GeT: Generative Target Structure Debiasing for Domain Adaptation

Can Zhang Gim Hee Lee
Department of Computer Science, National University of Singapore
can.zhang@u.nus.edu gimhee.lee@nus.edu.sg

Abstract

Domain adaptation (DA) aims to transfer knowledge from a fully labeled source to a scarcely labeled or totally unlabeled target under domain shift. Recently, semi-supervised learning-based (SSL) techniques that leverage pseudo labeling have been increasingly used in DA. Despite the competitive performance, these pseudo labeling methods rely heavily on the source domain to generate pseudo labels for the target domain and therefore still suffer considerably from source data bias. Moreover, class distribution bias in the target domain is also often ignored in the pseudo label generation and thus leading to further deterioration of performance. In this paper, we propose GeT that learns a non-bias target embedding distribution with high quality pseudo labels. Specifically, we formulate an online target generative classifier to induce the target distribution into distinctive Gaussian components weighted by their class priors to mitigate source data bias and enhance target class discriminability. We further propose a structure similarity regularization framework to alleviate target class distribution bias and further improve target class discriminability. Experimental results show that our proposed GeT is effective and achieves consistent improvements under various DA settings with and without class distribution bias. Our code is available at: https://lulusindazc.github.io/getproject/.

1. Introduction

Despite the remarkable advances of deep learning in the last decade [19, 58, 21, 27], the success of most deep learning-based works is based on the assumption that the data distributions of the train and test sets are similar, i.e. no domain shift. However, it is difficult to ensure no domain shift in the data distributions for many practical real-world scenarios. Consequently, many domain adaptation (DA) works [11, 25, 54, 50] have been proposed to alleviate the domain shift problem. Unsupervised DA (UDA) is the most commonly studied DA setting, where the goal is to transfer knowledge from the labeled source data to the unlabeled target data with domain shift. Other more challenging variants of UDA include partial-set DA (PDA) [5, 68, 12] where the target label space is a subset of the source label space, semi-supervised DA (SSDA) [55, 22, 33, 26] which assumes partial target data are labeled, etc.

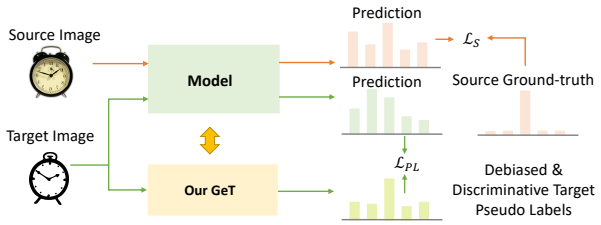

Figure 1. Our GeT is designed to generate source domain and class distribution debiased and discriminative pseudo labels for the target domain in various domain adaptation tasks.

Most DA approaches are often either based on learning domain-invariant feature representations or directly adopting SSL techniques for knowledge transfer. It is shown in [3, 43, 44] that the target error is bounded by the source error and the divergence between marginal distributions in the source and target domains. Inspired by the theoretical analysis, many works [38, 65, 56, 15, 61] propose to learn domain-invariant feature representations using a shared feature extractor to align the source and target domains. Nonetheless, feature alignment-based methods usually suffer from the potential risk of damaging intrinsic target data discrimination. On the other hand, some recent works [9, 72, 51] investigate the application of SSL techniques, e.g. MixMatch [4] in [51], Label Propagation [74] in [72], etc to strengthen discriminability on the unlabeled target domain. Although SSL-based DA methods can achieve competitive performance, they often suffer from source domain bias due to over reliance on the source domain for pseudo label generation. A recent work [36] proposes to deal with data bias using an auxiliary target domain-oriented classifier (ATDOC) based on pseudo labeling. It is shown that the proposed SSL regularization can work quite well in most DA scenarios.

In addition to source data bias, many DA approaches (in-

cluding ATDOC) suffer significant performance drop due to class distribution bias in the target domain. Several methods [66, 60, 23, 59] are proposed to alleviate class distribution bias with class-conditioned sampling [23], class-balanced self-training [60], etc. However, as discussed in [10, 73], these methods rely on domain-invariant representation learning that can hurt intrinsic data discrimination in the target domain. Consequently, a naive adoption of these methods on SSL-based DA can lead to unreliable pseudo labels that greatly degrade performance. Furthermore, many existing DA methods are often designed to be task-specific and may not be versatile enough to handle complex variants of the DA problem, e.g. PDA and SSDA.

As illustrated in Fig. 1, we propose GeT to generate debiased and discriminative pseudo labels to train the network on the DA tasks in this paper. Our GeT consists of an online target generative classifier and a structure similarity regularization. 1) Our *online target generative classifier* is a Gaussian mixture model (GMM). The class priors (i.e. mixture coefficients) and the means of the Gaussian components are the target features class distribution and prototypes, respectively. Intuitively, our generative classifier induces the target feature distribution into distinctive Gaussian components weighted by their respective class priors and thus alleviating source data bias and enhancing target class discriminability. We introduce a memory bank that resembles a replay buffer to efficiently store and update the class priors and feature prototypes for the classifier online in each mini-batch. 2) Our *structure similarity regularization* alleviates target class distribution bias and further improves target class discriminability. To this end, we introduce an auxiliary distribution implicitly constrained with entropy maximization to encourage balanced and discriminative pseudo labels. The final pseudo labels are obtained as a mixup of the pseudo labels generated by the target oriented generative classifier and the auxiliary distribution. We jointly optimize the auxiliary distribution, the pseudo labels and the network parameters in an iterative classification expectation maximization scheme.

We summarize our contributions as follows: 1) An online target oriented generative classifier is proposed to induce the distribution of the target features into distinctive Gaussian components weighted by the class priors to avoid class distribution and source data biases while enhancing class discriminability. 2) We introduce a structure similarity regularization that leverages an auxiliary distribution implicitly constrained with entropy maximization to avoid the severely biased model predictions. 3) A classification expectation maximization framework is designed to jointly optimize the generative classifier with the structure similarity regularization for pseudo labels generation and train the network with the generated pseudo labels. 4) Competitive results are achieved in various DA settings on several standard benchmark datasets.

2. Related Work

Domain Adaptation. Many recent deep DA works [15, 38, 65, 14] have been proposed based on *domain-invariant representation learning* using a shared feature extractor. *Marginal distribution alignment* [38, 65, 56] and *class conditional distribution alignment* [67, 39, 23] are two representative methods which minimize various divergence measures, e.g. $\mathcal{H}$-divergence [3] and maximum mean discrepancy (MMD) [18] to achieve invariance. Over the years, in contrast to the widely studied covariant shift, label shift assumption is proposed in some works [6, 2, 37, 71] from many views, e.g. setting a prior for the label distribution [52], learning marginal label distribution with the EM algorithm [6] and designing causal/non-causal models [57, 71, 2, 37]. Recent works [66, 60, 23, 59] exploit pseudo labels to improve the performance of DA models under class imbalance, but they still rely on learning domain-invariant representations. An auxiliary target classifier is proposed in [36] to solve the problems of highly unreliable pseudo labels and propagated errors, but it does not consider the more practical label shift problem. Motivated by the simplicity of their framework, we aim to learn an online target-oriented generative classifier to utilize the global target data structure for improving the quality of pseudo labels under both source domain and class distribution biases. In contrast to source-free DA [28, 30] that mostly freeze the source classifier during adaptation to preserve class information, our method uses a more robust pseudo-labeling strategy with the in-training source classifier optimized with data from both source and target domains.

