Label Hallucination for Few-Shot Classification

Technical Appendix

Yiren Jian, Lorenzo Torresani

Dartmouth College
yiren.jian.gr@dartmouth.edu, LT@dartmouth.edu

The supplementary material accompanying our main paper submission consists of:

- Source code implementing our approach, saved in file `code_label_hallucination.zip`. Please review `README.md` enclosed in the zip file for instructions on how to run the code.
- Several visualizations of label hallucination examples.
- Discussion and evaluation of strategies to reduce the computational cost of the training procedure.
- Analysis of performance as a function of the base dataset size.
- Stochastic finetuning using random mini-batches of the support set.
- Results for large number of novel classes (10-way and 20-way experiments) in each episode.
- Results when the base classes and the novel categories are far apart.
- Experiments with imbalanced base classes.
- Simultaneous recognition of base and novel classes.
- Ablation on knowledge distillation.
- Experiment with feature distillation.
- Description of datasets used in the experiments.
- Details of the network architectures.
- Explanation of hyperparameter and design choices.
- Optimization details.

Visualizations of Label Hallucination

As already discussed in the main paper, the intuition behind our approach is that even though the novel classes are not properly represented in the base dataset (due to the disjoint label spaces), many base images may include objects that resemble those of the novel classes or they may contain contextual elements (e.g., backgrounds or other objects) that tend to co-occur with the novel-class objects. In such cases, we would expect the soft pseudo-labels assigned by our method to reflect the presence of similar or contextual objects in the form of large classification scores for the novel class. In order to validate this hypothesis, in Figure 1, Figure 2, Figure 3 and Figure 4 of this document we show the base images receiving the largest pseudo-label scores for 4 distinct few-shot classification episodes. In each Figure, the 5 images with red frame in the first row are 1-shot training examples of the 5 novel classes within the episode. The names of the novel classes are listed in red. Underneath each novel example, we present a column of base images, showing the examples in the base dataset that received the largest pseudo-label score for that novel class. In order to show how the similarity to the novel class diminishes as the pseudo-label score gets smaller, we present base images corresponding to ranking positions 1–3, 2001-2003, and 5001-5003 in the list of base images sorted with respect to the classification score for the novel class. Above each base image we report the base class name (note that this information is used only during pretraining and not during the episode learning stage) and the novel-class probability assigned by the linear classifier in the pseudo-labeling stage. In Figure 1, we see that our method assigns a "malamute" probability of 0.73 to a base image that belongs to class "Arctic fox" due to the clear appearance similarities between these two classes. In Figure 2, "harvestman" base images are assigned large probability for the novel class "ant" because these two classes share contextual backgrounds, e.g., zoomed-in views of leaves or gravel. In Figure 2, base images of "green mamba" are assigned pseudo-label "nematode" because these two animals have similar body shapes, despite being very different in color. Similarly, in Figure 4 we see that the pseudo-label "bookshop" is assigned to base images that contain objects having similar spatial layouts, e.g., tobacco shops, or upright pianos. Our experiments demonstrate that retraining the whole network on the pseudo-labeled version of the base dataset yields a model that is tuned to recognize all of these contextual cues and appearance patterns that correlate with the novel class, and thus it produces improved accuracy on the few-shot categorization problem.

We would like to further clarify that contextual elements captured by our method are not limited to similar backgrounds. High-rank base images may represent examples containing foreground objects that have similar appearance to the few-shot images (e.g., the malamute image in Figure 1), or base examples including objects with shape akin

Figure 1: Examples of hallucinated labels. The first row shows 1-shot training images for 5 novel classes. Below each novel-class example we show the base images that have largest pseudo-label score for that novel class.

