Coarsely-labeled Data for Better Few-shot Transfer

Cheng Perng Phoo, Bharath Hariharan
Cornell University
{cpphoo, bharathh}@cs.cornell.edu

Abstract

Few-shot learning is based on the premise that labels are expensive, especially when they are fine-grained and require expertise. But coarse labels might be easy to acquire and thus abundant. We present a representation learning approach - PAS *that allows few-shot learners to leverage coarsely-labeled data available before evaluation. Inspired by self-training, we label the additional data using a teacher trained on the base dataset and filter the teacher's prediction based on the coarse labels; a new student representation is then trained on the base dataset and the pseudo-labeled dataset.* PAS *is able to produce a representation that consistently and significantly outperforms the baselines in 3 different datasets. Code is available at* *https://github.com/cpphoo/PAS*

1. Introduction

Large annotated datasets [4, 8, 19] have empowered the progress of visual recognition systems over the last decade. However, for many practically important recognition problems, annotations might require expertise and thus might be difficult to acquire. For example, to build a recognition system that identifies insect species, one would have to hire an entomologist to label hundreds of thousands of images from hundreds of species: an expensive, time-consuming affair.

This concern has sparked research in few-shot learning (FSL), which aims to train domain-specific *learners* that can learn new classes from very few examples. These learners are "meta-trained" on a large labeled dataset of "base" classes from the same domain. The hope is that this base dataset provides the learner with the right inductive biases for the domain of interest so that recognizing "novel" classes does not require quite as much labeled data. FSL is now an extremely active research area with a veritable array of recent results [33, 6, 12, 10, 16, 44, 38, 47, 30, 23]. Yet, existing FSL systems still lag far behind systems trained with large quantities of labeled training data. One might conjecture that the base dataset does not provide sufficient information about the novel classes.

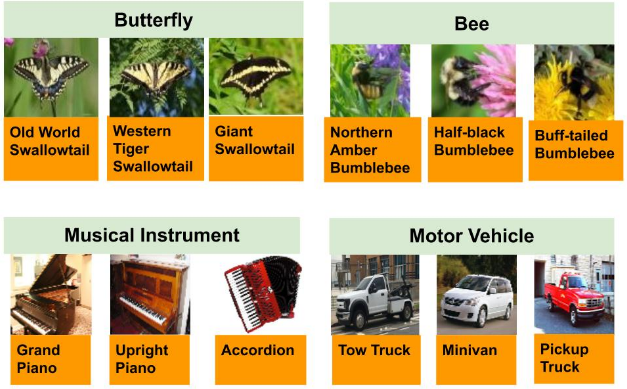

Figure 1. The top row represents 6 different fine-grained classes in iNat2019 and the bottom row consists of 6 fine-grained classes from tieredImagenet. Without domain expertise, one might find it challenging to distinguish the 6 different classes (orange) but identifying them based on their coarse labels (green) is intuitive.

One possible approach to address this issue is to leverage some *auxilliary* information about novel classes that might be more readily available. For example, recent work [26, 9, 35] uses *unlabeled* data from the novel classes: it is, after all, the *labels* that are expensive; data is often cheap. While such unlabeled data can inform the learner about the *data* distribution of the novel classes, they contain no information about the semantics of the class distinctions.

A potential source of auxiliary information about semantics is labeling at a coarser granularity, which might be easier to obtain than the actual labels of interest. Consider again the problem of insect classification. It is true that one would have to hire an entomologist or even a lepidopterist to help distinguish between the 3 species of butterflies in figure 1; these labels are therefore difficult to acquire. But a layperson would be able to distinguish between butterflies and bees. Labels at that coarse granularity can thus be crowd-sourced quite easily. This leads us to the following question: what if we had access to data from the novel classes that were weakly labeled with *easy-to-acquire coarse labels*?

Although such coarsely-labeled data are both readily available and potentially informative, no current FSL technique is capable of using this extra information. Class taxonomies have been explored in *traditional supervised learning* through hierarchical inference strategies, but it is unclear if these address the few-shot generalization problem. One could use these additional labels as an auxiliary loss in a multitask-training framework. However, it has been shown that multitask-training is not guaranteed to help all tasks [34]. Besides, multitask-training ignores the strong constraints that tie the coarse and fine labels together, thus missing out on vital semantic knowledge.

We propose a new few-shot learning approach that effectively leverages coarsely-labeled data. Following recent results, we focus on improving the feature representation, since this turns out to be crucial for FSL [38, 2, 11]. Inspired from recent work based on pseudo-labeling and self-training [26], we develop a representation learning approach named **P**arent-**A**ware **S**elf-training (PAS). Specifically, we use a classifier trained on the base dataset to provide *fine-grained pseudo-labels* to coarsely-labeled data. These pseudo-labels are *filtered* so that they are consistent with the coarse labels. These pseudo-labels will definitely be incorrect, because they will wrongly declare novel class examples to be from one of the base classes. However, they will *induce a fine-grained grouping* of the coarsely-labeled novel examples that is *consistent* with other fine-grained base classes with the same coarse labels. We then train with these pseudo-labels to produce a feature representation that hopefully captures the unknown novel class distinctions.

