Large-Scale Cross-Domain Few-Shot Learning

Jiechao Guan[1] , Manli Zhang[1], Zhiwu Lu[2] (✉)

[1] School of Information, Renmin University of China, Beijing, China
[2] Beijing Key Laboratory of Big Data Management and Analysis Methods, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
luzhiwu@ruc.edu.cn

Abstract. Learning classifiers for novel classes with a few training examples (shots) in a new domain is a practical problem setting. However, the two problems involved in this setting, few-shot learning (FSL) and domain adaption (DA), have only been studied separately so far. In this paper, for the first time, the problem of large-scale cross-domain few-shot learning is tackled. To overcome the dual challenges of few-shot and domain gap, we propose a novel Triplet Autoencoder (TriAE) model. The model aims to learn a latent subspace where not only transfer learning from the source classes to the novel classes occurs, but also domain alignment takes place. An efficient model optimization algorithm is formulated, followed by rigorous theoretical analysis. Extensive experiments on two large-scale cross-domain datasets show that our TriAE model outperforms the state-of-the-art FSL and domain adaptation models, as well as their naive combinations. Interestingly, under the conventional large-scale FSL setting, our TriAE model also outperforms existing FSL methods by significantly margins, indicating that domain gaps are universally present.

1 Introduction

Large-scale visual recognition has been the central focus of the computer vision research recently. It faces two major challenges: lack of sufficient training samples for each class and the significant domain gap between the training and test data. Both problems have long been identified, which leads to two extensively studied areas: few-shot learning (FSL) [1–3] and domain adaption (DA) [4, 5].

A FSL model is provided with a set of source classes and a set of target ones with no overlap in the label space between the two. Each source class has sufficient labeled samples, whereas each target class has only a few labeled samples. The goal of FSL is thus to learn transferable knowledge from the source classes, and develop a robust recognition model for recognizing novel/target object classes. Meta-learning based FSL methods [6–14] have shown state-of-the-art performance on several medium-scale benchmarks (e.g. miniImageNet [15]). Recently, large-scale FSL [16–18], which focuses on more challenging datasets such as the ImageNet 1K classes, starts to attract increasing attentions.

Different from FSL, domain adaptation (DA) [4, 5] aims to generalize a learned model to different domains but assumes that the class labels are shared

between the training and test domains. Recent DA methods focus on the unsupervised domain adaptation (UDA) setting, under which the target domain training data are unlabeled [19–24]. It is thus clear that both FSL and DA problems need to be solved by knowledge transfer. However, due to the different problem settings, they are largely treated as two separate problems.

In this paper, we argue that the two problems are actually closely intertwined, and thus should be solved jointly. In particular, existing large-scale FSL methods [16, 25, 17, 26, 27, 18] assume that both source and target classes come from the same domain. However, in real-world scenarios, the target novel classes are not only represented by a handful of examples, but also need to be recognized from a domain different from the domain of the source classes (see Figure 1). For example, it is assumed that an object recognizer is trained with the ImageNet 1K dataset and installed on a user's mobile phone. When the user comes across some new object categories during exploring a forest, he/she resorts to a FSL model to recognize the new classes. The domain gap between the images in ImageNet and the photos taken by her/his phone in the forest are clearly very different in style. Conventional FSL methods thus become inadequate because such domain changes between the source and target classes are not considered. We therefore define a new large-scale cross-domain FSL problem (see Figure 1). It is clearly more challenging than the FSL and DA problems on their own. Moreover, it is also noted that a naive combination of existing FSL and domain adaptation methods does not offer a valid solution (see Figure 4).

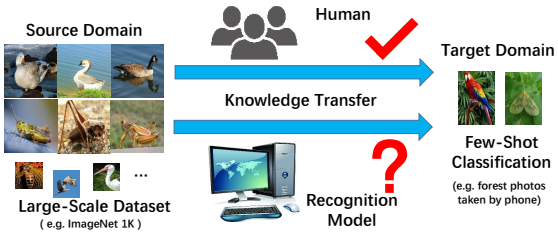


Fig. 1. Illustration of our proposed large-scale cross-domain few-shot learning setting.

To solve this challenging problem, we propose a novel Triplet Autoencoder (TriAE) model. As illustrated in Figure 2, it addresses both FSL and domain adaptation problems simultaneously by learning a joint latent subspace [28, 22, 29]. Intuitively, since class name semantic embedding is domain invariant, we utilize a semantic space (e.g., a word embedding space [30]) to learn the latent space. There are now three spaces: the visual feature space of the source domain, the visual feature space of the target domain, and the semantic space of both source and target classes. To construct the relationship among these three spaces, we choose to learn a shared latent subspace. Specifically, we leverage an encoder-decoder paradigm [31–34] between the semantic space and latent subspace for knowledge transfer (essential for FSL), and also learn encoder-decoder projections between the latent subspace and visual feature space. Domain alignment then takes place in the same latent subspace. The resultant model (see Figure 3) can be decomposed into a pair of dual autoencoders (one for modeling

source classes, and the other for target novel classes). We provide an efficient model optimization algorithm, followed by rigorous theoretical analysis. Extensive experiments on two large-scale cross-domain datasets show that our TriAE outperforms the state-of-the-art FSL and domain adaptation models. Importantly, it is noted that under the conventional large-scale FSL setting (i.e., without explicit domain gap), the proposed model also beats existing FSL models by significant margins. This indicates that the domain gap naturally exists in FSL as the source and target data contain non-overlapping classes, and needs to be bridged by a cross-domain FSL method.

Our contributions are summarized as follows: (1) We define a new large-scale cross-domain FSL setting, which is challenging yet common in real-world scenarios. (2) We propose a novel Triplet Autoencoder model, which seamlessly unifies FSL and domain adaptation into the same framework. (3) We provide an efficient model optimization algorithm for training the proposed TriAE model, followed by rigorous theoretical analysis. (4) The proposed TriAE model is shown to achieve the state-of-the-art performance on both new and conventional large-scale FSL problems. The code and dataset will be released soon.

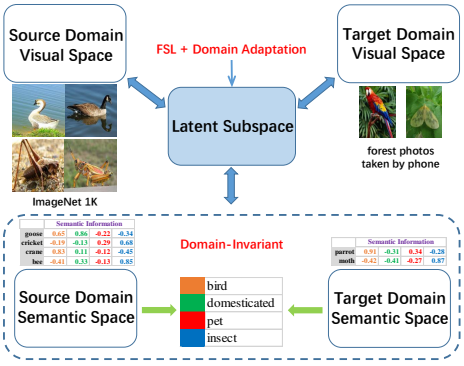


Fig. 2. Schematic of the proposed TriAE model for large-scale cross-domain FSL.

2 Related Work

Large-Scale FSL Meta-learning based approaches [6–14, 35–37] have achieved great success on small- or medium-scale FSL datasets. More recently, large-scale FSL [16–18, 25–27, 38] becomes topical. Squared Gradient Magnitude (SGM) [16] proposes a Feature Hallucination strategy to augment target classes' samples based on target class centroids. Covariance-Preserving Adversarial Augmentation Network (CP-AAN) [27] resorts to adversarial learning for covariance-preserving target feature hallucination. Instead of feature synthesis, Parameter Prediction from Activations (PPA) [17] aims to explore the relationship between the logistic regression parameters and activations in a deep neural network, and Large-Scale Diffusion (LSD) [26] constructs a large-scale semi-supervised graph over external data for label propagation. Knowledge Transfer with Class Hierarchy (KTCH) [18] constructs a tree-structured class hierarchy to formulate hierarchical classification loss functions for representation learning. However, existing large-scale FSL methods assume that both source and target classes come from the same domain. When they are deployed under the new cross-domain FSL setting, they are clearly beaten by our TriAE model (see Tables 1 and 2).

