Divide and Conquer: Two-Level Problem Remodeling for Large-Scale Few-Shot Learning

Mohamadreza Fereydooni *
Department of Computer Engineering
Sharif University of Technology
mrezafereydooni@gmail.com

Hosein Hasani*
Department of Computer Engineering
Sharif University of Technology
hosein.hasani@sharif.edu

Ali Razghandi
Department of Computer Engineering
Sharif University of Technology
ali.razghandi@sharif.edu

Mahdieh Soleymani Baghshah
Department of Computer Engineering
Sharif University of Technology
soleymani@sharif.edu

Abstract

Few-shot learning methods have achieved notable performance in recent years. However, few-shot learning in large-scale settings with hundreds of classes is still challenging. In this paper, we tackle the problems of large-scale few-shot learning by taking advantage of pre-trained foundation models. We recast the original problem in two levels with different granularity. At the coarse-grained level, we introduce a novel object recognition approach with robustness to sub-population shifts. At the fine-grained level, generative experts are designed for few-shot learning, specialized for different superclasses. A Bayesian schema is considered to combine coarse-grained information with fine-grained predictions in a winner-takes-all fashion. Extensive experiments on large-scale datasets and different architectures show that the proposed method is both effective and efficient besides its simplicity and natural problem remodeling. The code is publicly available at $https://github.com/mohamadreza99/divide\_and\_conquer$.

1 Introduction

Thanks to the success of deep neural networks, the visual object recognition problem has reached human performance in the past decade. However, traditional solutions require large datasets with hundreds of instances for each object. Moreover, the generalization capability of these methods is restricted and they cannot easily generalize to unseen classes of objects. In contrast, humans can learn and generalize about novel objects with considerably fewer samples. This challenge has been somewhat addressed in the few-shot learning paradigm using knowledge transfer mechanisms like meta-learning [23, 4]. Most of the existing few-shot learning methods have focused on problems with a limited number of target classes. Nevertheless, there is a recent growth of interest in large-scale few-shot learning as it covers a more realistic setting with a large number of target classes [12, 6, 2]. This framework could be more challenging since distinguishing different categories in the learned feature space becomes harder when there are hundreds of object categories with few samples.

Usually, large-scale few-shot learning problems encounter more particular objects with minimal variety in each class. Fine-grained object categories can naturally be grouped in larger clusters, forming coarse-grained superclasses with a higher variety of objects. Superclasses also can be recursively grouped and create a hierarchy of classes in a tree-like structure. This hierarchical

*Equal contribution

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

structure is provided in large-scale datasets like iNaturalist [25] and ImageNet [3] and could be used as an additional inductive bias for large-scale few-shot classification problems to improve their performance. Considering this hierarchical nature of object categories, humans can categorize the same object with different granularity corresponding to the different levels of this hierarchy. Interestingly, it is not required to treat all superclasses as unseen classes, since humans already know most of the coarse-grained categories. For example, consider showing a picture of a lynx to an adult person. One may face this species for the first time but it can easily be categorized as Feliformia, so other animal types and their descendants are completely excluded during inference.

Pre-trained foundation models have shown remarkable performance on visual object recognition tasks and have gained noticeable attention in the past few years [20, 1]. Their rich feature embedding space could be used for few-shot or even zero-shot learning tasks without additional training. However, the direct use of these pre-trained models in the large-scale few-shot setting is still challenging, and further considerations should be taken into account. Especially, when fine-grained object categories are very specific and differ from each other with few attributes, distinguishing a large number of unseen object categories becomes difficult.

Motivated by the aforementioned aspects, we propose a novel large-scale few-shot classifier using pre-trained foundation models. For simplicity, in this work, we consider only two levels of hierarchy. At the coarse-grained level, we apply a classifier to recognize superclasses. Although object superclasses are considered to be the same between train and evaluation, there exists a major subpopulation shift since evaluation base-classes are completely distinct from those of train. We design a new approach to improve the robustness to this subpopulation shift in the few-shot learning setting. The primary prediction task of large-scale few-shot learning should be taken at the fine-grained object categories. All of the extra information about the hierarchy structure and superclass inference is used to improve this final prediction. In this work, inspired by the human cognition system, we develop a Bayesian approach to incorporate superclass information as a prior for more precise inference at the base-class level. Figure 1 demonstrates the overall procedure of the proposed method during inference time.

Evaluating on large-scale datasets with different backbones shows that the proposed method is simple yet effective for large-scale few-shot classification problems. The main contributions of this paper are summarized as follows:

- We propose a novel large-scale few-shot learning method by leveraging the natural hierarchy of object categories and designing further inductive biases based on that.
- We utilize pre-trained foundation models as rich embedding networks to increase robustness to the distribution shift and also data efficiency.
- We develop a simple strategy for alleviating sub-population shift at the superclass level.
- We show that the proposed method can be easily employed in the free-training setting.

