Supplementary Material: Long-tail Recognition via Compositional Knowledge Transfer

Sarah Parisot Pedro M. Esperança Steven McDonagh Tamas J. Madarasz
Yongxin Yang Zhenguo Li

Huawei Noah's Ark Lab

1. Datasets and implementation details

ImageNet-LT [7] is a subset of the large-scale ImageNet dataset [2], subsampled such that class distributions follow a Pareto distribution with power value $\alpha = 6$. The dataset contains 116k training images from 1,000 categories, with class cardinality ranging from 5 to 1,280. The dataset is publicly available, and its usage is limited for research only (non-commercial or education purposes).

We trained our backbone encoder with parameters following the training setting most commonly used in the literature [5]: we train ResNext50 models, with cosine classifiers, for 90 epochs, with weight decay 0.0005, batch size 512, and learning rate initialised at 0.2 with cosine decay to 0. All sampling methods use identical parameter sets.

Places-LT is a subset of the large-scale scene classification dataset; Places [11] that is constructed in a similar fashion to the ImageNet-LT dataset [7]. The dataset comprises of 62.7K training images from 365 categories with class cardinality ranging from 5 to 4980. The dataset is publicly available, and its usage is limited to research only (non-commercial or education purposes).

We trained a ResNet152, using supervised pre-trained weights, towards direct comparison with state of the art methods. Due to the lack of publicly available pre-trained ResNet152 models, we carry out our backbone analysis and additional experiments using an unsupervised initialisation of ResNet101 architectures. All pre-trained weights are obtained by leveraging the full ImageNet dataset according to standard practice.

All models are trained following standard practice with regards to parameters commonly found in the literature [4, 7]. We use a batch size of 128, weight decay of 0.0005. The learning rate is set to 0.001 for the pre-trained backbone encoder, and 0.1 for the cosine classifier with a cosine decay to 0. We train ResNet152 models for 10 epochs, and ResNet101 models for 15 epochs.

Our method is implemented using PyTorch [8].

2. Societal impact

The potential benefits of low data regime tools often relate to reduction of data costs; collection, curation, storage and processing. Our approach in particular, can contribute to reducing recognition bias with regards to underrepresented classes, that involve rare or otherwise difficult to acquire training samples. Furthermore, our approach allows to adapt pre-trained models to reduce biases or introduce new classes without additional training steps, consequently improving environmental impact. In terms of risks; making models both readily available and quickly accessible for novel tasks, at low data and training costs, to individuals without domain expertise, in combination with potentially increased susceptibility to subtle prediction failures, may increase the risk of both models and their outputs being used incorrectly.

3. Additional results on the ImageNet dataset

We provide an ablation experiment showing the impact of changing prototype and classifier roles in Eqs. 4–6. Results are summarised in Table 1, where the first row report our original results (prototype to classifier weights attention). We can see that the best performance is achieved in our chosen configuration. The decreased performance in other configurations is to be expected, as less reliable few-shot classifiers and/or many-shot prototypes are relied on more heavily in other configurations.

We provide additional results on the ImageNet dataset in Table 2: we report performance using a model trained using uniform sampling, as well as a model trained using a balanced softmax loss [9] (*vs*. regular cross entropy loss). Our balanced softmax model is retrained using a cosine classifier, which explains our higher accuracy with regards to numbers reported in [9]. It may be observed that our approach continues to improve performance on rare classes, and in particular we note the high performance on this class

Table 1. Sensitivity results evaluating the impact of the prototype to classifier attention mechanism. $p \rightarrow w$ corresponds to the setting described in Eq. 4–6 of the manuscript, $w \rightarrow p$ inverts classifier and prototype roles, $w \rightarrow w$ and $p \rightarrow p$ carry out self attention with only one kind of class representation.

	Many-shot	Medium-shot	Few-shot	Total
$p \rightarrow w$	63.2	**52.1**	**36.9**	**54.2**
$w \rightarrow p$	**65.0**	50.1	28.9	52.9
$w \rightarrow w$	64.5	51.1	31.3	53.4
$p \rightarrow p$	62.1	50.1	31.4	52.1

group using a balanced softmax classifier. We further note how backbones influence our final overall performance (*e.g.* uniform *vs*. square root), highlighting the importance of training a high quality backbone. We state again that solutions that aim to learn better representations are complementary of our method, which focuses on handling the few-shot problem.

4. Attention mechanism class selection

In this section, we evaluate our model ability to select semantically relevant classes via our attention mechanism. To this end, we show in Figure 3 the classes with top 10 attention scores for five randomly selected classes. For simplicity, we consider $k = 0$, such that all classes are treated identically. To evaluate semantic similarity, we plot class semantic similarity in the WordNet hierarchy by computing the Leacock-Chodorow Similarity [6], which measures the shortest path distance between classes in the graph, while taking into account their depth in the taxonomy. We can see that classes with very similar categories are selected (i.e. dog breeds), allowing for transferring common properties across these classes.