Note that although cross-domain dataset (i.e. miniImageNet → CUB [39]) is used for FSL in [40], it is just used to evaluate the cross-dataset performance of conventional FSL methods. In contrast, we consider a dataset where the domain gap is much bigger (e.g. natural images vs. cartoon-like ones) and develop a model to specifically tackle the problem.

Domain Adaptation Domain adaptation [4, 5] aims to generalize the learned model to different domains. To alleviate the domain shift, some domain adaptation methods find a shared latent space that both source and target domains can be mapped into [41–45, 28, 22], to ensure that in the shared space the learned model cannot distinguish whether a sample is from the source or target domain [46, 47, 20, 22, 21]. Recently, adversarial learning [19–24] has also shown promising results on domain adaptation. Among these domain adaptation models, Few-shot Adversarial Domain Adaptation (FADA) [22] takes on board the few-shot domain adaptation setting, which is most similar to our proposed large-scale cross-domain FSL setting. But there is a clear difference: FADA assumes that the source and target domains share the same set of object classes, whereas in our proposed setting, the source and target domains consist of two non-overlapped sets of object classes (i.e. source and target classes). Moreover, FADA focuses on few-shot domain adaptation over medium-scale datasets (e.g. Office [48]), while we construct two new large-scale datasets from ImageNet2012/2010 and ImageNet2012/DomainNet (for cross-domain FSL), which are both more challenging yet more realistic for performance evaluation.

Domain-Invariant Semantic Space The semantic space has been regarded as domain-invariant to handle numerous machine learning problems where the source and target classes are non-overlapped or from different domains (e.g., zero-shot learning [49–52] in computer vision, and machine translation [53, 54] in natural language processing). However, few previous works have adopted the semantic space for solving the large-scale FSL problem. One exception is [18], but cross-domain FSL is not considered. Note that, after obtaining the class names in real-world application scenarios, it is trivial to project these class names into the semantic space [55, 30]. Therefore, we take advantage of this easily-accessible semantic information into our large-scale FSL optimization framework, for knowledge transfer from the source classes to the target ones.

3 Methodology

3.1 Problem Definition

We formally define the large-scale cross-domain FSL problem as follows. Let C_s denote the set of source classes and C_t denote the set of target classes ($C_s \bigcap C_t = \emptyset$). We are given a large-scale sample set $\mathcal{D}_s$ from source classes, a few-shot sample set $\mathcal{D}_t$ from target classes, and a test set $\mathcal{T}$ from target classes. For the large-scale sample set $\mathcal{D}_s$, we collect the visual features of all samples as $\mathbf{X}^s \in R^{d \times n_s}$, where d is the dimension of visual feature vectors and n_s is the number of samples. We further collect the semantic representations of all samples in $\mathcal{D}_s$ as $\mathbf{Y}^s \in R^{k \times n_s}$, where $\mathbf{y}_i^s$ (i.e. the i-th column vector of $\mathbf{Y}^s$) is set as a

k-dimensional class semantic embedding according to the source class label of the i-th sample ($i = 1, ..., n_s$). Similarly, the few-shot sample set $\mathcal{D}_t$ can be represented as $\mathbf{X}^t \in R^{d \times n_t}$ and $\mathbf{Y}^t \in R^{k \times n_t}$, where $n_t = K \times |C_t|$ ($n_t \ll n_s$) is the number of samples under the K-shot learning setting. In this paper, the class semantic embeddings of both source and target classes are extracted using the same word2vec model. Note that the source and target classes are assumed to come from different domains under our new FSL setting. For simplicity, we utilize the same source data pre-trained ResNet50 [56] to extract visual features for both source and target classes. The goal of large-scale cross-domain FSL is to obtain good classification results on the test set $\mathcal{T}$.

3.2 Latent Representation Learning over Source Classes

As shown in Figure 3, a Triplet Autoencoder is proposed to model the triple relationship among $\mathbf{X}^s$, $\mathbf{X}^t$, and $\mathbf{Y}^s/\mathbf{Y}^t$. Following the idea of training the well-known triplet network [57], this triplet relationship can be resolved by first modeling the relationship between $\mathbf{X}^s$ and $\mathbf{Y}^s$, and then modeling the relationship between $\mathbf{X}^t$ and $\mathbf{Y}^t$. In other words, we choose to decompose the proposed TriAE into a pair of dual autoencoders: one is designed for latent representation learning over source classes (see Eq. (1)), and the other is designed for the subsequent domain adaptation and FSL over target classes (see Eq. (6)).

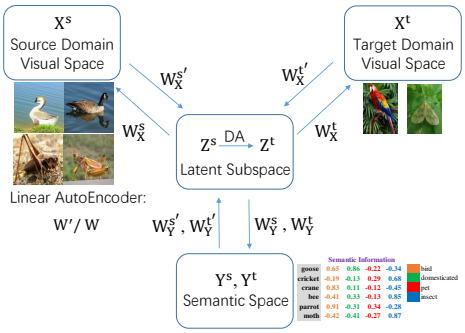


Fig. 3. Architecture of Triplet Autoencoder (TriAE). Note that only linear autoencoder is employed as the backbone model.

Concretely, for latent representation learning over source classes, we utilize a linear matrix projection and its transpose to mimic the encoder-decoder paradigm between the latent subspace and visual feature space/semantic embedding space. This results in a dual autoencoder model, which is optimized by minimizing the objective function:

$$\begin{aligned} F^{(s)} &= \|\mathbf{W}_X^{s'}\mathbf{X}^s - \mathbf{Z}^s\|_F^2 + \|\mathbf{X}^s - \mathbf{W}_X^s\mathbf{Z}^s\|_F^2 + \eta\|\mathbf{W}_X^s\|_F^2 \\ &+ \gamma(\|\mathbf{W}_Y^{s'}\mathbf{Y}^s - \mathbf{Z}^s\|_F^2 + \|\mathbf{Y}^s - \mathbf{W}_Y^s\mathbf{Z}^s\|_F^2 + \eta\|\mathbf{W}_Y^s\|_F^2) \end{aligned} \tag{1}$$

where $\mathbf{Z}^s \in R^{r \times n_s}$ is the latent representation of training samples from source classes, r is the dimensionality of the latent subspace, $\mathbf{W}_X^s (\in R^{d \times r})/\mathbf{W}_Y^s (\in R^{k \times r})$ is the projection matrix from the latent subspace to the visual feature space/semantic embedding space, η is a positive regularization parameter, and γ is a positive weighting coefficient that controls the importance of the two autoencoders (i.e. the first three terms and the last three terms). In this work, we empirically set $\eta = 0.001$.

