Enhancing Prototypical Networks for Few-Shot Learning

Xiao Chen
Stanford University
{markcx}@stanford.edu

Abstract

The goal of few-shot learning is to train a classifier for distinguishing images from novel classes based on a limited amount of labeled examples. Many existing studies on few-shot learning assume the within-domain setting, in which examples from base and novel classes come from the same domain. A related but more challenging problem is cross-domain few-shot learning, in which there is large domain discrepancy between base and novel classes. In this work, we tackle both problems by 1) fine-tuning the Prototypical Networks (ProtoNet) with a pre-trained EfficientNet as the backbone, 2) applying multiple data augmentation techniques, and 3) exploring various distance metrics and approximating prototypes via Gaussian distributions. Experimental results demonstrate that our approach achieves good performance on 5-way 5-shot learning tasks evaluated on CIFAR100 and *mini*Imagenet.

1 Introduction

Training a neural network model usually requires a large amount of labeled data. However, collecting labeled data in some categories or domains can be challenging. For example, it can be difficult to collect a large number of medical image data for certain rare diseases Irvin et al. (2019). In addition, labeling such data requires expertise and is expensive. The question arises: how can we build an image classifier using a limited amount of labeled data?

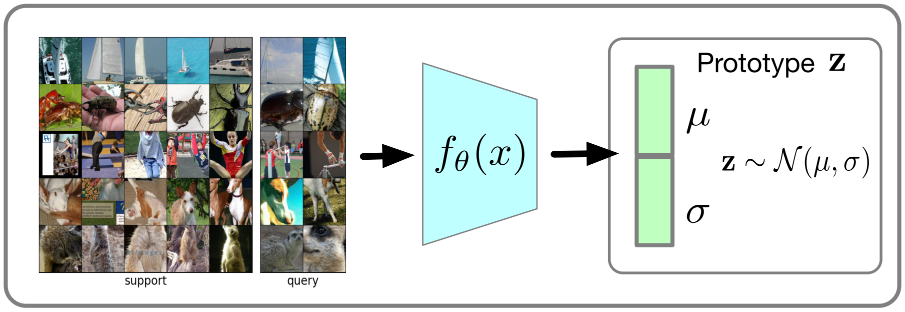

Figure 1: **Our approach:** We use transfer learning to pretrain weights from a relevant dataset as initialization. We used *mini*ImageNet and Omniglot best trained weights for within domain and cross domain baseline. We use EfficentNet $f_\theta(x)$ for feature extraction followed by a Prototypical Network to map a prototype per class using Gaussian approximation. The KL-distance is adopted between two multivariate Gaussians.

This problem is described as meta-learning or few-shot learning. Most existing approaches to this problem contain two steps: meta-training and meta-testing. In meta-training, a given dataset that has

submitted to CS231N 2020, Stanford, CA

a base set of classes is divided into support sets and query sets. A model is fitted on the support set to predict labels of examples in the query set. In meta-testing, a small number of training examples from novel classes are provided by a support set, and the goal is to use both the support set in meta-testing and the information extracted in the meta-training step to predict labels for examples contained in a meta-testing query set, which comes from novel class-set as well.

Two main challenges are in the few-shot learning problem: within- and cross-domain. In the former case, base and novel classes come from the same domain, but the specific types or classes in testing don't overlap with training examples. For example, both training and testing samples are hand-written characters but are from different languages. In the latter scenario, there is a significant domain shift between examples in base and novel classes. For example, natural images are given in the meta-training time, but the meta-testing task is classifying medical images.

We tackle the few-shot learning problem by combining techniques from transfer learning, meta-learning, Gaussian approximation, and data augmentation. Specifically, we use a EfficientNet Tan & Le (2019) pre-trained on ImageNet Deng et al. (2009) as the backbone, and we fine-tune the model weights to adapt to the domain of interest. The output of the EfficientNet is then passed to a linear layer that learns a sample's mean and variance. Such a structure can be considered as a variant of Prototypical Network (ProtoNet) Snell et al. (2017). Each class is represented by a prototype (or a prototype distribution), and a new image is assigned to the class whose prototype is the closest to the image in the learned embedding space. In addition, we also study the performance gain of applying a data augmentation technique called mixup Zhang et al. (2017), which linearly interpolates between training examples to generate new examples with their interpolated labels. Such a mixed augmentation can be alternatively interpreted as a form of "bootstrapping" because the limited observed examples can yield more samples by linear interpolations. Our approach is conceptualized in Figure 1. We test our method on both within- and cross-domain few-shot learning tasks.

