Long-Tail Learning with Language Model Guided Curricula

Mohammed Adnan
University of Calgary
adnan.ahmad@ucalgary.ca

Rahul Krishnan
University of Toronto
rahulgk@cs.toronto.edu

Yani Ioannou
University of Calgary
yani.ioannou@ucalgary.ca

Abstract

Real-world datasets often have class imbalance and follow a long-tail distribution, in contrast to curated datasets, such as CIFAR-10/100, MNIST, etc. Learning from long-tail distributed datasets is a challenging problem due to few representative samples from the tail classes, which makes it difficult for the model to learn robust representations. We posit that curriculum learning presents a viable route to iteratively learn good predictive models that better capture predictive signals about rare classes. We propose a simple method to leverage label hierarchies to craft curricula for learning. For real-world datasets, when the label hierarchy trees are not typically available and manually creating a hierarchy is tedious and expensive, we show that Large Language Models (LLMs) can be used to compose semantic information about the labels and generate label hierarchies to serve as curricula. We perform a thorough empirical evaluation of our method, showing that across different model architectures (ResNet, ViT, and ConvNext) and on multiple datasets (ImageNet, Places365-LT, iNaturalist, etc), we show that LLMs can be used to generate meaningful hierarchies. Our method improves performance on the long-tail classes and achieves state-of-the-art results on multiple large-scale datasets.

1 Introduction

In recent years, Machine Learning (ML) has made significant progress, demonstrating state-of-the-art results across a diverse set of tasks, often evaluated by benchmark datasets. For example, contemporary Deep Neural Network (DNN) models trained to address the computer vision task of visual classification can achieve state-of-the-art results when evaluated on benchmark datasets such as CIFAR10/100 [Krizhevsky, 2009], MNIST [LeCun et al., 2010], SVHN [Netzer et al., 2011], and ImageNet [Deng et al., 2009]. However, ML models and training methods evaluated on benchmark datasets such as these often do not generalize well to real-world applications/datasets. One reason for this is that benchmark datasets are often highly curated and fail to capture the complexity of real-world datasets completely. For example, CIFAR10/100, MNIST, SVHN, and ImageNet are perfectly or almost perfectly class balanced. In contrast, real-world datasets and problems usually exhibit a *long-tail distribution* — *i.e*., a few classes have most of the images, while many classes have few images. Learning from long-tail datasets is challenging because there are fewer training samples for the tail classes, potentially leading to overfitting the few samples available to the tail classes, thus biasing the model. Developing ML algorithms that can learn from long-tail datasets with reduced algorithmic bias and overfitting long-tail classes is important for the safe and reliable deployment of ML models in real-world applications.

We explore Curriculum Learning in the Label Space (CLLS) for long-tail learning, *i.e*., using label hierarchy for curriculum learning. Inspired by human learning, which is guided by the teaching curricula in educational institutes, Curriculum Learning (CL) similarly trains the model sequentially on a curriculum, *i.e*. learning sequentially from training samples (tasks), with increasing order of difficulty. CL divides the learning task into a sequence of tasks, each progressively becoming more difficult to learn, similar to how humans learn. It has been shown that CL has a faster training convergence rate, and models trained

NeurIPS 2024 MusIML Workshop

with CL generalize better [Weinshall and Amir, 2018]. However, a good notion of difficulty is difficult to specify. Devising heuristics to sort training samples based on difficulty is non-trivial and increases the time complexity of the training algorithm. Though there exists much work using CL in the literature, there is no clear consensus on the notion of sample/task difficulty, and methods often rely on hand-crafted curricula, making applying CL challenging. We propose a simple yet efficient workaround to overcome this limitation.

Our main contributions are as follows:

1. We propose a simple but efficient method to build curricula for CL using the label hierarchy available in some datasets. We demonstrate that CLLS can improve the generalization of the long-tail classes and help reduce the model bias. Against several existing baselines and different neural network architectures, our method significantly improves the predictive performance of models.
2. We demonstrate that LLMs can be used to extract rich semantic and contextual information about the label space and generate the label hierarchy tree for real-world datasets, eliminating the necessity of manually creating the hierarchy tree.
3. We show that CLLS is also effective in fine-tuning pre-trained models and can improve the performance on the long-tail classes compared to the state-of-the-art baselines.
4. We extend the state-of-the-art baseline for long-tail learning, PEL [Shi et al., 2024] and show that CLLS can be used on top of existing baselines to further improve the performance on long-tail classes.

