Towards Uncovering the Intrinsic Data Structures for Unsupervised Domain Adaptation using Structurally Regularized Deep Clustering

Hui Tang, Xiatian Zhu, Ke Chen, Kui Jia, and C. L. Philip Chen, Fellow, IEEE

Abstract—Unsupervised domain adaptation (UDA) is to learn classification models that make predictions for unlabeled data on a target domain, given labeled data on a source domain whose distribution diverges from the target one. Mainstream UDA methods strive to learn domain-aligned features such that classifiers trained on the source features can be readily applied to the target ones. Although impressive results have been achieved, these methods have a potential risk of damaging the intrinsic data structures of target discrimination, raising an issue of generalization particularly for UDA tasks in an inductive setting. To address this issue, we are motivated by a UDA assumption of structural similarity across domains, and propose to directly uncover the intrinsic target discrimination via constrained clustering, where we constrain the clustering solutions using structural source regularization that hinges on the very same assumption. Technically, we propose a hybrid model of Structurally Regularized Deep Clustering, which integrates the regularized discriminative clustering of target data with a generative one, and we thus term our method as SRDC++. Our hybrid model is based on a deep clustering framework that minimizes the Kullback-Leibler divergence between the distribution of network prediction and an auxiliary one, where we impose structural regularization by learning domain-shared classifier and cluster centroids. By enriching the structural similarity assumption, we are able to extend SRDC++ for a pixel-level UDA task of semantic segmentation. We conduct extensive experiments on seven UDA benchmarks of image classification and semantic segmentation. With no explicit feature alignment, our proposed SRDC++ outperforms all the existing methods under both the inductive and transductive settings. We make our implementation codes publicly available at https://github.com/huitangtang/SRDCPP.

Index Terms—Domain adaptation, deep clustering, inductive learning, image classification, semantic segmentation.

1 INTRODUCTION

In many practical applications of machine learning, the problem of interest is concerned with learning from data on a domain where, due to practical constraints and/or expenses, data annotations are difficult to acquire, and a standard supervised training cannot be readily applied; instead, labeled data on a different but related domain can be obtained relatively easily. This creates a learning scenario in which one is tempted to leverage the labeled data on the source domain to help learn machine learning models for a transferrable use on the unlabeled target domain, i.e., the problem of unsupervised domain adaptation (UDA) [1], [2]. UDA typically assumes a shared label space between the source and target domains, and its technical challenge arises from the assumed existence of distribution divergence between the two domains.

A rich literature of UDA research has been developed in the past decades [3]. Among them, mainstream methods [4], [5], [6], [7], [8], [9], [10] are motivated by the seminal theories [3], [11], [12] that bound the expected errors of classification models on the target domain by quantities involving classifier-induced divergence between feature distributions of the two domains, e.g., those recent ones based on adversarial training of deep networks [5], [6], [7]. Consequently, these methods strive to minimize domain divergence by learning aligned features between the two domains, such that classifiers trained on the features of source domain can be readily applied to the target ones. Despite the impressive results achieved, these methods have a potential risk of damaging the intrinsic data structures of target discrimination, as analyzed recently in [13]. We note that more importantly, this shortcoming would become severer in the practically more useful setting of inductive UDA analogous to the setting of inductive transfer learning in [4], where the objective is to learn classification models as off-the-shelf ones such that they can be used for held-out data sampled from the same target domain; adapting classifiers to the damaged discrimination of target data by feature alignment of existing methods would be less effective for inductive UDA, since the held-out target data still follow the undamaged, intrinsic data discrimination.

To overcome such limitations in existing methods, we first revisit the general UDA assumptions made in existing research [2], [12], [14], and summarize those in [14] as the structural similarity between the source and target domains, which includes the notions of domain-wise discrimination and class-wise closeness. Simply put, the former notion assumes the existence of intrinsic structures of discriminative data clusters in individual domains, and the latter one assumes that clusters of the two domains corresponding to the same
semantic class are geometrically close. These assumptions motivate us to consider a UDA approach that directly uncovers the intrinsic discrimination of target data via constrained clustering, where we propose to constrain the clustering solutions using structural source regularization hinging on the very same assumptions.