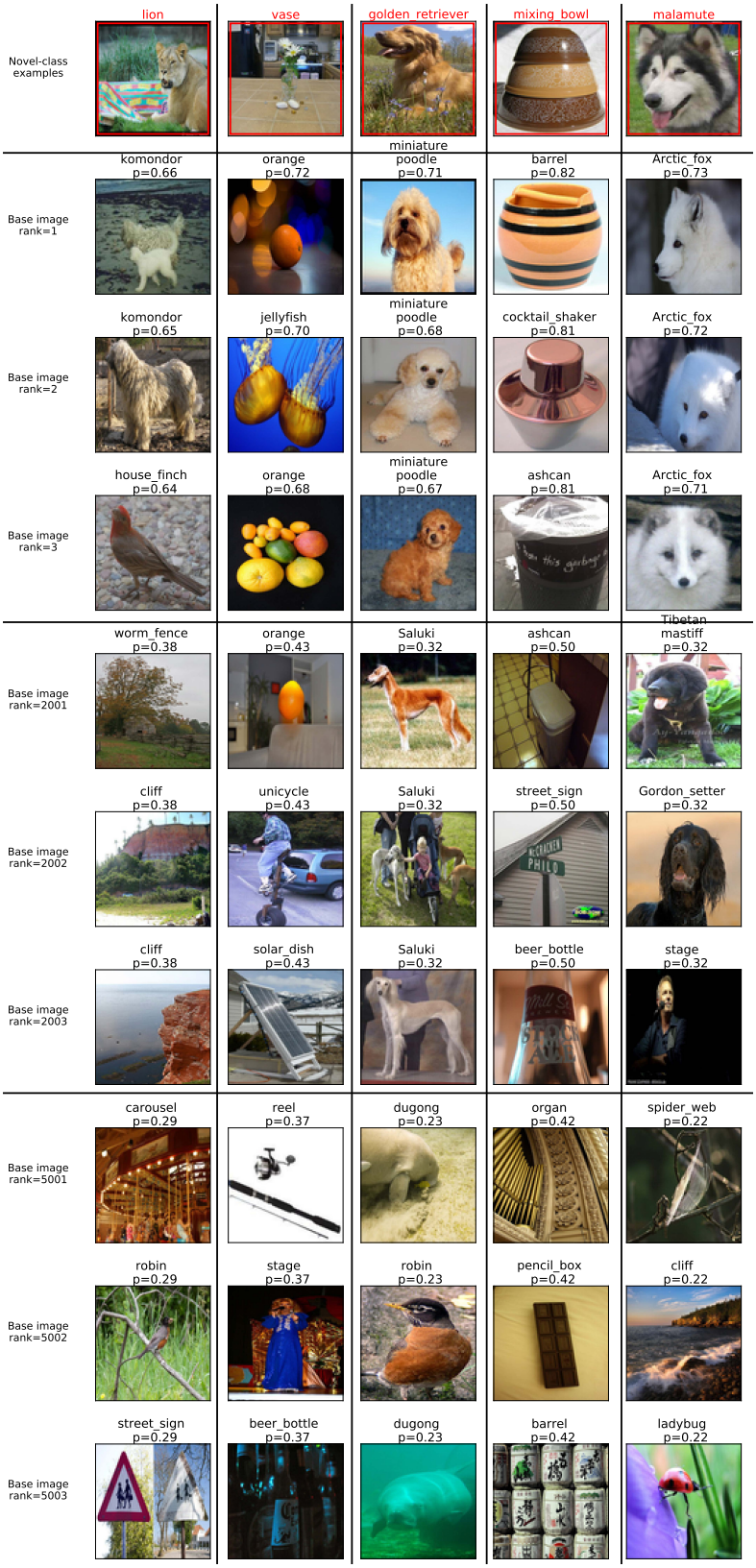


Figure 2: Examples of hallucinated labels. The first row shows 1-shot training images for 5 novel classes. Below each novel-class example we show the base images that have largest pseudo-label score for that novel class.