2 Curriculum Learning in the Label Space (CLLS)

Curriculum Learning lacks a formal definition of difficulty and a method to generate curriculum automatically [Weinshall and Amir, 2018]. Bengio et al. [2009] relied on manually crafted domain-specific curricula; for real-world datasets, crafting curricula manually is not feasible without domain knowledge/expertise. To address this issue, we propose to generate curricula based on the hierarchy tree of the labels. Often, the target labels are related and can be grouped into coarse/broad classes based on the label hierarchy tree; in this work, we explore leveraging the implicit tree hierarchy of labels for defining easy-to-difficult tasks (coarse-to-fine-grained classification).

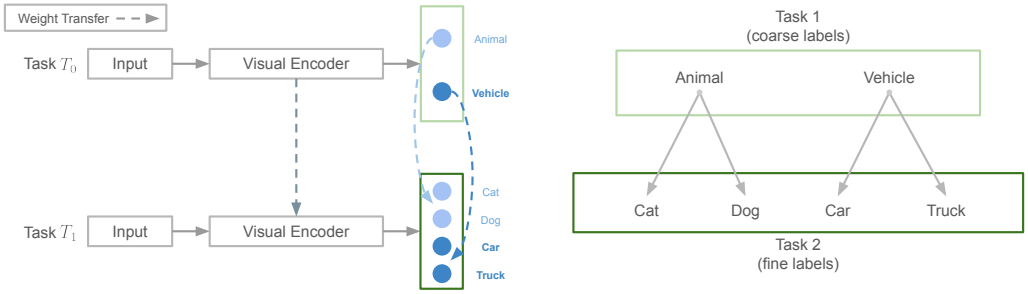
Notation. Let $\mathcal{T}(i)$ denote the label hierarchy tree, which maps the original fine labels Y_0, *i.e.* the leaf nodes of the tree, to Y_i, *i.e.*, the labels at height i in the label hierarchy tree, where the height is the number of edges from the leaf node to the particular root node. Let $\mathcal{D}_i = \{X, Y_i\}$ denote the training dataset and $\mathcal{M}_i = \{E^i, W^i\}$ denote the model with an encoder E^i and classifier head W^i for task $\mathbf{T_i}$. The model $\mathcal{M}_i$ is trained on task $\mathbf{T_i} : X \rightarrow Y_i$ and $\mathbf{C} = \{\mathbf{T}_n, \mathbf{T}_{n-1}, ..., \mathbf{T}_0\}$ denotes the curricula, *i.e.*, the model is sequentially trained on $\mathbf{T}_n$ to $\mathbf{T}_0$, where n is the total number of tasks in the curricula. K_i is the number of classes in dataset D_i. We denote the classifier head weights w_k^j for class k and task $\mathbf{T}_j$.

2.1 Proposed Method

Generating label hierarchy using LLMs. Since LLMs are trained on large training corpora, they can understand the contextual and semantic relations between the labels, enabling them to effectively grasp the nuances and connections within the data. We prompt LLMs with the original class labels (Y_0) of the dataset with the instruction to group the classes into broad categories based on visual and contextual similarities as conceptually illustrated in Figure 2. This generates the hierarchy structure of the label space ($\mathcal{T}$), which can be used to generate coarse labels Y_1. We can repeat this process to get more levels in the hierarchy tree. This substitutes the need to create, using domain expertise, the label hierarchy tree manually and thus enables us to leverage the implicit hierarchical information in the target labels to learn a more robust representation. Hierarchical information helps models understand the data distribution better, enabling them to make more informed predictions and decisions.