Comparing with state-of-the-art experiments, we achieve good performance on a 5-way 5-shot learning task based on CIFAR100, which gives about 63% accuracy. In general, our results are largely comparable with those of the state-of-the-art methods on multiple datasets such as MNIST, Omniglot, and miniImageNet. We find that mixup can help improve classification accuracy in a 10-way 5-shot learning task on CIFAR 100. In addition, fine-tuning only major blocks, rather than all parameters, within a pre-trained EfficientNet gives better performance.

2 Enhancing Few-shot Learning

Let a domain $(\mathcal{X}, \mathcal{Y})$ be defined as the joint distribution of image space $\mathcal{X}$ and label space $\mathcal{Y}$. Let (x, y) note a labeled example, in which y is the label for x. In cross-domain few-shot learning, the source domain $(\mathcal{X}_s, \mathcal{Y}_s)$ in meta-training is different from the target domain $(\mathcal{X}_t, \mathcal{Y}_t)$ in meta-testing, whereas in within-domain, the source and the target domains are the same. In meta-testing, a support set, $S = \{x_i, y_i\}_{i=1}^{K \times N}$, is provided such that for each one of the N base classes, there are K labeled examples. The goal is to train a classifier f_θ that minimizes $\mathbf{E}_{(x,y)\sim(\mathcal{X}_t,\mathcal{Y}_t)}[\ell(f_\theta(x; S, (\mathcal{X}_s, \mathcal{Y}_s)), y)]$ for a loss function ℓ, where the expectation is evaluated using the query set in meta-testing.

Our approach has three main components: applying mixup for data augmentation Zhang et al. (2017), fine-tuning a pretained EfficientNet Tan & Le (2019) for feature extraction, and using ProtoNet Snell et al. (2017) with Gaussian approximation for metric space learning. These components are discussed as follows.

Mixup for Data Augmentation. Data augmentation, especially in terms of more recent developments beyond flipping, cropping, and scaling, is an effective way to enhance data efficiency by both increasing the amount and improving the diversity of data Perez & Wang (2017); Lemley et al. (2017); Cubuk et al. (2019). In addition to standard data augmentation, we implement a technique called mixup Zhang et al. (2017) that takes linear combinations of any two training examples and their corresponding labels to generate new examples with labels. Specifically, a new training example $(\tilde{x}, \tilde{y})$ is generated according to $\tilde{x} = \lambda x_i + (1 - \lambda)x_j$ and $\tilde{y} = \lambda y_i + (1 - \lambda)y_j$, where (x_i, y_i) and (x_j, y_j) are training examples, y_i and y_j are labels in one-hot encoding, and the distribution of $\lambda \in [0, 1]$ is drawn from Beta(α, α) for $\alpha \in (0, \infty)$. We pick the value of $\alpha \in \{0.5, 0.8, 1, 1.5\}$ when it yields the best performance.

Incorporating Meta-Transfer Learning. The core idea of meta-transfer learning (MTL) Sun et al. (2019) is that networks trained on large datasets can learn to extract salient features such as blobs and edges, which are universal across various image classes. We make use of the meta-transfer learning by using the EfficientNet as our backbone. In the cross-domain setting, we substitute the weights with the best pre-trained weights obtained from our own training. We also experiment various configurations for fine-tuning the EfficientNet by freezing specific layers.

Exploring Distance Metrics. As the EfficientNet extracting the features embedding, we adopt a linear layer ψ to yield mean and variance per sample, i.e. $(\mu_i, \sigma_i) \leftarrow \psi(f_\theta(x_i))$. Then a prototype of a class k is expressed as an average of its mean and variance,

$$\mu^{(k)} = \tfrac{1}{|S_k|} \textstyle\sum_{(x_i,y_i)\in S_k} \mu_i, \quad \sigma^{(k)} = \tfrac{1}{|S_k|} \textstyle\sum_{(x_i,y_i)\in S_k} \sigma_i. \tag{1}$$