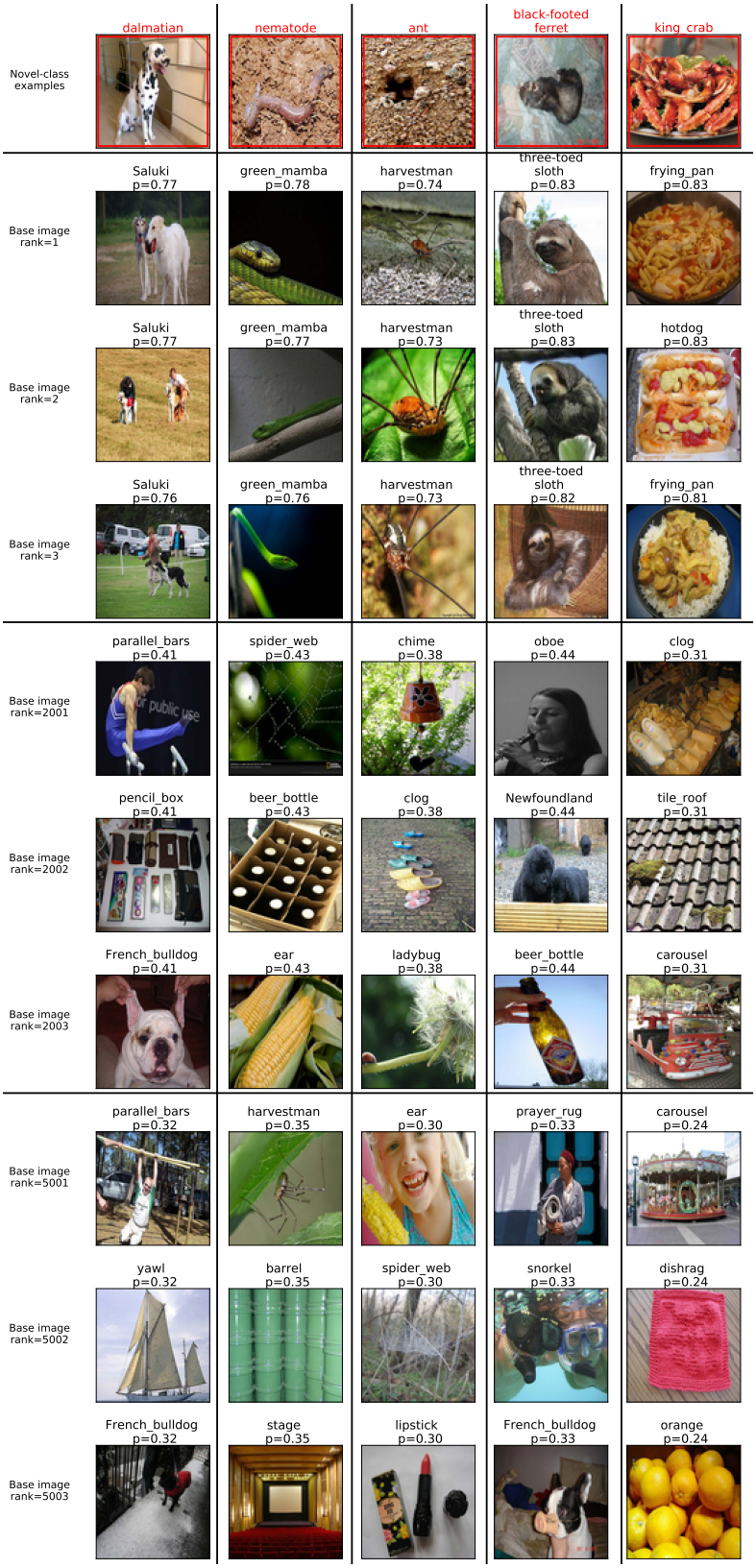


Figure 3: Examples of hallucinated labels. The first row shows 1-shot training images for 5 novel classes. Below each novel-class example we show the base images that have largest pseudo-label score for that novel class.

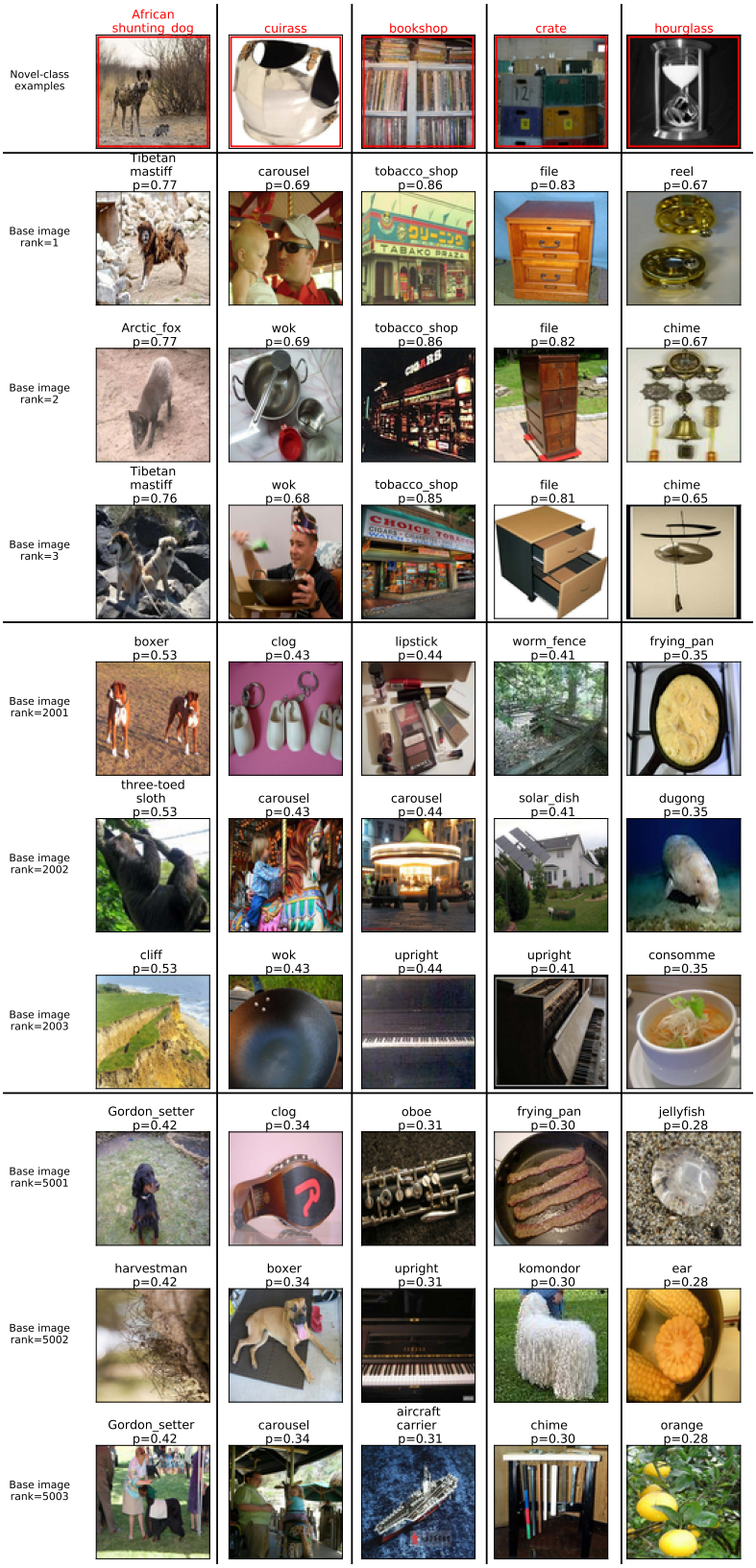


Figure 4: Examples of hallucinated labels. The first row shows 1-shot training images for 5 novel classes. Below each novel-class example we show the base images that have largest pseudo-label score for that novel class.