Semi-supervised Learning with Regularization. To leverage useful information from the unlabeled data, deep SSL introduces regularization as an auxiliary learning objective. Pseudo labeling [29], also known as self-training, serves as a simple and effective SSL baseline by generating pseudo labels for the unlabeled samples. A line of SSL works [63, 46, 75] propose different designs of the regularization. Early SSL methods [7] mainly involve Laplacian regularization, large margin regularization, etc. Recently, consistency regularization which enforces consistency between model predictions under different disturbances is becoming increasingly popular. Another widely used regularization strategy is minimum entropy [17] that aims to push model predictions to be sharp and prevent predicted label distribution from being too balanced. Moreover, regularization is also studied in the recent works on DA [13, 24, 9]. It is a special case of transductive SSL for efficiently improving domain alignment performance without explicitly designing DA strategies. In contrast, we study soft pseudo label regularization by learning target data distributions from

the layer-wise feature representations without any modification on the architecture nor applying perturbations on the data or model parameters.

Generative Classifier. It has been investigated in some works that inducing generative classifiers on the pretrained deep model for various tasks, e.g. speech recognition in [20], novelty detection in [31] and learning with noisy label in [32]. A previous UDA work [42] studies the stochastic classifier to improve the generalization ability of Maximum Classifier Discrepancy (MCD) [56] for feature alignment. Another related work [62] proposes a bi-directional prototype-oriented conditional transport approach to align the target features to the source prototypes. In contrast to these methods that focus on aligning feature distributions in two domains, we introduce the generative classifier as a regularization approach to improve the quality of pseudo labels and enhance the model robustness to the class imbalance.

3. Problem Formulation

Definitions. **DA** aims to deal with the domain shift between a set of labeled source data $\mathcal{D}_{\mathcal{S}} = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ and a set of target data $\mathcal{D}_{\mathcal{T}} = \mathcal{D}_{\mathcal{T}}^u \cup \mathcal{D}_{\mathcal{T}}^l$, where $|\mathcal{Y}_S| = C$ and $y_i^s \in \{1, \dots, C\}$ represent the source label space with C classes, and $\mathcal{D}_{\mathcal{T}}^u$ and $\mathcal{D}_{\mathcal{T}}^l$ denote the unlabeled and labeled target data, respectively. **UDA** and **PDA** assume empty target labeled set $\mathcal{D}_{\mathcal{T}}^l = \emptyset$ and unlabeled target set $\mathcal{D}_{\mathcal{T}}^u = \{x_i^t\}_{i=1}^{N_t^u}$. **SSDA** assumes partial target data are labeled, denoted by $\mathcal{D}_{\mathcal{T}}^l = \{x_i^t, y_i^t\}_{i=1}^{N_t^l}$. Furthermore, UDA and SSDA assume a common target and source label space, i.e. $\mathcal{Y}_U = \mathcal{Y}_S$, and PDA assumes that the target label space is a subset of the source label space, i.e. $\mathcal{Y}_U \subset \mathcal{Y}_S$. The objective is to predict the labels $\{\hat{y}^t\}$ of the unlabeled target samples $\{x^t\} \in \mathcal{D}_{\mathcal{T}}^u$ by utilizing the labeled source data $\mathcal{D}_{\mathcal{S}}$ and limited target labeled data $\mathcal{D}_{\mathcal{T}}^l$ if available. Based on the assumption from [41, 64], there exists a shared feature space across domains.

Objective. Our goal is to learn a network $g(\phi(x; \theta_f); \theta_g)$ that is able to handle the source data for the DA tasks with the SSL pseudo labeling approach under class distribution bias. $\phi(x; \theta_f) : x \mapsto f$ denotes the feature embedding function that maps x to the shared feature space f. $g(f; \theta_g) : f \mapsto \mathcal{Y}$ denotes the classifier that maps features f to the label space $\mathcal{Y}$.

4. Our Method

Overview. Our GeT is designed to generate debiased pseudo labels while improving model robustness towards noisy pseudo labels. As illustrated in Fig. 2, our GeT iteratively optimizes the pseudo label generation and trains the network using Classification Expectation Maximization (CEM). In the maximization step, we train the network with the generated pseudo labels, and updates the memory bank with the updated network. Specifically, we proposes a generative classifier to generate pseudo-labels $\hat{Y}_{\mathbb{M}}$ for alleviating data bias from the source domain and class bias in the target domain. The parameters of our generative classifier are efficiently updated by constructing two memory banks: 1) a feature prototypes $\{\mu_c\}_{c=1}^C$, and 2) a class prior distributions $\{\pi_c\}_{c=1}^C$. In addition to modeling the target data structure for improving the quality of pseudo labels, we further introduce the structure similarity regularization for improving model robustness towards noises. In the expectation step, we compute the predictive label distributions P_f and P_g with GMMs in the feature space $\mathcal{N}(x^t|\mu_f)$ and the output space $\mathcal{N}(x^t|\mu_g)$. The corresponding auxiliary target distributions Q_f and Q_g are defined with the empirical distribution of the samples being assigned to the clusters, and updated to enforce balanced assignments. In the classification step, we generate the optimal pseudo labels from the generative classifier and auxiliary distributions by mixing up data structure-wise knowledge and model-wise knowledge. Using the pseudo-labels $\hat{Y}_{\mathbb{M}}$ and target variables $Q_{\{f,g\}}$, we infer pseudo labels $\hat{y}_{\{f,g\}}$ for the target loss $\mathcal{L}_{KL}^{\{f,g\}}$ and additionally adopt the source loss $\mathcal{L}_S$ to optimize the network.

4.1. Generative Classifier

The presence of data bias from the source domain has been shown to degrade the quality of pseudo labels for model adaptation in the target domain [66, 60, 23, 59]. To alleviate data bias from the source domain, we propose a target domain-oriented classifier that can fully exploit the target data structure to generate reliable pseudo labels for the unlabeled target data. Although the prototype-based classifier [36] helps to alleviate the data bias from the source domain, it is sensitive to the class distributions in the target domain by favoring dominant classes over the minority classes. Consequently, the class-imbalance problem further motivates us to extend the prototype-based classifier to a Gaussian mixture model that can learn the intrinsic target data distributions for balancing the predictive label distributions.