We adopt the KL distance Kullback & Leibler (1951) to measure the closeness between query samples and prototypes because the KL has a closed form between two Gaussian distributions:

$$\begin{aligned}&\mathrm{KL}\big(\mathcal{N}(\mu_i, \sigma_i) || \mathcal{N}(\mu^{(k)}, \sigma^{(k)})\big)\\ =&\frac{1}{2}\Big(\log\frac{\det(\sigma^{(k)})}{\det(\sigma_i)} - d + \mathrm{Tr}\big(\big(\sigma^{(k)}\big)^{-1}\sigma_i\big) + (\mu^{(k)} - \mu_i)^\top \big(\sigma^{(k)}\big)^{-1}(\mu^{(k)} - \mu_i)\Big),\end{aligned} \tag{2}$$

where d is the dimension of the prototypes. We choose latent dimension $d = 64$ for the experiment to align with the settings suggested in ProtoNet Snell et al. (2017). We also incorporate the euclidean and cosine distance, as discussed in appendix A.2.

3 Experiments

3.1 Datasets

The datasets we used include MNIST LeCun et al. (1998), Omniglot Lake et al. (2015), CFIAR100 Krizhevsky et al. (2009), and *mini*ImageNet Vinyals et al. (2016). In the context of meta-learning, we partition each dataset such that there are no overlapping classes in the training and test sets.

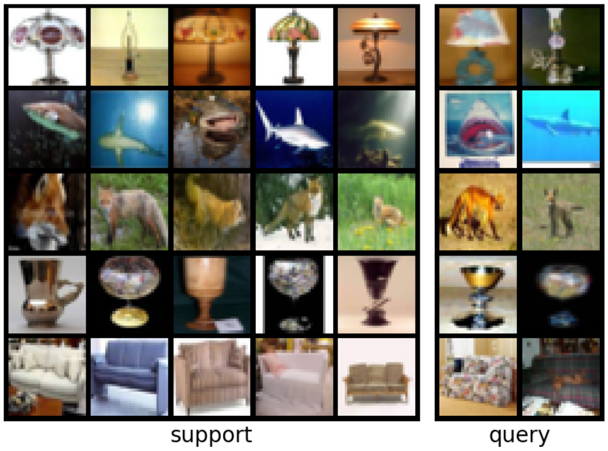

Figure 2: Sample images in a 5-way and 5-shot learning task based on CIFAR100. Rows correspond to image categories, and columns show examples within categories.

MNIST consists of 28×28 grayscale (1 channel) images of handwritten digits 0 through 9. We partition the digits 0 through 4 for training and digits 5 through 9 for testing. We also resize the images to 32×32 to fit the EfficientNet.

Omniglot Lake et al. (2015) dataset is one of 1623 types of handwritten characters from 50 different alphabets with 20 examples each. They are 28×28 gray-scale as well. In the training step, preprocessing includes a random horizontal flip, a random rotation of 90 degree, a color jittering of 5% in brightness, contrast, hue, and saturation, and resizing to 32×32. In the testing step, we resize images to 32×32.

CIFAR100 dataset has $32 \times 32 \times 3$ RGB colored images, containing 100 classes. We partition 67 classes for training and 33 classes for testing. Within the training set we randomly select 34 classes for the meta-train and remaining 33 classes for the meta-test (similar to the validation set in the setting of supervised learning), which follows the data split convention in Guo et al. (2019). We provide some examples in our training set from CIFAR100 in Figure 2.

***mini*ImageNet** is a subset (100 chosen classes) of ImageNet Deng et al. (2009) of images with 600 images per class. We use the partition of 64 classes for training, 16 for validation and 20 for testing, which follows the training convention from Ravi & Larochelle (2016) and Satorras & Estrach (2018). We show some examples in our training set from *mini*ImageNet in Figure 3. The images within a class are diverse in backgrounds and angles.

3.2 Results

We conducted experiments for few-shot learning under the within-domain setting (using CIFAR100, Omniglot, and *mini*ImageNet datasets) as well as the cross-domain setting (from Omniglot to MNIST). In the cross-domain case, weights are fine-tuned to improve performance. The primary performance metrics are test accuracies.