to that of the novel class (e.g., the green mamba in Figure 2, which resembles the shape of a nematode), or even examples matching in terms of spatial layout (e.g., the images of tobacco shops and upright pianos have similar spatial layout as the bookshop class in Figure 3). The pseudo-labeling is done by the linear classifier that is trained to distinguish the 5 one-shot novel images. Thus, the features used for ranking are those that make the 5 novel images as separate as possible. This may be shape, color, background, or texture depending on the specific novel images in the episode. For example, in Figure 1, it can bee seen from the top-rank base images of komondor that the classifier is giving high-score to images of animals that look like lions (the foreground in the novel class) even though they have background different from that of the lion image. This happens because the background in this case is not useful to discriminate the lion image from the other 4 novel images in this few-shot episode (e.g., the golden retriever has background similar to that of the lion). Similarly, we can see from the top-rank images retrieved for the golden retriever example in Figure 1, that the classifier is giving high-score to base images of dogs that have golden and curly hair, even though the backgrounds of these images are completely different from that of the golden retriever. Again this happens because golden and curly hair is a discriminative feature to distinguish the golden retriever from the other novel images. Conversely, the background is a poor feature for this episode (we note that both the malamute and the lion contain grass in the background, exactly like the golden retriever), and thus it is not used by the linear classifier that performs pseudo-labeling on the base dataset.

Speeding up the training

We discussed the computational cost in section Ablations of the main paper. Other finetuning methods [1,11,30] also have high latency, and in some cases even higher than ours. For example, while our method takes 0.35 seconds per gradient step, AssoAlign [1] takes 0.87 second. This is due to the more efficient optimization of our simple distillation loss compared to the more costly feature alignment in AssoAlign. It should also be noted that AssoAlign requires additional time to iterate through the base dataset to select related examples, and that this cost is not included in the reported runtime of 87 seconds (0.87 x 100 steps).

Furthermore, we found that we can lower the total training cost of our approach by 66% with negligible impact on accuracy by simply reducing the number of training iterations. This is demonstrated in Table 1 where we provide 5-shot test accuracies obtained by our model on the different benchmarks for a varying number of training steps (results are from a single run of 600 episodes w/ SKD-GEN0 pretraining). While in the main paper we report results when finetuning for 300 iterations, it can be seen from Table 1 that after 100 steps our model achieves already near-optimal performance. In the main paper we chose to finetune for 300 steps because for these datasets each epoch is about 300 steps. We did not tune the number of steps, and simply ran the method for one epoch. By reducing the optimization to 100 iterations, the total training time for our method is 35 seconds. Thus, our approach is faster to train compared to AssoAlign (which requires 87 seconds) and yields higher accuracy.

Table 1: Testing performances at different finetuning steps for 5-way 5-shot experiments.

steps	miniImageNet	FC100	CIFAR-FS
SKD-GEN0	82.04	64.24	89.38
20	81.26	64.07	88.68
40	83.12	65.53	89.47
60	83.93	66.06	89.94
80	84.51	66.57	90.05
100	84.79	66.90	90.21
120	84.90	67.16	90.18
140	84.94	67.19	90.24
160	84.99	67.29	90.15
180	85.05	67.36	90.19
200	85.06	67.32	90.17
220	84.97	67.19	90.18
240	85.04	67.27	90.22
260	85.04	67.31	90.16
280	85.04	67.29	90.16
300	84.93	67.27	90.16
320	84.99	67.30	90.13
340	84.88	67.33	90.13
360	84.83	67.26	90.16
380	84.84	67.25	90.13

Table 2: Testing performances with different number of training examples per class for 5-way 5-shot experiments.

examples per class	number of steps	miniImageNet	FC-100
50	100	84.73	66.54
100	100	84.97	66.18
200	100	85.08	66.79
300	100	85.54	67.08
600	300	85.60	67.20

Table 3: Testing performances with fewer number of training examples per base class for 5-way 5-shot experiments.

examples per class	number of steps	miniImageNet	FC-100
100	50	84.66	66.39
200	50	84.90	66.56
300	50	85.35	66.60

Another way to speed up the training is by reducing the number of base dataset examples used for pretraining and finetuning. To perform this ablation, we randomly sample images from the base dataset to form new base datasets with 50, 100, 200 and 300 examples per class (the full base dataset has 600 examples per class). Because of the reduced dataset sizes, we finetune our model for only 100 gradient steps (instead of 300). The training uses the same exact learning policy discussed in the main paper (learning rate: 0.025). We use SKD-GEN1 for pretraining for the 5-shot experiments shown in Table 2.

Next, we further reduce the number of finetuning steps to 50, but now use a larger learning rate (0.1 for miniImageNet

and 0.2 for FC-100). The results in Table 3 show that this configuration allows us to reduce the training cost to 1/6, at the expense of a negligible drop in accuracy: only 0.3% on miniImageNet and 0.6% on FC-100. Under this last configuration, our model can be trained in just 18 seconds, compared to the 87 seconds required by AssoAlign [1].

