Self-Supervised Prototypical Transfer Learning for Few-Shot Classification

Carlos Medina[*†] CARLOS.MEDINATEMME@EPFL.CH
Arnout Devos[*] ARNOUT.DEVOS@EPFL.CH
Matthias Grossglauser MATTHIAS.GROSSGLAUSER@EPFL.CH
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract

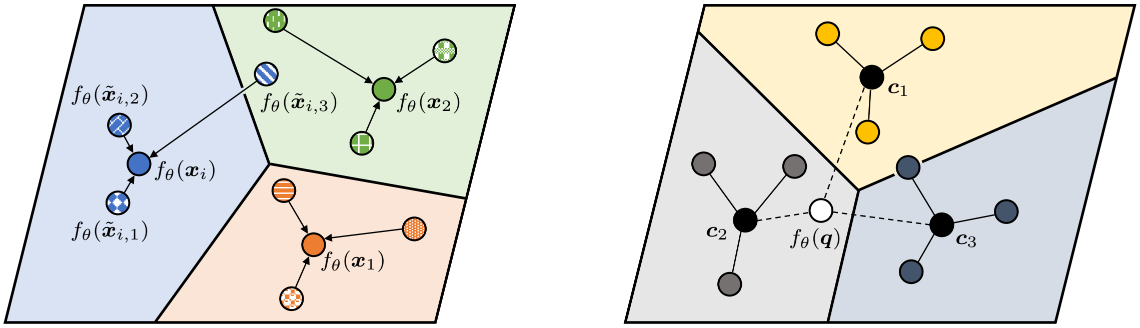
Recent advances in transfer learning and few-shot learning largely rely on annotated data related to the goal task during (pre-)training. However, collecting sufficiently similar and annotated data is often infeasible. Building on advances in self-supervised and few-shot learning, we propose to learn a metric embedding that clusters unlabeled samples and their augmentations closely together. This pre-trained embedding serves as a starting point for classification with limited labeled goal task data by summarizing class clusters and fine-tuning. Experiments show that our approach significantly outperforms state-of-the-art unsupervised meta-learning approaches, and is on par with supervised performance. In a cross-domain setting, our approach is competitive with its classical fully supervised counterpart. Code and pre-trained models are available on `github.com/indy-lab/ProtoTransfer`

1. Introduction

In few-shot classification (Fei-Fei et al., 2006) a classifier must adapt to distinguish novel classes not seen during training, given only a few examples (shots) of these classes. Meta-learning (Vinyals et al., 2016; Finn et al., 2017) is a popular approach for few-shot classification by mimicking the test setting during training through so-called episodes of learning with few examples from the training classes. However, several works (Chen et al., 2019; Guo et al., 2019) show that common (non-episodical) transfer learning outperforms meta-learning methods on the realistic cross-domain setting, where training and novel classes come from different distributions. Still, most few-shot classification methods still require much annotated data for pre-training. Recently, several unsupervised meta-learning approaches, constructing episodes via pseudo-labeling (Hsu et al., 2019) or image augmentations (Khodadadeh et al., 2019; Antoniou and Storkey, 2019), have addressed this problem. To our knowledge, unsupervised non-episodical techniques for transfer learning to few-shot tasks have not yet been explored.

Our approach ProtoTransfer performs self-supervised pre-training on an unlabeled source domain and can transfer to a few-shot target task. During pre-training, we minimize a pairwise distance loss in the embedding in order to cluster noisy augmentations of the same image around the original image. In the few-shot target task, in line with pre-training, we summarize class information in class prototypes for nearest neighbor inference similar to ProtoNet (Snell et al., 2017) and we support fine-tuning to improve performance when multiple examples are available per class.

[*] Equal contribution
[†] Most experiments by CM. Work performed as a semester project at EPFL-INDY supervised by AD,MG.


$f_\theta(\tilde{\boldsymbol{x}}_{i,2})$
$f_\theta(\tilde{\boldsymbol{x}}_{i,3})$
$f_\theta(\boldsymbol{x}_2)$
$f_\theta(\boldsymbol{x}_i)$
$f_\theta(\tilde{\boldsymbol{x}}_{i,1})$
$f_\theta(\boldsymbol{x}_1)$
$\boldsymbol{c}_1$
$\boldsymbol{c}_2$
$f_\theta(\boldsymbol{q})$
$\boldsymbol{c}_3$

(a) Self-Supervised Prototypical Pre-Training

(b) Prototypical Fine-Tuning & Inference

Figure 1: Self-Supervised Prototypical Transfer Learning. (a): In the embedding, original images $\boldsymbol{x}_i$ serve as class prototypes around which their Q augmentations $\tilde{\boldsymbol{x}}_{i,p}$ should cluster (b): Prototypes $\boldsymbol{c}_n$ are the means of embedded support examples for each class n and initialize a final linear layer for fine-tuning. An embedded query point $\boldsymbol{q}$ is classified via a softmax over the fine-tuned linear layer.

We highlight the main contributions and results of our approach:

1. We show that our approach exceeds the classification accuracy of state-of-the-art unsupervised meta-learning approaches on mini-ImageNet by 4% to 8% and has competitive performance on Omniglot.