Defining curricula. Given the hierarchy tree, we construct a sequence of tasks corresponding to each level in the tree. For label at height i, the classification task $\mathbf{T}_i$ learns a mapping from input images X to the corresponding labels in the hierarchy tree Y_i. Thus, using each level of the hierarchy tree, we can construct a sequence of *n* task $\mathbf{T}_0, ..., \mathbf{T}_n$, where $\mathbf{T}_0$ is the original classification task with fine-grained labels. We define the curricula to train the model as follows:

$$\mathbf{C} = [\mathbf{T}_n, \mathbf{T}_{n-1}, ..., \mathbf{T}_0] \quad \text{where } \mathbf{T}_i : X \rightarrow Y_i.$$



(a) Classifier Head Weight Transfer (b) Label Hierarchy Tree from LLM

Figure 1: **Schematic overview of the method for a curriculum with only two tasks.** (a) Encoder weights are transferred to the next task directly. Classifier head weights are transferred using the hierarchy tree, *i.e.*, the classifier head of the next task uses the weights of its corresponding parent for initialization. (b) Label hierarchy tree is generated using a LLM.

The model $\mathcal{M}$ is trained sequentially starting from $\mathbf{T_n}$, *i.e.*, the coarsest labels to $\mathbf{T}_0$, *i.e.*, the original fine-grained labels. Since the class imbalance in the dataset labels, and the number of classes themselves decrease for the tasks $\mathbf{T_n}$,...,$\mathbf{T}_1$ compared to the original task $\mathbf{T}_0$, task difficulty sequentially increases in the curricula $\mathbf{C}$. $\mathcal{M}_0$ is trained first on the task $\mathbf{T}_0$, which gives it a good prior for learning the subsequent task. Between each task, learned knowledge is transferred to the model $\mathcal{M}_i$ from the weights of $\mathcal{M}_{i+1}$, and used as an initialization for $\mathcal{M}_i$ to be trained from, right up to and including the final model $\mathcal{M}_0$.

Knowledge transfer between tasks. Model $\mathcal{M}_n = \{E^n, W^n\}$, where both the encoder and classifier head weights are randomly initialized, is first trained on task $\mathbf{T}_n$. After training on the first task, for the next task $\mathbf{T}_{n-1}$, the encoder weights E are transferred, whereas a new classifier head is initialized W^{n-1}. The classifier weights are initialized using the knowledge learned for the previous tasks. For a task $\mathbf{T}_i$, The new classifier head weights W^i are initialized using the previous classifier head weight W^{i+1}; weight vector w_k^i corresponding to class k in Y_i is initialized using the weight vector of its parent (coarser) class in W^{i+1} as shown in Figure 1a. The new model $\mathcal{M}_i = \{E, W^i\}$ is then trained on task $\mathbf{T}_i$, followed by a knowledge transfer to the next task $\mathbf{T}_{i-1}$ and so on. Since the classifier weights are not initialized randomly but instead use the knowledge learned in the previous task, this improves the training convergence. Most existing hierarchical learning methods only transfer the weights of the encoder to the subsequent task. As shown in Table 3, CLLS can improve the performance of baseline training (w/o curriculum learning).

Adapting for long-tail learning. The methodology described above is a general framework for CLLS; however, to improve long-tail learning, we also use the features from a pre-trained language model along with the knowledge transfer from the previous tasks. PEL [Shi et al., 2023] proposed to fine-tune a pre-trained model using adapter layers (parameter efficient). Prompts corresponding to each class ("a photo of ¡class name¿" is used to get feature f using the CLIP model [Radford et al., 2021]. We build up on PEL and show that using knowledge from the curricula can significantly enhance the tail classes' accuracy. Following the training methodology, we use adapter layers for fine-tuning, keeping the number of trainable parameters the same for a fair comparison. We train the adapter layers $\mathcal{A}$ and classifier head first sequentially on the curricula $\mathbf{C}$; adapter weights are shared between each task, whereas a weight vector w_k^i for task $\mathbf{T}_i$ is initialized using features from the CLIP model f_k^i corresponding to class k for task $\mathbf{T}_i$ and the weight vector of its parent class p from previous classifier head W^{i-1}, controlled by the mixing ratio β:

$$w_k^i = \beta * w_p^{i-1} + (1-\beta) * f_k^i.$$

This allows the classifier to use knowledge from both the CLIP model and the previous task. This is in contrast to the PEL method, which only relies on the feature of the CLIP model; our method, as demonstrated empirically, achieves better performance of long-tail classes using additional information learned from the curriculum.