The optimization problem $\min F^{(s)}$ can be solved by two alternating steps: 1) $\hat{\mathbf{W}}_X^s, \hat{\mathbf{W}}_Y^s = \arg\min_{\mathbf{W}_X^s, \mathbf{W}_Y^s} F^{(s)}(\mathbf{W}_X^s, \mathbf{W}_Y^s, \hat{\mathbf{Z}}^s)$; 2) $\hat{\mathbf{Z}}^s = \arg\min_{\mathbf{Z}^s} F^{(s)}(\hat{\mathbf{W}}_X^s, \hat{\mathbf{W}}_Y^s, \mathbf{Z}^s)$. In this work, $\hat{\mathbf{Z}}^s$ is initialized with the partial least squares (PLS) regression model [58]. Firstly, by setting $\frac{\partial F^{(s)}(\mathbf{W}_X^s, \mathbf{W}_Y^s, \hat{\mathbf{Z}}^s)}{\partial \mathbf{W}_X^s} = 0$ and $\frac{\partial F^{(s)}(\mathbf{W}_X^s, \mathbf{W}_Y^s, \hat{\mathbf{Z}}^s)}{\partial \mathbf{W}_Y^s} = 0$, we have the following two equations:

$$(\mathbf{X}^s\mathbf{X}^{s'} + \eta\mathbf{I})\mathbf{W}_X^s + \mathbf{W}_X^s(\hat{\mathbf{Z}}^s\hat{\mathbf{Z}}^{s'}) = 2\mathbf{X}^s\hat{\mathbf{Z}}^{s'} \tag{2}$$

$$(\mathbf{Y}^s\mathbf{Y}^{s'} + \eta\mathbf{I})\mathbf{W}_Y^s + \mathbf{W}_Y^s(\hat{\mathbf{Z}}^s\hat{\mathbf{Z}}^{s'}) = 2\mathbf{Y}^s\hat{\mathbf{Z}}^{s'} \tag{3}$$

Eq. (2)-(3) can both be solved efficiently by the Matlab built-in function 'Sylvester' with the Bartels-Stewart algorithm [59]. Secondly, by setting $\frac{\partial F^{(s)}(\hat{\mathbf{W}}_X^s, \hat{\mathbf{W}}_Y^s, \mathbf{Z}^s)}{\partial \mathbf{Z}^s} = 0$, we can obtain a linear equation:

$$(\hat{\mathbf{W}}_X^{s'}\hat{\mathbf{W}}_X^s + \gamma\hat{\mathbf{W}}_Y^{s'}\hat{\mathbf{W}}_Y^s + (1+\gamma)\mathbf{I})\mathbf{Z}^s = 2(\hat{\mathbf{W}}_X^{s'}\mathbf{X}^s + \gamma\hat{\mathbf{W}}_Y^{s'}\mathbf{Y}^s) \tag{4}$$

Since $\hat{\mathbf{W}}_X^{s'}\hat{\mathbf{W}}_X^s + \gamma\hat{\mathbf{W}}_Y^{s'}\hat{\mathbf{W}}_Y^s + (1+\gamma)\mathbf{I}$ is a positive definite matrix, Eq. (4) has one explicit unique solution.

3.3 Domain Adaptation and Few-Shot Learning over Target Classes

Once the optimal latent representation $\hat{\mathbf{Z}}^s$ is learned over the sufficient training samples from source classes, we further exploit it for the subsequent domain adaptation and FSL over target classes. Similar to most domain adaptation methods based on shared subspace learning [43–45, 28, 22], we choose to learn a latent subspace in which the learner is unable to distinguish whether a sample is from the source or target domain. The domain adaptation loss function is defined as:

$$F^{(a)} = \sum_{i=1}^{n_s}\sum_{j=1}^{n_t}[(\boldsymbol{\omega}'(\hat{\mathbf{z}}_i^s - \mathbf{z}_j^t))^2 + \lambda\|\hat{\mathbf{z}}_i^s - \mathbf{z}_j^t\|_2^2] \tag{5}$$

where $\hat{\mathbf{z}}_i^s$ is the optimal latent representation of i-th sample from the source domain (i.e. $\hat{\mathbf{Z}}^s = [\hat{\mathbf{z}}_1^s, ..., \hat{\mathbf{z}}_{n_s}^s]$), $\mathbf{z}_j^t$ is the latent representation of j-th sample from the target domain, $\boldsymbol{\omega}$ is a normalized linear projection/column vector (i.e. $\|\boldsymbol{\omega}\|_2 = 1$) to map the latent representations ($\hat{\mathbf{z}}_i^s$ & $\mathbf{z}_j^t$) into the real field R^1, and λ is a positive regularization parameter. In the above loss function, the first term aims to maximize the confusion between $\hat{\mathbf{z}}_i^s$ and $\mathbf{z}_j^t$ in the projected space R^1, and the second term aims to maximize the confusion in the original latent space.

For FSL over target classes, we propose another dual autoencoder model, similar to that used in Eq. (1). Let $\mathbf{Z}^t = [\mathbf{z}_1^t, ..., \mathbf{z}_{n_t}^t]$. The objective function is defined as:

$$\begin{aligned} F^{(t)} &= \|\mathbf{W}_X^{t'}\mathbf{X}^t - \mathbf{Z}^t\|_F^2 + \|\mathbf{X}^t - \mathbf{W}_X^t\mathbf{Z}^t\|_F^2 + \eta\|\mathbf{W}_X^t\|_F^2 \\ &+ \gamma(\|\mathbf{W}_Y^{t'}\mathbf{Y}^t - \mathbf{Z}^t\|_F^2 + \|\mathbf{Y}^t - \mathbf{W}_Y^t\mathbf{Z}^t\|_F^2 + \eta\|\mathbf{W}_Y^t\|_F^2) \end{aligned} \tag{6}$$

Algorithm 1: Triplet Autoencoder (TriAE)

Input: Visual features $\mathbf{X}^s, \mathbf{X}^t$; semantic representations $\mathbf{Y}^s, \mathbf{Y}^t$; parameters β, γ, λ

Output: $\hat{\mathbf{W}}_X^t, \hat{\mathbf{W}}_Y^t$

1. Initialize $\hat{\mathbf{Z}}^s$ with the PLS regression model [58];

while *a stopping criterion is not met* **do**

2. With the learned representation $\hat{\mathbf{Z}}^s$, find $\hat{\mathbf{W}}_X^s$ and $\hat{\mathbf{W}}_Y^s$ by solving Eqs. (2)-(3);

3. With the learned projections $\hat{\mathbf{W}}_X^s$ and $\hat{\mathbf{W}}_Y^s$, update $\hat{\mathbf{Z}}^s$ by solving Eq. (4);

end

4. Initialize $\hat{\mathbf{Z}}^t$ with the PLS regression model [58];

while *a stopping criterion is not met* **do**

5. With the learned representations $\hat{\mathbf{Z}}^s$ and $\hat{\mathbf{Z}}^t$, find $\hat{\boldsymbol{\omega}}$ in Eq. (8) according to Prop. 1;

6. With the learned representations $\hat{\mathbf{Z}}^t$, find $\hat{\mathbf{W}}_X^t$ and $\hat{\mathbf{W}}_Y^t$ by solving Eqs. (9)-(10);