We model the distribution of the feature embeddings randomly sampled from the target domain $f^t \sim \mathcal{D}_{\mathcal{T}}$ with a Gaussian mixture model given by:

$$p(f^t) = \sum_c \pi_c \mathcal{N}(f^t \mid \boldsymbol{\mu}_c, \Sigma_c), \tag{1}$$

where $\pi_c = p(y_c)$ is the class prior, i.e. mixing coefficient, and $\{\mu_c, \Sigma_c\}$ are the class prototype, i.e. mean and covariance of the Gaussian component $\mathcal{N}(f^t \mid \mu_c, \Sigma_c)$. We define our generative classifier as the posterior probability of class

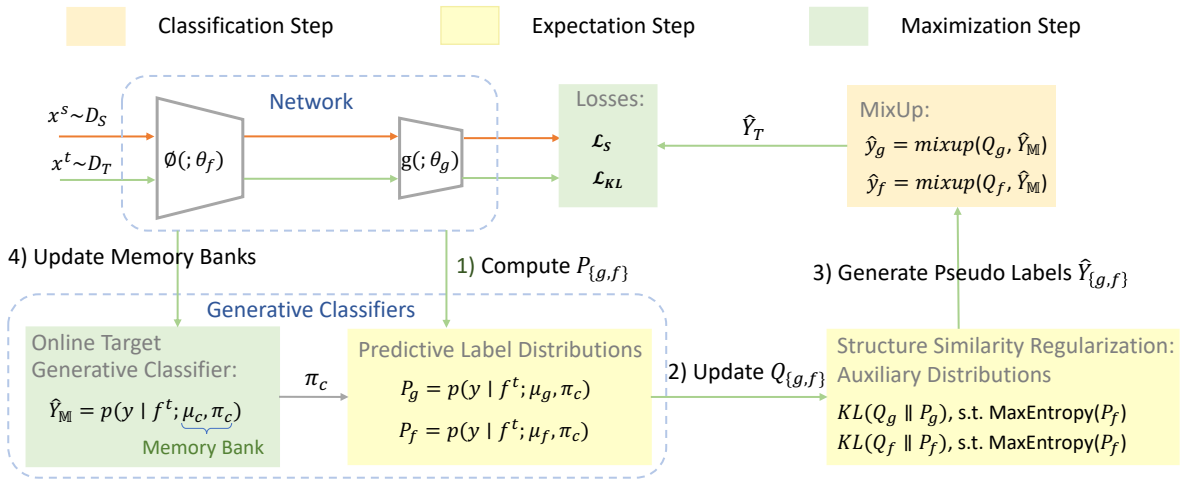

Figure 2. **Overview of GeT.** Our GeT consists of a *online target generative classifier* and a *structure similarity regularization* to generate debiased and discriminative pseudo labels for the supervision of the network on various DA tasks.

y_c given f^t:

$$p(y_c \mid f^t) = \frac{\pi_c p(f^t \mid \boldsymbol{\mu}_c, \Sigma_c)}{\sum_{c'} \pi_{c'} p(f \mid \boldsymbol{\mu}_{c'}, \Sigma_c)} = \sigma(\log \pi_c + \frac{s(f^t, \boldsymbol{\mu}_c)}{\tau}), \tag{2}$$

where $\sigma(\cdot)$ represents the softmax function, $s(\cdot,\cdot)$ measures the similarity between feature embeddings and prototypes (cosine similarity is adopted by default), and τ is a temperature hyperparameter analogous to the class covariance Σ_c. $\frac{s(\cdot,\cdot)}{\tau}$ gives the log-likelihood $\log p(f^t \mid \boldsymbol{\mu}_c, \Sigma_c)$.

4.2. Online Target Generative Classifier

We maintain an online target generative classifier using the features from the target domain, where the parameters, *i.e.* the class prior π_c and the class prototypes $\boldsymbol{\mu}_c$ over the entire distribution of the target embeddings are updated efficiently online using a memory bank. In contrast, many early works [35, 70] naively compute the feature cluster centroids based on unsupervised feature clustering, *e.g.* k-means, which requires the computationally expensive extraction of all the feature embeddings with the feature extractor for clustering.

Class priors π_c. Given the i^{th} batch of target data $\{f^t_{j,i}\}_{j=1}^{\mathcal{B}^t}$, we first generate the pseudo label $\hat{y}^t_{\mathbb{M},j} = \arg\max_c p(y \mid f^t_{j,i};\ \boldsymbol{\mu_{1:C}}, \pi_{1:C})$ of each data $f^t_{j,i}$ with the current class prototypes $\boldsymbol{\mu}_{1:C}$ and class prior $\pi_{1:C}$ in the memory bank. We then use $\hat{y}^t_{\mathbb{M},j}$ in a mixup strategy to generate more reliable pseudo labels to train the network $g(\phi(x;\theta_f);\theta_g)$ (*c.f.* Section 4.4 for more details). The class prior is updated as:

$$\pi_c \leftarrow (1-\gamma_\pi)\pi_c + \gamma_\pi \bar{P}, \quad \text{where} \quad \bar{P} = \frac{1}{\mathcal{B}^t}\sum_{j=1}^{\mathcal{B}^t} \frac{s(f^t_{j,i}, \boldsymbol{\mu}^g_c)}{\tau}, \tag{3}$$

$\boldsymbol{\mu}^g_c = \theta_{g_c} \backslash b_c$ is the weights of the linear classifier $g(.;\theta_g)$ corresponding the class c without the bias terms b_c, and γ_π is the memory decay coefficient. We refer to $\boldsymbol{\mu}^g_c$ as the classifier prototype. The intuition behind this formulation is that the label prior for each class can be estimated by averaging the likelihoods over all target features and the memory bank eases on the computational complexity. Note that we initialize the class prior as a uniform distribution to $\pi_c = \frac{1}{C}$.

Class prototypes $\boldsymbol{\mu}_c$. Different from previous works which compute the prototypes based on all features according to the class labels, we propose to construct the class prototypes $\boldsymbol{\mu}_{1:C}$ on the entire target feature space online using a memory bank. We derived the update of the learnable class prototypes $\boldsymbol{\mu}_{1:C}$ from the conditional distribution $p(\boldsymbol{\mu}_c \mid f^t_{1:i})$ based on the entire historical inputs $\{f^t_{1:i}, y_c\}$ as follows:

$$\log p(\boldsymbol{\mu}_c \mid f^t_{1:i}) \propto \log p(\boldsymbol{\mu}_c \mid f^t_{1:i-1}) + \log p(f^t_i \mid \boldsymbol{\mu}_c) := \boldsymbol{\mu}_c \leftarrow (1-\gamma_{\boldsymbol{\mu}}) * \boldsymbol{\mu}_c + \gamma_{\boldsymbol{\mu}} * \bar{\boldsymbol{\mu}}_c, \text{ where } \bar{\boldsymbol{\mu}}_c = \frac{1}{\mathcal{B}^t_c}\sum_{j=1}^{\mathcal{B}^t} \mathbb{1}_{c,j} f^t_{j,i}. \tag{4}$$

$\mathbb{1}_{c,j} = \mathbb{1}[p(y_c \mid f^t_{j,i}) \geq p(y_{c'} \mid f^t_{j,i}), \forall c' \in C]$ selects the features that give the highest class probability for class c to update the class prototype μ_c. γ_μ is the memory decay coefficient defined in the same way as γ_π.