Pre-training: We used EfficientNet-b0 with pre-trained weights on ImageNet as the backbone architecture and cross-entropy loss to classify images. We used standard data augmentation mentioned in Section 3.1 and mixup regularization Zhang et al. (2017) for the query set during training. For the ImageNet data we resize and crop them down to 224×224 RGB images to leverage the EfficientNet. The models were implemented in PyTorch Paszke et al. (2019).

Hyperparameters: We used the Adam Optimizer with weight decay of 10^{-4}. The initial learning was chosen to be 10^{-3} for training involving CIFAR100 and ImangeNet datasets, and 5×10^{-4} for cross domain settings. In addition, the learning rate is decayed by 0.5 every 30 epochs. The batch size was the same as the number of categories (i.e. number of ways) for all settings. All trainings were done for at least 200 epochs. The hyperparameters were chosen to match those in state-of-the-art studies Vinyals et al. (2016); Sun et al. (2019) for a meaningful comparison of results.

Table 1: **Test accuracies of few-shot learning within domain (CIFAR100 and Omniglot).** The distance metric used is Euclidean. We compare results of our approach on different datasets, denoted as "dataset (Ours)" using standard and mixup data augmentation techniques with the best results achieved in the literature. The accuracies are the highest obtained amongst all episodes. The best results are in bold. (†Results are obtained by reimplementation under the same settings.)

	Data Aug.	5-way 1-shot	5-way 5-shot	10-way 5-shot	20-way 5-shot
MTL Sun et al. (2019)	standard	**43.6%**	52.6%	✗	✗
TADAM Oreshkin et al. (2018)	standard	40.1%	56.1%	✗	✗
CIFAR100 (Ours) - Euc.	standard	42.3%	**61.9%**	48.6%	**39.1%**
CIFAR100 (Ours) - Euc.	mixup	40.9%	60.5%	**51.2%**	37.9%
CIFAR100 (Ours) - KL	standard	41.5%	61.0%	50.8%	38.6
Matching Net Vinyals et al. (2016)	standard	**98.1%**	98.9%	90.3%†	**98.5%**
ProtoNet Snell et al. (2017)	standard	97.4%	**99%**	92.6%†	97.2%
Omniglot (Ours) - Euc.	standard	93.5%	97.8%	93.8%	93.1%
Omniglot (Ours) - Euc.	mixup	91.8%	96.5%	94.3%	91.2%
Omniglot (Ours) - KL	standard	94.8%	98.1%	**94.5%**	90.2%

Results for Within-Domain Few-Shot Learning. We compare results on three different datasets with documented best performances in literature across various scenarios, shown in Table 1 on CIFAR 100 and Omniglot and in Table 2 on *mini*ImageNet. Overall, our approach performed the best in 5-way 5-shot scenarios across datasets. In particular, our method with standard data augmentation is almost 6% more accurate than TADAM Oreshkin et al. (2018).

On CIFAR100, our approach achieved the highest accuracy of 61.9% in 5-way 5-shot learning, better than 56.1% as reported in TADAM Oreshkin et al. (2018). When the number of classes (N-way) increases, we see the performance of our model reducing to 48.6% and 39.1% accuracy for 10-way 5-shot and 20-way 5-shot respectively. We believe that as the number of categories (or tasks) grows, it naturally becomes more difficult for a model to classify images accurately and quickly (with limited

Table 2: **Test accuracies of few-shot learning within domain.** (The 5-way classification scenario in *mini*Imagenet)

	1-shot	5-shot
Matching Net Vinyals et al. (2016)	46.6%	60.0%
ProtoNet Snell et al. (2017)	49.4%	68.2%
MAML Finn et al. (2017)	48.7%	63.1%
TADAM Oreshkin et al. (2018)	58.5%	76.7%
MTL Sun et al. (2019)	60.2%	74.3%
*mini*ImageNet (Ours) - Euc.	71.6%	**87.2%**
*mini*ImageNet (Ours) - KL	**72.9%**	85.5%

observations). On the *mini*ImageNet, our approach performs much better than existing studies both on the settings of euclidean and KL distances.

Data augmentation using mixup did not decrease accuracies by more than 3%. But it improved accuracies are on CIFAR100 in 10-way 5-shot scenario as well as on Omniglot in 10-way 5-shot scenarios using the euclidean distance.

