Cross-domain few-shot classification through feature confusion

Guoxiang Ye 1a, Yan Xia lb, Zhangwei Feng 1 and Feng Tian 1

1Chengdu Aircraft Industry (Group) Co., Ltd., Chengdu, China
a cacygx@gmail.com, byanxiahit@163.com

Abstract. Few-shot classification has made great progress in recent years which aims to solve new tasks using the prior knowledge obtained from a set of similar tasks with few labeled examples. However, the performance of current methods drops when domain shift exists between training and testing samples. Considering the situation where target domain examples are hard to collect, it is necessary to focus on the cross-domain problem. In this paper, we propose a simple method which embeds the domain adaptation method into the few-shot problem under a novel cross-domain few-shot setting, and focus on supervised setting in the target domain where few labeled data available for domain adaptation. Extensive empirical evidence shows the effectiveness of our method.

1. Introduction
Deep learning has made great success in image classification since 2012 which partly depends on the large amount of labeled data[1–5]. However, in many applications, it is hard to collect labeled or even unlabeled data for specific tasks. To tackle this problem, a series of problems and methods have been studied by researchers, such as few-shot learning[6–12], domain adaptation[13–21] and teach-student methods[22–24].

Focusing on training on a set of tasks with limited labeled data to acquire prior knowledge which can be used in novel tasks, few-shot learning received more and more attention in the past few years and performs quit well under n-way k-shot paradigm to solve few-shot problems. However, these few-shot learning methods are applied under the assumption that the testing data and training data are sampled from the same distribution. The assumption is quite strong in practice since the domain shift exists in many applications. Chen[26] evaluates the performance when domain shift exists between novel tasks and training tasks and show the limitation of current few-shot learning methods.

Current domain adaptation methods focus on training a classifier using labeled source domain data and unlabeled target domain data which performs well in the target domain. It highly relies on the large amount of labeled source data and unlabeled target data which makes it less useful in the circumstance of few-shot classification.

Inspired by the fancy methods for few-shot problem and domain adaptation methods for cross-domain problem, we propose a simple method which is based on ProtoNet[17] to prove that learning domain invariant features can improve the performance of meta-learning methods under cross domain circumstance. More specifically, we focus on the problem where sufficient labeled source domain examples and few target domain examples are available, and remove the constraints that testing classes and training classes are mutually exclusive. The experiments are conducted under supervised settings where a small number of examples can be used for domain adaptation.
The main contributions of this paper are presented as follows: 1) A simple method to solve cross-domain few-shot problem is proposed which proves domain invariant feature is beneficial for target domain tasks. 2) A novel setting is proposed that target domain classes and source domain classes can be identical, which indicates the proposed method can be applied to ‘synthetic to real’ problem where real examples are limited.

2. Related work
Our work mainly based on the idea from few-shot learning and domain adaptation, following is a brief introduction.

2.1. Few-shot learning
In the real world, relevant objects are continuously replaced by new ones, and it takes a lot of time or money to collect data. Few-shot learning aims to solve the problem by training on a set of tasks with few labeled examples which can obtain prior knowledge for other tasks. It usually follows n-way k-shot paradigm under the assumption that the testing tasks and training tasks are drawn from the same distribution. Roughly, three kind of methods are mainly studied: 1) Initialize based methods address the few-shot problem by adapting to new tasks quickly. Finn[15] and Li[16] see the problem as learning a good initialization so it can be adapted to new tasks with a small number of steps. 2) Metric based methods try to find a latent similarity space which can recognize different instances. Vinyals[14], Snell[17] and Sung[19] dive into this area based on different distance metric. 3) Augmentation based methods[20-21] try to learn a generator which plays a role as data augmentation in the novel task.