2 Related Work

Few-shot learning problem has been widely studied in recent years. Most of these methods leverage meta-learning approach to make learning from a few training samples feasible. Generally, these methods can be divided into two main categories, metric-based methods [26, 23, 15] and optimization-based methods [4, 21]. In this work, we utilize prototypical learning [23] which is one of the mostly used metric-based methods for few-shot learning because of its simplicity, effectiveness, and also being generative approach. Most of the classical few-shot learning methods are designed for small datasets like Cifar [10], Omniglot [11], and miniImageNet [26]. In the past few years, large-scale few-shot learning became more topical [12, 6]. To address data limitations, some works utilize generators for data augmentation [7, 27]. Some other works have proposed methods to infer a good initialization point for novel classes from a few training samples [18, 19, 2].

To increase data efficiency, some recent studies have focused on the power of transfer learning and using large and expressive backbones on large datasets before few-shot learning [9, 29, 24]. In addition to standard supervised learning, unsupervised [29, 24] and self-supervised learning [14] have shown to be effective approaches for providing a proper backbone for few-shot learning. More recently, well-received large foundation models such as CLIP [20] and DINO [1] have gained considerable attention for zero-shot [16, 5] and few-shot learning [30, 31]. For zero-shot settings, normally multimodal models are used to infer a prototype for a class based on its label.

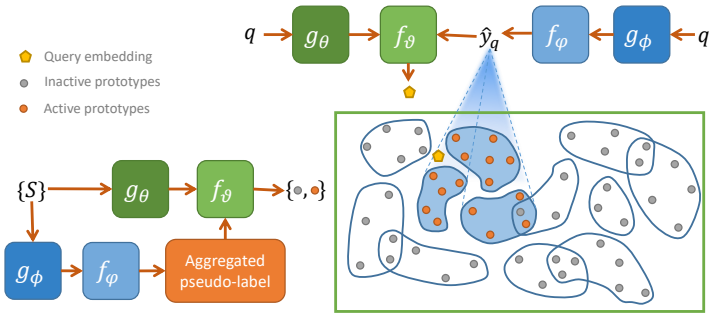

Figure 1: Overview of the proposed method. The coarse-grained (colored in blue) classifier restricts the scope of plausible base-classes in a top-down manner and increases the accuracy of the fine-grained classifier (colored in green) by reducing the number of active base-classes. Dark and light colors are used to indicate foundation model backbones (g_θ and g_ϕ) and classification heads (f_ϑ and f_φ) respectively.

Using some kind of knowledge graphs has shown to be an effective way to improve the performance of few-shot learning methods especially in large-scale settings [12, 2, 6, 28]. In [12], different classification heads are used for different levels of the hierarchy to learn a feature space that is suited for different granularity. Then, a simple nearest-neighbor strategy is shown to be an effective way for large-scale classification in the obtained feature space. Similar to these works, we also leverage a hierarchical knowledge graph as an inductive bias for large-scale few-shot learning. However, our work differs from the mentioned works as it utilizes a top-down Bayesian approach to provide coarse-grained priors for final fine-grained classification.

3 Proposed Method

3.1 Preliminaries

In hierarchical large-scale few-shot learning, we assume to have access to a large-scale dataset $\mathcal{D}_{source}$ and $\mathcal{D}_{target}$ respectively for the training and testing. Each dataset $\mathcal{D} = \{(x^i, y^i_{super}, y^i_{base})\}_{i=1}^{|\mathcal{D}|}$ is composed of three sets, (1) image samples $\mathcal{X} = \{x^i\}$, (2) superclass labels $\mathcal{Y}_{super} = \{y^i_{super}\}$ at the coarse-grained level, and (3) base-class labels $\mathcal{Y}_{base} = \{y^i_{base}\}$ at the fine-grained level. For the coarse-grained level, we assume that $\mathcal{Y}_{super,target} \subseteq \mathcal{Y}_{super,source}$, but for the fine-grained level we assume $\mathcal{X}_{source} \cap \mathcal{X}_{target} = \emptyset$ and $\mathcal{Y}_{base,source} \cap \mathcal{Y}_{base,target} = \emptyset$. Since the primary task is the few-shot learning at the fine-grained level, each y_{target} contains a few labeled samples. Furthermore, the proposed method does not require sufficient training data samples and by leveraging large foundation models we show that it is also possible to achieve acceptable performance in a free-training manner.

For few-shot learning, we employ prototypical learning as a meta-learning approach. Meta-training and meta-testing are performed in an episodic fashion. In each episode a random task $\mathcal{T}$ is sampled from $\mathcal{D}$, forming a N-way K-shot problem with N random classes and K support samples in each class. In each task, the samples of class c are split into support set and query set which we denote by $\mathcal{S}^c$ and $\mathcal{Q}^c$, respectively. The goal of meta-learning is to achieve good recognition performance on $\mathcal{Q}$ by fast adaptation on a few support samples $\mathcal{S}$.