7. With the learnt projections $\hat{\mathbf{W}}_X^t$, $\hat{\mathbf{W}}_Y^t$, and $\hat{\boldsymbol{\omega}}$, update $\hat{\mathbf{Z}}^t$ by solving Eq. (11);

end

8. Return $\hat{\mathbf{W}}_X^t$ and $\hat{\mathbf{W}}_Y^t$.

where η and γ are exactly the same as in Eq. (1). By combining Eq. (5) and Eq. (6) with a weighting coefficient β, we have the final loss function L:

$$L = F^{(t)} + \beta F^{(a)} \tag{7}$$

The optimization problem $\min L$ can be solved by three alternate steps: 1) $\hat{\boldsymbol{\omega}} = \arg_{\boldsymbol{\omega}} \min F^{(a)}(\boldsymbol{\omega}, \hat{\mathbf{Z}}^t)$; 2) $\hat{\mathbf{W}}_X^t, \hat{\mathbf{W}}_Y^t = \arg\min_{\mathbf{W}_X^t, \mathbf{W}_Y^t} F^{(t)}(\mathbf{W}_X^t, \mathbf{W}_Y^t, \hat{\mathbf{Z}}^t)$; 3) $\hat{\mathbf{Z}}^t = \arg\min_{\mathbf{Z}^t} L(\hat{\boldsymbol{\omega}}, \hat{\mathbf{W}}_X^t, \hat{\mathbf{W}}_Y^t, \mathbf{Z}^t)$. In this work, $\hat{\mathbf{Z}}^t$ is initialized with the PLS regression model [58]. Firstly, we find the best $\hat{\boldsymbol{\omega}}$ by solving the following optimization problem according to Prop. 1:

$$\hat{\boldsymbol{\omega}} = \arg_{\boldsymbol{\omega}} \min F^{(a)}(\boldsymbol{\omega}, \hat{\mathbf{Z}}^t) = \arg_{\boldsymbol{\omega}} \min \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} (\boldsymbol{\omega}'(\hat{\mathbf{z}}_i^s - \hat{\mathbf{z}}_j^t))^2 \tag{8}$$

Secondly, by setting $\frac{\partial F^{(t)}(\mathbf{W}_X^t, \mathbf{W}_Y^t, \hat{\mathbf{Z}}^t)}{\partial \mathbf{W}_X^t} = 0$ and $\frac{\partial F^{(t)}(\mathbf{W}_X^t, \mathbf{W}_Y^t, \hat{\mathbf{Z}}^t)}{\partial \mathbf{W}_Y^t} = 0$, we can obtain two equations:

$$(\mathbf{X}^t \mathbf{X}^{t'} + \eta \mathbf{I}) \mathbf{W}_X^t + \mathbf{W}_X^t (\hat{\mathbf{Z}}^t \hat{\mathbf{Z}}^{t'}) = 2 \mathbf{X}^t \hat{\mathbf{Z}}^{t'} \tag{9}$$

$$(\mathbf{Y}^t \mathbf{Y}^{t'} + \eta \mathbf{I}) \mathbf{W}_Y^t + \mathbf{W}_Y^t (\hat{\mathbf{Z}}^t \hat{\mathbf{Z}}^{t'}) = 2 \mathbf{Y}^t \hat{\mathbf{Z}}^{t'} \tag{10}$$

Thirdly, by setting $\frac{\partial L(\hat{\boldsymbol{\omega}}, \hat{\mathbf{W}}_X^t, \hat{\mathbf{W}}_Y^t, \mathbf{Z}^t)}{\partial \mathbf{Z}^t} = 0$, we have the following equation:

$$\begin{aligned} &[\hat{\mathbf{W}}_X^{t'} \hat{\mathbf{W}}_X^t + \gamma \hat{\mathbf{W}}_Y^{t'} \hat{\mathbf{W}}_Y^t + (1+\gamma)\mathbf{I}] \mathbf{Z}^t + \beta(\hat{\boldsymbol{\omega}} \hat{\boldsymbol{\omega}}' + \lambda \mathbf{I}) \mathbf{Z}^t (\mathbf{B}\mathbf{B}') \\ =& 2 \hat{\mathbf{W}}_X^{t'} \mathbf{X}^t + 2\gamma \hat{\mathbf{W}}_Y^{t'} \mathbf{Y}^t + \beta(\hat{\boldsymbol{\omega}} \hat{\boldsymbol{\omega}}' + \lambda \mathbf{I}) \hat{\mathbf{Z}}^s \mathbf{A}\mathbf{B}' \end{aligned} \tag{11}$$

where the formal definitions of $\mathbf{A}, \mathbf{B}$ are given in Prop. 1. Notably, according to Prop. 4, we can find the unique solution $\hat{\mathbf{Z}}^t$ of Eq. (11).

The complete algorithm for training our triplet autoencoder model is outlined in Algorithm 1. Once the optimal projections $\hat{\mathbf{W}}_X^t$ and $\hat{\mathbf{W}}_Y^t$ are learned, the class label of a test sample $\mathbf{x}^*$ is predicted as $l_{\mathbf{x}^*} = \arg\min_j \|\hat{\mathbf{W}}_X^{t'}\mathbf{x}^* - \hat{\mathbf{W}}_Y^{t'}\bar{\mathbf{y}}_j^t\|_2$, where $\bar{\mathbf{y}}_j^t$ is the semantic embedding of j-th target class using the word2vec model ($j = 1, ..., |C_t|$).

3.4 Theoretical Analysis

We finally give theoretical analysis for Algorithm 1. Specifically, Prop. 1 and Prop. 2 provide an efficient approach to finding the solution $\hat{\boldsymbol{\omega}}$ of Eq. (8), and Prop. 3 and Prop. 4 guarantee the solution uniqueness of Eqs. (9)-(11). Their proofs can be found in the suppl. material.

Proposition 1 $\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}(\boldsymbol{\omega}'(\hat{\mathbf{z}}_i^s - \hat{\mathbf{z}}_j^t))^2 = \boldsymbol{\omega}'(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})'\boldsymbol{\omega}$, *where* $\mathbf{A} = \begin{pmatrix} \mathbf{1}'_{n_t} & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{0} & \mathbf{1}'_{n_t} & ... & \mathbf{0} \\ ... & ... & ... & ... \\ \mathbf{0} & \mathbf{0} & ... & \mathbf{1}'_{n_t} \end{pmatrix} \in R^{n_s \times (n_t \times n_s)}, \mathbf{B} = (\mathbf{I}_{n_t}, \mathbf{I}_{n_t}, ..., \mathbf{I}_{n_t}) \in R^{n_t \times (n_t \times n_s)}$, $\mathbf{1}_{n_t}$ *is* n_t*-dimensional vector with all elements* 1*, and* $\mathbf{I}_{n_t} \in R^{n_t \times n_t}$ *is an identity matrix. Therefore, the solution* $\hat{\boldsymbol{\omega}}$ *of Eq. (8) is exactly the smallest eigenvector of* $(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})'$.

Proposition 2 $(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})' = \hat{\mathbf{Z}}^s\mathbf{A}\mathbf{A}'\hat{\mathbf{Z}}^{s'} - \hat{\mathbf{Z}}^s\mathbf{A}\mathbf{B}'\hat{\mathbf{Z}}^{t'} - \hat{\mathbf{Z}}^t\mathbf{B}\mathbf{A}'\hat{\mathbf{Z}}^{s'} + \hat{\mathbf{Z}}^t\mathbf{B}\mathbf{B}'\hat{\mathbf{Z}}^{t'}$. *Since* $\mathbf{A}\mathbf{A}' = n_t\mathbf{I}_{n_s}, \mathbf{B}\mathbf{B}' = n_s\mathbf{I}_{n_t}, \mathbf{B}\mathbf{A}' = (\mathbf{1}_{n_t}, \mathbf{1}_{n_t}, ..., \mathbf{1}_{n_t}) \in R^{n_t \times n_s}$*, and* $\mathbf{A}\mathbf{B}' = (\mathbf{B}\mathbf{A}')'$*, the computation of* $(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})'$ *has a linear time cost* $\mathcal{O}(r^2(n_t + n_s))$ $(r \ll n_t + n_s)$.