Technically, we propose a hybrid model of deep clustering that integrates a regularized discriminative clustering with a generative one. Among various deep network based clustering algorithms, we choose a simple but flexible framework, which performs clustering by minimizing the Kullback-Leibler (KL) divergence between the predictive label distribution of the network and an introduced auxiliary one. For discriminative part of our hybrid model, the structural source regularization is simply achieved by training the same network layers of classifier using labeled source data, i.e., a strategy of joint network training; for the generative part, we learn cluster centroids in the deep feature space to enable probabilistic data modeling, and the structural source regularization is achieved by making the centroids common to the source and target domains, where we borrow ideas from set transformer and learn the cluster centroids in a feed-forward manner using self-attentive feature interactions of training instances. We empirically observe that generative clustering modulates the feature space learning, which potentially enhances the uncovering of intrinsic discrimination by providing benefits complementary to the discriminative ones. We term our method of Structurally Regularized Deep Clustering as SRDC++, to emphasize both its hybrid nature and its extension to the method of SRDC proposed in the preliminary version of this work.

In the present paper, we also contribute in a second UDA task of semantic segmentation, which is to learn domain-adapted models to classify each pixel in an input image into one of multiple semantic classes. We note that by treating observations at each pixel of the image as a data instance, our proposed SRDC++ can be readily applied. To further improve the performance, we borrow ideas from existing methods and propose to enrich our UDA assumption of structural similarity with a third notion of layout-wise consistency, which states that the spatial layout of semantic segmentation maps is consistent between the source and target domains. Implementing this notion into the SRDC++ objective gives our method for domain-adapted semantic segmentation. We present extensive experiments on the benchmark datasets of Office-31, ImageCLEF-DA, Office-Home, VisDA-2017, and Digits for image classification, and on the UDA tasks among GTA5, SYNTHIA, and Cityscapes for semantic segmentation. Our proposed SRDC++ outperforms all the existing methods under both the setting of inductive UDA and the more conventional setting of transductive UDA, where domain-adapted models are directly evaluated on the target data that are involved in the training.

1.1 Relations with Existing Works

In this section, we organize our brief review of existing methods into the following three categories. We also discuss their relations with our proposed one.

Domain Adaptation for Image Classification. There exists a huge literature of UDA methods for image classification. We focus our review on those representative ones, particularly those by learning aligned deep features and those incorporating modern techniques of deep clustering. One may refer to for a comprehensive review of existing methods.

To achieve domain adaptation, the methods learn deep features to reduce the classical measure of maximum mean discrepancy (MMD) across domains. A strategy of adversarial training is subsequently used in to further reduce the domain discrepancy. These methods are designed for domain-level alignment; however, reducing discrepancies of conditional distributions towards category-level alignments are more desired for better UDA. To this end, multiplicative interactions of feature and category predictions are used in to achieve the goal. The methods do so in an alternative manner by aligning feature means of individual categories between the source and target data. More recently, show that discriminative structures of target data may be degraded by adversarial feature alignment; they apply spectral decompositions to the instance features, and penalize the singular values corresponding to the singular vectors that learn aligned features. We are motivated by the same degradation of intrinsic target discrimination; we instead propose SRDC++ as a regularized deep clustering solution to address the issue.

Unsupervised domain adaptation is by nature to cluster the unlabeled target data, given regularization from the labeled source data. As such, principles of unsupervised learning, e.g., the cluster assumption, are typically applicable to UDA tasks. The cluster assumption states that the classification boundaries should not pass through high-density regions, but instead lie in low-density regions. To enforce the cluster assumption, conditional entropy minimization is widely used in the UDA community. Most of these methods use clustering of target data to help improve feature alignment; they apply spectral decompositions to the instance features, and penalize the singular values corresponding to the singular vectors that learn aligned features. We are motivated by the same degradation of intrinsic target discrimination; we instead propose SRDC++ as a regularized deep clustering solution to directly uncover the intrinsic discrimination of target data.