We utilize pre-trained deep networks to provide feature embeddings for classification or few-shot learning tasks. The embedding networks of the superclass level and base-class level are denoted by g_ϕ and g_θ, respectively. We further employ additional MLP networks f_φ and f_ϑ to perform desired tasks (e.g. classification) based on raw features of the embedding backbones.

3.2 Coarse-Grained Classifier

Since target superclasses are not distinct from source superclasses, formal supervised learning approaches can be applied at this level. In the first step, we apply the standard cross-entropy loss to train classifier f_φ on top of the backbone g_ϕ. Since we are using the pre-trained network g_ϕ as a

feature extractor, it is not required training dataset to be large with sufficient samples. Furthermore, base-classes from the same superclass are aggregated together for this task. So, even if the original dataset is suited for few-shot learning, the aggregated dataset could be used for formal classification.

Although the set of superclasses is not novel in the testing phase, there exists a major challenge. The set of base-classes in each superclass are completely distinct between train and test datasets. To alleviate this challenge, we employ a semi-supervised learning approach. Using the fact that support samples of each base-class share the same (unknown) superclass label, it is possible to enhance coarse-grained prediction on each support sample, using the prediction probabilities of the other support samples. To this end, by using conditional independence of data samples and applying the Bayes rule two times, we obtain:

$$P(y|\mathcal{S}) = P(y|x^1, \ldots, x^K) = \frac{P(y)\prod_{i=1}^{K} P(x^i|y)}{P(x^1, \ldots, x^K)} = \frac{P(y)\prod_{i=1}^{K} P(y|x^i)P(x^i)}{P(y)^K P(x^1, \ldots, x^K)}. \quad (1)$$

So, to estimate superclass pseudo-labels for support samples, it is enough to compare the logarithm of conditional likelihoods for all superclasses and consider the highest one. Using Non-uniform prior for different superclasses, $\underset{y_{super}}{\operatorname{argmax}} \sum_{i=1}^{K} \log P(y_{super}|x^i) - (K-1)\log P(y_{super})$ defines the winner superclass. The prior probability of each superclass is considered to be proportional to the number of its base-classes in the training set. We refer to the parameters of approximators by ϕ' and φ' after fine-tuning on superclass pseudo-labels.

3.3 Fine-Grained Few-Shot Learner

The standard prototypical learning approach is used for few-shot object recognition at the fine-grained level. The coarse-grained information is utilized through two separate mechanisms:

1. Fine-grained MLP f_ϑ is conditioned by the final prediction of coarse-grained MLP f_φ. Given this additional information, few-shot learning at the fine-grained level is performed in a multi-task learning manner. We could consider superclasses as different tasks and employ different experts, i.e. a shared MLP network with specific conditions for each task, to learn these tasks in parallel.

2. To reduce the risk of misclassification in the large-scale setting, we restrict the possible target base-classes for each query sample by defining the most probable superclasses first. To this end, coarse-grained classification is first applied on the query sample, and top-w relevant superclasses are estimated. Given these estimations, irrelevant base-classes could be omitted by taking a winner-takes-all strategy.

Formally, we assign the probability of a query sample q belonging to the base class c by taking a generative approach. By applying the Bayes rule and using conditional independency, the posterior predictive probability can be calculated as follows (see Supplementary Materials 6 for more details):

$$P(y_{base} = c|q, \{\mathcal{S}\}, f_\vartheta, g_\theta, f_{\varphi'}, g_{\phi'}) \propto \sum_{\hat{y}_{super}=1}^{|\mathcal{Y}_{super}|} P(q|\mathcal{S}^c, f_\vartheta, g_\theta, \hat{y}^i_{super}) P(\hat{y}^i_{super}|q, f_{\varphi'}, g_{\phi'}). \quad (2)$$

We observed that the conditional likelihood $P(\hat{y}_{super}|q, f_{\varphi'}, g_{\phi'})$ for one class is usually dominant to the others. So, to increase the computational efficiency we assign the probability of zero to the dominated superclasses and approximate the final summation in 4 by the likelihood of the winner superclass (or top-w winner superclasses), resulting the class conditional probability $P(q|\mathcal{S}^c, f_\vartheta, g_\theta, f_\varphi, g_\phi, \hat{y}_{super,winner})$. We assume spherical Gaussian distribution for modeling the likelihood of each base-class. Given episode supports samples and by calculating prototypes as $\mathcal{P}_{\mathcal{S}^j} = \frac{1}{|\mathcal{S}^j|} \sum_{x^i \in \mathcal{S}^j} f_\vartheta\left(g_\theta(x^i), f_\varphi(g_\phi(x^i))\right)$, the final prediction for base-classes is derived by

normalizing through the softmax function:

$$P(y_{base} = c|q, \{\mathcal{S}\}, f_\vartheta, g_\theta, f_\varphi, g_\phi) \propto \frac{P(q|\mathcal{S}^c, f_\vartheta, g_\theta, f_\varphi, g_\phi, \hat{y}_{super,winner})}{\sum_{j \in \mathcal{Y}_{base}} P(q|\mathcal{S}^j, f_\vartheta, g_\theta, f_\varphi, g_\phi, \hat{y}_{super,winner})} = \frac{\exp\left(-\|f_\vartheta\left(g_\theta(q), f_\varphi(g_\phi(q))\right) - \mathcal{P}_{\mathcal{S}^c}\|^2\right)}{\sum_{j \in \mathcal{Y}_{base}} \exp\left(-\|f_\vartheta\left(g_\theta(q), f_\varphi(g_\phi(q))\right) - \mathcal{P}_{\mathcal{S}^j}\|^2\right)}. \tag{3}$$