Remark 1. Let $\mathbf{G} = \hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B}$. If we directly calculate $\mathbf{G}$ and then $\mathbf{G}\mathbf{G}'$, the total flops cost is $2rn_tn_s(n_t + n_s + r + 1)$ and the computation cost is $\mathcal{O}(rn_tn_s(n_t + n_s))$, which is much higher than that given by **Prop. 2**.

Proposition 3 *According to the eigenvalue decomposition of positive semi-definite matrices, we have* $\mathbf{X}^t\mathbf{X}^{t'} = \mathbf{U}_1\boldsymbol{\Sigma}_1\mathbf{U}_1'$, $\hat{\mathbf{Z}}^t\hat{\mathbf{Z}}^{t'} = \mathbf{U}_2\boldsymbol{\Sigma}_2\mathbf{U}_2'$*, as well as* $\mathbf{Y}^t\mathbf{Y}^{t'} = \mathbf{U}_3\boldsymbol{\Sigma}_3\mathbf{U}_3'$*, where* $\boldsymbol{\Sigma}_1 = diag(\lambda_1^{(1)}, ..., \lambda_d^{(1)})$, $\boldsymbol{\Sigma}_2 = diag(\lambda_1^{(2)}, ..., \lambda_r^{(2)})$*, and* $\boldsymbol{\Sigma}_3 = diag(\lambda_1^{(3)}, ..., \lambda_k^{(3)})$*. Let* $\mathbf{C} = (\frac{2}{\lambda_i^{(1)} + \eta + \lambda_j^{(2)}})_{d \times r}$ *and* $\mathbf{D} = (\frac{2}{\lambda_l^{(3)} + \eta + \lambda_j^{(2)}})_{k \times r}$*. Both Eq. (9) and Eq. (10) have and only have one solution:*

$$\hat{\mathbf{W}}_X^t = \mathbf{U}_1[(\mathbf{U}_1'\mathbf{X}^t\hat{\mathbf{Z}}^{t'}\mathbf{U}_2) \odot \mathbf{C}]\mathbf{U}_2' \quad \hat{\mathbf{W}}_Y^t = \mathbf{U}_3[(\mathbf{U}_3'\mathbf{Y}^t\hat{\mathbf{Z}}^{t'}\mathbf{U}_2) \odot \mathbf{D}]\mathbf{U}_2'$$

where $\odot$ *means Hardamard product of two matrices (i.e. element-wise product).*

Proposition 4 *Let* $\mathbf{H} = (1 + \gamma + n_s\beta\lambda)\mathbf{I} + \hat{\mathbf{W}}_X^{t'}\hat{\mathbf{W}}_X^{t} + \gamma\hat{\mathbf{W}}_Y^{t'}\hat{\mathbf{W}}_Y^{t} + n_s\beta\hat{\boldsymbol{\omega}}\hat{\boldsymbol{\omega}}'$. *Since* $\mathbf{B}\mathbf{B}' = n_s\mathbf{I}_{n_t}$ *and* $\mathbf{H}$ *is positive definite, Eq. (11) has and only has one solution:*

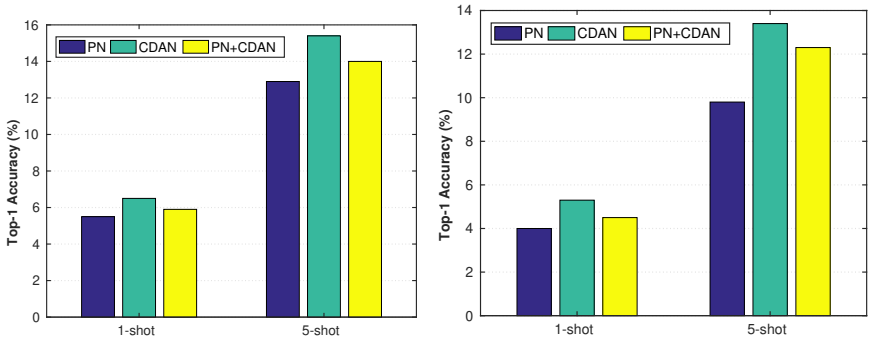
$$\hat{\mathbf{Z}}^t = \mathbf{H}^{-1}[2\hat{\mathbf{W}}_X^{t'}\mathbf{X}^t + 2\gamma\hat{\mathbf{W}}_Y^{t'}\mathbf{Y}^t + \beta(\hat{\boldsymbol{\omega}}\hat{\boldsymbol{\omega}}' + \lambda\mathbf{I})\hat{\mathbf{Z}}^s\mathbf{A}\mathbf{B}']$$

According to Prop. 2, the computation of $(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})'$ has a linear time cost. Given that $(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A} - \hat{\mathbf{Z}}^t\mathbf{B})' \in R^{r\times r}$, its smallest eigenvector can be found very efficiently. According to Prop. 1, finding the solution $\hat{\boldsymbol{\omega}}$ of Eq. (8) thus has a linear time cost. Moreover, Prop. 3 and Prop. 4 give explicit solutions for Eqs. (9)-(11) (and also guarantee the solution uniqueness). Note that Prop. 3 similarly holds for Eqs. (2)-(3).

4 Experiments

4.1 Experiment Setup

Datasets and Settings We take the 1,000 classes from ILSVRC2012 (ImageNet) as the source classes as in [34, 18], with 200 samples per class. Based on ILSVRC2012, we construct two large-scale datasets for evaluation. (1) **ImageNet 2012-2010**: The 360 classes from ILSVRC2010 (not included in ILSVRC 2012) are used as the target classes, with 150 samples per class. To construct a cross-domain dataset, we adopt pre-trained MSG-Net [60] for style transfer over all samples from ILSVRC2010 360 classes, and take the style-transferred samples from these 360 classes as the final target data. (2) **ImageNet2012-DomainNet**: We choose the Infograph dataset (all infographic images) in DomainNet [61], remove the overlapped ILSVRC2012 1K classes from the original 345 Infograph classes, and leave the non-overlapped 144 classes as the target domain (see more details in the suppl. material). With each target DomainNet category containing samples from dozens to hundreds, the target domain includes 20,661 images in total. As in [18], each dataset is split into three parts: a large-scale sample set of sufficient labeled samples from source classes, a K-shot sample set of few labeled samples from target classes, and a test set of the rest samples from target classes. Note that this dataset is more challenging than ImageNet2012-2010 since the target domain is real and the domain gap is bigger.
Visual, Semantic, and Latent Spaces As in [18], we utilize the ResNet50 [56] model pre-trained by us on the source ILSVRC2012 1K classes to extract 2,048-dimensional visual feature vectors. To obtain the semantic representations, we use the same 1,000-dimensional word vectors as in [34, 18], which are obtained by training a skip-gram text model on a corpus of 4.6M Wikipedia documents. In this work, the shared latent subspace is initialized by running PLS regression [58] with visual feature vectors and semantic word vectors as two groups of inputs. The dimension of the latent shared subspace is a hyperparameter, and we empirically set it as $r = 300$ in all experiments.



(a) ImageNet2012-2010 (b) ImageNet2012-DomainNet

Fig. 4. Comparative results among PN, CDAN, and PN+CDAN on two large-scale cross-domain datasets. For each method, an extra LR classifier is trained for large-scale classification.

