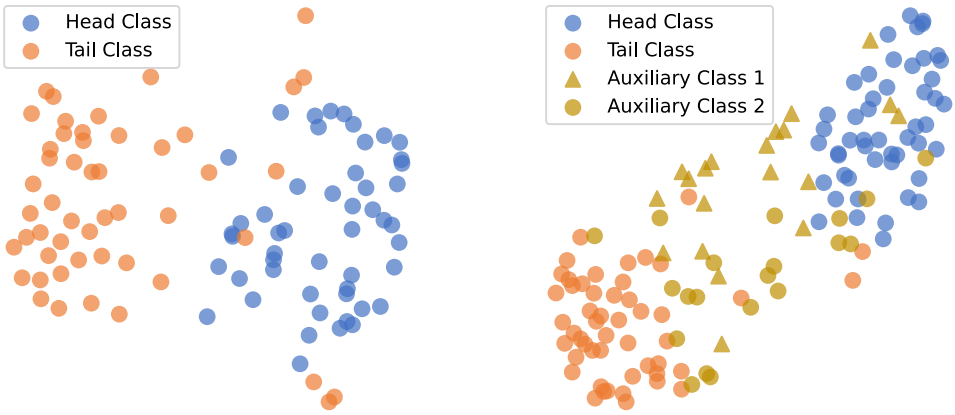
1) **A new perspective for long-tail learning from neighbor categories.** We investigate how to enhance long-tailed learning from open-set data, which is an understudied problem. Our pilot study (Sec. 3) highlights the granularity matters in long-tail learning (Sec. 3.2) and the need for auxiliary categories to improve generalization (Sec. 3.3). As shown in Fig. 2(c), traditional reweighting methods fail to generalize well. However, based on our finding in Sec. 3.2



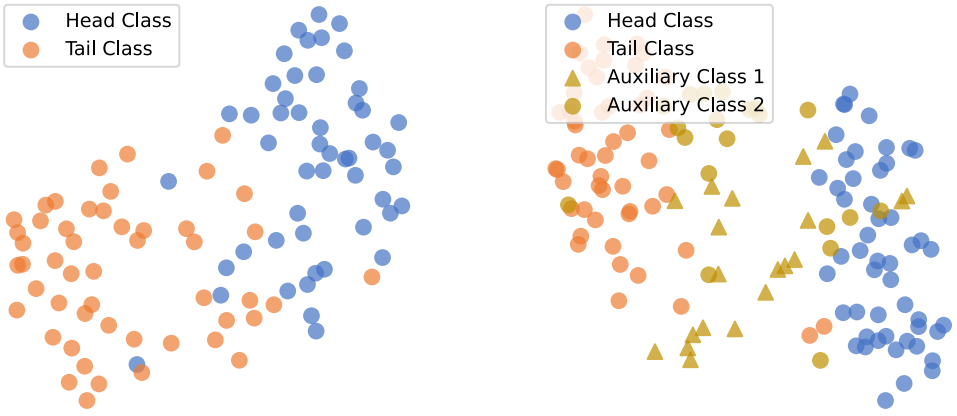
(a) Feature visualization of Kit Fox (Head) and Cougar (Tail).



(b) Feature visualization of Crane (Head) and White Stork (Tail).



(c) Feature visualization of Arctic Fox (Head) and Persian Cat (Tail).



(d) Feature visualization of African Hunting Dog (Head) and Cheetah (Tail).

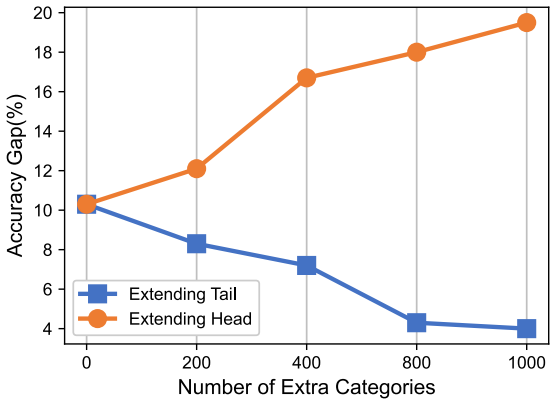


Figure 10. Effect of extending tail vs. extending head.

that increased granularity of training data benefits long-tail learning ((Fig. 3)), we apply auxiliary fine-grained categories, which leads to better separation of the target classes (Fig. 2(d)). We also conduct studies on how to select auxiliary categories: inappropriate auxiliary data can even hinder long-tail learning (Fig. 4), and there exists a trade-off between the similarity and diversity of auxiliary data (Sec. 3.3). We believe these insights are valuable to the community.