Results for Cross-Domain Few-Shot Learning. We studied the validity of our approach in cross domain learning, from Omniglot to MNIST in 5-way 1-shot and 5-way 5-shot scenarios. Although both datasets are images of handwritten objects, one is of characters and the other of digits. We find that initializing the network with the best trained weights from Omniglot and a selective (majority) of the blocks achieve around 10% performance improvements.

First, we obtain a baseline, shown as the MNIST baseline in Table 3, when training on digits 0-4 and testing on digits 5-9. We examine the effect of choosing different distance metrics. For 5-way 1-shot learning, cosine distance outperforms Euclidean distance marginally. We hypothesize this is because that when used in ProtoNet, Euclidean distance is equivalent to a linear transformation per class. Cosine distance does not have the effect, which allows it to perform better in fewer-shot learning scenarios.

Second, we utilize the best-trained weights from Omniglot dataset in Section 3.2 to train on MNIST dataset with different fine-tuning settings of the network blocks. The corresponding results are presented in the following ablation study.

Ablation study: We perform an ablation study directing training on different layers selectively. In particular, we focused on training only the final fully-connected (fc) layer, the last block (of sixteen blocks of EfficientNet-b0), the last block and the fully-connected layer, all sixteen blocks, and all layers (Table 3). Retraining the later layers (fc and last block) did not yield better results than the baseline. In fact, performances decreased. We hypothesize that this is because of the nature of the cross domain problem that only a new classifier retrained at the last layer does not improve performance. On the other hand, training all layers does not outperform (75.6% at its peak) retraining a majority of the model, achieving 77.4%, which is almost 9% higher than the better baseline result of 68.6%.

Table 3: **Test accuracies of few-shot learning cross domain. Omniglot → MNIST**. The experiment is 5-way classification. We compare Euclidean and cosine distance metrics and show results of different fine-tuning. The accuracies are the highest obtained amongst all episodes. The best results are in bold.

	Distance	Fine-tune	1-shot	5-shot
MNIST Baseline	Euc.	✗	48.3%	**68.6%**
	cos	✗	**52.7%**	64.23%
Omniglot Weights	Euc.	fc only	38.4%	54.9%
	Euc.	last block	47.6%	61.6%
	Euc.	last block + fc	40.0%	56.7%
	Euc.	all blocks	**62.9%**	**77.4%**
	Euc.	all blocks + fc	51.2%	75.6%

4 Conclusion

In this work, we set out to approach the problem of few-shot learning under both within- and cross-domain settings. Our model has two major ingredients: 1) fine-tuning ProtoNets with a pre-trained EfficientNet as the backbone and 2) applying a data augmentation technique called mixup. We test our model on within-domain tasks based on CIFAR100, Omniglot, MNIST, and miniImageNet and on a cross-domain task, which is transferring from Omniglot to MNIST. Our findings can be summarized as follows: i) Our model attains state-of-the-art performance ($\sim 6\%$ higher) on the 5-way 5-shot learning task based on CIFAR100. ii) Mixup can help improve accuracy in some classification tasks based on CIFAR100 (e.g. 10-way 5-shot setting). iii) The choice of distance metric plays a role in different learning scenarios. iv) Finetuning only the major blocks gives better performance ($\sim 10\%$ more) than finetuning all parameters within a pre-trained EfficientNet. As for future work, it would be fruitful to examine ways to improve performances of our method on these datasets in 10-way 5-shot and 20-way 5-shot scenarios.

References

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 113–123, 2019.

De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. L. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. *CoRR*, abs/1806.02817, 2018. URL http://arxiv.org/abs/1806.02817.

Guo, Y., Codella, N. C., Karlinsky, L., Smith, J. R., Rosing, T., and Feris, R. A new benchmark for evaluation of cross-domain few-shot learning. *arXiv preprint arXiv:1912.07200*, 2019.

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 590–597, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

LeCun, Y., Cortes, C., and Burges, C. J. The mnist database. *URL http://yann. lecun. com/exdb/mnist*, 1998.

Lemley, J., Bazrafkan, S., and Corcoran, P. Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, 5:5858–5869, 2017.

Oreshkin, B., López, P. R., and Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 721–731, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Perez, L. and Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2016. URL https://openreview.net/forum?id=rJY0-Kcll.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. One-shot learning with memory-augmented neural networks, 2016. URL https://openreview.net/forum?id=HJkPb0N9.