2. Compared to the fully supervised setting, our approach achieves competitive performance on mini-ImageNet and multiple datasets from the cross-domain transfer learning CDFSL benchmark, with the benefit of not requiring labels during training.

2. A Self-Supervised Prototypical Transfer Learning Algorithm

ProtoTransfer combines self-supervised model pre-training and supervised fine-tuning. Sections 2.1 and 2.2 detail our pre-training ProtoCLR and fine-tuning ProtoTune parts, respectively. Figure 1 illustrates the overall procedure.

2.1 Self-Supervised Prototypical Pre-Training: ProtoCLR

We frame every ProtoCLR pre-training update step as an N-way 1-shot classification task optimized by a noise-contrastive prototypical loss function. In this, we draw inspiration from recent progress in unsupervised meta-learning such as UMTRA (Khodadadeh et al., 2019) and self-supervised visual contrastive learning of representations (Ye et al., 2019; Chen et al., 2020). Algorithm 1 details ProtoCLR and it comprises the following parts:

- Batch generation (lines 4-10): Each mini-batch contains N random samples $\{\mathbf{x}_i\}_{i=1\ldots N}$ from the training set, which serve as support samples. For each support sample $\boldsymbol{x}_i$, Q different randomly transformed versions $\tilde{\boldsymbol{x}}_{i,q}$ are used as query samples. $Q = 3$ showed the best performance in our experiments (see Section 3.3). For the transformations we use augmentations similar to (Chen et al., 2020), which are listed in Appendix B.3.

Algorithm 1 Self-Supervised Prototypical Pre-Training (ProtoCLR)

```
1: input: batch size $N$, augmentations size $Q$, embedding function $f_\theta$, set of random
   transformations $\mathcal{T}$, step size $\alpha$, distance function $d[\cdot,\cdot]$
2: Randomly initialize $\theta$
3: while not done do
4:     Sample minibatch $\{\boldsymbol{x}_i\}_{i=1}^N$
5:     for all $i \in \{1, \dots, N\}$ do
6:         for all $q \in \{1, \dots, Q\}$ do
7:             draw a random transformation $t \sim \mathcal{T}$
8:             $\tilde{\boldsymbol{x}}_{i,q} = t(\boldsymbol{x}_i)$
9:         end for
10:    end for
11:    let $\ell(i,q) = -\log \frac{\exp(-d[f(\tilde{\boldsymbol{x}}_{i,q}), f(\boldsymbol{x}_i)])}{\sum_{k=1}^N \exp(-d[f(\tilde{\boldsymbol{x}}_{i,q}), f(\boldsymbol{x}_k)])}$
12:    $\mathcal{L} = \frac{1}{NP} \sum_{i=1}^N \sum_{q=1}^Q \ell(i,q)$
13:    $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$
14: end while
15: return embedding function $f_\theta(\cdot)$
```

- Noise-contrastive prototypical loss optimization (lines 11-13): The pre-training loss encourages clustering of augmented query samples $\{\tilde{\boldsymbol{x}}_{i,q}\}$ around their prototype $\boldsymbol{x}_i$ in the metric embedding. Although in the experiments below, we use Euclidean distance, the method is generic and works with any metric. We do not create artificial tasks as in meta-learning, and can thus use any batch size N. A larger batch size aids self-supervised representation learning (Chen et al., 2020) and gives a significant performance improvement before fine-tuning (ablation study in Section 3.3). The loss is minimized w.r.t the embedding parameters θ with stochastic gradient descent.

2.2 Supervised Prototypical Fine-Tuning: ProtoTune

We use a prototypical fine-tuning approach, which we refer to as ProtoTune. First, the class prototypes $\boldsymbol{c}_n$ are computed as the Euclidean mean of the class samples in the support set S: $\boldsymbol{c}_n = \frac{1}{|S_n|} \sum_{(\boldsymbol{x}_i, y_i=n) \in S} f_\theta(\boldsymbol{x}_i)$. Following Snell et al. (2017), nearest-neighbor classification with respect to $\boldsymbol{c}_n$ can be re-interpreted as a linear classifier applied to a learned representation $f_\theta(\mathbf{x})$. We initialize a final linear layer with weights $\mathbf{W}_n = 2\mathbf{c}_n$ and biases $b_n = -||\mathbf{c}_n||^2$. Then, this final layer is fine-tuned with a cross-entropy loss on samples from the support set, while keeping the embedding function parameters θ fixed.

3. Experiments

We carry out in-domain few-shot classification experiments on Omniglot (Lake et al., 2011) and mini-ImageNet (Vinyals et al., 2016). Also, in a more challenging setting, we evaluate on the 4 datasets (ChestX, ISIC, EuroSAT, CropDiseases) from the cross-domain few-shot learning benchmark (Guo et al., 2019). Appendix A elaborates on visualizing the embedding and generalization gap performance. Experimental details can be found in Appendix B.

Table 1: Accuracy (%) of unsupervised pre-training methods on N-way K-shot classification tasks on Omniglot and mini-Imagenet on a Conv-4 architecture. For detailed results, see Tables 5 and 6 in the Appendix. Results style: **best** and second best.