2) **Fully automated data acquisition.** Inspired by our findings, we develop a fully automated pipeline for auxiliary data acquisition. As detailed in Sec. 4.1, we utilize GPT-4 API to query neighbor categories for target classes. Then, we retrieve images from the Web and automatically filter these images. We will release all the associated code.

3) **A new balanced loss with neighbor silencing.** As shown in Sec. 4.2, we design a new balanced loss with neighbor silencing for improving long-tailed learning with auxiliary data, which mitigates the distraction of extra classes during training. After training, we directly mask out the classifier weights of auxiliary categories to obtain the final classifier. We find that this strategy works better than retraining a new one by linear probing.

D.2. Limitations

This paper proposes to balance feature learning on downstream long-tail datasets by using visually similar categories. While it has achieved decent performance, there are still the following limitations. First, we use LLM [28] to obtain the names of similar categories. This step depends on the capability of the large language model; if the model has not seen or is unfamiliar with our query, then this step will fail. Second, we obtain images through the web, but we find that some categories are difficult to obtain online, such as those related to the iNat18 categories. For some special categories, we may need to look for more specialized websites to crawl data.

Methods	Backbone	Overall	Many	Med.	Few
Training from scratch					
BCL [51]	ResNet-50	56.0	67.5	52.7	34.8
BCL† [51]	ResNet-50	59.8	68.6	58.1	41.2
PaCo [6]	ResNet-50	57.0	66.4	54.5	38.6
PaCo† [6]	ResNet-50	60.9	67.9	60.4	42.9
NCL [19]	ResNet-50	57.4	67.1	54.9	38.5
NCL† [19]	ResNet-50	61.4	68.1	61.2	43.3
Ours	ResNet-50	64.5	70.1	64.8	47.9
Fine-tuning pre-trained model (CLIP)					
BALLAD [25]	ViT-B	75.7	79.1	74.5	69.8
BALLAD† [25]	ViT-B	76.9	79.3	76.2	72.4
Decoder [42]	ViT-B	73.2	77.9	71.9	64.7
Decoder† [42]	ViT-B	75.2	78.1	74.9	68.6
Ours	ViT-B	**78.8**	**80.3**	**78.4**	**75.8**
Fine-tuning pre-trained model (DINOv2)					
Bal-BCE [44]	ViT-B	79.7	84.1	78.5	71.3
Bal-BCE† [44]	ViT-B	80.7	84.2	80.0	73.5
Ours	ViT-B	**82.0**	**84.7**	**81.5**	**76.2**

Table 9. Performance on ImageNet-LT. We report accuracy (%) of all methods under three pre-training paradigms. We also report the performance of adding the auxiliary data, which denotes by †.

D.3. Future Work

In future research, we consider collecting large-scale unlabeled data as an auxiliary dataset for downstream long-tail datasets and then using this dataset to balance feature learning. Since it is an unlabeled dataset, we can only consider its similarity to the downstream dataset, so compared to the data collection method in this paper, we can have feature learning on a larger scale. Secondly, we find that in a long-tailed distribution dataset, the distribution of superclasses also shows a long-tailed distribution in some datasets (*e.g.*, iNat18 [39]), we will also take into account the long-tail distribution of superclasses to achieve a better balance in feature learning.

D.4. Previous Methods on Auxiliary Data

As shown in Tab. 9, Tab. 10 and Tab. 11, we conduct experiments on more previous methods. We train these methods using the same auxiliary data, which is denoted by †, and conduct the comparison on three pre-training paradigms. The results show that the auxiliary data can enhance the performance of previous methods. Moreover, our methods can further take advantage of the auxiliary data and promote the performance. The potential reason might be that our method prevents the model from being overwhelmed by auxiliary classes, and ensure alignment with the objectives of the testing phase.