Finally, based on these probabilities, the cross-entropy loss is performed to optimize the parameters of g_θ and f_ϑ. Algorithms 1 and 2, formally summarize the procedure of training coarse and fine-grained classifiers in Supplementary Materials 6.

4 Experiments

In this section, we demonstrate the effectiveness of our proposed method through extensive experiments and compare them in different settings. We show that it is applicable for a range of model architectures from light backbones to heavy foundation models. Moreover, we explain how we redefined and used popular large-scale datasets to adapt them to our structure.

4.1 Datasets

In large-scale problems, preparing a proper large dataset with a large number of classes is a great deal of work due to the difficulty of the annotation procedure. So there are limited computer vision datasets that have more than 1000 labeled classes. Furthermore, there are additional challenges in the formation and structure of large-scale datasets. These datasets are usually general purpose and there is not enough coherence in class data points. Moreover, there is coupling among classes and interdependency. Hence, we select two multi-granular datasets, iNat Animalia and iNat Plantae [25] for evaluations. We take the finest granularity as the base-classes and one of the coarse-grained levels as the superclasses. We split all base-classes into source domain and target domain data. In order to better verify the performance of the proposed method, we limit our datasets to ones which were not used in the training procedure of the pre-trained backbones. There is a detailed explanation of dataset details in the Supplementary Material 6.3.

4.2 Backbones

Dino Giant is the largest model in the Dino v2 series [17]. Dino v2 models are trained in self-supervised learning in a new way of using knowledge distillation. This model produces very high-performance visual features in practice which could be used in different downstream tasks. These models are trained on a dataset of 142 M images. Another backbone of this family which we used is **Dino Large** with lower than $\frac{1}{3}$ of the giant version parameters.

ConvNext v2 is a fully convolutional network with comparative performance to vision transformers, introduced by Facebook research group [13]. We use their large backbone which is pre-trained on ImageNet-21k. **MobileNet v3** is a backbone we choose to illustrate the effectiveness of our method on a relatively small backbone [8]. We select the pre-trained weights of this model from [22].

4.3 Coarse-Grained Classifier

This classifier consists of g_ϕ frozen backbone which is Dino v2 Giant and a learnable linear classification layer with the output size of superclass labels $|\mathcal{Y}_{super}|$. It is trained on $\mathcal{D}_{source}$ data with a batch size of 64 in a supervised learning manner with their superclass labels $\mathcal{Y}_{super}$. After this, as the second phase of the Algorithm 1, we perform our specific semi-supervised learning by using 5-shot samples of target base-classes and estimating the group labels. Then we finetune the classification head f_φ on $\mathcal{D}_{super,target}$ with estimated pseudo-labels.

4.4 Fine-Grained Classifier

In the fine level, we used all the pre-trained backbones introduced in 4.2 as g_θ. Two fully connected layers are used at the top of these backbones with a GELU activation between them as f_ϑ. During training, g_θ is frozen for the CLIP and DINO families. However, the MobileNet backbone is fine-tuned since it is light. In the case of ConvNext, only normalization layer parameters are fine-tuned and the others remain frozen during training. We adopt prototypical learning in an episodic way, so we sample a 650-way 5-shot task $\mathcal{T}$ from $\mathcal{D}_{source}$ in the meta-training phase. Since the problem is large-scale, we examine to what extent the model can generalize by creating one task from $\mathcal{D}_{target}$ which contains all the base classes. In other words, we perform meta-testing with a large-scale task of $|\mathcal{Y}_{base,target}|$-way 5-shot and evaluate the classification accuracy with query samples of each class.

4.4.1 Baselines

To evaluate the effectiveness of two components introduced in 3.3, we make two different ablation baselines of our method with a simple modification of f_ϑ. In the first baseline, we just use embeddings as input but in the second one, the one-hot vector of the determined superclass is concatenated to embeddings. The true labels are used as conditioned superclasses during the training but during the testing phase superclasses are inferred from coarse-grained classifier. Furthermore, to assess the effectiveness of imposing a hierarchical structure, we set our main baseline to flat prototypical learning which lacks hierarchical knowledge. Lastly, we evaluate the richness of pre-trained networks by omitting the meta-training phase. In this free-training setting, we do not apply f_ϑ on embedding space, and prototypical learning is performed directly on raw embeddings.