We experiment with three different datasets, and compare representations that use such coarsely-labeled data and those that do not. We find that using coarsely-labeled data improves five-shot accuracy between 5 to 15 points on the challenging all-way classification setup. Our particular approach is also the best way to use this additional data, providing up to an average (across datasets) of 2 points improvement in five-shot accuracy compared to multitask training. All these gains vindicate the power of this additional information and the ability of our approach to use it.

2. Related Work

Few-shot Learning (FSL). We tackle FSL in our work. There are three main categorizations of FSL techniques: initialization-based approaches [6, 7, 28, 24, 31, 36, 16] build model initializations that can lead to rapid convergence on the base classes, positing that such initializations can also be good model initializations for the novel classes; metric learning approaches [41, 33, 37, 14, 10, 27] build a metric on the base dataset, assuming that base and novel share similar discriminative features; augmentation-based approaches [12, 43, 3] aim to learn augmentation mechanisms on the base dataset, postulating that base and novel classes share some class agnostic, intra-class variations. Most FSL techniques assume no access to the data from the novel classes when training the learners and solely hinge on the similarity between base and novel datasets. This critical assumption has led to underperformance of FSL techniques when the gap between base and novel datasets are large [11, 2]. To remedy this, we propose to use easy-to-acquire coarsely-labeled data from the novel classes in FSL.

FSL with Additional Data. Additional data have proven to be useful in FSL. The two most common setup that utilizes additional data are: semi-supervised FSL [29, 18, 48, 30, 42] and tranductive FSL [23, 5]. Different from our setup, the additional data in these setups are unlabeled and only available during evaluation. [20] operates in a setup similar to ours but they focus on developing specialized inference procedures using the class hierarchy. In particular, they build upon Prototypical Network [33] and seek to build learned prototypes for coarse classes during representation learning that can be used to refine the prototypes for fine-grained classes during evaluation. In contrast, our approach focuses on building representations that are agnostic to any inference methods. The two approaches focus on different aspects of the setup and can be combined.

FSL with Class Taxonomy. Leveraging class taxonomy or hierarchy is common in supervised machine learning [1]. In fact, a survey on hierarchical classification by Silla and Freitas [32] have shown that in a wide range of application domains, incorporating class hierarchy when building classifiers can yield performance gains. In FSL, class taxonomy has been used to build better techniques. Efforts include building specific ConvNet architectures using semantic relationships between base and novel classes [17, 25] or specialized inference procedures that leverage the class taxonomy [21, 22]. All these methods can be directly used in our setup but they do not consider the use of coarsely-labeled examples which could bring forth more improvements.

Self-training. Our approach for using coarsely-labeled data is closely related to self-training. Self-training is often used in semi-supervised learning. The idea is to use a teacher model trained on the labeled data to label the unlabeled data and train another student model on both the original labeled data and the pseudo-labeled data. This simple technique has been shown to improve ImageNet classification performance [45, 46]. Another venue where self-training is used is knowledge distillation where the goal is to compress a large teacher model by training a student to reproduce the teacher's predictions. Self-training has also been used in semi-supervised FSL [18, 42]. However, most of these approaches deploy self-training in a "closed set" setup, i.e., the set of classes is fixed, and the unlabeled data comes from this same set of classes. Thus, there is a significant chance the pseudo-labels are actually correct (though [45] do note that noisy pseudo-labels help).

Our approach moves away from the closed set setup and labels the coarsely-labeled *novel class* data with fine-grained *base class* pseudo-labels *that are guaranteed to be incorrect*: an uncommon scenario. The only related work here is [26] in which the authors adapt feature representations to far off domains by training a network to replicate pseudo-labels produced by an unrelated classifier from a distant source domain. The authors observe that this is effective if the *groupings induced by the pseudo-labels match class distinctions in the novel domain*. However, the authors only use unlabeled data during representation learning resulting in potentially poor pseudo-labels, and as such resort to additional tricks such as self-supervised learning techniques. In contrast, our coarsely-labeled data leads to much better pseudo-labels, removing the need for such tricks.

3. Problem Setup

Our setup is illustrated in figure 2. We assume that we have a taxonomy of classes with two levels, a set of *fine-grained* classes that are more challenging to annotate (C) and a set of *coarse-grained classes* that are easier to annotate (P). The classes we are interested in recognizing are the former. Every fine-grained class c is associated to a single coarse-grained class $p(c)$, i.e., the taxonomy is a tree. The fine-grained classes are split into base classes C_{base} and novel classes C_{novel}. Similar to the traditional few-shot classification setup, the goal is to build learners that can quickly learn to recognize novel classes C_{novel} each of which has very few training images.