Accuracy as a function of the base dataset size

Here we analyze how the accuracy of our system varies as we reduce the size of the base dataset available for pretraining and finetuning. We perform this a study by training SKD (the baseline) as well as our method on a subset of tieredImageNet obtained by sampling only 1/10-th of the examples from each base class. This produced a base dataset having roughly the same size as the other benchmarks considered in our experiments (since the whole tieredImageNet is about 10 times bigger than the other datasets). We can see from Table 4 that while our method produced a gain of +0.96% over the baseline on the full-size dataset (see Table 1 in the main paper), the improvement becomes +1.72% on the smaller dataset. Intuitively this makes sense. If the base dataset is very large, the embedding learned on the base dataset will be of high quality and will produce good generalization to novel classes even with a simple method such as linear regression. Conversely, when the base dataset is smaller there will be more benefit in adapting the embedding to the specific patterns of the novel classes. Our approach provides an effective way to achieve this goal by tuning the representation via pseudolabeling of the base dataset.

Table 4: Experiments of 5-way 5-shot on 1/10 of tieredImageNet

	full tieredImageNet	1/10 of tieredImageNet
SKD	85.97+/-0.63	80.60+/-0.82
ours	86.93+/-0.60	82.32+/-0.80

Stochastic finetuning using random mini-batches of the support set

Benchmarks for few-shot learning such as miniImageNet, CIFAR-FS and FC100 can support at most 20-way classification problems, since their meta test sets include only 20 classes. Furthermore, the most common experimental setups on these datasets are "5-way 1-shot" and "5-way 5-shot". For such cases, as well as for the 20-way and 10-way experiments we showed above, the entire support set (consisting of a few hundreds images) can be fed to the GPU in each step. Thus, prior methods (e.g., RFS, SKD, IER) use the whole support set in each learning step. We follow their experimental setup and feed the model with the whole support set.

However, we note that our method does not strictly require the whole support set at each learning step. Like any other method based on stochastic-gradient optimization, we could form mini-batches containing only a subset of examples from the support set. To demonstrate that this strategy does not degrade the accuracy of our approach, we adopt this mini-batch optimization on a "20-way 10-shot" miniImageNet experiment, where we purposely use a "5x" data augmentation in order to make the support set too large to be fed to the GPU. With this 5x factor, the support set contains a total of 20x10x5=1000 examples. In each step of finetuning we randomly sample 125 images (from this support set of 1000 images) and draw 125 base images to construct our mini-batch. Under this setting, our method Label-Halluc achieves an accuracy of 65.17% versus the 61.18% obtained by SKD-GEN1. Finally, we also note that, for both Label-Halluc and SKD-GEN1, these numbers are higher than those we reported in our previous response where the experimental setup was 20-way 10-shot but without data augmentation (the support set included 200 images in each episode). This suggests that random sampling combined with data augmentation is a winning strategy over using the entire support set without data augmentation in each iteration.

Table 5: 10-way and 20-way experiments on FC100.

FC-100	10-way 5-shot	20-way 10-shot
SKD-GEN1	46.66+/-0.49	37.55+/-0.22
ours	**49.63+/-0.48**	**42.21+/-0.22**

Table 6: 10-way and 20-way experiments on CIFAR-FS.

CIFAR-FS	10-way 5-shot	20-way 10-shot
SKD-GEN1	79.01+/-0.80	71.33+/-0.30
ours	**80.99+/-0.75**	**75.40+/-0.31**

Table 7: 10-way and 20-way experiments on miniImageNet.

miniImageNet	10-way 5-shot	20-way 10-shot
SKD-GEN1	79.01+/-0.55	71.33+/-0.20
ours	**80.99+/-0.53**	**75.40+/-0.19**

Results for large number of novel classes in each episode

Here we apply Label-Halluc (our method pretrained w/ SKD) on "10-way 5-shot" and "20-way 10-shot" classification problems on FC-100, CIFAR-FS and miniImageNet. We use a 5x data augmentation for both SKD and our method in the "10-way 5-shot" setting, the same data augmentation strategy used in the main paper. Due to the limitation of GPU memory, for the "20-way 10-shot" setting, we do not use data augmentation for SKD or our method. The following results use the same learning policy discussed in the main paper, with the exception of finetuning for 2 epochs in the case of FC-100 and CIFAR-FS. From the results in Table 5, Table 6, Table 7 we can observe that when the classification problem involves a higher number of classes, our approach yields even larger gains over the baseline transfer learning method (SKD-GEN1) compared to the standard 5-way 1-shot and 5-way 5-shot settings.

Results when the base classes and the novel categories are far apart

In this section we study the behavior of our method when the semantic gap between the base dataset and the novel classes is substantial.

We start by considering an experimental scenario where the base dataset contains only animal classes, while the novel dataset has some novel animal classes and some common objects.There are 16 animal classes within the 64 base classes of miniImageNet. We created a base dataset including only these animal classes, thus producing a reduced base dataset of only 16x600=9600 examples. Using this base dataset we pretrained the model as 16-way classifier with SKD-GEN0, and then we finetuned the network on this base dataset with Label-Halluc (our method). Table 8 reports the accuracy of novel-class recognition for both our approach and a linear classifier trained on top of SKD-GEN0.

Table 8: Experiments on miniImageNet, with pretraining done on a base set containing only animal classes. The novel categories include animal classes and other common classes.