Remarks: Our generative classifier mitigates source domain bias since it is based solely on the target features. Furthermore, our Bayesian formulation of the classifier encourages discriminative features with the class prior π_c.

4.3. Structure Similarity Regularization

Although our generative classifier can mitigate source data bias and encourage discriminative features, it still risks wrong assignments of features from the scarce classes under class distribution bias. To this end, we introduce a structure similarity regularization which alternates between optimizing the KL-divergence of an auxiliary distribution to the predictive label distributions and the network parameters to encourage balanced and discriminative assignments of features into their respective classes.

Classifier label distribution P_g. We estimate the label distribution $P_g := p(y \mid f_j^t;\, \boldsymbol{\mu}_{1:C}^g,\, \pi_{1:C})$ from the features of the unlabeled target data $\{f_j^t\}_{j=1}^{N_t^u}$, the classifier prototypes $\boldsymbol{\mu}_{1:C}^g$ and class prior distribution $\pi_{1:C}$. A naive optimization of P_g with the pseudo labels $\hat{Y}_{\mathbb{M}}^t$ produced by our generative classifier can cause degenerate solutions where data from scarce classes are assigned wrongly due to class distribution bias. Motivated by [16], we introduce an auxiliary distribution Q_g and minimize the following loss:

$$\mathcal{L}_{\text{KL}}^g = \frac{1}{N_t^u}\,\text{KL}(Q_g \| P_g) + \sum_{c=1}^{C} \bar{Q}_{g_c} \log \bar{Q}_{g_c} \qquad (5)$$

over θ_f, θ_g and Q_g. The first term compute the KL-divergence between the discrete posteriors P_g and Q_g. The second term plays the role of confidence penalty by encouraging entropy maximization of the label distribution in the target domain. $\bar{Q}_{g_c} = \frac{1}{N_t^u}\sum_{j=1}^{N_t^u} Q_g(y_c \mid f_j^t)$ is defined to be the target class proportions. Intuitively, the unlabeled data are more likely to be assigned to the prototypes corresponding to the dominant classes or the prototypes that are much closer to the target features. The empirical label distribution $\bar{Q}_g$ of the regularized auxiliary distribution is enforced to have balanced assignments by the second term, which is equivalent to using the KL-divergence between $\bar{Q}_g$ and a uniform prior distribution. We minimize the loss $\mathcal{L}_{\text{KL}}^g$ using an alternating optimization based on the following two steps:

a) Pseudo-label generation. We fix the network parameters $\{\theta_f, \theta_g\}$ and P_g to estimate the auxiliary distribution Q_g. The closed-form solution of Q_g can be derived by setting the gradient of the optimization objective from Eq. 5 as zero, *i.e.*:

$$Q_g(y_c \mid f_j^t) = \frac{P_g(y_c \mid f_j^t)/(\sum_{j=1}^{N_t^u} P_g(y_c \mid f_j^t))^{\frac{1}{2}}}{\sum_{c'=1}^{C} P_g(y_{c'} \mid f_j^t)/(\sum_{j=1}^{N_t^u} P_g(y_{c'} \mid f_j^t))^{\frac{1}{2}}}. \qquad (6)$$

b) Network retraining. By fixing Q_g, the second term in Eq. 5 reduces to a constant value and thus giving rise to a cross entropy loss using Q_g as the soft label for network optimization:

$$\min_{\theta_f, \theta_g} -\frac{1}{N_t^u}\sum_{j=1}^{N_t^u}\sum_{c=1}^{C} Q_g(y_c \mid f_j^t) \log P_g(y_c \mid f_j^t). \qquad (7)$$

Note that the pseudo-label generation step will be included in the C-step and the network retraining step will be included in the M-step of the final CEM optimization shown in the next section.

Embedding label distribution P_f. Based on the clustering assumption in the feature space, we also introduce a set of learnable embedding prototypes $\boldsymbol{\mu}_{1:C}^f$ to discover the target feature discrimination. We compute the label distribution $P_f := p(y \mid f_j^t;\, \boldsymbol{\mu}_{1:C}^f,\, \pi_{1:C})$ and introduce an auxiliary distribution Q_f. We then minimize the following loss:

$$\mathcal{L}_{\text{KL}}^f = \frac{1}{N_t^u}\,\text{KL}(Q_f \| P_f) + \sum_{c=1}^{C} \bar{Q}_{f_c} \log \bar{Q}_{f_c}. \qquad (8)$$

over θ_f, $\boldsymbol{\mu}_{1:C}^f$, and Q_f. The regularization term is defined as $\bar{Q}_{f_c} = \frac{1}{N_t^u}\sum_{j=1}^{N_t^u} Q_f(y_c \mid f_j^t)$. $\boldsymbol{\mu}_{1:C}^f$ are re-initialized at each epoch using the class prototypes $\boldsymbol{\mu}_{1:C}$ from the memory bank. We apply the same alternating optimization strategy for $\mathcal{L}_{\text{KL}}^g$ on $\mathcal{L}_{\text{KL}}^f$, where the auxiliary distribution Q_f is used as the soft label for optimizing the feature embedding network parameters θ_f and embedding prototypes $\boldsymbol{\mu}_{1:C}^f$.

4.4. Optimization

Given the labeled samples $\{x_j^s, y_j^s\}$ from the source domain $\{X_S, Y_S\}$ and unlabeled samples $\{x_j^t\}$ from the target domain X_T, our GeT model is alternatively optimized by the CEM steps:

***E-Step*:** Compute the posterior probabilities classifier $P_g := p(y \mid f_j^t;\, \boldsymbol{\mu}_{1:C}^g,\, \pi_{1:C})$ and embedding $P_f := p(y \mid f_j^t;\, \boldsymbol{\mu}_{1:C}^f,\, \pi_{1:C})$ label distributions from the current batch of target features $\{f_{j,i}^t\}_{j=1}^{\mathcal{B}^t}$, class prior $\pi_{1:C}$, and classifier $\boldsymbol{\mu}_{1:C}^g$ and embedding $\boldsymbol{\mu}_{1:C}^f$ prototypes.