Evaluation Metric and Hyperparameter Unlike the n-way K-shot evaluation protocol widely used in the conventional FSL setting [9, 14], we choose to evaluate the performance over all target classes (but not over a subset of target classes for one trial), similar to that in other large-scale FSL works [16, 18]. Top-1 accuracy over the test set is used for each trial, and average Top-1 accuracy is computed. Note that we *reproduce the results of all baseline methods* under our new FSL setting, and thus it is still fair to make comparison between the proposed TriAE and other FSL/domain adaptation methods.

Algorithm 1 for training our TriAE model has three hyperparameters to tune: β (see Eq. (11)), γ (see Eqs. (4) and (11)), and λ (see Eq. (11)). Note that the K-shot sample set is not used to directly compute the classification loss for training our TriAE model. Instead, we select the hyperparameters using the Top-1 classification accuracy computed over the K-shot sample set. Additional experiments in the suppl. material show that the influence of the hyperparameters on our model's performance is small.

Compared Methods We select representative/latest FSL and domain adaptation baselines. (1) **FSL Baselines**: We first select the latest large-scale FSL methods including SGM [16], PPA [17], LSD [26], and KTCH [18]. Moreover, the latest meta-learning-based FSL models (e.g., Prototypical Network (PN) [13], Matching Network (MN) [11], MetaOptNet [62], and Baseline++ [40]) are also selected as baselines. Note that PN, MN, and MetaOptNet are designed and evaluated under the n-way K-shot setting. When extending them to our proposed setting, we replace their backbone with ResNet50 for fair comparison: the n-way K-shot setting is still used for model training over the source classes, but when adapting to the target classes, the learned model is only used to extract the visual features of the few-shot sample set and thus a logistic regression (LR) classifier has to be trained for finally recognizing the target classes (as in [26]). In addition, another baseline is obtained by conducting the naive Nearest Neigh-

Table 1. Comparative accuracies (%, top-1) for large-scale cross-domain FSL on ImageNet2012-2010. 'LR' means that an extra LR classifier is trained for large-scale classification. 'K' denotes the number of shots per target class. ‡ highlights that the extra LR is *even stronger* than recent FSL-based classifiers (see Figure 4).

Model	LR?	K=1	K=2	K=5	K=10	K=20
NN	w/o	3.5	5.3	8.3	11.3	13.7
MN [11]	w	3.4	5.2	8.7	12.6	14.8
PN [13]	w	5.5	7.6	12.9	13.8	15.0
MetaOptNet [62]	w	5.7	7.7	12.8	13.9	15.2
Baseline++ [40]	w/o	4.9	7.4	11.7	16.0	19.0
PPA [17]	w/o	4.1	6.4	11.8	15.1	17.6
SGM [16]	w/o	3.9	6.3	12.1	16.2	19.1
LSD [26]	w/o	5.7	8.0	12.4	15.6	19.0
KTCH [18]	w/o	5.9	8.7	13.5	17.0	19.4
CoGAN [20]	w	5.5	8.1	12.2	16.9	19.3
ADDA [21]	w	5.4	7.3	11.8	14.5	15.7
CDAN‡ [24]	w	6.5	9.8	15.4	20.6	24.2
AFN‡ [63]	w	7.0	9.9	15.7	20.5	24.2
TriAE (ours)	w/o	**7.9**	**11.6**	**17.3**	**22.4**	**26.3**

bor (NN) search based on the pretrained ResNet50. (2) **Domain Adaptation Baselines**: We further compare with the latest unsupervised domain adaptation methods such as Coupled Generative Adversarial Network (CoGAN) [20], Adversarial Discriminative Domain Adaptation (ADDA) [21], Conditional Adversarial Domain Adaptation (CDAN) [24], and Adaptive Feature Norm (AFN) [63]. We first evaluate these domain adaptation methods with ResNet50 as backbone for visual feature learning and then conduct LR classification for FSL.

We assume that *naively combining FSL and domain adaptation* for large-scale cross-domain FSL is not effective and this is the place where our major technical novelty lies. To validate this, we conduct experiments by directly combining a representative FSL method (i.e. PN [13]) and a representative domain adaptation method (i.e. CDAN [24]). Concretely, we utilize CDAN to train a feature extractor and then apply the feature extractor to PN (denoted as PN+CDAN). Under the large-scale FSL evaluation protocol, we have to train an extra LR classifier for final recognition. The comparative results in Figure 4 show that adding FSL to domain adaptation even causes performance degradation (see PN+CDAN vs. CDAN). Moreover, it is also observed that under the large-scale FSL evaluation protocol, a basic classifier like LR even yields better classification performance than FSL-based classifiers. In the following experiments, we thus ignore the naive combination of FSL and domain adaptation as a baseline.

4.2 Comparative Results

The comparative results of large-scale cross-domain FSL on the two datasets are presented in Tables 1 and 2. We can make the following observations: (1) Our

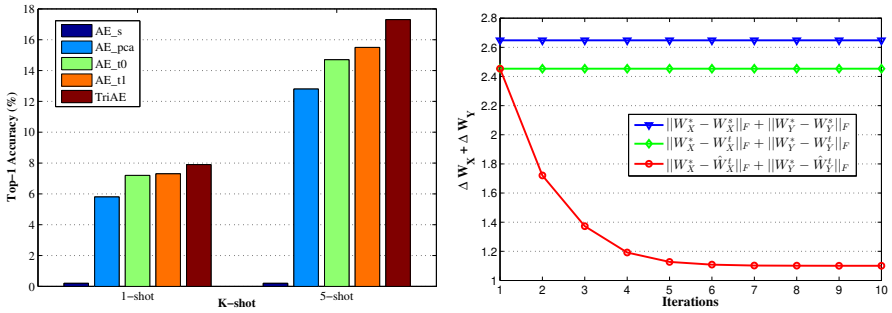
Table 2. Comparative accuracies (%, top-1) for large-scale cross-domain FSL on ImageNet2012-DomainNet. ‡ highlights that the extra LR is *even stronger* than recent FSL-based classifiers.

Model	LR?	K=1	K=2	K=5	K=10	K=20
NN	w/o	2.5	4.0	7.4	10.3	13.0
MN [11]	w	3.1	4.3	8.0	11.9	14.1
PN [13]	w	3.7	4.8	9.5	12.7	14.4
MetaOptNet [62]	w	3.1	4.8	9.6	12.8	14.9
Baseline++ [40]	w/o	3.2	4.6	9.5	13.2	16.2
PPA [17]	w/o	3.8	5.8	11.2	14.9	17.4
SGM [16]	w/o	3.6	5.9	11.5	15.5	20.8
LSD [26]	w/o	3.2	4.4	9.0	12.5	15.5
KTCH [18]	w/o	3.7	5.1	10.7	15.1	18.6
CoGAN [20]	w	5.2	7.4	12.4	15.2	18.9
ADDA [21]	w	3.3	4.6	8.2	11.2	14.4
CDAN‡ [24]	w	5.3	7.5	13.4	17.7	22.1
AFN‡ [63]	w	3.9	6.1	10.8	15.3	20.5
TriAE (ours)	w/o	**6.4**	**9.3**	**16.0**	**20.2**	**24.8**

TriAE model significantly outperforms the state-of-the-art large-scale FSL methods [16, 17, 26, 18], because of explicitly solving the domain adaptation problem under the new cross-domain FSL setting. (2) Our TriAE model also clearly outperforms the latest unsupervised domain adaptation (UDA) methods [24, 20, 21, 63]. In particular, the improvements achieved by our TriAE model over these UDA methods on the more challenging ImageNet2012-DomainNet dataset are larger. This suggests that our model can better cope with the large domain gap in the dataset. (3) The latest UDA methods (i.e. CDAN and AFN) clearly yield better results than existing large-scale FSL methods. This is interesting because they were originally designed for a shared class label space across the source and target domains. This result seems to suggest that solving the domain gap problem is more critical under the new cross-domain FSL setting. (4) The latest meta-learning-based FSL methods (i.e. MetaOptNet and Baseline++) generally perform worse than existing large-scale FSL methods, indicating that these approaches are not suitable for the proposed more challenging FSL setting.