	SKD-GEN0	Label-Halluc
1-shot	51.17	**54.51**
5-shot	69.07	**74.31**

By comparing the results above with those in Table 1 of our main paper, we can observe that both the baseline method and Label-Halluc produce lower accuracies when pretrained on the smaller base dataset containing only animal classes. This is due to the fact that the representation learned from the base dataset is skewed towards animal categories and thus produces poorer generalization for novel categories that are outside this domain. Furthermore, the base dataset is now considerably smaller and therefore both the pretraining as well as the finetuning procedure have fewer training examples at their disposal. However, we can observe that, under this scenario, Label-Halluc produces an even larger gain over SKD-GEN0 compared to when using all base classes of miniImageNet: the improvement in 1-shot accuracy enabled by our method is now +3.34% versus the +0.96% observed in Table 1. This makes intuitive sense since, under this setting, the pretrained representation lacks generality and thus it becomes particularly beneficial to adapt the backbone to the characteristics of the novel classes. These results suggest that our pseudolabeling method can successfully adapt the representation also in this extreme scenario where all base classes fall within a niche domain.

Next, we form an experimental setup where the base categories and the novel classes are completely unrelated. We use the 16 classes of animals from the 64 class of miniImageNet base set to construct our new base dataset. The set of novel classes is now restricted to 11 classes of non-animals out of the original 20 classes. During meta-testing, we sample 5 classes from those 11 novel classes to construct our 5-way episodes. To accelerate the evaluation time, we follow the shorthened-learning strategy already explored during the rebuttal. This reduces the finetuning steps of Label-Halluc from 300 steps to 160 steps. We evaluate by averaging performance over 400 episodes. Table 9 summarizes the performance of the baseline model SKD-GEN0 and our Label-Halluc, both trained and evaluated on these reduced base dataset and novel set.

Table 9: Experiments on miniImageNet, with pretraining done on a base set containing only animal classes. The novel categories include only *non*-animal classes.

	SKD-GEN0	Label-Halluc
1-shot	39.38	**43.56**
5-shot	56.78	**64.99**

Compared to the the results presented for the case when the difference between the base and novel datasets is "large but not extreme" (16 classes of animal base classes and 20 classes of novel classes), we observe that both methods suffer performance drops. This is understandable given that the base dataset now includes only animals while the novel classes are all non-animals. However, we can noticed that the gain of Label-Halluc over SKD-GEN0 has grown even further: in the 5-shot setting it is now +8.21% vs the +5.24% of the'large difference" setting and the +2.42% of the original setting involving all 64 classes in the base dataset and 20 classes in the novel set. This further confirms that when the base classes and novel classes are further apart, the finetuning of the original embedding becomes increasingly important to retain good performance.

We also consider the opposite scenario where we use the 48 base classes of non-animals in miniImageNet to form the base dataset. We then evaluate according to two distinct novel sets: (1) a novel set consisting of all 20 novel classes (including both animal and non-animal classes) and (2) a novel set that has only 9 classes of animals. We evaluate the two approaches over 400 5-way episodes under the 5-shot setting with 160 finetuning steps for Label-Halluc. Table 10 reports the results.

Table 10: Experiments on miniImageNet, with pretraining done on a base set containing only *non*-animal classes. The novel categories include only animal classes.

novel set	SKD-GEN0	Label-Halluc
20 classes	78.33	**81.39**
9 classes of animals	65.73	**69.76**

We recall that under the original setting considered in the paper (i.e., a base set containing all 64 classes and a novel set with all 20 classes) the performance gain of Label-Halluc over SKD-GEN0 was 2.42%. We see from Table 10 that this gain increases to 3.06% when the base set includes only 48 non-animal classes and the novel set contains all 20 classes. However, understandably, both methods achieve lower accuracies compared to the original setting due to the reduced and more skewed base dataset. Finally, the gain of Label-Halluc over SKD-GEN0 grows to 4.03% when the novel

set is the most distant from the base set (i.e., the novel set contains only 9 animal classes and the base set includes 48 non-animal classes).

Experiments with imbalanced base classes

In this section we study our method in a setting where the base dataset contains large imbalance levels, with a few majority classes resulting in a skewed representation of base classes in the batch. To simulate this scenario, we re-sampled miniImageNet to form a new base dataset where 48 classes have much lower representation than the other 16 classes. Specifically, in each of the 48 classes contains only 60 examples, while the remaining 16 classes include 600 examples each. Thus, the total number of examples is 16x600+48x60=12480. We use this base dataset for both SKD as well as our Label-Halluc. Note that we did not make any attempts at rebalancing the classes during pretraining or finetuning. Because mini-batches are formed by random sampling from , the majority classes tend to be more represented than the others during learning. We see from Table 11 that Label-Halluc works well even in this scenario of imbalanced classes, and it produces substantial improvements over the baseline: +1.5% in the 1-shot setting and +3.2% in the case of 5-shot recognition.

Table 11: Experiments for a manually-constructed imbalanced version of miniImageNet.

	SKD-GEN0	Label-Halluc
1-shot	56.61+/-0.79	**58.11+/-0.75**
5-shot	75.02+/-0.83	**78.34+/-0.84**

Recognition of base and novel classes

Our work is aimed at improving few-shot classification by means of the base dataset, without any attempt at preserving good performance on the original base classification problem. However, in some settings, it may be beneficial to retain good recognition performance of the base classes while learning to recognize the novel categories. Thus, we performed a study under this setting on miniImageNet using 5 examples per novel class. We experimented with two variants.