5. Visualizing the impact of knowlege transfer

Fig 1 shows the impact of knowledge transfer on class representations, visualising how our knowledge transfer process adjusts class representations. We can see that after transfer, the few-shot class representation is pushed towards more accurate class prototypes, while common class representations remain close to their learned classifier. We also note that it illustrates how prototypes and trained classifier representations can differ, highlighting the advantage of combining these two representations.

6. Additional results on the Places dataset

6.1. Backbone analysis

Further to our ImageNet-LT backbone analysis, we provide here further study on training strategies of interest, additionally for the places dataset. We carry out this study

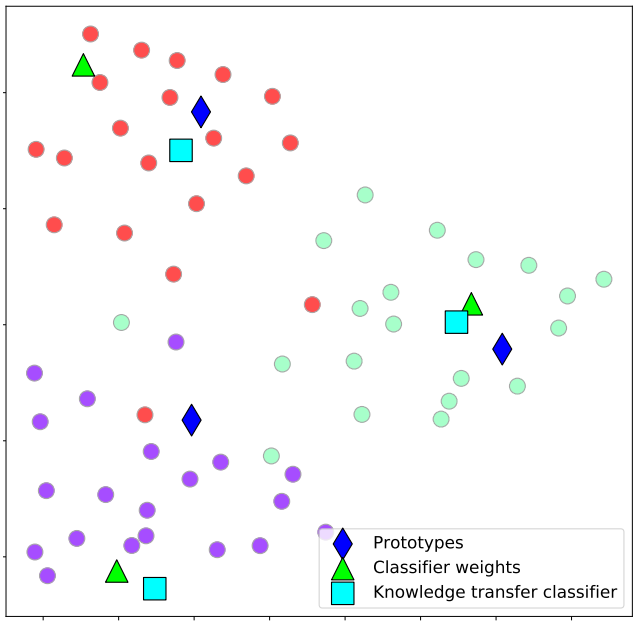

Figure 1. UMAP representations of validation samples from a few-shot class [red cluster] and the top-2 closest classes in terms of knowledge transfer attention score [purple (many shot class) and green (medium shot) clusters]. For each class, we also plot prototypes, classifiers and final classifier.

on the ResNet101 backbone, and consider: a) the sampling strategy (square root or uniform), and b) the choice of pre-trained weights (supervised or unsupervised pre-training on ImageNet). Due to the smaller size of the dataset, experiments in the literature on Places-LT typically initialise the model backbone encoder with weights pre-trained, in a supervised manner, on the entire ImageNet dataset.

Here, we additionally consider usage of an unsupervised initialisation, with a model pre-trained on ImageNet using the SimCLR [1] contrastive learning method. Unsupervised pre-training has been shown to achieve superior transfer learning performance in certain circumstances [3]. In addition, our key incentive is the fact that, in contrast to standard supervised pre-training, SimCLR relies on a *normalised, distance-based* representation learning process, which has higher compatibility with our cosine classifier strategy.

To evaluate which training strategy yields higher quality backbones, we consider the same initialisation criteria discussed in the main manuscript: We compute training and validation prototypes, and measure accuracy on both the training (to evaluate underfitting and memorisation), and test sets. Our analysis is reported in Figure 2. Firstly, in terms of sampling strategies, we note a limited impact overall, with square root yielding higher quality prototypes with supervised initialisation, and uniform sampling having a very slight edge when using an unsupervised initialisation.

Table 2. Classification accuracies on ImageNet-LT. All methods use a ResNext50 backbone. * models trained with a normalised classifier.

Method	Classifier type	Many-shot	Medium shot	Few shot	Total
Uniform sampling	Cosine classifier	**69.2**	43.0	15.4	49.2
	ensemble$\left(w^{\mathrm{h}}(0), w^{\mathrm{h}}(20), w^{\mathrm{h}}(100)\right)$	62.9	48.9	33.7	**52.2**
Balanced softmax	Cosine classifier	**64.2**	49.2	32.8	52.7
	ensemble$\left(w^{\mathrm{h}}(0), w^{\mathrm{h}}(20), w^{\mathrm{h}}(100)\right)$	61.4	**49.9**	**38.8**	**52.8**