Method (N,K)	**(5,1)**	**(5,5)**	**(20,1)**	**(20,5)**	**(5,1)**	**(5,5)**	**(5,20)**	**(5,50)**
	Omniglot				**mini-ImageNet**			
Training (scratch)	52.50	74.78	24.91	47.62	27.59	38.48	51.53	59.63
CACTUs-MAML	68.84	87.78	48.09	73.36	39.90	53.97	63.84	69.64
CACTUs-ProtoNets	68.12	83.58	47.75	66.27	39.18	53.36	61.54	63.55
UMTRA	83.80	95.43	**74.25**	**92.12**	39.93	50.73	61.11	67.15
AAL-ProtoNets	84.66	89.14	68.79	74.28	37.67	40.29	-	-
AAL-MAML++	**88.40**	**97.96**	70.21	88.32	34.57	49.18	-	-
ULDA-ProtoNet	-	-	-	-	40.63	55.41	63.16	65.20
ULDA-MetaOptNet	-	-	-	-	40.71	54.49	63.58	67.65
ProtoTransfer (ours)	88.00	96.48	72.27	89.08	**45.67**	**62.99**	**72.34**	**77.22**
Supervised training								
MAML	94.46	98.83	84.60	96.29	46.81	62.13	71.03	75.54
ProtoNet	97.70	99.28	94.40	98.39	46.44	66.33	76.73	78.91
Pre+Linear	94.30	99.08	86.05	97.11	43.87	63.01	75.46	80.17

3.1 Few-shot Image Classification: Omniglot and mini-ImageNet

In Table 1, we report performance on the mini-ImageNet and Omniglot benchmarks. Information about other approaches and how they compare to ours can be found in Section 4. On mini-ImageNet, ProtoTransfer outperforms all other state-of-the-art unsupervised pre-training approaches by at least 4% up to 8% On Omniglot, ProtoTransfer shows competitive performance with the unsupervised meta-learning approaches.

3.2 Cross-domain Few-Shot Learning: CDFSL benchmark

We report results on the CDFSL benchmark in Table 2. For comparison to unsupervised meta-learning, we include our results on UMTRA-ProtoNet and its fine-tuned version UMTRA-ProtoTune (Khodadadeh et al., 2019). Both use our augmentations instead of those from (Khodadadeh et al., 2019). For further comparison, we include ProtoNet (Snell et al., 2017) for supervised few-shot learning and Pre+Mean-Centroid and Pre+Linear as the best-on-average performing transfer learning approaches from Guo et al. (2019). As the CDFSL benchmark presents a large domain shift w.r.t. mini-ImageNet, all model parameters are fine-tuned. ProtoTransfer consistently outperforms its meta-learned counterparts by at least 0.7% up to 19% and performs on par with the supervised transfer learning approaches. Notably, on the dataset with the largest domain-shift (ChestX), ProtoTransfer outperforms all other approaches.

Table 2: Accuracy (%) of methods on N-way K-shot (N,K) classification tasks of the CDFSL benchmark (Guo et al., 2019). Both methods with unsupervised pre-training (UnSup) and without are listed. All models are trained on mini-ImageNet with ResNet-10. For detailed results, see Table 7 in the Appendix. Results style: **best** and second best.

Method	**UnSup**	**(5,5)**	**(5,20)**	**(5,50)**	**(5,5)**	**(5,20)**	**(5,50)**
			ChestX			**ISIC**	
ProtoNet		24.05	28.21	29.32	39.57	49.50	51.99
Pre+Mean-Centroid		26.31	30.41	34.68	47.16	56.40	61.57
Pre+Linear		25.97	31.32	35.49	**48.11**	**59.31**	**66.48**
UMTRA-ProtoNet	✓	24.94	28.04	29.88	39.21	44.62	46.48
UMTRA-ProtoTune	✓	25.00	30.41	35.63	38.47	51.60	60.12
ProtoTransfer (ours)	✓	**26.71**	**33.82**	**39.35**	45.19	59.07	66.15
			EuroSat			**CropDiseases**	
ProtoNet		73.29	82.27	80.48	79.72	88.15	90.81
Pre+Mean-Centroid		**82.21**	87.62	88.24	87.61	93.87	94.77
Pre+Linear		79.08	**87.64**	**91.34**	**89.25**	**95.51**	**97.68**
UMTRA-ProtoNet	✓	74.91	80.42	82.24	79.81	86.84	88.44
UMTRA-ProtoTune	✓	68.11	81.56	85.05	82.67	92.04	95.46
ProtoTransfer (ours)	✓	75.62	86.80	90.46	86.53	95.06	97.01

Table 3: Accuracy (%) of methods on N-way K-shot classification tasks on Mini-ImageNet with a Conv-4 architecture for different batch sizes, number of queries (Q) and optional finetuning (FT). Detailed results in Appendix Table 8. Results style: **best** and second best.