***C-Step*:** Fixing the network parameters $\{\theta_f, \theta_g\}$, we solve the pseudo label generation step on the objectives from Eq. 5 and Eq. 8 to get the auxiliary distributions $Q_{\{g,f\}}$. We then fully utilize data structure knowledge from both domains to get the final pseudo labels $\hat{Y}_T = \{\hat{y}_g^t, \hat{y}_f^t\}$ by applying mixup on the soft labels from $Q_{\{f,g\}}$ and the pseudo-labels $\hat{Y}_{\mathbb{M}}^t$ from our generative classifier, i.e.:

$$\hat{y}_{\{g,f\},j}^{t,c} = (1-\gamma_Q) Q_{\{g,f\}}(y_c \mid f_j^t) + \gamma_Q \hat{y}_{\mathbb{M},j}^{t,c}, \tag{9}$$

where γ_Q is the coefficient for the mixup.

***M-Step*:** Fixing $\hat{Y}_T = \{\hat{y}_g^t, \hat{y}_f^t\}$, we use the gradient ascent to update the network parameters θ_f and θ_g, and the learnable embedding prototypes $\boldsymbol{\mu}_{1:C}^f$:

$$\begin{aligned} &\max_{\theta_f, \theta_g, \boldsymbol{\mu}_{1:C}^f} \frac{1}{N_s} \sum_{j=1}^{N_s} \sum_{c=1}^{C} y_j^{s,c} \log p_\theta(y_j^{s,c} \mid x_j^s) + \\ &\frac{1}{N_t^u} \sum_{j=1}^{N_t^u} \sum_{c=1}^{C} \hat{y}_{g,j}^{t,c} \log P_g(y_c \mid f_j^t) + \hat{y}_{f,j}^{t,c} \log P_f(y_c \mid f_j^t), \end{aligned} \tag{10}$$

where $p_\theta(y \mid x^s) = g(\phi(x^s; \theta_f); \theta_g)$ denotes the output label predictions of source data x^s from the network. The second and third terms are adapted from the network retraining step mentioned in the previous section.

5. Experiment

5.1. Datasets and Experimental Setting

Datasets. **Office-31** [53] includes three domains: Amazon (A), DSLR (D) and Webcam (W), and contains a total of 4,110 images covering 31 categories. A combination of six pairs of source-target domain settings are evaluated. **Office-Home** [1] includes 4 domains: Artistic (Ar), Clip Art (Cl), Product (Pr) and Real-World (Re) with 65 categories, where there are ∼15,500 images in total. **VisDA-2017** [49] is a challenging dataset due to the big domain shift between the synthetic images (152,397 images from VisDA) and the real images (55,388 images from COCO). **DomainNet-126** is constructed in [55] by selecting 126 classes across 4 domains, i.e. Real (R), Clipart (C), Painting (P) and Sketch (S), from the largest UDA dataset DomainNet [48].

Implementation details. Following [36], we use ResNet-50 pretrained on the ImageNet as the backbone and utilize a mini-batch SGD with momentum 0.9 and weight decay $1e^{-3}$. The learning rate follows the schedule as $\eta_i = \eta_0(1 + \omega \frac{i}{I_{max}})^{-\alpha}$, where $\omega = 10$, $\alpha = 0.75$, and η_0 is the initial learning rate. We set $\eta_0 = 0.001$ for the target-specific bottleneck layers and $\eta_0 = 0.01$ for the classifier. We set $\gamma_{\bar{\mu}} = 0.9$ and perform the sensitivity analysis on γ_Q and $\gamma_{\bar{P}}$. We set the temperature hyper-parameter τ to 1 empirically. We adopt the imbalanced target class setting by following [59], where only thirty percent of data from the first $\lceil C/2 \rceil$ classes are kept to simulate class imbalance in the target domain.

5.2. Comparison with Baselines

Closed-set UDA. Tabs. 1 and 2 study the closed-set UDA setting using OfficeHome and Office-31 datasets under the standard setting. We evaluate our method by combining it with three base models for comparisons: 1) Source-only model, trained with only labeled source data; 2) CDAN+E, a UDA model with an additional domain alignment loss to train the model; 3) MixMatch serving as the SSL base model. We first study the regularization methods integrated with the Source-only model. BNM and MCC perform consistently better than the entropy regularization method, i.e. MinEnt, due to their design on the encouragement of prediction diversity generally ignored by entropy minimization. ATDOC that uses the target-oriented classifier significantly outperforms pseudo labeling (PL) and BNM, which verifies the importance of regularizing target predictions. As shown in Tabs. 1 and 2, our GeT that uses the target structure similarity regularization consistently achieves state-of-the-art performance. When combined with CDAN+E, all baselines show better results as the model is jointly optimized with an additional domain alignment loss. Our GeT is able to further improve the performance and obtain the best average accuracy compared with other regularization methods. Under the SSL framework with MixMatch adopted as the base model, our GeT also boosts the performance when it is adopted as a pseudo label generation module. It is further shown in Tab. 1 and Tab. 2 that GeT can achieve competitive results as some state-of-the-art UDA methods, *e.g.* SCDA and DALN, with no explicit feature alignment.

Semi-supervised DA. We follow the experiment setting in [55] to evaluate SSDA on the DomainNet-126 dataset. As shown in Tab. 3: 1) 1-shot represents one labeled instance is available for each class in the target domain, and 2) 3-shot means we have access to three target labels per class. From the reported results, pseudo-labeling and BNM achieve the same second best performance in the 3-shot setting while BNM performs better in the 1-shot setting. The overall average accuracy of ATDOC-NC indicates that the performance of nearest centroid classifier relies heavily on the assumption of balanced target clusters to assign pseudo labels. We also include the prior state-of-the-art SSDA method, *e.g.* S^{3}D, for comparison, which outperforms the SSL regularization approaches. By contrast, our GeT achieves the highest average accuracy among all the compared methods, which shows that our design on the target data structure learning indeed improves data discrimination.

Table 1. Classification accuracy (%) on Office-Home for UDA (ResNet-50).