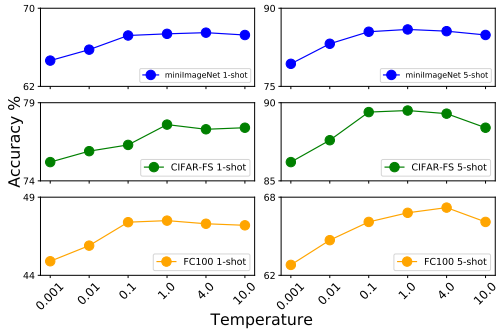
The first involves attaching the unmodified original base classifier (i.e., the last layer of the network) to the backbone obtained after finetuning on pseudo-labels. As expected, the resulting performance is very low: the base classification top-1 accuracy drops to 2.3%, which is barely above random chance (1.56%). This is understandable given that the linear classifier is applied without any form of retraining to a representation that has been heavily modified by finetuning on pseudo-labeled data.

We consider a second setting, similar to that used in multi-task learning. During finetuning, we append a 64-way linear classifier optimized for base classification (using the original base labels), in addition to the linear classifier trained with pseudo-labels. Note that the gradient back-propagated from the classification loss on base labels is stopped at the second-to-last layer. Thus, it is used only to update the base classifier. Conversely, the gradient computed from the pseudo-labels is back-propagated all the way through the network. This setup adds a negligible cost to our finetuning procedure since only one additional layer is updated. However, it provides the benefit of keeping the base classifier updated with respect to the representation learned from the pseudo-labels. We found that this strategy yields a top-1 accuracy of 77.81% on the original 64-way classification problem, while not degrading at all the performance on the few-shot novel-class recognition problem (since the gradient from the base labels is not used to update the backbone). Note that the original network trained exclusively on the base labels achieves an accuracy of 83.93% on the 64-way classification problem. This demonstrates that this strategy allows us to obtain a representation optimized for novel-class few-shot recognition that retains good accuracy on the original classification problem (only 6.12% lower than that achieved with a network exclusively optimized for base classification performance).

Ablation on knowledge distillation

The knowledge distillation loss in Eq. 3 of our submission involves a temperature hyper-parameter to soften the logits in the softmax. By default, we use a temperature value of 4.0. Setting the temperature to extremely small values effectively converts the soft-labels into hard ones, while large values of temperature smooth the soft-labels. As shown in fig. 5, our method is fairly robust to the choice of temperature, unless this is set to a very small value.

Figure 5: Ablation study on different temperature hyper-parameter values used in knowledge distillation when finetuning the network on the pseudo-labeled base examples.



Experiment with feature distillation

As shown in Equation 3, our proposed method performs knowledge distillation between the output logits of base examples and produced pseudo labels. A natural question is whether it is actually necessary to predict the pseudo labels for base examples, or if it is sufficient to perform feature distillation, i.e, minimizing the distance between $f_{\Theta^{base}}(x)$ and $f_{\Theta}(x)$ where x is from $\mathcal{D}^{base}$ in Equation 3. To answer

this question, we changed the first term of Equation 3 in the main paper into

$$E_{x,y\in\mathcal{D}^{base}} MSELoss(f_{\Theta}(x), f_{\Theta'}(x)) \quad (1)$$

to perform feature distillation. We found that this leads to a degradation in accuracy on miniImageNet 5-way 5-shot: 80.89%, compared to the 83.18% produced by SKD-GEN1 and the 85.60% achieved by our Label-Halluc. We believe that this happens because feature distillation excessively constrains the backbone to produce features similar to those learned from the base dataset at the expense of performance on the novel set. This result emphasizes the importance of predicting the pseudo-labels for base examples by using the classifier learned on the novel classes.

Overview of datasets

miniImageNet [10] is one of the most commonly used benchmark for few-shot classification. It is a subset of ImageNet [3] containing images downsampled to a resolution of 84×84 pixels. It includes 100 classes, each class with 600 examples. It is further divided into 64 classes for meta-training, 16 classes for meta-validation and 20 classes for meta-testing. **tieredImageNet** [7] is another subset of ImageNet. It contains a total of 608 classes, with 351 classes used for meta-training, 97 classes for meta-validation and the remaining 160 classes for meta-testing. **CIFAR-FS** [1] is derived from CIFAR-100 dataset. The original 100 classes are split into 64 classes for meta-training, 16 classes for meta-validation and 20 classes for meta-testing. **FC-100** [5] is also obtained from CIFAR-100. It has 60 classes for meta-training, 20 classes for meta-validation and 20 classes for meta-testing.

Network Architecture

To make fair comparison to recent works [9, 6, 8], we adopt the popular ResNet-12 [4] as our backbone. The network has 4 residual blocks, each containing 3 convolutional layers with 3×3 convolution. A 2×2 max-pooling is applied at the end of each of the first three blocks and an average-pooling is used in the last one. Our ResNet-12 is identical to those used in RFS [9], SKD [6] and IER [8]. Thus the number of channels for 4 residual block is set to (64,160,320,640).

Explanation of hyperparameter and design choices

In this section we provide further details about the hyperparameter values used in our experiments, including the number of batches, and the ratio of base vs novel images in the optimization.