Training	**Testing**	**batch size**	**Q**	**FT**	**(5,1)**	**(5,5)**	**(5,20)**	**(5,50)**
n.a.	ProtoNet	n.a.	n.a.	no	27.05	34.12	39.68	41.40
UMTRA	MAML	$N(=5)$	1	yes	39.93	50.73	61.11	67.15
UMTRA	ProtoNet	$N(=5)$	1	no	39.17	53.78	62.41	64.40
ProtoCLR	ProtoNet	50	1	no	44.53	62.88	70.86	73.93
ProtoCLR	ProtoNet	50	3	no	44.89	**63.35**	72.27	74.31
ProtoCLR	ProtoTune	50	3	yes	**45.67**	62.99	**72.34**	**77.22**

3.3 Ablation Study: Batch Size, Number of Queries and Fine-Tuning

We conduct an ablation study of ProtoTransfer's components in Table 3. Starting from ProtoTransfer we successively remove components to arrive at the equivalent UMTRA-ProtoNet which shows similar performance to the original UMTRA approach (Khodadadeh et al., 2019) on mini-ImageNet. Importantly, UMTRA-ProtoNet uses our own augmentations. Our approach benefits most from larger batch sizes, increasing the number of queries to 3 and fine-tuning on the target domain. Increasing the batch size beyond 50 or the number of query images beyond 3 did not further improve the classification accuracy.

4. Related Work

Within unsupervised meta-learning, both CACTUs (Hsu et al., 2019) and UFLST (Ji et al., 2019) alternate between models for clustering for support and query set generation and employing standard meta-learning. In contrast, our method unifies self-supervised clustering and inference in a single model. Khodadadeh et al. (2019) propose an unsupervised model-agnostic meta-learning approach (UMTRA), where artifical N-way 1-shot tasks are generated by randomly sampling N support examples from the training set and generating N corresponding queries by augmentation. Antoniou and Storkey (2019) (AAL) generalize this approach to more support shots by randomly grouping augmented images into classes for classification tasks. ULDA (Qin et al., 2020) induce a distribution shift between the support and query set by applying different types of augmentations to each.

ProtoTransfer uses an un-augmented support sample, similar to Khodadadeh et al. (2019), but extends to several query samples for better gradient signals and steps away from artificial few-shot task sampling by using larger batch sizes, which is key to learning stronger embeddings. Several works exploit self-supervision either as an auxiliary self-supervised loss alongside the supervised meta-learning episodes (Gidaris et al., 2019; Liu et al., 2019) or to initialize a model prior to supervised meta-learning on the source domain (Chen et al., 2019; Su et al., 2019). In contrast, we do not require labels during training.

Similar to ProtoTune, Triantafillou et al. (2020) also initialize a final layer with prototypes after supervised meta-learning, but always fine-tune all parameters of the model.

Several recent works use noise contrastive losses to learn embedding functions (Ye et al., 2019; Chen et al., 2020; He et al., 2019). Most similar to our approach is Ye et al. (2019). They propose a per-batch contrastive loss that minimizes the distance between an image and an augmented version of it. Different to us, they do not generalize to using multiple augmented query images per prototype, use 2 extra fully connected layers during training, and use cosine instead of Euclidean distance. Concurrently, Li et al. (2020) also use a prototype-based contrastive loss. In contrast, they compute the prototypes as centroids after clustering augmented images via k-Means. They also separate learning and clustering steps, which ProtoTransfer achieves in a single step.

5. Conclusion

In this work, we proposed ProtoTransfer for few-shot classification. ProtoTransfer performs transfer learning from an unlabeled source domain to a target domain with only a few labeled examples. Our experiments show that on mini-ImageNet it outperforms all prior unsupervised few-shot learning approaches by a large margin. On a more challenging cross-domain few-shot classification benchmark, ProtoTransfer shows similar performance to fully supervised approaches. Our ablation studies show that large batch sizes are crucial to learning good representations for downstream few-shot classification tasks and that parametric fine-tuning on target tasks can significantly boost performance.

Acknowledgements

This work received support from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 754354.

References

Antreas Antoniou and Amos Storkey. Assume, Augment and Learn: Unsupervised Few-Shot Meta-Learning via Random Labels and Data Augmentation. *arXiv preprint arXiv:1902.09884*, 2019.

Da Chen, Yuefeng Chen, Yuhong Li, Feng Mao, Yuan He, and Hui Xue. Self-Supervised Learning For Few-Shot Image Classification. *arXiv preprint arXiv:1911.06045*, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv preprint arXiv:2002.05709*, 2020.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A Closer Look at Few-shot Classification. In *ICLR 2019 : 7th International Conference on Learning Representations*, 2019.

Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC). *arXiv preprint arXiv:1902.03368*, 2019.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019.

Terrance DeVries and Graham W Taylor. Improved Regularization of Convolutional Neural Networks With Cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Li Fei-Fei, Rob Fergus, and Pietro Perona. One-Shot Learning of Object Categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

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. JMLR. org, 2017.

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.

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. *arXiv preprint arXiv:1912.07200*, 2019.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. *arXiv preprint arXiv:1911.05722*, 2019.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12 (7):2217–2226, 2019.

Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised Learning via Meta-Learning. In *ICLR 2019 : 7th International Conference on Learning Representations*, 2019.