Method	Backbone	Overall	Many	Med.	Few
Training from scratch					
BCL [51]	ResNet-50	71.8	70.1	71.6	72.3
BCL† [51]	ResNet-50	72.9	70.3	72.9	73.4
PaCo [6]	ResNet-50	73.2	70.4	72.8	73.6
PaCo† [6]	ResNet-50	73.8	70.5	74.0	74.3
NCL [19]	ResNet-50	74.2	72.0	74.9	73.8
NCL† [19]	ResNet-50	74.8	72.3	75.5	74.5
Ours	ResNet-50	**75.9**	**74.9**	**76.2**	**75.7**
Fine-tuning pre-trained model (CLIP)					
BALLAD [25]	ViT-B	75.0	77.5	75.9	73.1
BALLAD† [25]	ViT-B	77.3	78.1	77.9	76.2
Ours	ViT-B	**80.9**	**79.6**	**80.1**	**82.1**
Fine-tuning pre-trained model (DINOv2)					
Bal-BCE [44]	ViT-B	84.8	85.5	85.4	83.9
Bal-BCE† [44]	ViT-B	85.6	85.9	85.8	85.1
Ours	ViT-B	**87.0**	**86.4**	**87.4**	**86.7**

Table 10. Performance on iNaturalist 2018. We report accuracy (%) of all methods under three pre-training paradigms. We also report the performance of adding the auxiliary data, which denotes by †.

Method	Backbone	Overall	Many	Med.	Few
Training from scratch					
PaCo [6]	ResNet-152	41.2	36.1	47.9	35.3
PaCo† [6]	ResNet-152	42.9	37.2	48.5	40.9
Ours	ResNet-152	**44.7**	47.0	**47.1**	**44.7**
Fine-tuning pre-trained model (CLIP)					
BALLAD [25]	ViT-B	49.5	49.3	50.2	48.4
BALLAD† [25]	ViT-B	50.6	50.1	51.0	50.4
Decoder [42]	ViT-B	46.8	50.6	46.8	39.6
Decoder† [42]	ViT-B	48.8	50.8	48.4	45.8
Ours	ViT-B	**52.4**	**51.6**	**53.0**	**52.3**
Fine-tuning pre-trained model (DINOv2)					
Bal-BCE [44]	ViT-B	49.4	49.1	50.8	46.9
Bal-BCE [44]	ViT-B	44.9	49.3	51.5	47.5
Ours	ViT-B	**50.8**	**49.4**	**52.4**	**49.2**

Table 11. Performance on Places-LT. We report accuracy (%) of all methods under three pre-training paradigms. We also report the performance of adding the auxiliary data, which denotes by †.

D.5. Web Searching Data of Original Category

We aim to explore whether web searching images of the same categories as the original dataset can benefit long-tail learning. Using the original category names, we collect corresponding images. During training, we ensure the same number of images across experiments. As shown in Tab. 12 and Tab. 13, our results indicate that web searching images of the same categories do not improve performance, likely due to a significant distribution gap between the online images and the original dataset. However, when curated data (CD) is used, we observe a significant performance boost. Furthermore, incorporating images from neighboring categories leads to greater improvement.

Methods	Many	Medium	Few	Overall
Baseline	84.3	78.3	71.1	79.6
+ OC	84.4	75.4	65.9	77.6
+ NC	84.7	81.5	76.2	82.0
+ CD	84.4	83.7	83.2	83.9
+ CD + NC	85.4	85.0	84.8	85.1

Table 12. Neighbor Category (NC) v.s. Original Category (OC) from web searching. Results are obtained on ImageNet-LT. CD denotes the curated data from ImageNet-1k [33].