3 Results

We evaluate our modified PEL method (PEL + CLLS) on ImageNet-LT, Places365-LT and iNaturalist-2018 datasets. As shown in Table 1, Table 2 (Appendix A) and Table 4 (Appendix A), our method improves

Table 1: **Evaluation on the Places-LT dataset**. Test accuracy is calculated for all the classes (*overall*), classes with more than 100 training images (*many*), with 20–100 images (*medium*), and with less than 20 images (*few*). Our method achieves overall accuracy comparable to the existing state-of-the-art method whilst improving the performance on the long-tail classes by a significant margin. Mean accuracy along with std. deviation over three runs w/ diff random initializations are reported.

		Test Accuracy (%)			
Method	**Backbone**	Overall	Many (¿100)	Medium (20–100)	Few (¡20)
Trained from scratch (w/ pre-trained backbone)					
OLTR [Liu et al., 2019]	ResNet-152	35.9	44.7	37.0	25.3
cRT [Kang et al., 2020]	ResNet-152	36.7	42.0	37.6	24.9
LWS [Kang et al., 2020]	ResNet-152	37.6	40.6	39.1	28.6
MiSLAS [Zhong et al., 2021]	ResNet-152	40.4	39.6	43.3	36.1
DisAlign [Zhang et al., 2021]	ResNet-152	39.3	40.4	42.4	30.1
ALA [Zhao et al., 2021]	ResNet-152	40.1	43.9	40.1	32.9
PaCo [Cui et al., 2021]	ResNet-152	41.2	36.1	47.9	35.3
LiVT [Xu et al., 2023]	ViT-B/16	40.8	48.1	40.6	27.5
Fine-tuned from pre-trained model					
BALLAD [Ma et al., 2021]	ViT-B/16	49.5	49.3	50.2	48.4
Decoder [Wang et al., 2023]	ViT-B/16	46.8	-	-	-
LPT [Dong et al., 2023]	ViT-B/16	50.1	49.3	52.3	46.9
PEL [Shi et al., 2024]	ViT-B/16	**52.2**	51.7	53.1	50.9
Ours	ViT-B/16	51.7 ± 0.1	51.3 ± 0.3	52.8 ± 0.1	$\mathbf{51.8 \pm 0.1}$

the performance on the long-tail classes while keeping the accuracy on the dominant classes almost same. Places-LT contains 62.5K images from 365 classes, from a maximum of 4980 to a minimum of 5 images per class. iNaturalist-2018 consists of 437.5K images distributed across 8142 species, with the number of images per species varying from as few as 2 to as many as 1000. For the Places365-LT dataset, the performance on the long-tail classes is improved with curricula designed by the language model, confirming our hypothesis that language models can be used to extract label hierarchy trees. For all three datasets, the accuracy of the classes with less than 20 images was improved, thus reducing the model bias. We provide an additional evaluation of metrics used for imbalanced datasets. Additional results on ImageNet-LT and iNaturalist-2018 datasets are shown in the appendix. For the ImageNet dataset, the accuracy of the majority classes decreases slightly; we suspect this might be because the ViT model was pre-trained on the ImageNet-21k dataset and thus was overfit on the majority classes.

4 Conclusion

In this work, we explored leveraging curriculum learning in the label space for learning long-tail datasets. This formulation of curricula provides a simple yet effective way to design curricula using language models. We demonstrated for the Places365-LT dataset that LLMs can create curricula based on contextual similarities and thus substitute the expensive process of creating a label hierarchy tree manually (*e.g.* WordNet). Our method improved the state-of-the-art LT baselines on multiple large-scale LT datasets, such as ImageNet-LT, Places-LT, and iNaturalist2018 datasets, specifically for the tail classes. For the Places-LT dataset, our method improved the accuracy on tail classes from 50.9% to 51.8%, on ImageNet-LT from 73.4% to 73.9% and on iNaturalist-2018 dataset from 82.2% to 82.7%. One limitation of our method is that our work relies on using LLMs to extract the label tree hierarchy. To extract a meaningful label tree hierarchy, the label must be in the training corpus of LLMs. For some task-specific datasets, it is possible that labels were not present in the training corpus, and thus the label tree extracted with the LLM will not be reliable. In future work, we will evaluate the method for domain-specific datasets, such as medical imaging and histopathology.