Zilong Ji, Xiaolong Zou, Tiejun Huang, and Si Wu. Unsupervised Few-shot Learning via Self-supervised Training. *arXiv preprint arXiv:1912.12178*, 2019.

Siavash Khodadadeh, Ladislau Boloni, and Mubarak Shah. Unsupervised Meta-Learning for Few-Shot Image Classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10132–10142, 2019.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In *ICLR 2015 : International Conference on Learning Representations 2015*, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification With Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1097–1105, 2012.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One-shot Learning of Simple Visual Concepts. *Cognitive Science*, 33(33), 2011.

Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical Contrastive Learning of Unsupervised Representations. *arXiv preprint arXiv:2005.04966*, 2020.

Shikun Liu, Andrew Davison, and Edward Johns. Self-Supervised Generalisation with Meta-Auxiliary Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1677–1687, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 7:1419, 2016.

Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3235–3246, 2018.

Tiexin Qin, Wenbin Li, Yinghuan Shi, and Yang Gao. Unsupervised Few-shot Learning via Distribution Shift-based Augmentation. *arXiv preprint arXiv:2004.05805*, 2020.

Sachin Ravi and Hugo Larochelle. Optimization as a Model for Few-Shot Learning. In *ICLR 2017 : International Conference on Learning Representations 2017*, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical Networks for Few-Shot Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4077–4087, 2017.

Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. When Does Self-Supervision Improve Few-Shot Learning? *arXiv preprint arXiv:1910.03560*, 2019.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples. In *ICLR 2020 : Eighth International Conference on Learning Representations*, 2020.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 Dataset, a Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions. *Scientific data*, 5:180161, 2018.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, and Daan Wierstra. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3630–3638, 2016.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2097–2106, 2017.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised Embedding Learning via Invariant and Spreading Instance Feature. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6210–6219, 2019.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random Erasing Data Augmentation. In *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Appendix A. Task Generalization Gap

To compare the generalization of ProtoCLR with its supervised embedding learning counterpart ProtoNet (Snell et al., 2017), we visualize the learned embedding spaces with t-SNE (Maaten and Hinton, 2008) in Figure 2. We compare both methods on samples from 5 random classes from the training and testing sets of mini-ImageNet. In Figures 2a and 2b we observe that, for the same training classes, ProtoNet shows more structure. Comparing all subfigures in Figure 2, ProtoCLR shows more closely related embeddings in Figures 2a and 2c than ProtoNet in Figures 2b and 2d.

The classes in the t-SNE plots are a random subset of classes from the mini-ImageNet base classes (classes 1-5) and the mini-ImageNet novel classes (classes 6-10). Their corresponding labels are the following:

1. n02687172 aircraft carrier
2. n04251144 snorkel
3. n02823428 beer bottle
4. n03676483 lipstick
5. n03400231 frying pan
6. n03272010 electric guitar
7. n07613480 trifle
8. n03775546 mixing bowl
9. n03127925 crate
10. n04146614 school bus

Each of the t-SNE plots in Figure 2 shows 500 randomly selected embedded images from within those classes.

These visual observations are supported numerically in Table 4. Self-supervised embedding approaches, such as UMTRA and our ProtoCLR approach, show a much smaller task generalization gap than supervised ProtoNet. ProtoCLR shows virtually no classification performance drop. However, supervised ProtoNet suffers a significant accuracy reduction of 6% to 12%.

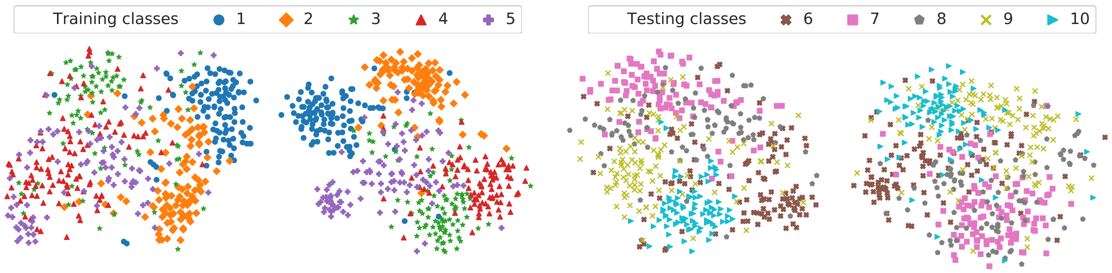
Appendix B. Experimental Details

B.1 Datasets

B.1.1 In-domain experiments

For our in-domain experiments we used the popular few-shot datasets Omniglot (Lake et al., 2011) and mini-ImageNet (Vinyals et al., 2016).

Omniglot consists of 1623 handwritten characters from 50 alphabets and 20 examples per character. Identical to Vinyals et al. (2016) the grayscale images are resized to 28x28.



(a) ProtoCLR training (b) ProtoNet training (c) ProtoCLR testing (d) ProtoNet testing

Figure 2: t-SNE plots of trained embeddings on 5 classes from the training and testing sets of mini-ImageNet. Trained embeddings considered are self-supervised ProtoCLR and supervised 20-way 5-shot ProtoNet.