Methods	Many	Medium	Few	Overall
Baseline	49.2	51.3	46.1	49.5
+ OC	49.3	46.3	40.1	46.2
+ NC	49.4	52.4	49.2	50.8
+ CD	51.0	54.0	53.2	52.8
+ CD + NC	52.3	55.5	54.3	54.1

Table 13. Neighbor Category (NC) v.s. Original Category (OC) from web searching. Results are obtained on Place-LT. CD denotes the curated data from Places [49].

iNat18	Many	Medium	Few
Non-Filter	85.3	83.1	79.4
Filter-DINOv2	86.4	87.4	86.7
Filter-CLIP	86.3	87.5	86.5

Table 14. Effects of data filtering.

iNat18	Many	Medium	Few
Original	86.4	87.4	86.7
LLAMA	86.2	87.6	86.4
Public Corpus	85.9	86.1	84.5

Table 15. Validate alternative data acquisition tools..

D.6. Effects of data filtering

In Tab. 14, we conducted experiments to validate the data filtering module. Without filtering, performance drops significantly due to the inclusion of irrelevant data. Meanwhile, replacing DINOv2 with CLIP has minimal impact on model performance.

D.7. Validate alternative data acquisition tools

In Tab. 15, we validate the use of open-source language models and public datasets. **(1)** Using a publicly available model like Llama resulted in no significant performance change, indicating that our system does not rely on GPT-4, as the querying task is relatively simple. **(2)** When retrieving images via text from fixed public datasets (LAION + DataComp1B + ImageNet21K), certain category images

iNat18	Many	Medium	Few
IN-LT	84.2	78.4	71.2
iNat18	85.6	85.8	84.1

Table 16. Average performance of previous methods using DINOv2..

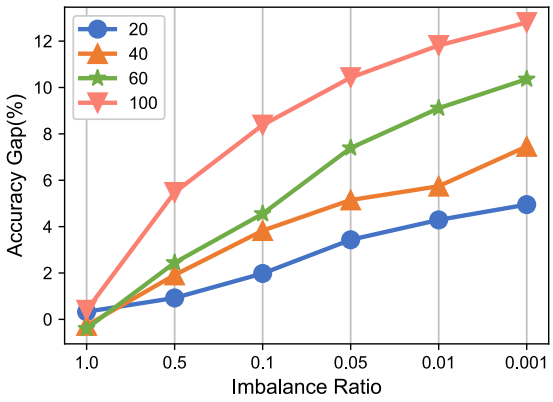


Figure 11. Effect of granularity *vs*. imbalance ratio using DINOv2.

could not be queried, and data diversity was limited, leading to a performance drop. However, the results remain reasonably strong, further demonstrating the robustness and generalizability of our approach.

D.8. Analysis using DINOv2

We conduct the analysis on DINOv2. **(1)** We use DINOv2 to reproduce four classic and state-of-the-art long-tail learning methods [2,7,14,5]. As shown in Tab. 16, we report the average results on ImageNet-LT and iNat18, where iNat18 continues to demonstrate a more balanced performance across many and tail categories. **(2)** As shown in Fig. 11, we conduct the same experiment as in Fig. 3 using DINOv2 and observe that the trend remains consistent as in the paper. **(3)** Further, our visualizations (Fig. 2, Fig. 6, and Fig. 1(Appendix)) are all based on DINOv2.

References

[1] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *CVPR*, 2022. 1

[2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019. 8

[3] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *JAIR*, 2002. 1, 8

[4] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: rebalanced mixup. In *ECCV Workshops*, 2020. 1, 8

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020. 5, 1

[6] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *ICCV*, 2021. 1, 5, 6

[7] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 1, 2, 3, 5, 6, 7, 8

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv:1708.04552*, 2017. 1, 8

[9] Bowen Dong, Pan Zhou, Shuicheng Yan, and Wangmeng Zuo. LPT: Long-tailed prompt tuning for image classification. In *ICLR*, 2023. 1

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 3, 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1

[14] Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. Distilling virtual examples for long-tailed recognition. In *ICCV*, 2021. 1

[15] Ahmet Iscen, Alireza Fathi, and Cordelia Schmid. Improving image recognition by retrieving from web-scale image-text data. In *CVPR*, 2023. 8