References

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 715–724, October 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Bowen Dong, Pan Zhou, Shuicheng Yan, and Wangmeng Zuo. Lpt: Long-tailed prompt tuning for image classification, 2023.

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, 2021.

Jeffrey L. Elman. Learning and development in neural networks: the importance of starting small. *Cognition*, 48:71–99, 1993. URL `https://api.semanticscholar.org/CorpusID:2105042`.

Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6626–6636, June 2021.

iNaturalist 2018 competition dataset. iNaturalist 2018 competition dataset. `https://github.com/visipedia/inat_comp/tree/master/2018`, 2018.

Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting, 2019.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition, 2020.

Muhammad Gul Zain Khan, Muhammad Ferjad Naeem, L Gool, D Stricker, F Tombari, and Muhammad Zeshan Afzal. Introducing language guidance in prompt-based continual learning. *ICCV*, pages 11429–11439, August 2023.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

M. Pawan Kumar, Ben Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Neural Information Processing Systems*, 2010. URL `https://api.semanticscholar.org/CorpusID:1977996`.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Jun Li, Zichang Tan, Jun Wan, Zhen Lei, and Guodong Guo. Nested collaborative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6949–6958, June 2022.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world, 2019.

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, 2021.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment, 2021.

George A. Miller. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. URL https://aclanthology.org/H94-1111.

Pietro Morerio, Jacopo Cavazza, Riccardo Volpi, Rene Vidal, and Vittorio Murino. Curriculum dropout, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

Terence D. Sanger. Neural network learning control of robot manipulators using gradually increasing task difficulty. *IEEE Trans. Robotics Autom.*, 10:323–333, 1994. URL https://api.semanticscholar.org/CorpusID:35187967.

Jiang-Xin Shi, Tong Wei, Zhi Zhou, Xin-Yan Han, Jie-Jing Shao, and Yu-Feng Li. Parameter-efficient long-tailed recognition, 2023.

Jiang-Xin Shi, Tong Wei, Zhi Zhou, Xin-Yan Han, Jie-Jing Shao, and Yu-Feng Li. Parameter-efficient long-tailed recognition, 2024. URL https://openreview.net/forum?id=GT57SN8xt9.

Samarth Sinha, Animesh Garg, and Hugo Larochelle. Curriculum by smoothing, 2021.

Burrhus Frederic Skinner. *The behavior of organisms: An experimental analysis*. BF Skinner Foundation, 2019.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey, 2022.

Min-Kook Suh and Seung-Woo Seo. Long-tailed recognition by mutual information maximization between latent features and ground-truth labels. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32770–32782. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/suh23a.html.

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, 2023.

Daphna Weinshall and Dan Amir. Theory of curriculum learning, with convex loss functions. December 2018.

Zhengzhuo Xu, Ruikang Liu, Shuo Yang, Zenghao Chai, and Chun Yuan. Learning imbalanced data with vision transformers, 2023.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification, 2023.

Dingwen Zhang, Deyu Meng, Chao Li, Lu Jiang, Qian Zhao, and Junwei Han. A self-paced multiple-instance learning framework for co-saliency detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2361–2370, June 2021.

Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition, 2022.

Yan Zhao, Weicong Chen, Xu Tan, Kai Huang, and Jihong Zhu. Adaptive logit adjustment loss for long-tailed visual recognition, 2021.

Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition, 2021.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9716–9725, 2019. URL `https://api.semanticscholar.org/CorpusID:208637033`.

Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition, 2022.