Before encountering the novel classes, the learner fits its parameters in a *representation learning* phase. In this phase, similar to FSL, we assume that the learner has access to a large annotated base dataset D_{rep}^{fine}:

$$D_{rep}^{fine} = \{(x_i, y_i, p_i)\}_{i=1}^{n} \tag{1}$$

where x_i is the image, y_i is the base class label and p_i is the coarse label associated to base class y_i. Different from conventional FSL, we assume that the learner has access to an additional set of coarsely-labeled examples D_{rep}^{coarse}:

$$D_{rep}^{coarse} = \{(x_j, p_j)\}_{j=1}^{n'} \tag{2}$$

that contains images x_j from some of the novel classes but weakly annotated with coarse label p_j.

We define the representation set as $D_{rep} = D_{rep}^{fine} \cup D_{rep}^{coarse}$. We assume that we know the parents for the *base* classes, so that D_{rep}^{fine} can also be decorated with coarse labels. We also assume that only a subset of the novel classes, C_{novel}^{seen} are "seen" by the learner through D_{rep}^{coarse} (with only coarse labels). The remainder of the novel classes are "unseen" ($C_{novel}^{unseen} = C_{novel} - C_{novel}^{seen}$) .

After the representation learning phase, the learner goes into the *evaluation* phase where it gets a small reference set $D_{ref} = \{(x_j, y_j)\}_{j=1}^{n_{ref}}$ of novel class examples x_j and their corresponding label y_j. In our experiments, D_{ref} is disjoint from D_{rep}^{coarse} though this is not necessary. Using D_{ref}, the learner must train a classifier for the novel classes, which will be evaluated on a *completely unseen, unlabeled* query set of novel class examples D_{query}: *We stress that the coarse label of reference and query examples are not revealed during evaluation.*

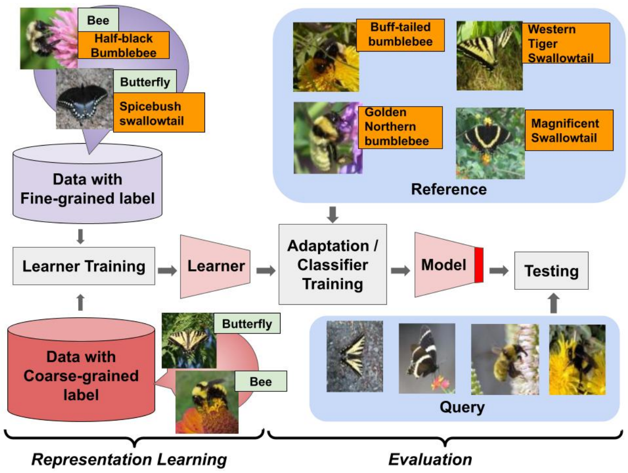

Figure 2. Problem setup. There are two levels of labels in our setup - the coarse label (light green) and the fine-grained label (orange). During representation learning, the learner learns from data with fine-grained and coarse labels (purple) and data with coarse-grained labels that can be labeled as one of the novel classes (red). Upon receiving the reference images where only the fine-grained label is available, the learner has to produce a model that can recognize the query images at the fine-grained level.

For most of our experiments, we assume that each novel class has a base class as its sibling in the taxonomy. We explore the scenario where some novel classes are not related to any base classes in section 6.2.2.

4. Methodology

The goal is to build learners that can output a classification model f_θ parametrized by θ upon receiving a small D_{ref}. We assume f_θ consists of two components: a feature extractor $\phi_\theta(\cdot)$ that maps an input image x into $\mathbb{R}^d$ and a classification model $h_\theta(\cdot)$ that maps $\phi_\theta(x)$ to the predicted probabilities $\mathbb{P}_\theta(y|x)$. In general, the feature extractor ϕ_θ would be learned during representation learning and kept fixed during few-shot evaluation to avoid overfitting.

4.1. Parent-Aware Self-training, PAS

We learn our feature representation ϕ_{θ^*} as follows:

1. Learn a teacher model f_{θ_0} on D_{rep}^{fine} via minimizing cross entropy loss (with respect to the base classes).
2. Use the teacher model f_{θ_0} to "pseudo-label" the coarsely-labeled dataset D_{rep}^{coarse}. Crucially, we use the

coarse labels to *filter* the pseudo-labels:

$$D_{rep}^{pseudo} = \{(x_j, p_j, \bar{y}_j)\}_{j=1}^{n'} \tag{3}$$

$$\bar{y}_j = g(f_{\theta_0}(x_j), p_j) \quad \forall (x_j, p_j) \in D_{rep}^{coarse} \tag{4}$$

where g filters the pseudo-labels $f_{\theta_0}(x_j)$ based on the coarse-label p_j (section 4.1.1).