[16] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020. 2, 5, 8

[17] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017. 3

[18] Alexander C Li, Ellis Brown, Alexei A Efros, and Deepak Pathak. Internet explorer: Targeted representation learning on the open web. In *ICML*, 2023. 8

[19] Jun Li, Zichang Tan, Jun Wan, Zhen Lei, and Guodong Guo. Nested collaborative learning for long-tailed visual recognition. In *CVPR*, 2022. 5, 6

[20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 8

[21] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. In *ICDM*, 2006. 8

[22] 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*, 2019. 2, 3, 5, 8, 1, 4

[23] Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton van den Hengel. Retrieval augmented classification for long-tail visual recognition. In *CVPR*, 2022. 8

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5, 1

[25] Teli Ma, Shijie Geng, Mengmeng Wang, Jing Shao, Jiasen Lu, Hongsheng Li, Peng Gao, and Yu Qiao. A simple long-tailed recognition baseline via vision-language model. *arXiv:2111.14745*, 2021. 5, 6

[26] Oquab Maxime, Darcet Timothée, Moutakanni Théo, Vo Huy, Szafraniec Marc, Khalidov Vasil, Fernandez Pierre, Haziza Daniel, Massa Francisco, El-Nouby Alaaeldin, Assran Mahmoud, Ballas Nicolas, Galuba Wojciech, Howes Russell, Huang Po-Yao, Li Shang-Wen, Misra Ishan, Rabbat Michael, Sharma Vasu, Synnaeve Gabriel, Xu Hu, Jegou Hervé, Mairal Julien, Labatut Patrick, Joulin Armand, and Bojanowski Piotr. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023. 2, 3, 4, 5, 1

[27] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*, 2020. 3, 4

[28] OpenAI. Gpt-4 technical report. *arXiv:2303.08774*, 2023. 4, 1, 5

[29] Sarah Parisot, Pedro M Esperança, Steven McDonagh, Tamas J Madarasz, Yongxin Yang, and Zhenguo Li. Long-tail recognition via compositional knowledge transfer. In *CVPR*, 2022. 1

[30] 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 *CVPR*, 2022. 1

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 5, 8

[32] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses, 2021. 3

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 5, 6

[34] Dvir Samuel and Gal Chechik. Distributional robustness loss for long-tail learning. In *ICCV*, 2021. 1

[35] Jiang-Xin Shi, Tong Wei, Zhi Zhou, Jie-Jing Shao, Xin-Yan Han, and Yu-Feng Li. Long-tail learning with foundation model: Heavy fine-tuning hurts. In *ICML*, 2024. 6, 7

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[37] Cecilia Summers and Michael J Dinneen. Improved mixed-example data augmentation. In *WACV*, 2019. 8

[38] Changyao Tian, Wenhai Wang, Xizhou Zhu, Jifeng Dai, and Yu Qiao. VL-LTR: learning class-wise visual-linguistic representation for long-tailed visual recognition. In *ECCV*, 2022. 8

[39] 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*, 2018. 2, 3, 5, 1

[40] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, 2019. 8

[41] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *ICLR*, 2021. 1

[42] Yidong Wang, Zhuohao Yu, Jindong Wang, Qiang Heng, Hao Chen, Wei Ye, Rui Xie, Xing Xie, and Shikun Zhang. Exploring vision-language models for imbalanced learning. *arXiv:2304.01457*, 2023. 5, 6

[43] Liuyu Xiang, Guiguang Ding, Jungong Han, et al. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *ECCV*, 2020. 1

[44] Zhengzhuo Xu, Ruikang Liu, Shuo Yang, Zenghao Chai, and Chun Yuan. Learning imbalanced data with vision transformers. In *CVPR*, 2023. 5, 6, 7, 1, 2, 3

[45] Sihao Yu, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Zizhen Wang, and Xueqi Cheng. A re-balancing strategy for class-imbalanced classification based on instance difficulty. In *CVPR*, 2022. 1

[46] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 1, 5, 8

[47] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 1, 5, 8

[48] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *CVPR*, 2021. 1, 8

[49] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 2017. 5, 1, 6

[50] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, 2020. 2, 5, 8

[51] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *CVPR*, 2022. 1, 5, 6