A Appendix / supplemental material

A.1 Background

Curriculum Learning. Inspired by how humans learn, CL involves training models in a meaningful order of data defined by curricula — usually starting from easier-to-learn and moving on to difficult-to-learn data. The concept of CL has been studied from a behavioural approach [Skinner, 2019] and computational approach [Elman, 1993]. Although the idea of CL in supervised learning is not new [Sanger, 1994], CL has gained traction in recent years. Bengio et al. [2009] proposed training models by gradually increasing the difficulty of the data samples — defined by a curriculum — during the training process. CL can be broadly categorized into five categories [Soviany et al., 2022]: Balanced CL, Progressive CL, Self-Paced CL and Implicit CL. In Self-Paced CL (SPL) [Kumar et al., 2010], training loss is regularized to give greater significance to points that better fit the current learner's hypothesis. While SPL does not explicitly require a pre-defined curriculum, it introduces new challenges. SPL is difficult to optimize and more susceptible to over-fitting [Weinshall and Amir, 2018]. Progressive CL, instead of training the model on easy to difficult samples, trains the model sequentially on easy-difficult tasks [Morerio et al., 2017]. Implicit CL does not change the easy-to-hard schedule; instead, it changes the training method to make training easier. For example, Sinha et al. [2021] proposed to deblur the convolution activation map during the training gradually. Balanced CL, in addition to training from easy-to-hard samples, also enforces the selection of training samples to be balanced [Zhang et al., 2015].

Long-Tail Learning. Most of the methods for long-tail learning can be divided into four categories — Data Manipulation, Representation Learning, loss adjustment, and fine-tuning pre-trained models. Data manipulation methods use data augmentation and re-sampling techniques to learn better representations for the long-tail classes [Kang et al., 2020, Zhou et al., 2019]. Loss adjustment techniques use label frequency to re-weight the loss [Hong et al., 2021, Kang et al., 2019, Lin et al., 2018] and adjust the logits [Menon et al., 2021, Zhang et al., 2021] based on the label frequency. Most of the recent work within this area has focused on fine-tuning a model pre-trained on a large and diverse dataset. Ensemble-based methods learn to use multiple expert models trained on different data sub-groups to learn robust representation for long-tail recognition [Zhang et al., 2022, Zhou et al., 2019]. BALLAD [Ma et al., 2021] leverages contrastive vision-language models for long-tailed recognition. It first fine-tunes a vision-language backbone on the target long-tail datasets, followed by using an additional adapter layer to enhance the representations of tail classes. PEL [Shi et al., 2024] uses a pre-trained ViT [Dosovitskiy et al., 2021] and CLIP [Radford et al., 2021] model and introduces a small number of task-specific parameters by adopting the design of any existing parameter-efficient fine-tuning method. CLIP feature embeddings are used to initialize the classifier head and improve the convergence rate. In this work, we extend the PEL without CL approach and show that it further improves the performance on long-tail classes on multiple datasets.

LLMs in Vision. Since the emergence of LLMs, different methods have been proposed to implicitly use the information learned by the language model for solving downstream vision tasks. Vision-Language Models (VLMs) have gained popularity because of their ability to learn rich representation from limited data [Shi et al., 2023]. Outside of VLM, language model guidance for vision tasks has been explored recently, including open-set learning and metric learning. Khan et al. [2023] explored the use of language model prompting to reduce the effect of catastrophic forgetting in the vision model. Methods in open-set learning learn a vision encoder to map to the same embedding space as the language model. The model can then generalize to new classes by matching it with the embeddings of the class names without requiring labelled visual training data. Language models have also been explored in interpretability in computer vision; Yang et al. [2023] proposed language-guided bottlenecks to improve the concept bottlenecks, which leverages GPT-3 to define a large space of possible bottlenecks. However, to the best of our knowledge, the use of language models for curriculum learning has not been studied so far in this manner. Our work aims to bridge this gap, and in this paper, we explore the applications of language models for curriculum learning, in particularly for long-tail classification.