3. Learn a new student model f_{θ^*} on D_{rep}^{base} and D_{rep}^{pseudo}:

$$\begin{aligned}\min_{\theta} \; & \frac{1}{n} \sum_{(x_i, y_i, p_i) \in \mathcal{D}_{rep}^{fine}} l_{CE}(f_\theta(x_i), y_i) \\ + & \frac{1}{n'} \sum_{(x_j, \bar{y}_j, p_j) \in \mathcal{D}_{rep}^{pseudo}} l_{KL}(f_\theta(x_j), \bar{y}_j)\end{aligned} \tag{5}$$

where l_{CE} is the cross entropy loss, l_{KL} is the Kullback-Leibler divergence.

After representation learning, the student's feature extractor ϕ_{θ^*} can be used to extract features for training the downstream classifier on the reference set.

4.1.1 Filtering Function g

Consider a data point x with coarse label p. The pseudo-labels produced by the teacher $\mathbb{P}_{\theta_0}(y = k|x) = f_{\theta_0}(x)$ need not be consistent with the coarse labels. This is especially true for the coarsely-labeled novel class examples, since these are sampled from a different data distribution as compared to the base classes. We therefore *filter* the pseudo-labels to encourage consistency between the pseudo-labels $f_{\theta_0}(x)$ and the coarse labels p. To do so, we first *zero out* the predicted probabilities for fine-grained labels that are inconsistent with p to produce an unnormalized probability vector $\bar{s}$:

$$\bar{s}[k] = \begin{cases} 0 & \text{if } p(k) \neq p \\ \mathbb{P}_{\theta_0}(y = k|x) & \text{otherwise} \end{cases} \tag{6}$$

We then renormalize $\bar{s}_j$ to construct the filtered soft pseudo-label:

$$g(f_{\theta_0}(x), p) = \frac{\bar{s}}{\sum_k \bar{s}[k]} \tag{7}$$

Intuitively, the filtering function ensures that an example with coarse label p would only have non-zero probability mass for base classes associated to coarse label p.

4.2. Inference Strategy

During evaluation, a variety of inference methods [33, 10] can be used along with the student's representation during inference. For simplicity, we decided to use classifiers based on the nearest class prototype [33]. For each class k we compute the class prototype:

Setup	Base	Novel- Seen	Novel- Unseen	Super- category
iNat2019-CL	398	126	119	50
tieredImageNet-CL	498	60	50	34
CIFAR-100-CL	40	40	20	20

Table 1. Class distribution of the benchmarks introduced in this paper.

$$\bar{c}_k = \frac{1}{\sum_j \mathbb{I}[y_j = k]} \sum_{x_j \in D_{ref}: y_j = k} \frac{\phi(x_j)}{||\phi(x_j)||_2} \tag{8}$$

The class probability of a query examples x_i is computed via measuring the cosine similarity between $\phi(x_i)$ and $\bar{c}_k$:

$$\mathbb{P}(y = k|x_i) \propto exp\left\{ \frac{\bar{c}_k^T \phi(x_i)}{||c_k||_2 \cdot ||\phi(x_i)||_2} \right\} \tag{9}$$

To accommodate the use of cosine similarity, we use a cosine classifier[10] as our default classification model $h(\cdot)$ when training the teacher and the student.

5. Experimental Setup

5.1. Benchmark and Datasets

Since our problem setup is new and requires additional coarsely labeled examples during representation learning, we set up new benchmarks from three existing datasets: iNaturalist [39], TieredImageNet [29] and CIFAR100 [15]. In these new benchmarks, we ensure (via re-splitting the classes between base and novel) that every novel class has a sibling base class. We also make sure that there are at least two novel classes associated to a single coarse label to ensure that the coarse label does not automatically give away the fine label. We present the class distribution of these datasets in table 1 and some relevant information below:

1. **iNat2019-CL.** We construct this benchmark from the iNaturalist 2019 (iNat2019) competition dataset [39] - a fine-grained animal species classification dataset with a natural taxonomy (We use the genera level labels as supercategory). After removing species and genera with insufficient examples, we split each genus into base, novel-seen and novel-unseen.
2. **TieredImageNet-CL.** TieredImageNet [29] comes with 34 high level supercategories but different supercategories are split into base and novel in the original benchmark. To reflect the assumption that novel and base classes share coarse labels, we resplit each supercategory into base, novel-seen and novel-unseen.

3. **CIFAR-100-CL.** CIFAR-100 [15] contains 100 classes of images that can be grouped evenly into 20 supercategories. We split each supercategory into 2/2/1 for base, novel-seen and novel-unseen.

For each dataset, we split examples of each class into three buckets: 60%/20%/20%. For base classes and novel-seen classes, the 60% split is used to construct the representation set. For novel-seen and novel-unseen classes, the two 20% splits are used to form D_{ref} and D_{query} respectively.