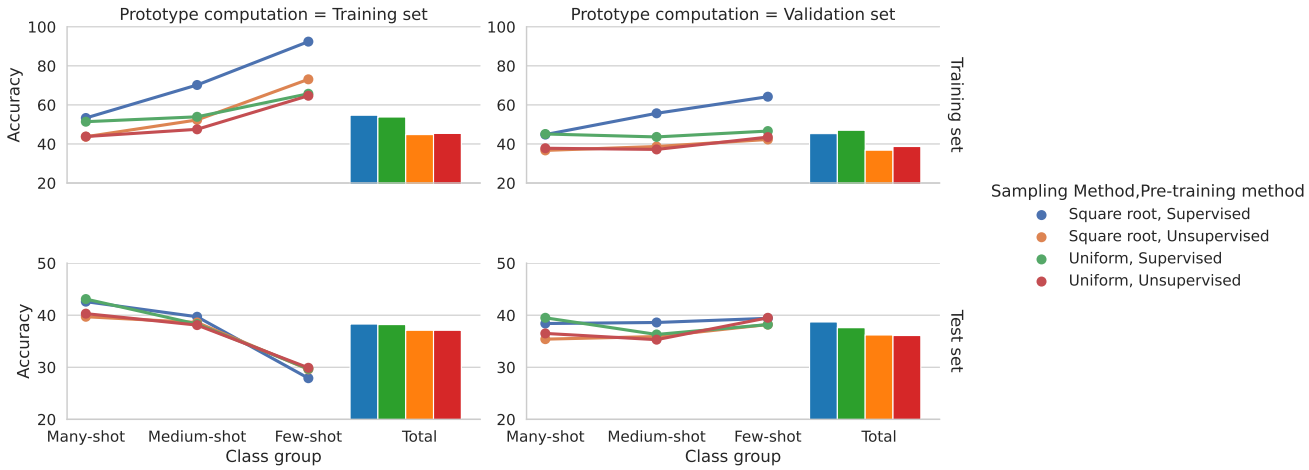

Figure 2. Influence of data sampling strategies and pre-training weight choice on class separability for the Places-LT dataset. Prototype-based prediction accuracy is computed with respect to three different class groups on the training and test sets, with prototypes computed on training and validation sets.

We can see that models trained using a supervised initialisation achieve better training accuracy, suggesting underfitting when using an unsupervised initialisation. We note that the supervised models achieve better performance on classes with sufficient data, while better performance is obtained for few-shot classes when using unsupervised models.

Further to this, we seek to analyse the compatibility between classifiers and prototypes, as well as the sharpness of our attention mechanism. To evaluate this we compute, for each class prototype, the cosine distance to all cosine classifiers weights. Firstly, we evaluate the number of class prototypes which are not closest to their corresponding cosine weight (*i.e.* the prototype from class A is closer to the classifier from class B, instead of the classifier from class A). Selecting the wrong class suggests poorer compatibility, reducing the accuracy of our attention mechanism and knowledge transfer process. We report this result in Table 3, showing that square root models achieve better compatibility, and that the supervised model yields the worst results.

In addition to this, we evaluate how sharp cosine similarity distributions are between a given prototype and classifier weights. Intuitively, one seeks sharp distributions, with only a handful of classes possessing high similarity with the prototype of interest so as to only transfer knowledge from the most relevant classes. This is visualised in Figure 4, where it may be observed that backbones relying on unsupervised weights tend to obtain sharper, and therefore more selective distributions. While square root and uniform sampling appear to behave similarly, we note that uniform sampling yields slightly sharper distributions, giving it a slight edge again.

Table 3. Number of classes where prototypes are not closest to their corresponding cosine classifier for multiple backbones.

Backbone	Mismatch count
Uniform, Supervised	39
Uniform, Unsupervised	17
Square root, Supervised	0
Square root, Unsupervised	0

In light of this, we expect models initialised with unsupervised weights to achieve stronger performance due to their higher prototype / classifier compatibility.

6.2. Ablation experiments

We provide additional detailed results on the Places-LT dataset for all studied ResNet101 backbones in Table 4. As was conjectured in the previous section, we achieve better

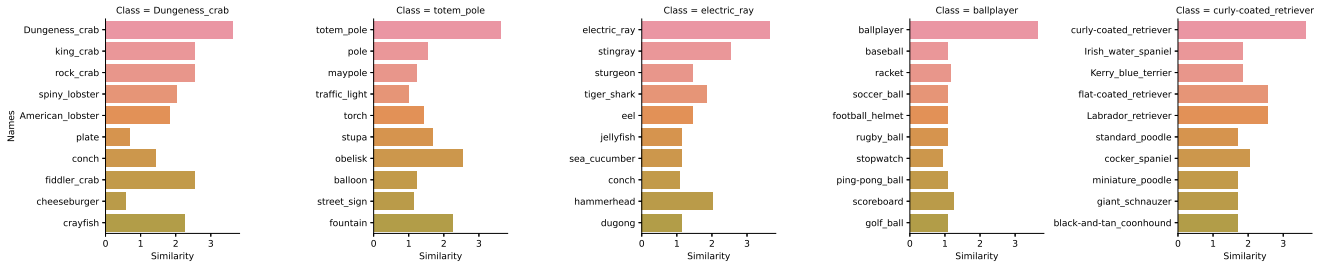

Figure 3. For five randomly selected classes, we report the ten nearest classes in terms of cosine similarity, with respect to their semantic similarity according to the WordNet taxonomy.