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Mean
ResNet-50	44.9	66.3	74.3	51.8	61.9	63.6	52.4	39.1	71.2	63.8	45.9	77.2	59.4
MinEnt [17]	51.0	71.9	77.1	61.2	69.1	70.1	59.3	48.7	77.0	70.4	53.0	81.0	65.8
BNM [13]	56.7	77.5	81.0	67.3	76.3	77.1	65.3	55.1	82.0	73.6	57.0	84.3	71.1
MCC [24]	56.3	77.3	80.3	67.0	77.1	77.0	66.2	55.1	81.2	73.5	57.4	84.1	71.0
PL	54.1	74.1	78.4	63.3	72.8	74.0	61.7	51.0	78.9	71.9	56.6	81.9	68.2
ATDOC-NC [36]	54.4	77.6	80.8	66.5	75.6	75.8	65.9	51.9	81.1	72.7	57.0	83.5	70.2
GeT	59.4	**79.6**	**82.9**	**71.4**	**79.8**	**79.8**	69.7	56.2	83.5	73.9	60.1	86.0	73.5
CDAN+E [40]	54.6	74.1	78.1	63.0	72.2	74.1	61.6	52.3	79.1	72.3	57.3	82.8	68.5
+ BNM [13]	58.1	77.2	81.1	67.5	75.3	77.2	65.5	56.8	82.6	74.1	59.9	84.6	71.7
+ MCC [24]	58.9	77.6	80.7	67.0	75.1	77.1	65.8	56.8	82.2	73.9	59.8	84.5	71.6
+ PL	57.3	76.6	79.2	66.6	74.0	76.6	66.1	53.6	81.0	74.3	58.9	84.2	70.7
+ ATDOC-NC [36]	55.9	76.3	80.3	63.8	75.7	76.4	63.9	53.7	81.7	71.6	57.7	83.3	70.0
+ GeT	**60.5**	78.8	82.6	69.1	79.7	78.8	**69.5**	**59.3**	**84.6**	**75.2**	**62.3**	**88.0**	**74.0**
SAFN [68]	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
SHOT [35]	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
SCDA [34]	57.5	76.9	80.3	65.7	74.9	74.7	65.5	53.6	79.8	74.5	59.6	83.7	70.5
DALN [8]	57.8	79.9	82.0	66.3	76.2	77.2	66.7	55.5	81.3	73.5	60.4	85.3	71.8
FixBi [47]	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7

Table 2. Accuracy (%) on Office-31 for UDA (ResNet-50). [†: average accuracy except D ↔ W.]

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg.	Avg.†
ResNet-50	78.3	70.4	57.3	93.4	61.5	98.1	76.5	66.9
MinEnt [17]	90.7	89.4	67.1	97.5	65.0	**100.0**	85.0	78.1
MCC [24]	92.1	94.0	74.9	98.5	75.3	**100.0**	89.1	84.1
BNM [13]	92.2	94.0	74.9	98.5	75.3	**100.0**	89.2	84.1
PL	88.7	89.1	65.8	98.1	66.6	99.6	84.7	77.6
ATDOC-NC [36]	95.2	91.6	74.6	99.1	74.7	**100.0**	89.2	84.0
ATDOC-NA [36]	94.4	94.3	75.6	98.9	75.2	99.6	89.7	84.9
GeT	95.4	95.4	76.6	**99.1**	77.0	**100.**	90.6	86.0
CDAN+E [40]	94.5	94.2	72.8	98.6	72.2	**100.0**	88.7	83.4
+ MCC [24]	94.1	94.7	75.4	99.0	75.7	**100.0**	89.8	85.0
+ BNM [13]	94.9	94.3	75.8	99.0	75.9	**100.0**	90.0	85.2
+ PL	91.5	93.1	72.5	97.8	72.7	99.8	87.9	82.4
+ ATDOC-NC [36]	96.3	93.6	74.3	**99.1**	75.4	**100.0**	89.8.	84.9
+ ATDOC-NA [36]	95.4	94.6	77.5	98.1	77.0	99.7	90.4	86.1
+ GeT	**96.7**	**95.8**	**78.6**	**99.1**	**77.8**	**100.**	**91.2**	**87.2**
MixMatch [4]	88.5	84.6	63.3	96.1	65.0	99.6	82.9	75.4
w/ PL	89.0	86.0	65.8	96.2	65.6	99.6	83.7	76.6
w/ ATDOC-NC [36]	91.3	86.4	66.0	97.4	64.4	99.4	84.1	77.0
w/ ATDOC-NA [36]	92.1	91.0	70.9	98.6	76.2	99.6	88.1	82.6
w/ GeT	93.1	92.7	71.8	98.8	77.0	99.6	88.65	83.3
SHOT [35]	94.0	90.1	74.7	98.4	74.3	99.9	88.6	83.3
SCDA [34]	95.2	94.2	75.7	98.7	76.2	99.8	90.0	85.3
DALN [8]	95.4	95.2	76.4	99.1	76.5	**100.**	90.4	85.9

Partial-set UDA. We follow the partial-set UDA setting in [36] and evaluate performance on the OfficeHome dataset in Tab. 4 by selecting the first 25 classes as the label space for the unlabeled target data. PDA suffers from both the data bias and the label distribution shift, i.e., two domains have mismatched label space. ATDOC shows relative better results than other SSL regularization baselines as well as the prior state-of-the-art PDA methods, *i.e.* RTNet$_{adv}$. MCC and BNM show comparable performance as MinEnT in the standard setting, but MinEnT achieves better results due to its superiority from the prediction diversity. Similarly, the structural similarity regularization in our GeT can penalize the over-confident predictions and shows effectiveness in improving performance.

Imbalanced Target Distribution. We further evaluate our method for the closed-set UDA and PDA under the imbalanced target label distribution scenario. As shown in Tab. 5, when the target domain is added with class distribution bias, the performance of all methods is inferior to their corresponding standard models suffering only from the data bias, *e.g.* the performance of ATDOC deteriorates to be even more inferior than BNM. Our Get achieves the best results in all DA tasks and shows superior resilience to severe label distribution shift.

5.3. Model Analysis

Ablation study. We conduct ablation study on Office-31 and VisDA-2017 for UDA in Tab. 6 to examine the effect of each component on our GeT. The base model is ATDOC-NC where a target-oriented prototype classifier is used to generate pseudo labels. 1) *Online update strategy for the probabilistic model* (i.e. w/o $\mathcal{L}_{\text{KL}}^{f,g}$). We present results by directly using the predictions $\hat{y}_{\mathbb{M}}^t$ from the online updated generative classifier as pseudo labels. Compared to the pure target feature classifier (NC), our GeT improves $+1.1\%$ average accuracy on VisDA-2017 by modeling feature distributions with the generative classifier. It verifies the effectiveness of our online update strategy for the probabilistic model. 2) *Effect of feature structure regularization*. We examine the effect of each loss by removing each feature discrimination objective formulated by KL-divergence. We first evaluate $\mathcal{L}_{\text{KL}}^{g}$ in the label space (i.e. w/o $\mathcal{L}_{\text{KL}}^{\tilde{f}}$). It is shown the auxiliary distribution variable $Q_{g,f}$ can bring better performance than the oracle hard supervisions. The ensemble of feature-level regularization further improves $+1.7\%$ on VisDA-2017, thus showing the effectiveness of our learnable embedding prototypes for improving target feature discrimination. 3) *Mixed soft labels*. We further analyze how pseudo labels generated from the maintained generative classifier improve upon the auxiliary label $Q_{g,f}$ with mixed supervisions (i.e. w/o $\hat{Y}_T$). We can see there is a performance drop (VisDA-2017: $83.4\% \rightarrow 82.0\%$) by

Table 3. Classification accuracy (%) on DomainNet-126 for SSDA (ResNet-34).