5.2. Evaluation Protocol

We report the top-1 per class accuracy averaged across each class for all datasets (to avoid issues arising from class imbalance in iNat2019). We consider two evaluation protocols - all-way-k-shot and 5-way-k-shot classification for $k = 1, 5$. When sampling 5-way classes for evaluation, we restrict the maximum number of supercategories in each single classification task to 3 to ensure that there are at least two classes that share the same supercategory in a single task and simply identifying the supercategory alone does not yield good performance. Regardless of 5-way or all-way, we construct a classification task by sampling k different reference examples from each novel classes and then evaluate the performance of a model on the whole query set. The process is repeated 1000 times for all-way classification and 10,000 times for 5-way classification (following [47]) to generate statistically meaningful comparisons. In addition, we also consider the all-shot setup where we use all the examples in the 20% split for all-way classification.

5.3. Comparisons

To assess PAS's representations, we establish a few representations for comparisons. These representations are trained similarly to PAS (same architecture with cosine classifiers [10]) but with different loss functions:

1. **Baseline.** Here the representation is simply obtained via training the model to classify the fine-grained examples on D_{rep}^{fine}.
2. **Repr-Coarse.** Similar to Repr-Fine. This representation is the feature extractor of a ConvNet trained to classify the examples from both D_{rep}^{fine} and D_{rep}^{coarse} into their respective supercategories.
3. **Self-training.** This representation is trained similarly to PAS except that the filtering function g is removed when generating the pseudo-labels.
4. **Repr-Multi.** This multi-tasked representation is produced by training a ConvNet with two cosine classifier heads - one for classifying the fine-grained label and another for the supercategory:

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{n} \sum_{(x_i, y_i) \in \mathcal{D}_{rep}^{base}} l_{CE}(f_\theta(x_i), y_i) \\ + & \frac{C}{n + n'} \sum_{(x_i, p_i) \in \mathcal{D}_{rep}} l_{CE}(f_\theta(x_i), p_i) \end{aligned} \tag{10}$$

where C is set to 1 for simplicity

We also get an **upper bound** for representation learning techniques by training a classifier on a fully labeled dataset consisting of all training examples from both base and novel-seen classes (obtained by adding fine-grained novel class labels to D_{rep}^{coarse} and combining it with D_{rep}^{fine}); the classifier head is discarded and the feature extractor is used for few-shot transfer as with the other methods. This feature representation is the best representation one can get from the task; hence it is an upper bound for representation learning-based FSL technniques.

When evaluating the representations, we deploy the same inference procedure (sec 4.2) for fair comparison. In addition, we compare PAS' representation (with nearest class prototype inference) to two recent few-shot learners that deploy more sophisticated inference strategies: MetaOptNet [16] and FEAT [47] for 5-way classification. These learners are trained assuming no knowledge of the coarse labels (i.e. trained on D_{rep}^{fine} without the coarse labels) since they were initially developed for the conventional few-shot learning setup. We use ResNet18 [13] as the backbone for all methods and defer training details to the supplementary materials.

6. Experimental Results

6.1. Coarsely-labeled Data Improve FSL

We present the all-way classification result on all novel classes C_{novel} in table 2 and 5-way classification result in table 3. We observe the following:

1. **Data from novel classes improves representation even without labels**: Methods that use the additional data (Repr-Multi, Self-training, PAS) outperform the Baseline that is trained only on the base classes. This is true even for Self-training, which uses *only the novel class data and not the coarse labels*. The performance of Self-training confirms the findings in [26].
2. **Learners trained with coarsely-labeled data outperform those trained without**: The addition of coarse label information significantly helps: Among representation learning approaches, Repr-Multi and PAS both outperform the Baseline and Self-training, which do not use the coarse label information on all-way classification; for 5-way classification, Repr-Multi slightly underperforms Self-training but PAS

	iNat2019-CL								
	Novel			Novel-seen			Novel-unseen		
Method	k=1	5	all	k=1	5	all	k=1	5	all
Baseline	20.46	39.22	57.22	28.68	50.68	67.25	28.14	50.37	67.49
Repr-Coarse	19.89	29.32	41.72	33.50	44.62	57.62	28.09	40.32	51.39
Self-training	22.94	42.17	59.69	33.18	54.79	69.85	29.95	**52.11**	**69.87**
Repr-Multi	24.72	41.42	57.34	38.24	56.77	70.72	**32.03**	51.21	65.88
PAS	**25.21**	**43.27**	**61.04**	**39.06**	**58.76**	**73.63**	30.91	51.85	69.12
Upper Bound	27.30	47.98	64.20	41.64	64.61	75.36	30.71	53.77	72.29

	tieredImageNet-CL								
	Novel			Novel-seen			Novel-unseen		
Method	k=1	5	all	k=1	5	all	k=1	5	all
Baseline	32.16	53.36	68.97	41.22	62.92	77.19	54.19	75.50	85.51
Repr-Coarse	25.69	37.19	49.76	38.14	48.83	62.55	41.64	55.70	66.32
Self-training	35.49	57.26	70.87	48.12	69.11	**80.60**	**54.71**	**75.89**	**86.08**
Repr-Multi	37.16	57.27	70.20	49.54	68.38	80.28	53.28	72.94	83.31
PAS	**38.11**	**59.08**	**71.84**	**50.60**	**69.52**	80.40	53.18	74.68	85.12
Upper Bound	42.86	65.68	76.71	60.03	80.67	87.14	55.94	76.96	86.55