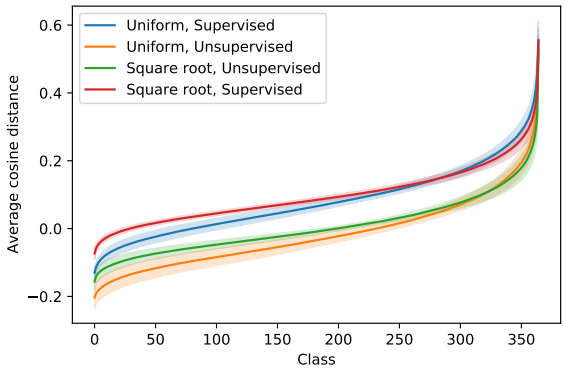

Figure 4. Influence of data sampling strategies and pre-training weight choice on prototype compatibility and our attention mechanism. We report, for sampling and pre-training weights considered, the average over all prototypes of the sorted cosine similarity between a class prototype and cosine classifier weights of all classes.

performance using an unsupervised initialisation, and, remarkably, that performance is equivalent between the two sampling strategies with 40.2 total accuracy.

Interestingly, the backbone exhibiting the poorest performance combines a supervised initialisation with uniform sampling, resulting in the weakest performance in almost all settings. Nonetheless, we note that all backbone configurations outperform state of the art method [10] when employing a ResNet101 backbone.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2

[2] 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. Ieee, 2009. 1

[3] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5414–5423, 2021. 2

[4] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. *arXiv preprint arXiv:2012.00321*, 2020. 1

[5] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019. 1

[6] Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998. 2

[7] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 1

[8] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 1

[9] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. *arXiv preprint arXiv:2007.10740*, 2020. 1

[10] 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*, pages 2361–2370, 2021. 4, 5

[11] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1

Table 4. Detailed classification accuracies and ablations on the Places-LT dataset. Bold numbers highlight the best performing classifier type per backbone.

Method	Many-shot	Medium shot	Few shot	Total
Supervised initialisation, square root				
Prototypes	42.6	39.7	27.9	38.3
Cosine classifier	**47.5**	35	19.7	36.3
Classifier + prototypes	46.6	37.2	22.1	37.4
$w^h(20)$	41.6	30.4	**44.8**	37.5
$w^h(100)$	27.8	**46.9**	33.2	37.1
ensemble($w^{hc}(20)$,$w^{hc}(100)$) (**continual**)	36.2	37.8	41.9	38.1
ensemble($w^h(0)$,$w^h(20)$,$w^h(100)$)	40.8	40.1	34.9	**39.3**
Unsupervised initialisation, square root				
Prototypes	39.7	38.6	29.6	37.1
Cosine classifier	**48.4**	33.8	18.2	35.8
Classifier + prototypes	45.7	38.4	25.1	38.2
$w^h(20)$	42.3	32.6	**44.3**	38.6
$w^h(100)$	31.0	**47.2**	34.2	38.6
ensemble($w^{hc}(20)$,$w^{hc}(100)$) (**continual**)	38.5	38.4	39.5	38.6
ensemble($w^h(0)$,$w^h(20)$,$w^h(100)$)	41.6	41.4	35.1	**40.2**
Supervised initialisation, uniform				
Prototypes	43.1	38.3	29.7	38.2
Cosine classifier	**48.1**	25.7	10.0	30.5
Classifier + prototypes	47.4	35.0	21.0	36.5
$w^h(20)$	42.2	34.3	**39.8**	38.3
$w^h(100)$	31.6	**44.1**	31.9	37.0
ensemble($w^{hc}(20)$,$w^{hc}(100)$) (**continual**)	34.9	37.7	37.0	36.5
ensemble($w^h(0)$,$w^h(20)$,$w^h(100)$)	40.6	39.7	34.8	**39.0**
Unsupervised initialisation, uniform				
Prototypes	40.3	38.1	29.9	37.1
Cosine classifier	**48.9**	27.0	13.2	32.0
Classifier + prototypes	46.4	37.0	24.1	37.7
$w^h(20)$	43.0	35.2	**41.5**	39.4
$w^h(100)$	33.5	**45.8**	34.5	39.0
ensemble($w^{hc}(20)$,$w^{hc}(100)$) (**continual**)	38.9	38.9	37.7	38.7
ensemble($w^h(0)$,$w^h(20)$,$w^h(100)$)	42.0	41.7	34.3	**40.2**
Disalign R101 [10]	39.1	42.0	29.1	38.5