Ratio of base vs novel images. In all our experiments we used a constant 1:1 ratio for the base and novel images in our finetuning optimization. We used this ratio because it is the simplest choice when merging samples from two sets and because we wanted to adopt a single common setting for all benchmarks. However, we found that tuning this hyperparameter on the different datasets further improves the performance of our method. For example, for 5-way 1-shot classification on FC-100, a 5:1 ratio of base:novel images (miniImageNet with 3:1) yields superior results as shown in Table 12.

Table 12: Tuning the hyper-parameter to control the ratio between the base and novel examples in the mini-batch can lead to better results.

	FC-100 5-way 1-shot	miniImageNet 5-way 1-shot
SKD	46.5	66.54
ours (1:1)	47.3	67.50
ours (k:1)	**48.0** (k=5)	**68.35** (k=3)

Number of batches. For a fair comparison, we chose to adopt the same settings as in prior work. For 5-way, 5-shot (5x5=25 examples) classification, prior transfer learning methods [46, 33, 37] use a 5x augmentation leading to 125 novel examples. In order to have an equal proportion of base and novel images in each batch (1:1 ratio), our method samples another 125 base examples to form a batch size of 250. For the experiments reported in the paper we kept the values fixed across all datasets for both ratio and batch size, in order to demonstrate that the empirical gains of our method come from our pseudo-labeling strategy rather than from tuned hyperparameters. Similarly for 5-way, 1-shot (5x1=25 distinct examples) classification, we use 25x augmentation, as in prior work, leading to 125 novel examples, and then we randomly sample another 125 base examples to form a mini batch of size 250.

Sampling of the base images. For all the experiments in the paper, in each iteration we select 125 samples from the entire base dataset according to a uniform random sampling strategy. We experimented with different selection strategies but found them to be inferior to random sampling. For example, the second to last row in Table 6 of the main paper shows the result of an ablation where we adopted the selection method of AssoAlign to train our method. While this selection strategy has been shown to yield improvements for their feature euclidean/adversarial alignment, it causes a drop in accuracy for our method compared to uniform sampling.

Optimization details

For the embedding training, we use the public code implementations of SKD [6] and IER [8]. For the training of the linear classifiers on top of frozen features, we use LogisticRegression with the LBFGS optimizer from scikit-learn package [2], as in RFS [9]. When finetuning, we use the SGD optimizer with a momentum of 0.9 and a weight decay of $5e^{-4}$ across all experiments in the four benchmarks. The learning rate for the network up to the penultimate layer is set to 0.025 while the final classification layer uses a learning rate of 0.05.

We use mini-batches of 250 examples. For the 5-way 5-shot classification in miniImageNet, CIFAR-FS and FC100, we generate 5 views from each of the 25 novel images in the support in order to obtain 125 examples. We use the same data augmentation transformations employed in prior works [9, 6, 8]). We fill the remaining half the mini-batch by

sampling 125 distinct examples from the base dataset with associated pseudo labels. The finetuning runs for 1 epoch in order to complete a pass over the entire base dataset for each episode. This amount to $\sim$300 steps. Similarly in 5-way 1-shot classification, each mini-batch has a support set of 125 novel examples (5 distinct images augmented 25 times) and 125 pseudo-labeled base examples.

The tieredImageNet dataset is much larger than the other three datasets and the images have much higher resolution. Thus, a finetuning procedure that iterates through the whole base dataset for each episode is intractable. Thus, we finetune for 200 steps for both the 1-shot and the 5-shot settings.

Each of these four benchmarks includes a meta-training set (the base dataset), a meta-validation set and a meta-testing set (the novel-class dataset) organized in episodes. The meta-validation set is only used for searching hyperparameters. The experiments are carried out on a desktop server with Intel i9-9960X CPU and four NVIDIA RTX-2080Ti GPUs.

References

[1] Bertinetto, L.; Henriques, J. F.; Torr, P.; and Vedaldi, A. 2019. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*.

[2] Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; Layton, R.; VanderPlas, J.; Joly, A.; Holt, B.; and Varoquaux, G. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*.

[3] Deng, J.; Dong, W.; Socher, R.; Li, L.; Kai Li; and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*.

[4] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[5] Oreshkin, B. N.; López, P. R.; and Lacoste, A. 2018. TADAM: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*.

[6] Rajasegaran, J.; Khan, S.; Hayat, M.; Khan, F. S.; and Shah, M. 2020. Self-supervised Knowledge Distillation for Few-shot Learning. *arXiv preprint arXiv:2006.09785*.

[7] Ren, M.; Ravi, S.; Triantafillou, E.; Snell, J.; Swersky, K.; Tenenbaum, J. B.; Larochelle, H.; and Zemel, R. S. 2018. Meta-Learning for Semi-Supervised Few-Shot Classification. In *International Conference on Learning Representations*.

[8] Rizve, M. N.; Khan, S.; Khan, F. S.; and Shah, M. 2021. Exploring Complementary Strengths of Invariant and Equivariant Representations for Few-Shot Learning. *arXiv preprint arXiv:2103.01315*.

[9] Tian, Y.; Wang, Y.; Krishnan, D.; Tenenbaum, J. B.; and Isola, P. 2020. Rethinking Few-Shot Image Classification: A Good Embedding is All You Need? In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*.

[10] Vinyals, O.; Blundell, C.; Lillicrap, T.; kavukcuoglu, k.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems*.