	CIFAR-100-CL								
	Novel			Novel-seen			Novel-unseen		
Method	k=1	5	all	k=1	5	all	k=1	5	all
Baseline	20.32	33.24	42.67	25.50	39.95	50.45	34.37	51.80	64.00
Repr-Coarse	31.56	38.90	47.87	45.74	53.36	63.10	37.52	50.65	55.20
Self-training	25.68	42.43	54.93	32.96	51.42	63.30	38.24	**57.51**	**69.50**
Repr-Multi	**34.99**	46.30	55.07	**49.18**	60.51	69.20	**39.00**	53.69	61.20
PAS	**35.00**	**48.42**	**58.37**	48.57	**61.95**	**72.65**	37.92	54.91	65.10
Upper Bound	51.83	64.97	69.17	73.75	85.02	85.45	36.53	56.25	70.30

Table 2. Average top-1 per class accuracy of various representations across 1000 runs. For each novel categories, we use k=1, 5 and all reference examples. Best performing entries that leverage coarsely-labeled data are bolded. 95% confidence intervals are omitted for brevity. The full table can be found in the supplementary materials.

performs comparatively to Self-training. Further, PAS with simple nearest prototypes inference can outperform MetaOptNet and FEAT (except on iNat2019-CL) on 5-way classification. These observations validate our hypothesis, that easy-to-acquire coarse labels can significantly improve FSL.

3. **PAS is the strongest representation overall**: On all-way-5-shot classification, PAS significantly outperforms Repr-Multi by **1.92 points** on average, and yields a **8.31 points** gain over the Baseline; On 5-way-5-shot classification, PAS outperforms Repr-Multi by **1.17 points** and the Baseline by **3.24 points**. All these results show that even though multitask-training can be used to leverage coarsely annotations, it is not as effective as PAS . In conclusion, with coarse annotations, PAS is an extremely effective way of improving FSL.

To unpack the performance gains, we also evaluate the different representations separately on the novel-seen and novel-unseen classes (Table 2). To ensure that the classification tasks are truly fine-grained, we remove supercategories that only have one child when splitting the novel classes for tieredImageNet-CL (we report the performance on novel-unseen for CIFAR-100-CL for completeness even though there is only one novel-unseen class per supercategory). We observe, as expected, that the performance

	iNat2019-CL		tieredImageNet-CL		CIFAR-100-CL	
Method	k=1	5	k=1	5	k=1	5
MetaOpt	59.32 ± 0.22	72.92 ± 0.20	59.12 ± 0.20	73.96 ± 0.16	51.57 ± 0.21	63.90 ± 0.18
FEAT	**62.76 ± 0.22**	**76.45 ± 0.20**	67.60 ± 0.21	82.05 ± 0.15	55.65 ± 0.21	71.05 ± 0.17
Baseline	57.07 ± 0.20	73.68 ± 0.19	65.17 ± 0.20	81.09 ± 0.15	51.28 ± 0.20	67.01 ± 0.17
Repr-Coarse	54.43 ± 0.19	65.29 ± 0.18	56.92 ± 0.19	68.28 ± 0.17	57.76 ± 0.18	67.18 ± 0.15
Self-training	60.19 ± 0.22	75.82 ± 0.20	**68.35 ± 0.21**	**83.42 ± 0.14**	57.92 ± 0.21	**73.76 ± 0.16**
Repr-Multi	59.06 ± 0.20	73.74 ± 0.19	66.51 ± 0.20	81.76 ± 0.15	**60.81 ± 0.19**	72.48 ± 0.15
PAS	59.74 ± 0.21	74.88 ± 0.20	**68.02 ± 0.20**	**83.26 ± 0.15**	**60.82 ± 0.19**	73.37 ± 0.15
Upper Bound	62.64 ± 0.22	78.52 ± 0.19	73.03 ± 0.20	87.34 ± 0.12	72.78 ± 0.22	84.34 ± 0.13

Table 3. Average 5-way-k-shot top-1 accuracy and 95% confidence intervals of various few-shot learners and our representations across 10000 runs. Top performing entries (excluding Upper Bound) are bolded.