Domain Adaptation for Semantic Segmentation. Existing domain adaptation strategies for semantic segmentation are similar to those for image classification, with additional consideration such as constraints on the spatial layout of output segmentation maps. For example, Hoffman et al. adopt adversarial domain adaptation at both pixel and feature levels, by applying cycle-consistency and semantic consistency losses. In information bottleneck is utilized to eliminate nuisance factors and maintain pure semantic information in features, thus promoting feature alignment and stabilizing adversarial training. Vu et al. enforce structural consistency across domains by conditional entropy minimization and distribution matching in terms of weighted self-information maps. Our extension of SRDC++ for semantic segmentation...
incorporates these established domain knowledge into our regularized deep clustering framework. As far as we know, we are the first to use a direct clustering solution and achieve superior performance on the semantic segmentation task.

**Transductive or Inductive Domain Adaptation.** Existing research on UDA tasks does not consider the nuanced difference between the transductive and inductive settings. Comparisons are usually made on different benchmarks by following the setting conventions. For example, for image classification, existing methods report results in the transductive setting for the benchmarks of Office-31 [22], ImageCLEFDA [23], Office-Home [24], and VisDA-2017 [25], and report results in the inductive setting for the benchmark of Digits. The experiments on the semantic segmentation benchmarks of GTA5 [29], SYNTHIA [30], and Cityscapes [31] are also reported in the inductive setting. It is arguably more useful to study the inductive setting of UDA, since once trained in this setting, the adaptation models can be directly used for hold-out test sets. In this work, we report comprehensive experiments in both the inductive and transductive settings. We expect that our results contribute to the community as the new benchmarks.

**1.2 Contributions**

A preliminary version of this work has been published as an oral presentation in [19], where we have proposed the basic strategy of Structurally Regularized Deep discriminative Clustering (SRDC) for UDA tasks in the transductive setting. We re-state its main technical contributions as follows.

- To address a potential issue of damaging the intrinsic data discrimination by explicitly learning domain-aligned features, we propose a method of SRDC [19] that makes use of source-regularized, deep discriminative clustering to directly uncover the intrinsic structures of target discrimination. The method is motivated by our assumption of structural similarity between the two domains.
- We technically achieve SRDC based on a simple but flexible framework of deep clustering, which minimizes the KL divergence between the distribution of network prediction and an auxiliary one; replacing the auxiliary distribution with that of ground-truth labels of source data implements the structural source regularization via a simple strategy of joint network training.

In the present paper, we improve the method in [19] as a hybrid model of SRDC++, and extend SRDC++ for a second UDA task of semantic segmentation. We organize our evaluations in both the inductive and transductive UDA settings. SRDC++ achieves the new state of the art across a range of UDA benchmarks in both the settings. Our new contributions are summarized as follows.

- We propose a hybrid model of SRDC++ that integrates regularized discriminative clustering with a generative one. Regularized generative clustering is achieved by learning in a feed-forward manner a set of cluster centroids common to the source and target domains. We empirically verify that our learning of cluster centroids modulates the feature space learning, rather than align the features across domains; it enhances the uncovering of intrinsic data structures by providing benefits complementary to the discriminative one.
- Our proposed SRDC++ can be readily applied to a pixel-level UDA task of semantic segmentation. To further improve the performance, we enrich our UDA assumption of structural similarity with a third notion of layout-wise consistency. Implementing this notion into the learning objective of SRDC++ gives our method for domain-adapted semantic segmentation. Efficacy of this extension is verified empirically.
- To the best of our knowledge, we organize, for the first time, a comprehensive evaluation of different UDA methods in both the inductive and transductive settings. Across a range of UDA benchmarks for image classification and semantic segmentation, our proposed SRDC++ is consistently superior to existing ones under both the settings. Careful ablation studies also reveal the internal mechanism of SRDC++.