Method	C→S		P→C		P→R		R→C		R→P		R→S		S→P		Avg.	
	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
ResNet-34	54.8	57.9	59.2	63.0	73.7	75.6	61.2	63.9	64.5	66.3	52.0	56.0	60.4	62.2	60.8	63.6
MinEnt [17]	56.3	61.5	67.7	71.2	76.0	78.1	66.1	71.6	68.9	70.4	60.0	63.5	62.9	66.0	65.4	68.9
BNM [13]	58.4	62.6	69.4	72.7	77.0	79.5	69.8	73.7	69.8	71.2	61.4	65.1	64.1	67.6	67.1	70.3
MCC [24]	56.8	60.5	62.8	66.5	75.3	76.5	65.5	67.2	66.9	68.1	57.6	59.8	63.4	65.0	64.0	66.2
PL	62.5	64.5	67.6 1	70.7	78.3	79.3	70.9	72.9	69.2	70.7	62.0	64.8	67.0	68.6	68.2	70.2
ATDOC-NC [36]	58.1	62.2	65.8	70.2	76.9	78.7	69.2	72.3	69.8	70.6	60.4	65.0	65.5	68.1	66.5	69.6
GeT	**66.7**	**67.8**	**73.9**	75.8	**82.0**	**82.8**	**76.1**	**77.6**	**72.5**	**73.9**	**66.8**	67.1	**69.8**	**73.6**	**72.2**	**73.9**
MME [55]	56.3	61.8	69.0	71.7	76.1	78.5	70.0	72.2	67.7	69.7	61.0	61.9	64.8	66.8	66.4	68.9
APE [26]	56.7	63.1	72.9	76.7	76.6	79.4	70.4	76.6	70.8	72.1	63.0	67.8	64.5	66.1	67.6	71.7
S^3D [69]	60.8	64.4	73.4	75.1	79.5	80.3	73.3	75.9	68.9	72.1	65.1	66.7	68.2	70.0	69.9	72.1

Table 4. Classification accuracy (%) on Office-Home for PDA (ResNet-50).

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Mean
ResNet-50	43.5	67.8	78.9	57.5	56.2	62.2	58.1	40.7	74.9	68.1	46.1	76.3	60.9
MinEnt [17]	45.7	73.3	81.6	64.6	66.2	73.0	66.0	52.4	78.7	74.8	56.7	80.8	67.8
BNM [13]	54.6	77.2	81.1	64.9	67.9	72.8	62.6	55.7	79.4	70.5	54.7	77.6	68.2
MCC [24]	54.1	75.3	79.5	63.9	66.3	71.8	63.3	55.1 5	78.0	70.4	55.7	76.7	67.5
PL	51.9	70.7	77.5	61.7	62.4	67.8	62.9	54.1	73.8	70.4	56.7	75.0	65.4
ATDOC-NC [36]	59.5	80.3	83.8	71.8	71.6	79.7	70.6	59.4	82.2	78.4	61.1	81.5	73.3
ATDOC-NA [36]	60.1	76.9	84.5	72.8	71.2	80.9	73.9	61.8	83.8	77.3	60.4	80.4	73.7
GeT	**61.4**	**81.2**	**85.9**	**74.0**	**74.1**	**82.3**	**75.8**	**63.9**	**85.3**	**79.6**	**63.7**	**84.6**	**75.8**
SAFN [68]	58.9	76.3	81.4	70.4	73.0	77.8	72.4	55.3	80.4	75.8	60.4	79.9	71.8
RTNet$_{adv}$ [12]	63.2	80.1	80.7	66.7	69.3	77.2	71.6	53.9	84.6	77.4	57.9	85.5	72.3

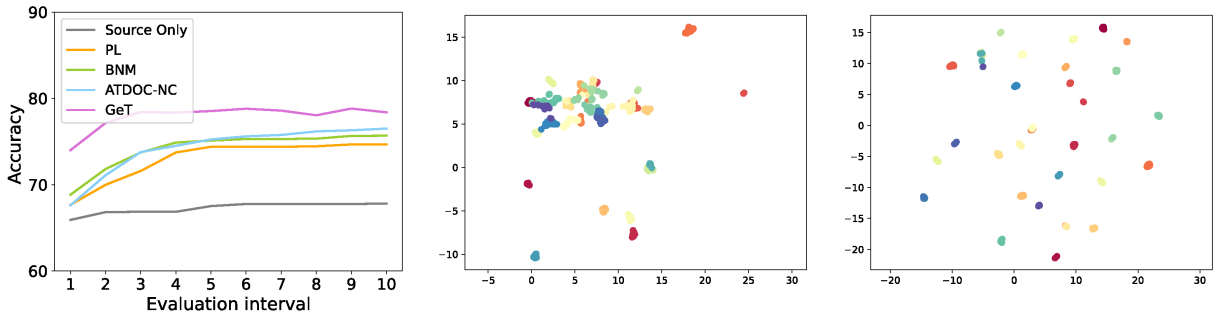

Figure 3. a) Comparison of convergence for the Ar → Pr task on Office-Home. Feature visualization with b) Source-only and c) GeT for the task A → W on Office-31. Note that different classes are denoted by different colors.

Table 5. Classification accuracy (%) on Office-Home for UDA and PDA under the imbalanced target distribution (ResNet-50).

Setting	UDA					PDA				
Method	A→ C	C→P	P→R	R→A	Avg.	A→ C	C→P	P→R	R→A	Avg.
ResNet-50	44.3	62.4	72.6	64.3	59.5	49.1	59.9	76.1	70.2	63.8
BNM [13]	55.9	70.9	78.7	70.4	67.5	53.8	63.7	78.9	70.7	67.6
MCC [24]	48.5	66.8	75.1	67.6	63.1	52.4	61.1	75.7	70.1	65.6
PL	53.8	68.5	76.5	69.2	65.4	49.9	61.5	72.8	68.0	62.9
ATDOC-NC [36]	52.5	72.5	78.6	69.7	67.1	55.6	71.8	81.6	73.9	70.9
GeT	**56.1**	**74.8**	**81.9**	**71.6**	**70.2**	**57.8**	**73.8**	**85.9**	**77.2**	**75.3**

Table 6. Classification accuracy (%) of GeT on Office-31 and VisDA-2017 under different variants. (ResNet-50)

	NC	GeT w/o $\mathcal{L}_{\mathrm{KL}}^{\{f,g\}}$	GeT w/o $\mathcal{L}_{\mathrm{KL}}^{f}$	GeT w/o $\hat{Y}_T$	GeT
Office-31	84.0	85.1	84.5	85.6	86.0
VisDA-2017	80.3	81.4	80.6	82.0	83.4

removing the guidance from the online generative classifier. The mixed soft labels performs better than each separate supervision from $\hat{Y}_{\mathbb{M}}$ and $Q_{g,f}$.

Convergence comparison. We study the convergence of GeT by plotting test accuracy versus iteration number for Ar → Pr task on Office-Home in Fig. 3. Comparing GeT with other baselines, GeT converges more quickly and the performance remains stable thereafter. This observation demonstrates that our GeT can provide more reliable pseudo labels in the early stage and performs better regularization on target data for discrimination.