Dataset	Before Filtering	After Filtering
iNat2019-CL	0.4258	0.7260
tieredImageNet- CL	0.4620	0.7352
CIFAR-100-CL	0.3695	0.8293

Table 4. Adjusted Mutual Information (AMI) of the predicted class identities of examples in D_{rep}^{coarse} and their ground truth identities. AMI has a theoretical range of [0, 1] with higher value signifying stronger alignment between the prediction and ground truth.

gains are largest on the novel-seen classes. However, even on the novel-unseen classes we do observe performance gains from Self-training, Repr-Multi and PAS for iNat and CIFAR-100 (though on TieredImageNet-CL, the gains disappear). This suggests that using coarsely labeled data can potentially help *even for completely unseen novel classes*.

6.1.1 The Effect of Filtering

From table 2, we observe that PAS significantly outperforms Self-training. As reported in [26], the key to good transferrability of self-trained student representation relies on the alignment between the grouping induced by the teacher and the ground truth of the additional data. We posit that filtering has strengthened the alignment and thus yields a superior result. To validate this, we investigate the generated pseudo-labels on D_{rep}^{coarse} by the teacher before and after filtering. Specifically, we use the most probable prediction of the pseudo-label to "label" each example in D_{rep}^{coarse}. Then, as in [26], we evaluate the induced grouping by measuring the adjusted mutual information (AMI) [40] between the induced clustering and the ground truth. Table 4 shows that the AMI increases significantly with filtering, indicating a stronger alignment between the grouping induced by the filtered pseudo-labels and the ground truth class distinctions as compared to the original pseudo-labels. We believe that this alignment results in a cleaner signal for training the student's representation.

6.2. Analyses

In this section, we analyze PAS on iNat2019-CL. Unless explicitly stated, we report performance on all novel classes.

6.2.1 Reducing the number of Base Classes.

Few-shot learners rely on a large diverse base dataset. The usage of coarsely-labeled data sets up the possibility of reducing this dependence. In this subsection, we investigate the effect of reducing the number of base classes. To start, we remove the fine-grained labels of 2/3 of the base classes in iNat2019-CL while keeping their coarse labels. This reduces the number of base classes from 398 to only 144. As a result, the number of novel classes during evaluation (126 + 119 = 245) becomes significantly more than the effective number of classes available during representation learning (144 fine-grained, 50 coarse-grained).

We report the results on this benchmark in table 5. We find that when the number of base classes is substantially reduced, the accuracy of the baseline drops by 6 to 10 points. In contrast, we observe that with the aid of coarsely-labeled data, PAS experiences a much smaller performance degradation. PAS is thus less reliant on the availability of large amounts of fine-grained labels on the base dataset.

6.2.2 Effect of Unseen Supercategories

So far we have assumed that each novel class shares a coarse label with a base class. However, it is crucial that few-shot learners generalize to completely unseen parts of the class taxonomy. To test how PAS works in this setting, we constructed another modification of iNat2019-CL: we randomly chose a fifth of the coarse categories and removed all labels associated with these supercategories in the representation set. In particular, we removed all base classes that

Large Reduction in Base Classes		
Method	k=1	5
Baseline - Original	20.46	39.22
PAS- Original	25.31	43.27
Baseline	14.19 (↓ 6.26)	28.37 (↓ 10.84)
PAS	21.80 (↓ 3.41)	37.31 (↓ 5.96)

Table 5. Average k-shot performance of different representations evaluated on the original iNat2019-CL novel classes. PAS- Original and Baseline - Original are trained on the original base dataset. (↓) indicates absolute amount of degradation due to reduced base classes. 95% confidence intervals can be found in the supplementary materials. See section 6.2.1 for more details.

Removing Some Coarse Labels		
Method	k=1	5
Baseline	18.15	35.72
Repr-Multi	21.92 (+ 3.77)	37.23 (+ 1.50)
PAS	23.14 (+ 4.99)	40.73 (+ 5.00)

Table 6. Average k-shot performance (on iNat2019-CL novel classes) of various representations trained on a base dataset with unseen supercategories. (+) indicates absolute improvement from the Baseline. 95% confidence intervals can be found in the supplementary materials. See section 6.2.2 for details.

belong to this set of supercategories, and we removed the coarse labels corresponding to these supercategories from the coarsely-labeled dataset. Note that in this setup, the connection between base and novel classes is weakened which could impact the performance of PAS.

We adapt PAS to this setting by simply using this newly-unlabeled data with *unfiltered* pseudo-labels when training the student representation. In table 6, we observe that with this modification, PAS is still able to leverage all the available data to yield considerable performance gains. PAS outperforms both the baseline as well as Repr-Multi, indicating the efficacy of PAS in leveraging additional data that are less related to the base dataset.