A.2 Experimental Details and Additional Results

Datasets. We ran experiments on multiple large-scale datasets, such as ImageNet [Deng et al., 2009], ImageNet-LT [Liu et al., 2019], Places-LT [Liu et al., 2019], and iNaturalist-2018 [iNaturalist 2018 competition dataset] using different types of visual backbones (Convolutional Neural Network, ViT [Dosovitskiy et al., 2021] and ConvNext [Liu et al., 2022]) to show that our method can improve the generalization

Table 2: **Evaluation on the ImageNet-LT dataset**. Accuracy is calculated for all the classes (overall), classes with more than 100 images (many), classes with 20–100 images (medium), and classes with less than 20 images (few). As shown in the table, our method achieves overall accuracy comparable to the existing state-of-the-art method but improves the performance on the long-tail classes by a significant margin. Mean accuracy along with std. deviation over five runs w/ diff random initializations are reported.

		Test Accuracy (%)			
Method	**Backbone**	Overall	Many (¿100)	Medium (20–100)	Few (¡20)
Trained from scratch (w/ pre-trained backbone)					
cRT [Kang et al., 2020]	ResNet-50	47.3	58.8	44.0	26.1
LWS [Kang et al., 2020]	ResNet-50	47.7	57.1	45.2	29.3
MiSLAS [Zhong et al., 2021]	ResNet-50	52.7	62.9	50.7	34.3
LA [Menon et al., 2021]	ResNet-50	52.7	62.9	50.7	34.3
DisAlign [Zhang et al., 2021]	ResNet-50	52.9	61.3	52.2	31.4
BCL [Zhu et al., 2022]	ResNet-50	56.0	-	-	-
PaCo [Cui et al., 2021]	ResNet-50	57.0	-	-	-
NCL [Li et al., 2022]	ResNet-50	57.4	-	-	-
LiVT [Xu et al., 2023]	ViT-B/16	60.9	73.6	56.4	41.0
Fine-tuned from pre-trained model					
BALLAD [Ma et al., 2021]	ViT-B/16	75.7	79.1	74.5	69.8
Decoder [Wang et al., 2023]	ViT-B/16	73.2	-	-	-
PEL [Shi et al., 2024]	ViT-B/16	**78.3**	81.3	77.4	73.4
Ours	ViT-B/16	77.8 ± 0.2	80.7 ± 0.1	76.6 ± 0.4	$\mathbf{73.9} \pm \mathbf{0.2}$

of visual backbones across different architecture types. The ImageNet dataset contains 1.2 million images of 1000 different object categories; Places-LT contains 62.5K images from 365 classes, from a maximum of 4980 to a minimum of 5 images per class. iNaturalist-2018 consists of 437.5K images distributed across 8142 species, with the number of images per species varying from as few as 2 to as many as 1000. The ImageNet dataset is organized according to the WordNet hierarchy [Miller, 1994]; to extract the label hierarchy for the ImageNet dataset, we use the WordNet synsets. The ImageNet-LT dataset, which is a long-tail version of the ImageNet dataset, contains 115.8K images from 1000 categories, with a maximum of 1280 images per class and a minimum 5 images per class. The iNaturalist-2018 dataset has information about the 'genus', 'kingdom', 'order', etc. for each of the species, which can be used to extract a label hierarchy tree. For the Places-LT dataset, we use a LLM to group the classes and generate a label hierarchy tree. Following the evaluation scheme of Shi et al. [2024], we report overall accuracy along with three splits of classes: *many* (>100 images/class), *medium* (20–100 images/class) and *few* (<20 images/class).

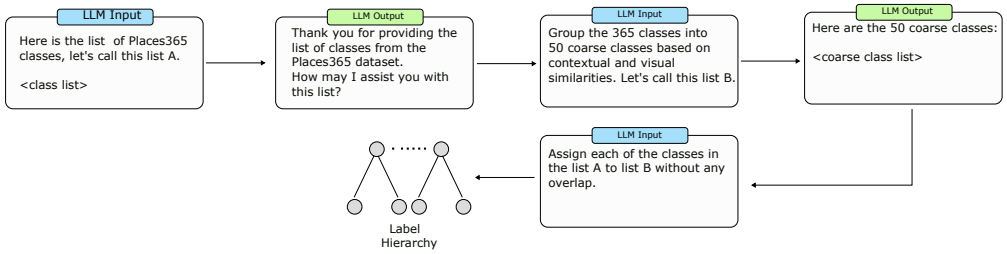


Figure 2: **LLM prompt for Places365-LT dataset.** LLM is given the list of classes with the instruction of grouping them into coarse classes based on contextual similarities. The hierarchy tree generated by the LLM is then used for curriculum learning.