4.3 Further Evaluation

Ablation Study Our full TriAE model can be simplified as follows: (1) When only the large-scale sample set from source classes is used for projection learning, we can obtain $\hat{\mathbf{W}}_X^s, \hat{\mathbf{W}}_Y^s$ and then utilize these two projections directly for image classification over target classes. That is, our TriAE degrades to the model proposed in Section 3.2, denoted as AE_s. (2) When only the few-shot sample set from target classes is used to learn $\mathbf{W}_X^t, \mathbf{W}_Y^t$ but without updating $\hat{\mathbf{Z}}^t$, our TriAE degrades to one ablative model AE_t0. (3) We can further introduce the alternate optimization steps from Eqs. (9)–(11) into AE_t0, resulting in another



(a) Ablation study (b) Convergence analysis

Fig. 5. (a) Ablative results under the large-scale cross-domain 1-shot and 5-shot settings. (b) Convergence analysis of our TriAE model under the large-scale cross-domain 5-shot setting.

ablative model AE_t1 (i.e. TriAE with $\beta = 0$). Note that the semantic information is used in AE_s, AE_t0, and AE_t1. (4) We take on board the fourth ablative model AE_pca (i.e. TriAE with $\gamma = 0$), which utilizes PCA [64] to reduce the dimension of $\mathbf{X}^s$ ($\mathbf{X}^t$) and obtain the latent representation $\mathbf{Z}^s$ ($\mathbf{Z}^t$), without exploiting the semantic embedding. After mapping the visual features of few-shot target-class images into the latent subspace for generating target class representations, AE_pca projects a test image into the latent subspace for NN search. We conduct the ablation study under the large-scale cross-domain FSL setting. The ablative results in Figure 5(a) show that: (i) The unsatisfactory performance of AE_s indicates that there does exist a large gap between the source and target domains; (ii) The marginal improvements achieved by AE_t1 over AE_t0 validate the effectiveness of the alternate optimization steps used for training our TriAE; (iii) The dominance of our TriAE over AE_pca demonstrates the advantage of introducing the semantic embedding into FSL; (iv) The performance gains obtained by our TriAE over AE_t1 validate the effectiveness of our linear domain adaptation strategy.

Convergence Analysis To provide convergence analysis for our TriAE model, we define three baseline projection matrices: (1) $\mathbf{W}_X^*, \mathbf{W}_Y^*$ – learned by AE_t0 with the whole labeled data from target classes (i.e. both few-shot sample set and test set); (2) $\mathbf{W}_X^s, \mathbf{W}_Y^s$ – learned by AE_s only with the large-scale sample set from source classes; (3) $\mathbf{W}_X^t, \mathbf{W}_Y^t$ – learned by AE_t0 only with the few-shot sample set from target classes. As we have mentioned, $\hat{\mathbf{W}}_X^t, \hat{\mathbf{W}}_Y^t$ are learned by our TriAE model using the large-scale sample set and few-shot sample set. We can directly compare $\mathbf{W}_X^s/\mathbf{W}_Y^s$, $\mathbf{W}_X^t/\mathbf{W}_Y^t$, and $\hat{\mathbf{W}}_X^t/\hat{\mathbf{W}}_Y^t$ to $\mathbf{W}_X^*/\mathbf{W}_Y^*$ by computing the matrix distances among them. Note that $\mathbf{W}_X^*, \mathbf{W}_Y^*$ are considered to be the best projection matrices for recognizing the target classes. The results in Figure 5(b) show that: (i) Our TriAE algorithm converges very quickly; (ii)

Table 3. Comparative accuracies (%, top-1) for large-scale conventional FSL on ImageNet2012-2010. As in [18], the original ImageNet2012-2010 is used, without style transfer.

Model	LR?	K=1	K=2	K=5	K=10	K=20
NN	w/o	8.2	11.4	16.6	20.6	23.4
MN [11]	w	7.0	10.1	18.5	24.9	26.2
PN [13]	w	9.9	15.2	21.8	25.2	28.5
MetaOptNet [62]	w	10.2	16.4	22.3	26.8	30.3
Baseline++ [40]	w/o	10.5	16.4	25.2	32.1	38.0
PPA [17]	w/o	15.1	21.4	25.6	28.0	30.7
SGM [16]	w/o	14.8	21.4	33.0	39.1	43.4
LSD [26]	w/o	17.8	22.2	29.0	33.7	38.3
KTCH [18]	w/o	20.2	27.3	36.6	41.8	45.0
CoGAN [20]	w	10.1	16.3	25.0	32.5	38.2
ADDA [21]	w	10.0	15.4	22.2	25.5	27.2
CDAN [24]	w	14.7	21.1	31.5	38.9	43.4
AFN [63]	w	16.4	23.7	34.1	41.0	44.7
TriAE (ours)	w/o	**20.5**	**27.8**	**37.6**	**43.7**	**48.7**

$\hat{\mathbf{W}}_X^t, \hat{\mathbf{W}}_Y^t$ get closer to $\mathbf{W}_X^*, \mathbf{W}_Y^*$ with more iterations and finally become the closest to $\mathbf{W}_X^*, \mathbf{W}_Y^*$.

Conventional FSL In this paper, style transfer is performed on ImageNet2012-2010 to construct a large-scale cross-domain dataset. When style transfer is removed, the cross-domain FSL setting becomes the conventional FSL one. For comprehensive comparison, we also present the results of large-scale conventional FSL on the ImageNet2012-2010 dataset without the added domain change in Table 3. We can observe that our TriAE model still clearly beats all latest FSL and domain adaptation methods. This results suggest that domain gap naturally exists when the target data contains different classes from the source data.

5 Conclusion

We have defined a new large-scale cross-domain FSL setting, which is challenging yet common in real-world scenarios. To overcome the large-scale cross-domain FSL challenge, we propose a Triplet Autoencoder model, which can address both FSL and domain adaptation problems by learning a joint latent subspace. We further provide an efficient model optimization algorithm, followed by rigorous theoretical algorithm analysis. The proposed model is shown to achieve state-of-the-art performance on both new and conventional large-scale FSL problems.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (61976220 and 61832017), and Beijing Outstanding Young Scientist Program (BJJWZYJH012019100020098).