## 2 Problem Statement

In unsupervised domain adaptation (UDA), we assume a labeled set of examples \( \{(x_i^s, y_i^s)\}_{i=1}^n \) from a source domain \( S \), and an unlabeled set \( \{x_i^t\}_{i=1}^n \) from a target domain \( T \). The two domains share a common label space \( Y \). Let \( |Y| = K \), and we have \( y^t \in \{1, 2, \ldots, K\} \) for any source instance \( x^s \). There exist two typical settings in the literature of UDA: transductive UDA aims to learn prediction models that directly assign labels to the target instances \( \{x_i^t\}_{i=1}^n \) to \( y^t \), and inductive UDA is to measure performance of the learned models on held-out sets of instances that are sampled from the same \( T \). By convention, the two settings are respectively considered in the UDA tasks for object classification [4], [5], [6], [49] and semantic segmentation [21], [51], [52], [53].

### 2.1 Motivations for Uncovering the Intrinsic Target Discrimination

Given the discrepancies between the source and target domains, domain adaptation is less feasible without certain assumptions on their similarities. In fact, domain closeness is generally assumed in UDA either theoretically [2], [12] or intuitively [14]. In this work, we summarize the assumptions in [14] as the structural similarity between the source and target domains, which includes the following notions of domain-wise discrimination and class-wise closeness.

- **Domain-wise discrimination** assumes that there exist intrinsic structures of data discrimination in individual domains, i.e., data in either source or target domains are discriminatively clustered corresponding to the shared label space.
- **Class-wise closeness** assumes that clusters of the two domains corresponding to the same class label are geometrically close.

Based on these assumptions, many of existing works [4], [6], [34], [36], [37], [55] take the strategy of learning aligned feature representations between the two domains, such that classifiers trained on source features can be readily
applied to the target ones. However, such a strategy has a potential risk of damaging the intrinsic data discrimination on the target domain, as discussed in some recent works [13], [14], [59]. We note that more importantly, classifiers adapting to the damaged discrimination of target data would be less effective for tasks of inductive UDA, since in inductive UDA, the learned classifiers are expected to be used as off-the-shelf models for held-out target data, which still follow the undamaged, intrinsic data discrimination. As such, it is ideal that domain-adapted classifiers should be consistent when learning with different source domains, and they should not deviate too much from the oracle one obtained, e.g., by training on target samples with ground-truth labels, which is deemed to represent the intrinsic target discrimination.

Based on the above analysis, we are motivated to directly recover the intrinsic target discrimination via discriminative clustering of the target data. To leverage the labeled source data, we propose to constrain the clustering solutions using structural source regularization that hinges on our assumed structural similarity across domains. We note that quite a few recent methods [38], [46], [48] consider clustering of structural similarity across domains. We note that quite a few recent methods consider clustering of structural similarity across domains. In order to uncover the intrinsic discrimination of the target domain, we opt for direct clustering of target instances with structural regularization from the source domain. Among various clustering methods [15], [16], [17], we choose a flexible framework of deep discriminative clustering [17], which minimizes the Kullback-Leibler (KL) divergence between the predictive label distribution of the network and an introduced auxiliary one; by replacing the auxiliary distribution with that of ground-truth labels of source data, we easily implement the structural source regularization via a simple strategy of joint network training. We also enhance structural regularization with soft selection of less divergent source examples. Technical details are presented as follows.

### 3 The Proposed Method

#### 3.1 Deep Discriminative Target Clustering with Structural Source Regularization

In order to uncover the intrinsic discrimination of the target domain, we opt for direct clustering of target instances with structural regularization from the source domain. Among various clustering methods [15], [16], [17], we choose a flexible framework of deep discriminative clustering [17], which minimizes the Kullback-Leibler (KL) divergence between the predictive label distribution of the network and an introduced auxiliary one; by replacing the auxiliary distribution with that of ground-truth labels of source data, we easily implement the structural source regularization via a simple strategy of joint network training. We also enhance structural regularization with soft selection of less divergent source examples. Technical details are presented as follows.