2.2. Domain adaptation
In the deep era, researchers mostly focus on three kinds of methods: 1) Network-based methods reuse the knowledge obtained from the source domain, such as network structure and parameters, to solve the task in the target domain. Fine-tune maybe the most commonly used method among them. 2) Metric-based methods try to find a uniform latent space to represent the instances both from the source domain and the target domain. Tzeng[6] and Long[7,9] focus on confusing source data and target data in the latent space. Bousmalis[8] emphasizes the importance of explicitly modelling the unique feature of each domain and aims to solve the synthetic to real problem. 3) Adversarial-based methods[10,11] embed adversarial learning for learning more transferable feature representations.

3. Few-shot domain adaptation
In this section, we present the method we proposed as well as related notations and definitions.

3.1. Notations
Some notations used in this paper are introduced which mainly follow the setting as Pan[25] and Chen[17] adopt.

In few-shot learning, we follow the n-way k-shot paradigm where n-way means n classes for a task and k-shot means k samples per class for training. At each epoch, multi-episodes are sampled for training. For ProtoNet, a small support set of k labeled examples $S = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ can be used to compute prototype for the corresponding class during training. $S_k$ denotes the set of examples labeled with class k.

In domain adaptation, a domain $D = \{\mathcal{X}, P(\mathcal{X})\}$ consists of two part: a feature space $\mathcal{X}$ and a marginal probability distribution $P(\mathcal{X})$, where $\mathcal{X} = (x_1, \ldots, x_n) \in \mathcal{X}$. A task $T = \{y, f(\mathcal{X})\}$ consists of two parts: a label space $\mathcal{Y}$ and an objective function $f(\mathcal{X})$ which can also be viewed as a conditional probability function $P(y|\mathcal{X})$.

3.2. Problem formulation
Given a source domain $D_s$ where sufficient labeled examples available, and a target domain $D_t$ where
just few labeled examples available. Few-shot domain adaptation aims to learn a meta-learner which has a good performance in tasks \(\{T^1, ..., T^m\}\) from the target domain by learning transferable prior knowledge from source domain tasks \(\{T^1_s, ..., T^n_s\}\). In addition, we remove the constraints that testing classes and training classes are mutually exclusive.

3.3. Method

The method used in this paper adapts the meta-learner to target domain with the help of few target domain labeled data for domain adaptation. Training episodes are formed by randomly selecting a subset of classes from the \(Ds\) and \(Dt\), then choosing a subset from \(Ds\) as support set and a subset of the remainder from \(Ds\) as query set. Under the supervised setting, data collected for training of each episode come from the same class. Our model architecture is shown as Figure.1.

![Model architecture for few-shot domain adaptation](image)

### Figure.1 Model architecture for few-shot domain adaptation

In this architecture, the loss function consists of two parts: \(L_{\text{domain}}\) is used for minimizing the distance between the source and target distribution in the latent space, and \(L_{\text{proto}}\) is used for minimizing the distance between prototypes and query representation in latent space.

Given a representation \(f_\phi: x \rightarrow \mathbb{R}^d\), the distance between source data points and target data points can be computed as followed:

\[
L_{\text{domain}} = \left\| \frac{1}{|X^s|} \sum_{x \in X^s} f_\phi(x) - \frac{1}{|X^t|} \sum_{x \in X^t} f_\phi(x) \right\|^2
\]

(1)

For \(L_{\text{proto}}\), we follow the setting in ProtoNet[17] as well, where \(c_k\) represent the prototype of class \(k\), \(d(\cdot)\) represents Euclidean distance function to compute distance between prototype and query points.

\[
L_{\text{proto}} = -\log P_\phi(y = k|x)
\]

(2)

\[
P_\phi(y = k|x) = \frac{\exp(-d(f_\phi(x),c_k))}{\sum_{k'} \exp(-d(f_\phi(x),c_{k'}))}
\]

(3)

\[
c_k = \frac{1}{|S_k|} \sum_{(x_i,y_i) \in S_k} f_\phi(x_i)
\]

(4)

Through combining minimize loss items above, we can obtain prior knowledge that can be used in the target domain.

\[
L = L_{\text{proto}} + \lambda L_{\text{domain}}
\]

Where \(\lambda\) determines how strongly we would like to confuse the domains.
4. Experiments

4.1. Setup

Office-Home is widely used for visual domain adaptation, which consists of images from 4 different domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr) and Real World (Rw). It contains around 15500 images from 65 different categories. In this paper, we take Ar and Cl as source domain, Pr and Rw as target domain. Meanwhile, we split the dataset into 3 parts, which include 25 classes (base set), 20 classes (validation set) and 20 classes (novel set) of objects respectively.