Satorras, V. G. and Estrach, J. B. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJj6qGbRW.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 2017-December:4078–4088, 2017. ISSN 10495258.

Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 403–412, 2019.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114, 2019.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pp. 3630–3638, 2016.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

A Appendix

A.1 Related Work

Current literature for few-shot learning can largely be divided into three categories: 1) memory-based methods that summarize information contained in training examples in external memory units, 2) optimization-based methods that first learn baseline models and then take gradient steps to fine tune the baseline models to fit novel categories, and 3) metric-based methods that learn embedding functions so that images from the same novel class are close in some embedding space.

Memory-based methods (e.g Santoro et al. (2016)) typically use recurrent-based models to sequentially process training images and summarize the contextual task information in an external memory unit. Memory-based models are expressive, but they can be data-inefficient as it can be challenging to summarize all useful information contained in the training set by a low dimensional vector.

Optimization-based methods are often regarded as "learning to learn", as they learn general initialization such that a good classifier for a novel class can be obtained from taking a few gradient steps from the initialization. MAML Finn et al. (2017, 2018) is a prominent example in this category. While these methods achieve state-the-art performance and are robust to noise Finn et al. (2017), they sometimes pose optimization challenges as these models often involve inner gradient steps.

Metric-based methods (e.g. ProtoNet Snell et al. (2017) and Matching Networks Vinyals et al. (2016)) learn embedding functions such that examples from each class are close in the embedding space according to some distance metric. The embedding function is usually represented by neural networks, and the distance metric usually takes the form of Euclidean or cosine distance. These methods are simple to train but can be limited in terms of model expressiveness.

We base our model on ProtoNet due to its simplicity and robustness. Different from Snell et al. (2017), we use a pre-trained EfficientNet as the backbone for extracting features from images and fine-tune its weights during the process. We also combine data augmentation techniques and different distance metric with ProtoNet. In addition, we show the potential of cross-domain few shot learning with our proposed adjustments.

A.2 Extending Prototypical Networks

ProtoNet uses a non-linear mapping f_θ: $\mathbb{R}^D \to \mathbb{R}^d$ to map an image into an embedding space, in which a prototypical representation $c_k \in \mathbb{R}^d$ for a class k is calculated as

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\theta(x_i), \tag{3}$$

where $|S_k|$ is the set of examples labeled with class k. For a new example x, the probability of this example belonging to class $y = k$ is computed using the softmax function as

$$p_\theta(y = k \mid x) = \frac{\exp\left(-\mathsf{dist}(f_\theta(x), c_k)\right)}{\sum_{k'} \exp\left(-\mathsf{dist}(f_\theta(x), c_{k'})\right)}, \tag{4}$$

where dist is a distance metric. When the Euclidean distance is used, that is

$$\mathsf{dist}_{\text{Euclidean}}\left(f_\theta(x), c_k\right) = \|f_\theta(x) - c_k\|_2^2. \tag{5}$$

Notice there is an equivalent linear representation:

$$p_\theta(y = k \mid x) = \frac{\exp\left(w_k^\top f_\theta(x) + b_k\right)}{\sum_{k'} \exp\left(w_{k'}^\top f_\theta(x) + b_{k'}\right)} \tag{6}$$

where $w_k = 2c_k$ and $b_k = -c_k^\top c_k$. Because it optimize the prototype c_k, we can ignore the $f_\theta(x)^\top f_\theta(x)$ in (5).

In addition to the Euclidean distance, we also consider a form of cosine distance defined as

$$\begin{aligned} &\mathsf{dist}_{\text{cosine}}(f_\theta(x_i), f_\theta(x_j)) \\ &= 1 - \frac{f_\theta(x_i)^\top f_\theta(x_j)}{\max\{||f_\theta(x_i)||_2 \cdot ||f_\theta(x_j)||_2, \epsilon\}} \end{aligned} \tag{7}$$

for small values of $\epsilon = 10^{-8}$ to avoid numerical issues.