Table 4: Accuracy (%) of N-way K-shot (N,K) classification tasks from the training and testing split of mini-ImageNet. Following Snell et al. (2017), ProtoNet is trained with 30-way 1-shot for 1-shot tasks and 20-way K-shot otherwise. All results use a Conv-4 architecture. All results show 95% confidence intervals over 600 randomly generated episodes.

Training	**Testing**	**Data**	**(5,1)**	**(5,5)**	**(5,20)**	**(5,50)**
ProtoNet	ProtoNet	Train	53.74 ± 0.95	79.09 ± 0.69	85.53 ± 0.53	86.62 ± 0.48
ProtoNet	ProtoNet	Val	46.62 ± 0.82	67.34 ± 0.69	76.44 ± 0.57	79.00 ± 0.53
ProtoNet	ProtoNet	Test	46.44 ± 0.78	66.33 ± 0.68	76.73 ± 0.54	78.91 ± 0.57
UMTRA	ProtoNet	Train	41.03 ± 0.79	56.43 ± 0.78	64.48 ± 0.71	66.28 ± 0.66
UMTRA	ProtoNet	Test	38.92 ± 0.69	53.37 ± 0.68	61.69 ± 0.66	65.12 ± 0.59
ProtoCLR	ProtoNet	Train	45.33 ± 0.63	63.47 ± 0.58	71.51 ± 0.51	73.99 ± 0.49
ProtoCLR	ProtoNet	Test	44.89 ± 0.58	63.35 ± 0.54	72.27 ± 0.45	74.31 ± 0.45

1028 characters are used for training and 423 for testing. We do not use the 172 validation set classes.

Mini-ImageNet is a subset of the ILSVRC-12 dataset (Russakovsky et al., 2015), which contains 60,000 color images that we resized to 84x84. For comparability we use the splits introduced by Ravi and Larochelle (2017) over 100 classes with 600 images each. 64 classes are used for pre-training and 20 for testing. We do not use the 16 validation set classes.

B.1.2 Cross-domain experiments

We evaluate all cross-domain experiments the CDFSL-benchmark (Guo et al., 2019). It comprises four datasets with decreasing similarity to mini-ImageNet. In order of similarity, they are plant disease images from CropDiseases (Mohanty et al., 2016), satellite images from EuroSAT (Helber et al., 2019), dermatological images from ISIC2018 (Tschandl et al., 2018; Codella et al., 2019) and grayscale chest x-ray images from ChestX (Wang et al., 2017). Training was performed on the mini-ImageNet dataset as described above with an image size of 224x224 to be comparable with the CDFSL benchmark results.

B.2 Architecture and Optimization Parameters

In the following, we describe the experimental details for the individual experiments. We deliberately stay close to the parameters reported in prior work and do not perform an extensive hyperparameter search for our specific setup, as this can easily lead to performance overestimation compared to simpler approaches (Oliver et al., 2018)).

B.2.1 In-Domain Experiments

Our mini-ImageNet and Omniglot experiments use the Conv4 architecture proposed in Vinyals et al. (2016) for comparability. Its four convolutional blocks each apply a 64-filters 3x3 convolution, batch normalization, a ReLU nonlinearity and 2x2 max-pooling. The pre-training mostly mirrors Snell et al. (2017) and uses Adam (Kingma and Ba, 2015) with an initial learning rate of 0.001, which is multiplied by a factor of 0.5 every 25000 iterations. Training stops after 20.000 iterations without improvement in training loss. We use a batch size of 50. We do not use a validation set.

B.2.2 Cross-Domain Experiments

Our experiments on the CDFSL-Challenge are based on the code provided by Guo et al. (2019). Following Guo et al. (2019), we use a ResNet10 architecture that is pre-trained on mini-Imagenet for 400 epochs with Adam (Kingma and Ba, 2015) and the default learning rate of 0.001 for best comparability with the results reported in Guo et al. (2019). The batch size for self-supervised pre-training is 50. We do not use a validation set.

B.2.3 Prototypical Fine-Tuning

During the fine-tuning stage we add a fully connected classification layer after the embedding function and initialize as described in Section 2.2. We split the support examples into batches of 5 images each and perform 15 fine-tuning epochs with Adam (Kingma and Ba, 2015) and an initial learning rate of 0.001. For mini-ImageNet and Omniglot only the last fully connected layer is optimized, while for the CDFSL benachmark experiments the embedding network is adapted as well.