4.5 Results

Table 1 shows the top-1 and top-5 accuracy of the Dino v2 Giant on iNat Animalia and iNat Plantae datasets. Since the pseudo-label accuracy for support samples is higher than top-1 accuracy, by fine-tuning on these pseudo-labels the overall accuracy is increased, especially in iNat Plantae dataset.

Table 1: Accuracy of the coarse-grained classifier on $\mathcal{D}_{super,target}$, before and after semi-supervised fine-tuning.

Setting / Dataset	Top-1	Top-5	Pseudo-Label Acc.	Top-1 Fine Tuned	Top-5 Fine Tuned	Relative Gain Top-1	Relative Gain Top-5
iNat Animalia	87.38	97.26	95.86	88.88	97.83	+1.7	+0.6
iNat Plantae	79.68	92.45	90.51	87.86	97.1	+10.3	+5

Evaluation results at the fine-grained level of iNat Animalia and iNat plantae datasets are shown in Tables 2 and 3. As seen, the proposed method is comparatively better than the flat baseline in different backbones and settings. However, some differences are observed. Overall, the relative gain is more considerable in weaker baselines. This indicates that large and expressive foundation models are more robust, so even flat prototypical learning without additional conditioning may result in high accuracy. Table 3 demonstrates the results of the free-training setting. By comparing these results with those of Table 2, it is observed that stronger foundation models are more reliable to use in free-training settings compared to weaker counterparts, especially in iNat Plantae dataset.

5 Conclusion

In this study, we have focused on large-scale few-shot learning problems by proposing a Bayesian hierarchical approach. The proposed method has two main components designed for different levels of hierarchy. The first component is a coarse-grained classifier equipped with a novel semi-supervised learning method to alleviate the inherent sub-population shift problem. The second component is a conditional fine-grained classifier which is designed for object recognition in the presence of a large amount of novel base-classes. A Bayesian approach is considered to combine these components through a top-down mechanism. More importantly, using pre-trained foundation models makes large-scale classification in a low-data regime feasible by increasing data efficiency. Since in our large-scale setting, the base-classes are very specific and their names or samples are almost new for

Table 2: Top-1 accuracy of prototypical learning on $\mathcal{D}_{base,target}$.

Backbone \ Dataset	iNat Animalia							iNat Plantae						
	Baseline	Top-W Superclass				Relative	Oracle	Baseline	Top-W Superclass				Relative	Oracle
	(Flat)	1	2	5	10	Gain	Superclass	(Flat)	1	2	5	10	Gain	Superclass
DINO Large	53.52	*50.95*	*52.83*	*53.55*	***53.56***	+6.3	57.31	72.87	*68.48*	*71.44*	***72.68***	*72.94*	+0.2	77.91
DINO Large Cond.		*54.83*	*56.33*	***56.87***	*56.8*		60.12		*68.57*	*71.71*	***73.02***	*73.02*		77.99
DINO Giant	61.68	*58.64*	*60.77*	*61.55*	***61.65***	+2	64.4	74.77	*69.66*	*72.86*	***74.49***	*74.74*	+0.3	79.11
DINO Giant Cond.		*60.07*	*62.19*	*62.89*	***62.92***		65.48		*69.96*	*73.23*	*74.58*	***74.96***		79.54
CLIP	33.88	***39.12***	*38.9*	*37.36*	*36.29*	+16.9	44.96	38.48	*51.54*	*49.63*	***45.75***	*43.19*	+31.1	58.9
CLIP Cond.		*39.48*	*39.53*	*39.57*	***39.61***		43.48		*50.26*	*50.4*	***50.45***	*50.45*		52
Mobilenet V3	41.33	*43.7*	***44.39***	*43.77*	*43.15*	+8.8	50.2	37.15	*49.88*	*47.94*	***44.37***	*41.4*	+30.5	57.72
Mobilenet V3 Cond.		*43.6*	*44.39*	*44.8*	***44.95***		47.88		*48.22*	*48.46*	***48.49***	*48.47*		50.41
ConvNext V2	46.09	*46.16*	*47.49*	***47.69***	*47.57*	+6.7	52.83	61.02	*61.94*	*63.2*	***62.6***	*62.18*	+2.4	70.16
ConvNext V2 Cond.		*45.85*	*47.75*	*48.81*	***49.19***		51.68		*61.04*	*62.42*	***62.47***	*62.47*		68.08

Table 3: Top-1 accuracy of prototypical free-training on $\mathcal{D}_{base,target}$.