The KL-divergence is a natural distance metric to use between distributions. We give a short derivation of a general form of equation (2). Suppose there are two normal distribution: q_1 is $\mathcal{N}(\mu_1, \Sigma_1)$ and q_2 is $\mathcal{N}(\mu_2, \Sigma_2)$. We have

$$\mathsf{dist}_{\mathrm{KL}}(q_1, q_2) = \mathrm{KL}(q_1 || q_2) = \mathbb{E}_{q_1}(\log \frac{q_1}{q_2}) \tag{8}$$

$$\begin{aligned} = \frac{1}{2}\mathbb{E}_{q_1}\Big[&- \log\det(\Sigma_2) - (z-\mu_2)^\top \Sigma_2^{-1}(z-\mu_2) \\ &+ \log\det(\Sigma_1) + (z-\mu_1)^\top \Sigma_1^{-1}(z-\mu_1)\Big] \end{aligned} \tag{9}$$

$$\begin{aligned} = \frac{1}{2}\Big[&\log\frac{\det(\Sigma_2)}{\det(\Sigma_1)} - \underbrace{d}_{=\mathrm{Tr}(\mathbf{I})} + \mathrm{Tr}(\Sigma_2^{-1}\Sigma_1) \\ &+ (\mu_1-\mu_2)^\top \Sigma_2^{-1}(\mu_1-\mu_2)\Big], \end{aligned} \tag{10}$$

when the last equality holds as we replace the random variable z with the mean μ_1 and $(z-\mu_1)(z-\mu_1)^\top$ with the covariance Σ_1 since it takes expectation of q_1. For simplicity, we just consider the diagonal covariance in our implementation.

When we take a closer look at the KL distance, i.e. $\mathsf{dist}_{\mathrm{KL}}\Big(\psi\big(f_\theta(x)\big), \mathcal{N}(\mu^{(k)}, \sigma^{(k)})\Big)$, the last term of equation (10) can be interpreted as $(\mu_x - \mu^{(k)})^\top \big(\sigma^{(k)}\big)^{-1}(\mu_x - \mu^{(k)})$ which is a form of Mahalanobis distance De Maesschalck et al. (2000). If we only consider this part, the equation (4) can be viewed as a Gaussian mixture model, i.e.

$$p(y=k|x) = \frac{\pi_k \mathcal{N}(\mu^{(k)}, \sigma^{(k)})}{\sum_{k'} \pi_{k'} \mathcal{N}(\mu^{(k')}, \sigma^{(k')})}, \tag{11}$$

where π_k and $\pi_{k'}$ are the prior probabilities of class k and k'. It is a soft-membership classification comparing with the hard-membership classification using the euclidean distance.

A.3 Examples from *miniImagenet*

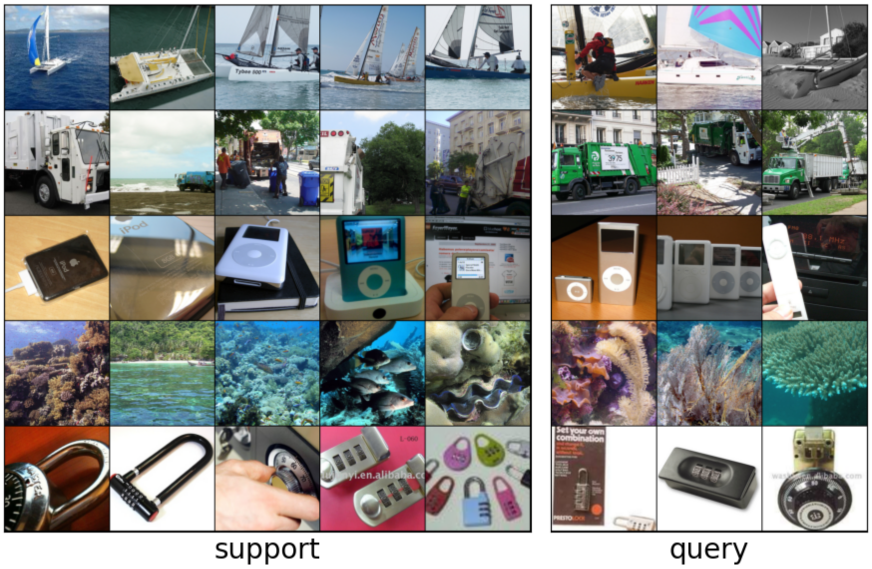

Figure 3: Sample images in a 5-way 5-shot learning task based on *mini*ImageNet. Rows correspond to image categories, and columns show examples within categories.