B.3 Augmentations

CDFSL transforms For the CDFSL-benchmark (Guo et al., 2019) experiments we employ the same augmentations as Chen et al. (2020), as these have proven to work well for ImageNet (Russakovsky et al., 2015) images of size 224x224. They are as follows:

1. Random crop and resize: `scale` $\in [0.08, 1.0]$, `aspect ratio` $\in [3/4, 4/3]$, Bilinear filter with `interpolation = 2`
2. Random horizontal flip
3. Random ($p = 0.8$) color jitter: `brightness = contrast = saturation = 0.8`, `hue=0.2`
4. Random ($p = 0.2$) grayscale
5. Gaussian blur, random radius $\sigma \in [0.1, 0.2]$

mini-ImageNet transforms For the mini-Imagenet experiments we used lighter versions of the Chen et al. (2020) augmentations, namely no Gaussian blur, lower color jitter strengths and smaller rescaling and cropping ranges. They are as follows:

1. * Random crop and resize: `scale` $\in [\mathbf{0.5}, 1.0]$, `aspect ratio` $\in [3/4, 4/3]$, Bilinear filter with `interpolation = 2`
2. Random horizontal flip
3. * Random vertical flip
4. * Random ($p = 0.8$) color jitter: `brightness = contrast = saturation = 0.4`, `hue=0.2`
5. Random ($p = 0.2$) grayscale

Omniglot transforms For Omniglot we use a set of custom augmentations, namely random resizing and cropping, horizontal and vertical flipping, Image-Pixel Dropout (Krizhevsky et al., 2012) and Cutout (DeVries and Taylor, 2017). They are as follows:

1. Resize to a size of 28x28 pixels
2. Random and resize: `scale` $\in [0.6, 1.4]$, `aspect ratio` $\in [3/4, 4/3]$, Bilinear filter with `interpolation = 2`
3. Random horizontal flip
4. Random vertical flip
5. Random ($p = 0.3$) dropout
6. Random erasing of a rectangular region in an image (Zhong et al., 2020), setting pixel values to `0`: `scale` $\in [0.02, 0.33]$, `aspect ratio` $\in [0.3, 3.3]$

B.4 Results With Full Confidence Intervals & References Validation Set

Table 5: Accuracy (%) of methods on N-way K-shot classification tasks on Omniglot and a Conv-4 architecture. Results style: **best** and <u>second best</u>.

Method (N,K)	**(5,1)**	**(5,5)**	**(20,1)**	**(20,5)**
		Omniglot		
Training (scratch)	52.50 ± 0.84	74.78 ± 0.69	24.91 ± 0.33	47.62 ± 0.44
CACTUs-MAML[1]	68.84 ± 0.80	87.78 ± 0.50	48.09 ± 0.41	73.36 ± 0.34
CACTUs-ProtoNets[1]	68.12 ± 0.84	83.58 ± 0.61	47.75 ± 0.43	66.27 ± 0.37
UMTRA[2]	83.80 ± -	95.43 ± -	**74.25** ± -	**92.12** ± -
AAL-ProtoNets[3]	84.66 ± 0.70	89.14 ± 0.27	68.79 ± 1.03	74.28 ± 0.46
AAL-MAML++[3]	<u>88.40</u> ± 0.75	**97.96** ± 0.32	70.21 ± 0.86	88.32 ± 1.22
ProtoTransfer (ours)	**88.95** ± 0.49	<u>96.67</u> ± 0.20	<u>73.13</u> ± 0.36	<u>89.46</u> ± 0.18
Supervised training				
MAML[1]	46.81 ± 0.77	62.13 ± 0.72	71.03 ± 0.69	75.54 ± 0.62
ProtoNet	46.44 ± 0.78	66.33 ± 0.68	76.73 ± 0.54	78.91 ± 0.57
Pre+Linear	43.87 ± 0.69	63.01 ± 0.71	75.46 ± 0.58	80.17 ± 0.51

[1] Hsu et al. (2019)

[2] Khodadadeh et al. (2019)

[3] Antoniou and Storkey (2019)

Table 6: Accuracy (%) of methods on N-way K-shot classification tasks mini-Imagenet and a Conv-4 architecture. Results style: **best** and second best.

Method (N,K)	**(5,1)**	**(5,5)**	**(5,20)**	**(5,50)**
		Mini-ImageNet		
Training (scratch)	27.59 ± 0.59	38.48 ± 0.66	51.53 ± 0.72	59.63 ± 0.74
CACTUs-MAML[1]	39.90 ± 0.74	53.97 ± 0.70	63.84 ± 0.70	69.64 ± 0.63
CACTUs-ProtoNets[1]	39.18 ± 0.71	53.36 ± 0.70	61.54 ± 0.68	63.55 ± 0.64
UMTRA[2]	39.93 ± -	50.73 ± -	61.11 ± -	67.15 ± -
AAL-ProtoNets[3]	37.67 ± 0.39	40.29 ± 0.68	-	-
AAL-MAML++[3]	34.57 ± 0.74	49.18 ± 0.47	-	-
ULDA-ProtoNets[4]	40.63 ± 0.61	55.41 ± 0.57	63.16 ± 0.51	65.20 ± 0.50
ULDA-MetaOptNet[4]	40.71 ± 0.62	54.49 ± 0.58	63.58 ± 0.51	67.65 ± 0.48
ProtoTransfer (ours)	**45.67** ± 0.79	**62.99** ± 0.75	**72.34** ± 0.58	**77.22** ± 0.52
Supervised training				
MAML[1]	46.81 ± 0.77	62.13 ± 0.72	71.03 ± 0.69	75.54 ± 0.62
ProtoNet	46.44 ± 0.78	66.33 ± 0.68	76.73 ± 0.54	78.91 ± 0.57
Pre+Linear	43.87 ± 0.69	63.01 ± 0.71	75.46 ± 0.58	80.17 ± 0.51

[1] Hsu et al. (2019)
[2] Khodadadeh et al. (2019)
[3] Antoniou and Storkey (2019)
[4] Qin et al. (2020)

Table 7: Accuracy (%) of methods on N-way K-shot classification tasks of the CDFSL benchmark (Guo et al., 2019). All models are trained on mini-ImageNet with ResNet-10. Results style: **best** and second best.