Backbone \ Dataset	iNat Animalia							iNat Plantae						
	Baseline	Top-W Superclass				Relative	*Oracle*	Baseline	Top-W Superclass				Relative	Oracle
	(Flat)	1	2	5	10	Gain	*Superclass*	(Flat)	1	2	5	10	Gain	Superclass
DINO Large	38.1	*38.17*	**38.48**	*38.28*	*38.17*	+1	42.1	67.42	*63.66*	66.25	*67.39*	***67.52***	+0.2	72.55
DINO Giant	35	*34.65*	**35.23**	*35.13*	*35.1*	+0.7	38.44	64.37	*60.64*	63.06	*64.13*	***64.4***	+0.1	69.26
CLIP	17.09	***24.52***	23.04	*21*	*19.61*	+43.5	28.12	22.82	***39.97***	35.75	*30.93*	*28.06*	+75.2	45.45
ConvNext V2	34.65	*36.48*	**36.84**	*36.37*	*36.2*	+6.3	41.8	55.2	*58.05*	**58.49**	*57.49*	*56.4*	+6	65.57

foundation models, a zero-shot approach may result in significantly poor performance, and solving this issue could be a good direction for future studies.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[2] Jingjing Chen, Linhai Zhuo, Zhipeng Wei, Hao Zhang, Huazhu Fu, and Yu-Gang Jiang. Knowledge driven weights estimation for large-scale few-shot image recognition. *Pattern Recognition*, 142:109668, 2023.

[3] 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.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[5] Yunhao Ge, Jie Ren, Andrew Gallagher, Yuxiao Wang, Ming-Hsuan Yang, Hartwig Adam, Laurent Itti, Balaji Lakshminarayanan, and Jiaping Zhao. Improving zero-shot generalization and robustness of multi-modal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11093–11101, 2023.

[6] Jiechao Guan, Manli Zhang, and Zhiwu Lu. Large-scale cross-domain few-shot learning. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[7] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE international conference on computer vision*, pages 3018–3027, 2017.

[8] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[9] Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9068–9077, 2022.

[10] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[11] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[12] Aoxue Li, Tiange Luo, Zhiwu Lu, Tao Xiang, and Liwei Wang. Large-scale few-shot learning: Knowledge transfer with class hierarchy. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7212–7220, 2019.

[13] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[14] Yuning Lu, Liangjian Wen, Jianzhuang Liu, Yajing Liu, and Xinmei Tian. Self-supervision can be a good few-shot learner. In *European Conference on Computer Vision*, pages 740–758. Springer, 2022.

[15] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and P. Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

[16] Zachary Novack, Julian McAuley, Zachary Chase Lipton, and Saurabh Garg. Chils: Zero-shot image classification with hierarchical label sets. In *International Conference on Machine Learning*, pages 26342–26362. PMLR, 2023.

[17] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[18] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5822–5830, 2018.

[19] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7229–7238, 2018.

[20] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[21] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2016.

[22] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[23] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[24] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 266–282. Springer, 2020.

[25] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[26] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[27] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7278–7286, 2018.

[28] Tz-Ying Wu, Pedro Morgado, Pei Wang, Chih-Hui Ho, and Nuno Vasconcelos. Solving long-tailed recognition with deep realistic taxonomic classifier. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 171–189. Springer, 2020.

[29] Hui Xu, Jiaxing Wang, Hao Li, Deqiang Ouyang, and Jie Shao. Unsupervised meta-learning for few-shot learning. *Pattern Recognition*, 116:107951, 2021.

[30] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15211–15222, 2023.

[31] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision*, pages 493–510. Springer, 2022.

6 Supplementary Material

6.1 Algorithms

Algorithms 1 and 2 demonstrate the pseudo-code of the training procedure in coarse-grained and fine-grained classifiers.

Algorithm 1 Coarse-grained classifier (training)

Input: dataset $\mathcal{D}_{source}$ and $\mathcal{D}_{target}$, backbone g_ϕ, classification head f_φ
Output: backbone $g_{\phi'}$, classification head $f_{\varphi'}$

while not converged **do** ▷ Supervised learning
  sample batch $\{(x^i, y^i_{super})\}_{i=1}^{B} \sim \mathcal{D}_{source}$
  $\hat{y}^i_{super} \leftarrow f_\varphi(g_\phi(x^i))$ ▷ Do this for all of the batch samples
  update ϕ and φ based on $\frac{1}{B}\sum_{i=1}^{B} \mathcal{L}(y^i_{super}, \hat{y}^i_{super})$
end while
initialize $\phi' \leftarrow \phi, \varphi' \leftarrow \varphi$
for $c \in \mathcal{Y}_{base,target}$ **do** ▷ Semi-supervised learning
  take all support samples $\mathcal{S}^c$
  $\tilde{y}_{super} \leftarrow \text{argmax} \prod_{x^i \in \mathcal{S}^c} f_{\varphi'}(g_{\phi'}(x^i))$
  $\hat{y}^i_{super} \leftarrow f_{\varphi'}\left(g_{\phi'}\left(x^i\right)\right)$ ▷ Do this for all of the support samples
  update ϕ' and φ' based on $\frac{1}{|\mathcal{S}^c|}\sum_{i=1}^{|\mathcal{S}^c|} \mathcal{L}(\tilde{y}_{super}, \hat{y}^i_{super})$
end for

Algorithm 2 fine-grained classifier (training)