Visualization. We visualize the target features learned by Source Only and GeT for the task A → W on Office-31 in Fig. 3 with UMAP [45] plot. It is obvious that our GeT provides better prototypes for target discrimination with clear boundaries among classes.

Sensitivity Analysis. We also evaluate the sensitivity of our model to two hyper-parameters γ_π and γ_Q, i.e. the

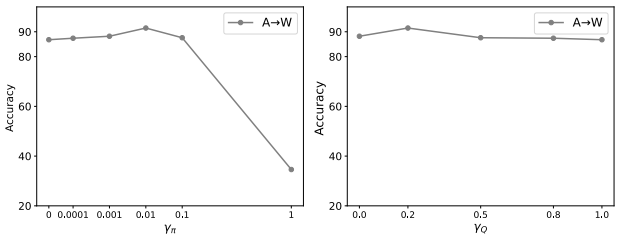

Figure 4. Average accuracy of our GeT with a) varying γ_π when $\gamma_Q = 0.2$, and b) varying γ_Q when $\gamma_\pi = 0$ on A→W in imbalanced Office-31.

memory decay for the class prior and the mixup coefficient for the soft labels in Fig. 4. Particularly, we choose γ_π from [0,1] by setting γ_Q to 0.2. we then vary the value of γ_Q over the range $\{0,0.2,0.5,0.8,1\}$ with $\gamma_\pi = 0$. When γ_Q is set to 0 or 1, it is equivalent to the single supervision from the model output or the target-oriented classifier. The mixed pseudo labels produce better results as they combine the source domain knowledge learned from the model and the target domain knowledge. It can be observed that the accuracy of our model is not sensitive to both hyper-parameters in a relative wide range.

6. Conclusion

In this paper, we propose a new target structure regularization approach for the DA tasks to deal with the source data bias and class distribution bias problems. We provide a new perspective of enhancing target data discrimination by formulating a learnable generative classifier, where the parameters are updated efficiently online in mini-batches. To further uncover the debiased target feature discrimination, we introduce the structure similarity regularization on the model predictions and the embeddings by an auxiliary distribution and a set of learnable embedding prototypes. Extensive experiments demonstrate that our GeT outperforms other regularization methods, and some DA models with explicit feature alignment on several DA tasks with large class distribution bias.

Acknowledgement. This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-024), and the Tier 2 grant MOE-T2EP20120-0011 from the Singapore Ministry of Education.

References

[1] https://www.imageclef.org/2014/adaptation/.

[2] Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2018.

[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2985–2994, 2019.

[6] Yee Seng Chan and Hwee Tou Ng. Word sense disambiguation with distribution estimation. In *IJCAI*, volume 5, pages 1010–5. Citeseer, 2005.

[7] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[8] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7181–7190, 2022.

[9] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2090–2099, 2019.

[10] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International conference on machine learning*, pages 1081–1090. PMLR, 2019.

[11] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.

[12] Zhihong Chen, Chao Chen, Zhaowei Cheng, Boyuan Jiang, Ke Fang, and Xinyu Jin. Selective transfer with reinforced transfer network for partial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12706–12714, 2020.

[13] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3941–3950, 2020.

[14] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9944–9953, 2019.

[15] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[16] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE international conference on computer vision*, pages 5736–5745, 2017.

[17] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

[18] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Hynek Hermansky, Daniel PW Ellis, and Sangita Sharma. Tandem connectionist feature extraction for conventional hmm systems. In *2000 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1635–1638. IEEE, 2000.

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[22] Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, pages 934–940, 2020.

[23] Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 4816–4827. PMLR, 2020.

[24] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *European Conference on Computer Vision*, pages 464–480. Springer, 2020.

[25] Guoliang Kang, Yunchao Wei, Yi Yang, Yueting Zhuang, and Alexander Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. *Advances in Neural Information Processing Systems*, 33:3569–3580, 2020.

[26] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *European conference on computer vision*, pages 591–607. Springer, 2020.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[28] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020.

[29] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[30] Jonghyun Lee, Dahuin Jung, Junho Yim, and Sungroh Yoon. Confidence score for source-free unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 12365–12377. PMLR, 2022.

[31] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

[32] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *International Conference on Machine Learning*, pages 3763–3772. PMLR, 2019.

[33] Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *European Conference on Computer Vision*, pages 382–403. Springer, 2020.

[34] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. Semantic concentration for domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9102–9111, 2021.

[35] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[36] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2021.

[37] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pages 3122–3130. PMLR, 2018.

[38] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[39] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.

[40] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

[41] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017.

[42] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9111–9120, 2020.

[43] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

[44] Yishay Mansour and Mariano Schain. Robust domain adaptation. *Annals of Mathematics and Artificial Intelligence*, 71(4):365–380, 2014.

[45] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[46] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[47] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1094–1103, 2021.

[48] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

[49] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

[50] Pedro O Pinheiro, Negar Rostamzadeh, and Sungjin Ahn. Domain-adaptive single-view 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7638–7647, 2019.

[51] Danila Rukhovich and Danil Galeev. Mixmatch domain adaptaion: Prize-winning solution for both tracks of visda 2019 challenge. *arXiv preprint arXiv:1910.03903*, 2019.

[52] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[53] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[54] Aadarsh Sahoo, Rutav Shah, Rameswar Panda, Kate Saenko, and Abir Das. Contrast and mix: Temporal contrastive video domain adaptation with background mixing. *Advances in Neural Information Processing Systems*, 34, 2021.

[55] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019.

[56] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[57] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 459–466, 2012.

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[59] Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33:19276–19289, 2020.

[60] Shuhan Tan, Xingchao Peng, and Kate Saenko. Class-imbalanced domain adaptation: an empirical odyssey. In *European Conference on Computer Vision*, pages 585–602. Springer, 2020.

[61] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8725–8735, 2020.

[62] Korawat Tanwisuth, Xinjie Fan, Huangjie Zheng, Shujian Zhang, Hao Zhang, Bo Chen, and Mingyuan Zhou. A prototype-oriented framework for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34, 2021.

[63] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[64] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

[65] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[66] Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International Conference on Machine Learning*, pages 6872–6881. PMLR, 2019.

[67] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 5423–5432, 2018.

[68] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1426–1435, 2019.

[69] Jeongbeen Yoon, Dahyun Kang, and Minsu Cho. Semi-supervised domain adaptation via sample-to-sample self-distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1978–1987, 2022.

[70] Xiangyu Yue, Zangwei Zheng, Shanghang Zhang, Yang Gao, Trevor Darrell, Kurt Keutzer, and Alberto Sangiovanni Vincentelli. Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13834–13844, 2021.

[71] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827. PMLR, 2013.

[72] Yabin Zhang, Bin Deng, Kui Jia, and Lei Zhang. Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 781–797. Springer, 2020.

[73] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019.

[74] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.

[75] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. 2002.