CL Baselines. To demonstrate that our method can generalize better than conventional training, *i.e.*, without CL, we train multiple baseline models (ResNet-50, ViT-B/16 [Dosovitskiy et al., 2021], and ConvNext [Liu et al., 2022]) on multiple datasets to compare the generalization performance of the trained

Table 3: **Improvement over baselines with CLLS.** Comparison of top-1 accuracy (%) between training without CL (baselines) and with our method, CLLS. As shown in the table, models trained with CLLS generalize better on multiple datasets across different architectures. ViT and ConvNext require large-scale datasets for training and thus not feasible to train them on the CIFAR-100 dataset. We re-train the baselines using the hyper-parameters and training methodology reported in the literature.

Backbone	**Method**	**ImageNet**	**iNat-2018**
ResNet	Baseline	75.4	59.0
	CLLS	**76.3**	**61.2**
ViT-B/16	Baseline	78.9	50.1
	CLLS	**79.5**	**55.8**
ConvNext-T	Baseline	79.6	60.4
	CLLS	**81.2**	**65.9**

Table 4: **Evaluation on the iNaturalist-2018 dataset**. Test accuracy is calculated for all the classes (*overall*), classes with more than 100 training images (*many*), with 20–100 images (*medium*), and with less than 20 images (*few*). Our method achieves overall accuracy comparable to the existing state-of-the-art method whilst improving the performance on the long-tail classes by a significant margin. Mean accuracy along with std. deviation over three runs w/ diff random initializations are reported.

		Test Accuracy (%)			
Method	**Backbone**	Overall	Many (¿100)	Medium (20–100)	Few (¡20)
Trained from scratch (w/ pre-trained backbone)					
cRT [Kang et al., 2020]	ResNet-50	65.2	69.0	66.0	63.2
LWS [Kang et al., 2020]	ResNet-50	65.9	65.0	66.3	65.5
MiSLAS [Zhong et al., 2021]	ResNet-50	71.6	73.2	72.4	70.4
DiVE	ResNet-50	69.1	70.6	70.0	67.6
DisAlign [Zhang et al., 2021]	ResNet-50	69.5	61.6	70.8	69.9
ALA	ResNet-50	70.7	71.3	70.8	70.4
RIDE	ResNet-50	72.6	70.9	72.4	71.3
BCL [Zhu et al., 2022]	ResNet-50	71.8	-	-	-
PaCo [Cui et al., 2021]	ResNet-50	73.2	70.4	72.8	73.6
NCL [Li et al., 2022]	ResNet-50	74.2	72.0	74.9	73.8
GML [Suh and Seo, 2023]	ResNet-50	74.5	-	-	-
LiVT [Xu et al., 2023]	ViT-B/16	76.1	78.9	76.5	74.8
Fine-tuned from pre-trained model					
Decoder [Wang et al., 2023]	ViT-B/16	59.2	-	-	-
LPT [Dong et al., 2023]	ViT-B/16	76.1	-	-	79.3
PEL [Shi et al., 2024]	ViT-B/16	80.4	74.0	80.3	82.2
Ours	ViT-B/16	**80.9** $\pm$ **0.2**	74.0 $\pm$ 0.3	80.9 $\pm$ 0.3	**82.7** $\pm$ **0.2**

models. For all the datasets, we create a curriculum with only two tasks, *i.e.*, coarse and fine (original) labels; we re-train all the baselines based on the using the hyper-parameters reported in the original paper. As shown in Table 3, our method can improve generalization across a range of different architectures and datasets (both long-tail and roughly class-balanced) for visual classification, validating the effectiveness of our method.