Method	**UnSup**	**(5,5)**	**(5,20)**	**(5,50)**	**(5,5)**	**(5,20)**	**(5,50)**
			ChestX			**ISIC**	
ProtoNet[*]		24.05 ± 1.01	28.21 ± 1.15	29.32 ± 1.12	39.57 ± 0.57	49.50 ± 0.55	51.99 ± 0.52
Pre+Mean-Centroid[*]		26.31 ± 0.42	30.41 ± 0.46	34.68 ± 0.46	47.16 ± 0.54	56.40 ± 0.53	61.57 ± 0.66
Pre+Linear[*]		25.97 ± 0.41	31.32 ± 0.45	35.49 ± 0.45	**48.11** ± 0.64	**59.31** ± 0.48	**66.48** ± 0.56
UMTRA-ProtoNet	✓	24.94 ± 0.43	28.04 ± 0.44	29.88 ± 0.43	39.21 ± 0.53	44.62 ± 0.49	46.48 ± 0.47
UMTRA-ProtoTune	✓	25.00 ± 0.43	30.41 ± 0.44	35.63 ± 0.48	38.47 ± 0.55	51.60 ± 0.54	60.12 ± 0.50
ProtoTransfer (ours)	✓	**26.71** ± 0.46	**33.82** ± 0.48	**39.35** ± 0.50	45.19 ± 0.56	59.07 ± 0.55	66.15 ± 0.57
			EuroSat			**CropDiseases**	
ProtoNet[*]		73.29 ± 0.71	82.27 ± 0.57	80.48 ± 0.57	79.72 ± 0.67	88.15 ± 0.51	90.81 ± 0.43
Pre+Mean-Centroid[*]		**82.21** ± 0.49	87.62 ± 0.34	88.24 ± 0.29	87.61 ± 0.47	93.87 ± 0.68	94.77 ± 0.34
Pre+Linear[*]		79.08 ± 0.61	**87.64** ± 0.47	**91.34** ± 0.37	**89.25** ± 0.51	**95.51** ± 0.31	**97.68** ± 0.21
UMTRA-ProtoNet	✓	74.91 ± 0.72	80.42 ± 0.66	82.24 ± 0.61	79.81 ± 0.65	86.84 ± 0.50	88.44 ± 0.46
UMTRA-ProtoTune	✓	68.11 ± 0.70	81.56 ± 0.54	85.05 ± 0.50	82.67 ± 0.60	92.04 ± 0.43	95.46 ± 0.31
ProtoTransfer (ours)	✓	75.62 ± 0.67	86.80 ± 0.42	90.46 ± 0.37	86.53 ± 0.56	95.06 ± 0.32	97.01 ± 0.26

[*] Results from Guo et al. (2019)

Table 8: Accuracy (%) of methods on N-way K-shot (N, K) classification tasks on mini-ImageNet with a Conv-4 architecture for different training image batch sizes, number of training queries (Q) and optional finetuning on target tasks (FT). UMTRA-MAML results are taken from Khodadadeh et al. (2019), where UMTRA uses AutoAugment (Cubuk et al., 2019) augmentations. All results are reported with 95% confidence intervals over 600 randomly generated test episodes. Results style: **best** and <u>second best</u>.

Training	**Testing**	**batch size**	**Q**	**FT**	**(5,1)**	**(5,5)**	**(5,20)**	**(5,50)**
n.a.	ProtoNet	n.a.	n.a.	no	27.05 ± 0.56	34.12 ± 0.59	39.68 ± 0.59	41.40 ± 0.59
UMTRA*	MAML	5	1	yes	39.93 ± -	50.73 ± -	61.11 ± -	67.15 ± -
UMTRA	ProtoNet	5	1	no	39.17 ± 0.53	53.78 ± 0.53	62.41 ± 0.49	64.40 ± 0.46
ProtoCLR	ProtoNet	50	1	no	44.53 ± 0.60	62.88 ± 0.54	70.86 ± 0.48	73.93 ± 0.44
ProtoCLR	ProtoNet	50	3	no	44.89 ± 0.58	**63.35** ± 0.54	<u>72.27</u> ± 0.45	<u>74.31</u> ± 0.45
ProtoCLR	ProtoNet	50	5	no	<u>45.00</u> ± 0.57	63.17 ± 0.55	71.70 ± 0.48	73.98 ± 0.44
ProtoCLR	ProtoNet	50	10	no	44.98 ± 0.58	62.56 ± 0.53	70.78 ± 0.48	73.69 ± 0.44
ProtoCLR	ProtoTune	50	3	yes	**45.67** ± 0.76	<u>62.99</u> ± 0.75	**72.34** ± 0.58	**77.22** ± 0.52

* Khodadadeh et al. (2019)