VisDA-2017 is a synthetic-to-real dataset for visual domain adaptation which consists of two domains: Synthetic(Syn) and Real World(Rw). It contains over 280k images from 12 different classes for the classification task. In this paper, we form a subset of this dataset by randomly picking 1000 images per class from the source domain(Syn) and 500 images from the target domain(Real). We call the subset as Mini-VisDA.

We compare our method with ProtoNet in the n-way k-shot paradigm under both setting, where n is set to 5 in all experiments. Additionally, we form our method in a n-way k-shot d-adapt paradigm, where d-da means d examples for domain adaptation. The implementation details in this paper is followed as Chen[26] adopted. For each class, we pick k labeled examples as support set and 16 examples as query set at each episode. Both methods take a conv-4 model as backbone which contains four convolutional layers. The input image size is set to 84x84. The parameter $\lambda$ is computed by $\frac{1-e^{-10(\text{epoch})/\text{max _epoch}}}{1+e^{-10(\text{epoch})/\text{max _epoch}}}$, where epoch represents the current training epoch, max _epoch means the total number of training epochs. All experiments are trained from scratch using Adam optimizer with learning rate $10^{-3}$.

4.2. Results

The experiments are conducted under the supervised setting which only use a small number of labeled examples from the target domain for domain adaptation. The support set and query set for training are both randomly collected from the source domain.

4.2.1. Office-Home

The support set and query set for training are picked from source domain base classes and the domain adaptation set is picked from the target domain base classes for the experiments conducted on Office-Home dataset. The examples in target validation set are used for choosing the best model. We set k as 5 for both methods and d as 5 for our method. We evaluate the performance both on base set and novel set in the target domain. The results on Office-Home are reported in Table 1 and Table 2.

| Table 1. Accuracy(%) evaluated on target domain base set of Office-Home. |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | Ar->Pr          | Ar->Rw          | Cl->Pr          | Cl->Rw          | Avg             |
| ProtoNet        | 60.38           | 55.43           | 64.31           | 57.18           | 59.33           |
| Ours            | 61.69           | 55.80           | 65.96           | 57.78           | 60.31           |

| Table 2. Accuracy(%) evaluated on target domain novel set of Office-Home. |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | Ar->Pr          | Ar->Rw          | Cl->Pr          | Cl->Rw          | Avg             |
| ProtoNet        | 52.64           | 41.02           | 59.30           | 47.36           | 50.08           |
| Ours            | 50.89           | 39.72           | 59.70           | 46.03           | 49.09           |

Our method outperforms ProtoNet on target domain base set where testing classes are the same as classes used in the meta-training stage, but performs no better on target novel set where testing classes are unseen during training. It is reasonable as we explicitly minimize the distance between source domain points and target domain points in the latent space at the class level, which makes the prior knowledge biased to the adaptation class.
4.2.2. Mini-VisDA
We only consider one circumstance where synthetic data as source domain and real data as target domain for Mini-VisDA dataset. Hence, the support set and query set are collected from the source domain and the domain adaptation set are collected from the target domain. In this paper, three settings of the number of training examples are considered: 5-shot 5-adapt, 10-shot 10-adapt and 15-shot 15-adapt. The results are reported in Table 3.

|                      | 5-shot 5-DA | 10-shot 10-DA | 15-shot 15-DA |
|----------------------|-------------|---------------|--------------|
| **ProtoNet**         | 42.47 ± 0.65| 46.27 ± 0.65  | 50.52 ± 0.64 |
| **Ours**             | 44.02 ± 0.61| 49.64 ± 0.56  | 52.56 ± 0.59 |

Our method outperforms ProtoNet on Mini-VisDA under all settings as on Office-Home. Both methods can get better performance when support set and domain adaptation set get bigger for more information being extracted with more data.