References

1. Li, F., Fergus, R., Perona, P.: One-shot learning of object categories. TPAMI **28** (2006) 594–611
2. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: CVPR. (2018) 4367–4375
3. Oreshkin, B., López, P.R., Lacoste, A.: Tadam: Task dependent adaptive metric for improved few-shot learning. In: Advances in Neural Information Processing Systems. (2018) 721–731
4. Pan, S.J., Yang, Q.: A survey on transfer learning. TKDE **22** (2010) 1345–1359
5. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks **22** (2011) 199–210
6. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: ICML. (2016) 1842–1850
7. Andrychowicz, M., Denil, M., Colmenarejo, S.G., Hoffman, M.W., Pfau, D., Schaul, T., de Freitas, N.: Learning to learn by gradient descent by gradient descent. In: Advances in Neural Information Processing Systems. (2016) 3981–3989
8. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: ICLR. (2017)
9. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML. (2017) 1126–1135
10. Munkhdalai, T., Yu, H.: Meta networks. In: ICML. (2017) 2554–2563
11. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems. (2016) 3630–3638
12. Bertinetto, L., Henriques, J.F., Valmadre, J., Torr, P.H.S., Vedaldi, A.: Learning feed-forward one-shot learners. In: Advances in Neural Information Processing Systems. (2016) 523–531
13. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems. (2017) 4077–4087
14. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: CVPR. (2018) 1199–1208
15. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. IJCV **115** (2015) 211–252
16. Hariharan, B., Girshick, R.B.: Low-shot visual recognition by shrinking and hallucinating features. In: ICCV. (2017) 3037–3046
17. Qiao, S., Liu, C., Shen, W., Yuille, A.L.: Few-shot image recognition by predicting parameters from activations. In: CVPR. (2018) 7229–7238
18. Li, A., Luo, T., Lu, Z., Xiang, T., Wang, L., Wen, J.: Large-scale few-shot learning: knowledge transfer with class hierarchy. In: CVPR. (2019) 7212–7220
19. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. (2014) 2672–2680
20. Liu, M., Tuzel, O.: Coupled generative adversarial networks. In: Advances in Neural Information Processing Systems. (2016) 469–477
21. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR. (2017) 2962–2971
22. Motiian, S., Jones, Q., Iranmanesh, S.M., Doretto, G.: Few-shot adversarial domain adaptation. In: Advances in Neural Information Processing Systems. (2017) 6673–6683

23. Hong, W., Wang, Z., Yang, M., Yuan, J.: Conditional generative adversarial network for structured domain adaptation. In: CVPR. (2018) 1335–1344
24. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems. (2018) 1647–1657
25. Qi, H., Brown, M., Lowe, D.G.: Low-shot learning with imprinted weights. In: CVPR. (2018) 5822–5830
26. Douze, M., Szlam, A., Hariharan, B., Jégou, H.: Low-shot learning with large-scale diffusion. In: CVPR. (2018) 3349–3358
27. Gao, H., Shou, Z., Zareian, A., Zhang, H., Chang, S.: Low-shot learning via covariance-preserving adversarial augmentation networks. In: Advances in Neural Information Processing Systems. (2018) 983–993
28. Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization. In: ICCV. (2017) 5716–5726
29. Klys, J., Snell, J., Zemel, R.: Learning latent subspaces in variational autoencoders. In: Advances in Neural Information Processing Systems. (2018) 6444–6454
30. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. (2013) 3111–3119
31. Ranzato, M., Boureau, Y., Chopra, S., LeCun, Y.: A unified energy-based framework for unsupervised learning. In: AISTATS. (2007) 371–379
32. Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contractive auto-encoders: Explicit invariance during feature extraction. In: ICML. (2011) 833–840
33. Reed, S.E., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: ICML. (2016) 1060–1069
34. Kodirov, E., Xiang, T., Gong, S.: Semantic autoencoder for zero-shot learning. In: CVPR. (2017) 4447–4456
35. Sun, Q., Liu, Y., Chua, T.S., Schiele, B.: Meta-transfer learning for few-shot learning. In: CVPR. (2019) 403–412
36. Jamal, M.A., Qi, G.J.: Task agnostic meta-learning for few-shot learning. In: CVPR. (2019) 11719–11727
37. Zhang, J., Zhang, M., Lu, Z., Xiang, T., Wen, J.: AdarGCN: Adaptive aggregation GCN for few-shot learning. arXiv preprint arXiv:2002.12641 (2020)
38. Guan, J., Lu, Z., Xiang, T., Li, A., Zhao, A., Wen, J.R.: Zero and few shot learning with semantic feature synthesis and competitive learning. TPAMI (2020) 1–14
39. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology (2011)
40. Chen, W., Liu, Y., Kira, Z., Wang, Y.F., Huang, J.: A closer look at few-shot classification. In: ICLR. (2019)
41. Long, M., Ding, G., Wang, J., Sun, J., Guo, Y., Yu, P.S.: Transfer sparse coding for robust image representation. In: CVPR. (2013) 407–414
42. Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: ICML. (2013) 10–18
43. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML. (2015) 97–105
44. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.S.: Domain-adversarial training of neural networks. JMLR **17** (2016) 59:1–59:35
45. Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Information bottleneck learning using privileged information for visual recognition. In: CVPR. (2016) 1496–1505

46. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: ICCV. (2015) 4068–4076
47. Ganin, Y., Lempitsky, V.S.: Unsupervised domain adaptation by backpropagation. In: ICML. (2015) 1180–1189
48. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV. (2010) 213–226
49. Zhao, A., Ding, M., Guan, J., Lu, Z., Xiang, T., Wen, J.R.: Domain-invariant projection learning for zero-shot recognition. In: Advances in Neural Information Processing Systems. (2018) 1019–1030
50. Huo, Y., Guan, J., Zhang, J., Zhang, M., Wen, J.R., Lu, Z.: Zero-shot learning with few seen class samples. In: ICME. (2019) 1336–1341
51. Liu, G., Guan, J., Zhang, M., Zhang, J., Wang, Z., Lu, Z.: Joint projection and subspace learning for zero-shot recognition. In: ICME. (2019) 1228–1233
52. Li, A., Lu, Z., Guan, J., Xiang, T., Wang, L., Wen, J.R.: Transferrable feature and projection learning with class hierarchy for zero-shot learning. IJCV (2020) 1–18
53. Garcia, E.M., Tiedemann, J., España-Bonet, C., Màrquez, L.: Word's vector representations meet machine translation. In: EMNLP 2014 Workshop on Syntax, Semantics and Structure in Statistical Translation. (2014) 132–134
54. McCann, B., Bradbury, J., Xiong, C., Socher, R.: Learned in translation: Contextualized word vectors. In: Advances in Neural Information Processing Systems. (2017) 6297–6308
55. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR 2013 Workshop. (2013)
56. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016) 770–778
57. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition. (2015) 84–92
58. Rosipal, R., Krämer, N.: Overview and recent advances in partial least squares. In: International Statistical and Optimization Perspectives Workshop "Subspace, Latent Structure and Feature Selection". (2005) 34–51
59. Bartels, R.H., Stewart, G.W.: Solution of the matrix equation ax+xb=c [F4] (algorithm 432). Commun. ACM **15** (1972) 820–826
60. Zhang, H., Dana, K.: Multi-style generative network for real-time transfer. arXiv preprint arXiv:1703.06953 (2017)
61. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: ICCV. (2019) 1406–1415
62. Lee, K., Maji, S., Ravichandran, A., Soatto, S.: Meta-learning with differentiable convex optimization. In: CVPR. (2019) 10657–10665
63. Xu, R., Li, G., Yang, J., Lin, L.: Unsupervised domain adaptation: An adaptive feature norm approach. CoRR **abs/1811.07456** (2018)
64. Maćkiewicz, A., Ratajczak, W.: Principal components analysis (pca). Computers & Geosciences **19** (1993) 303–342