Input: dataset $\mathcal{D}_{source}$ and $\mathcal{D}_{target}$, backbone g_θ, classification head f_ϑ
Output: backbone $g_{\theta'}$, classification head $f_{\vartheta'}$

while not converged **do** ▷ Conditional prototypical learning
  sample $\{\mathcal{S}^j, \mathcal{Q}^j\}_j \sim \mathcal{D}_{source}$ and create a N-way K-shot task $\mathcal{T}$
  calculate class conditional probabilities based on Eq. 4
  calculate class probabilities based on Eq. 3
  update θ' and ϑ' based on cross-entropy loss
end while

6.2 Base-Class Probability Derivation

We assign the probability of a query sample q belonging to the base class c by taking a generative approach. By applying the Bayes rule and using conditional independency, the posterior predictive probability can be calculated as follows:

$$
\begin{aligned}
P(y_{base} = c|q, \{\mathcal{S}\}, f_\vartheta, g_\theta, f_\varphi, g_\phi) \propto \\
\sum_{\hat{y}_{super} \in \mathcal{Y}_{super}} P(y_{base} = c|q, \{\mathcal{S}\}, f_\vartheta, g_\theta, f_\varphi, g_\phi, \hat{y}_{super}) P(\hat{y}_{super}|q, \{\mathcal{S}\}, f_\vartheta, g_\theta, f_\varphi, g_\phi) \propto \\
\sum_{\hat{y}_{super} \in \mathcal{Y}_{super}} P(y_{base} = c|q, \{\mathcal{S}\}, f_\vartheta, g_\theta, \hat{y}_{super}) P(\hat{y}_{super}|q, \{\mathcal{S}\}, f_\varphi, g_\phi) \propto \\
\sum_{\hat{y}_{super} \in \mathcal{Y}_{super}} P(q|\mathcal{S}^c, f_\vartheta, g_\theta, \hat{y}_{super}) P(\hat{y}_{super}|q, f_{\varphi'}, g_{\phi'}).
\end{aligned}
\tag{4}
$$

The last equation is obtained by assuming uniform prior on base-class probabilities and applying the Bayes rule and conditional independence:

$$
\begin{aligned}
P(y_{base} = c|q, \{\mathcal{S}\}, f_\vartheta, g_\theta, f_\varphi, g_\phi) &\propto P(q|y_{base} = c, \{\mathcal{S}\}, f_\vartheta, g_\theta, f_\varphi, g_\phi) P(y_{base} = c) \\
&\propto P(q|\mathcal{S}^c, f_\vartheta, g_\theta, f_\varphi, g_\phi).
\end{aligned}
\tag{5}
$$

6.3 Dataset Details

6.3.1 Collecting and Preparing Datasets

Our proposed method requires hierarchy at the meta-train stage, so we focus on large-scale datasets with a pre-defined hierarchical structure. As we define in our problem setting in section 3.1, the base-class sets in meta-train and meta-test datasets are disjoint but the superclasses of meta-test are a subset of meta-train superclasses. In this way, we pose a sub-population shift between meta-train and meta-test and measure the robustness of the model in the few-shot adaptations to meta-test data. So, we make our datasets from large-scale hierarchical ones and adapt them to our problem setting as follows.

iNat Animalia [25] is a part of iNaturalist-2021 dataset which has 10k classes. For each fine-grained class, all of its ancestors in the hierarchy are presented according to taxonomic rank. The coarsest level in the iNat is the kingdom and it contains Animalia, Plantae, and Fungi at its first level of hierarchy. We do not include Fungi classes in our experiments since it does not contain sufficient classes to be large-scale. We refer to the Animalia part of iNaturalist2021 as iNat Animalia which contains a vast range of animal species. In order to define superclasses, we take the fifth depth of the hierarchy tree known as the Family level, resulting in 74 superclasses. After pruning small-size superclasses, 3228 base-classes remain in total which are divided into sets of meta-train with 1728 base-classes and meta-test with 1500 base classes.

iNat Plantae [25] is the Plantae part of the iNaturalist2021 dataset which we explain in the previous paragraph. It contains a vast range of plant species including 2705 classes (after pruning small-size superclasses) which we divide into sets of meta-train and meta-test with 1450 and 1255 classes, respectively. There are 58 shared superclasses between meta-train and meta-test classes which we select from the Family level of the taxonomic rank. Table 4 summarizes the details of datasets.

Table 4: Quantitative Details of iNat Animalia and iNat Plantae datasets.

	# of Superclasses	# of Source Domain Base-Classes	# of Target Domain Base-Classes
iNat Animalia	74	1728	1500
iNat Plantae	58	1450	1255

6.3.2 Dataset Challenges

Datasets with a large number of classes may possess many challenges. The overlap between classes, interdependence, abstract classes, incorrectly annotated data, classes that share a concept, etc., are all possible. This causes inevitable problems in large-scale object recognition and many curation efforts are demanded to clean up datasets. Hence, most large datasets cannot be directly used, as the yielded results cannot be reliable.