5. Conclusion
In this paper, we have proposed a simple method for cross-domain few-shot problem based on the idea that minimizing distance between source and domain points in the latent space can learn domain invariant representations for meta-learner. Specifically, we aim to solve the problem where only few labeled data available in the target domain for adaptation but sufficient labeled data available in the source domain for meta-training. Furthermore, our method can be used for ‘synthetic to real’ problem to solve real world problems with the help of simulation data. The extensive experiments on Office-Home and Mini-VisDA prove the effectiveness of our method. In this paper, we focus on the feasibility of confusing source domain points and target domain points in the latent space to solve supervised cross-domain few-shot problem. In order to tackle more real world problems, unsupervised setting where labeled data in the target domain unavailable should be considered, and we need to consider the problem where tasks from the target domain different from tasks from the source domain.

References
[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
[2] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
[3] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
[5] Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L. J., ... & Murphy, K. (2018). Progressive neural architecture search. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 19-34).
[6] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., & Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474.
[7] Long, M., Cao, Y., Wang, J., & Jordan, M. I. (2015). Learning transferable features with deep adaptation networks. arXiv preprint arXiv:1502.02791.
[8] Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., & Erhan, D. (2016). Domain separation networks. In Advances in neural information processing systems (pp. 343-351).
[9] Long, M., Zhu, H., Wang, J., & Jordan, M. I. (2017, August). Deep transfer learning with joint adaptation networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 2208-2217). JMLR. org.

[10] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The Journal of Machine Learning Research, 17(1), 2096-2030.

[11] Long, M., Cao, Z., Wang, J., & Jordan, M. I. (2018). Conditional adversarial domain adaptation. In Advances in Neural Information Processing Systems (pp. 1640-1650).

[12] Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., & Wang, B. (2019). Moment matching for multi-source domain adaptation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1406-1415).

[13] Huang, G., Larochelle, H., & Lacoste-Julien, S. (2019). Centroid networks for few-shot clustering and unsupervised few-shot classification. arXiv preprint arXiv:1902.08605.

[14] Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. In Advances in neural information processing systems (pp. 3630-3638).

[15] Finn, C., Abbeel, P., & Levine, S. (2017, August). 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.

[16] Li, Z., Zhou, F., Chen, F., & Li, H. (2017). Meta-sgd: Learning to learn quickly for few-shot learning. arXiv preprint arXiv:1707.09835.

[17] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In Advances in neural information processing systems (pp. 4077-4087).

[18] Gidaris, S., & Komodakis, N. (2018). Dynamic few-shot visual learning without forgetting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4367-4375).

[19] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1199-1208).

[20] Wang, Y. X., Girshick, R., Hebert, M., & Hariharan, B. (2018). Low-shot learning from imaginary data. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7278-7286).

[21] Gao, H., Shou, Z., Zareian, A., Zhang, H., & Chang, S. F. (2018). Low-shot learning via covariance-preserving adversarial augmentation networks. In Advances in Neural Information Processing Systems (pp. 975-985).

[22] Miyato, T., Maeda, S. I., Koyama, M., & Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE transactions on pattern analysis and machine intelligence, 41(8), 1979-1993.

[23] Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Advances in neural information processing systems (pp. 1195-1204).

[24] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 113-123).

[25] Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.

[26] Chen, W. Y., Liu, Y. C., Kira, Z., Wang, Y. C. F., & Huang, J. B. (2019). A closer look at few-shot classification. arXiv preprint arXiv:1904.04232.