6.2.3 Effect of Coarser Labels

Even though the distinction at the genus rank for iNat2019 is rather clear-cut (as shown in figure 1), one can also easily obtain coarser labels that corresponds to higher taxonomic rank. For instance, one can recognize bees and butterflies as insects which corresponds to the kingdom of the species; one might also label bees and wasps as insects that have transparent wings (Hymenoptera) which corresponds to the order of the species.

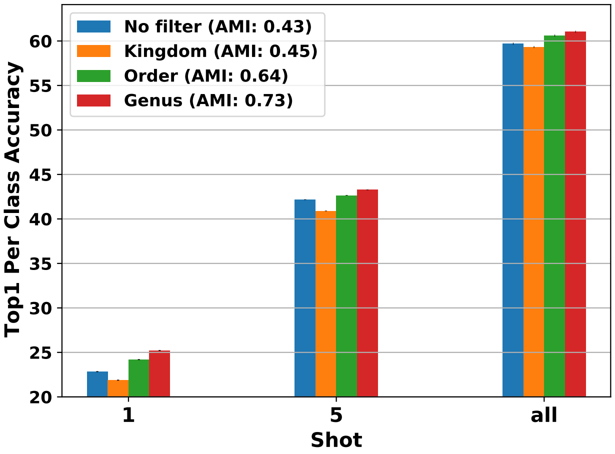

Figure 3. Performance of PAS with various class taxonomies. We observe that PAS with coarser labels yields less performance gains

In this section, we investigate how using these coarser labels would affect PAS. We conjecture that the coarser labels would dilute the effect of filtering and thus leading to degradation in performance. To investigate, we look into three taxonomic ranks available in iNat2019 (in decreasing order of coarseness): Kingdom (5 supercategories), Order (27 supercategories) and Genus (50 supercategories). Indeed, we find that the coarser labels are less effective, though we still see considerable gains from the "Order" level coarse labels (Figure 3). These gains roughly correlate with the AMI of the predicted class labels induced by the filtered pseudo-labels (with different coarse labels) and the ground truth.

Additional Analyses: We show the following additional results in the supplementary: (a) PAS can bring more improvements when coupled with a semi-supervised inference approach when the coarse labels of the reference set is available, and (b) the strength of PAS' representation is correlated with the amounts of coarsely-labeled data. For more details, please see the supplementary materials.

7. Conclusion

We investigate the use of coarsely-labeled data in building more transferrable representations for few-shot learning. We found that representations that are built using the additional coarsely-labeled examples are significantly better than their counterparts when evaluated under 1-shot and 5-shot classification in three different datasets. We develop a new representation learning technique - PAS that leverages self-training and parent consistent filtering to achieve stronger representations, bringing forth enormous improvement to few-shot learning.

Acknowledgements: This work was funded by the DARPA Learning with Less Labels program (HR001118S0044).

References

[1] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12506–12515, 2020. 2

[2] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2

[3] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[4] 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

[5] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 2

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135, 2017. 1, 2

[7] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527, 2018. 2

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1

[9] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8059–8068, 2019. 1

[10] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 1, 2, 4, 5

[11] Yunhui Guo, Noel CF Codella, Leonid Karlinsky, John R Smith, Tajana Rosing, and Rogerio Feris. A new benchmark for evaluation of cross-domain few-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[12] 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. 1, 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[14] Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *Advances in Neural Information Processing Systems*, pages 4003–4014, 2019. 2

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4, 5

[16] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019. 1, 2, 5

[17] 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 Conference on Computer Vision and Pattern Recognition*, pages 7212–7220, 2019. 2

[18] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems*, pages 10276–10286, 2019. 2

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[20] Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Prototype propagation networks (ppn) for weakly-supervised few-shot learning on category graph. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019. 2

[21] Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Learning to propagate for graph meta-learning. In *Advances in Neural Information Processing Systems*, pages 1037–1048, 2019. 2

[22] Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Many-class few-shot learning on multi-granularity class hierarchy. *IEEE Transactions on Knowledge and Data Engineering*, 2020. 2

[23] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 1, 2

[24] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. 2

[25] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 441–449, 2019. 2

[26] Cheng Perng Phoo and Bharath Hariharan. Self-training for few-shot transfer across extreme task differences. *arXiv preprint arXiv:2010.07734*, 2020. 1, 2, 3, 5, 7

[27] 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. 2

[28] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 2

[29] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 2, 4

[30] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding propagation: Smoother manifold for few-shot classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[31] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2

[32] Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011. 2

[33] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017. 1, 2, 4

[34] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020. 2

[35] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. When does self-supervision improve few-shot learning? In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

[36] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2019. 2

[37] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2

[38] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[39] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 4

[40] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010. 7

[41] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 2

[42] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12836–12845, 2020. 2

[43] 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. 2

[44] Davis Wertheimer and Bharath Hariharan. Few-shot learning with localization in realistic settings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6558–6567, 2019. 1

[45] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. 2

[46] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019. 2

[47] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020. 1, 5

[48] Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12856–12864, 2020. 2