ImageNet-21k is among the most famous large-scale datasets. It is the full version of imageNet-1k [3], however, there are a lot of failure cases that prevent us from using it. As illustrated in Figure 2, in the banana class in this dataset there are, for example, images of ripe bananas, unripe bananas, banana trees (without fruit), banana boxes, banana peels, and so on. In addition, there are classes with the same identity and similar concepts. For instance, ImageNet-21k has distinct classes like "dwarf banana", "edible banana", "japanese banana", and "plantain banana".

6.4 Additional Results and Visualizations

Figure 3 visualizes the t-SNE embeddings of all superclasses and base-classes from iNat Animalia and iNat Plantae datasets using the Dino Giant backbone. Figure 4 summarizes the top-1 accuracy diagrams of Tabels 2 and 3.

Figure 2: A collection of samples in one of the imageNet-21k classes that relate to bananas samples

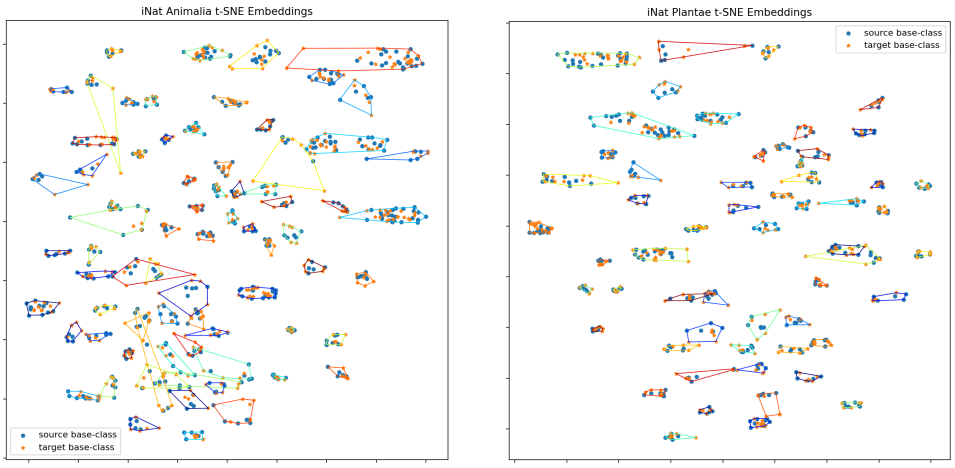

Figure 3: 2D visualization of iNat Animalia and iNat Plantae embeddings using t-SNE. Superclasses are shown by convex hulls.

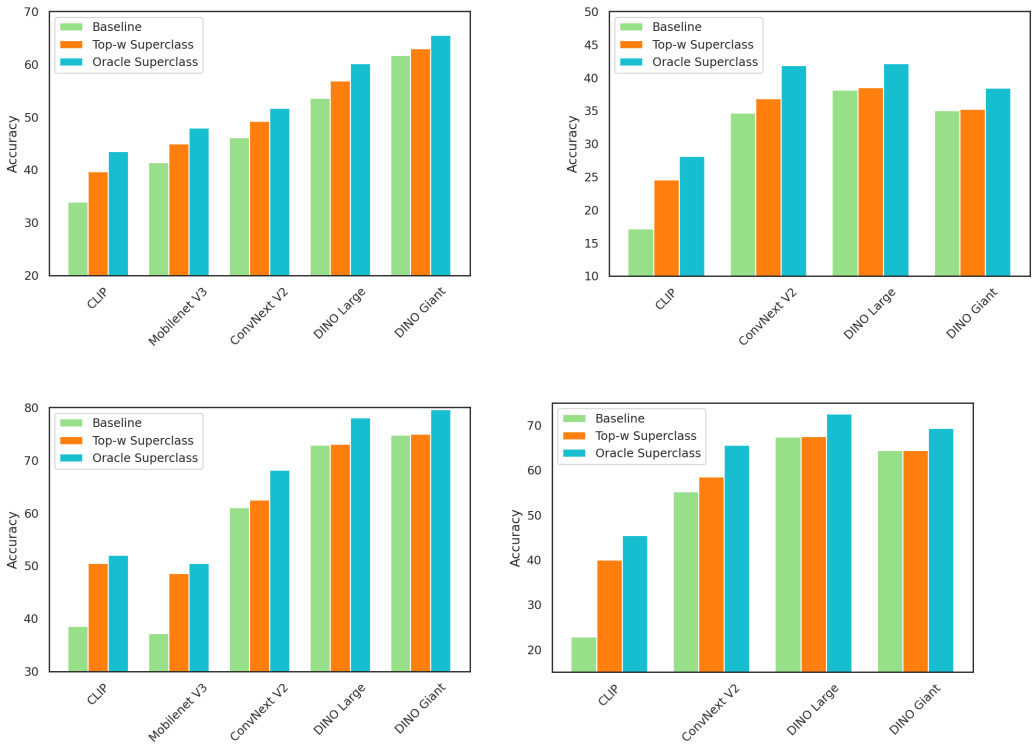

Figure 4: Top-1 accuracy of different backbones in iNat Animalia (top) and iNat Plantae (bottom) datasets. The right column demonstrates the free-training scenario.