#### 3.1.1 Deep Discriminative Target Clustering

Given the unlabeled \( \{x_i^n\}_{i=1}^n \), the network \( f \circ \varphi \) outputs probability vectors of \( P_t = \{p_t^n\}_{n=1}^N \). To implement deep discriminative clustering [17], [56], we first introduce an auxiliary counterpart \( Q_t = \{q_t^n\}_{n=1}^N \). The proposed method then alternates in (1) updating \( Q_t \), and (2) using the updated \( Q_t \) as labels to train the network to update parameters \( \{\theta_{\varphi}, \theta_f\} \), by optimizing the following objective of deep discriminative clustering

\[
\min_{Q_t}(\theta_{\varphi}, \theta_f) \mathcal{L}_{f \circ \varphi}^* = \text{KL}(Q_t^* || P_t^*) + \sum_{k=1}^K \alpha_k^* \log \phi_k^*,
\]
where $\text{KL}(\cdot\|\cdot)$ defines KL divergence between the two discrete probability distributions $P^t$ and $Q^t$, which is spelled out as

$$\text{KL}(Q^t\|P^t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{k=1}^{K} q_{i,k}^t \log \frac{q_{i,k}^t}{p_{i,k}^t},$$

and $\phi_k^t$ in the second term is computed as

$$\phi_k^t = \frac{1}{n_t} \sum_{i=1}^{n_t} q_{i,k}^t.$$

The use of second term in Eq. (1) is to encourage entropy maximization of the averaged $K$ probability predictions over the $n_t$ instances, such that the assignments of $\{p_i^t\}_{i=1}^{n_t}$ (via $\{q_i^t\}_{i=1}^{n_t}$) into the $K$ clusters are balanced; otherwise as suggested by [57], degenerate solutions would be obtained that merge clusters by removing cluster boundaries. Given the lack of prior knowledge about target label distribution, we simply rely on this term to account for a uniform one.

Optimizing Eq. (1) takes the following alternating steps.

- **Auxiliary distribution update.** Fix network parameters $\{\theta_s, \theta_f\}$ and $\{p_i^s\}_{i=1}^{n_s}$ of target instances. Setting the (approximate) gradient of Eq. (1) as zero, the following closed-form solution is given by [17]

$$q_{i,k}^t = \frac{p_{i,k}^t / (\sum_{i'=1}^{n_s} p_{i',k}^t)}{\sum_{k'=1}^{K} p_{i,k'}^t / (\sum_{i'=1}^{n_s} p_{i',k'}^t)^2}.$$  \hspace{1cm} (2)

- **Network update.** By fixing $Q^t$, this step is equivalent to training the network via a cross-entropy loss using $Q^t$ as labels, giving rise to

$$\min_{\theta_s, \theta_f} -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{k=1}^{K} q_{i,k}^t \log p_{i,k}^t.$$  \hspace{1cm} (3)

**Remarks.** Given unlabeled target data alone, the objective (1) itself is not guaranteed to have sensible solutions to uncover the intrinsic discrimination of target data, since the auxiliary distribution $Q^t$ could be arbitrary whose optimization is subject to no proper constraints. To guarantee sensible solutions, deep clustering methods [17, 58] usually employ an additional reconstruction loss as a data-dependent regularizer. In this work, we introduce the following structural source regularization that serves a similar purpose as that of the reconstruction ones used in [17, 58].

### 3.1.2 Structural Source Regularization by Learning a Common Model of Classifier

Based on the UDA assumption of structural similarity made in Section 2, we propose to regularize the clustering of target data simply by training the same network $f \circ \varphi$ using labeled source data, i.e., a strategy of joint network training. Note that the $K$-way classifier $f$ defines hyperplanes that partition the feature space $Z$ into regions whose number is bounded by $2^K$, and $K$ ones among them are uniquely responsible for the $K$ classes. Given that the two domains share the same label space, joint training would ideally push instances of the two domains from any class into a same region in $Z$, thus implicitly achieving adaptation between the two domains.

Technically, for the labeled source data $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, we simply replace the auxiliary distribution in Eq. (1) with that formed by the ground-truth labels $\{(y_i^s)_{i=1}^{n_s}\}$, resulting in a supervised network training via cross-entropy minimization

$$\min_{\theta_s, \theta_f} -\frac{1}{n_s} \sum_{i=1}^{n_s} \log p_{i,y_i^s}^s.$$  \hspace{1cm} (4)

Supervised training enables us to learn a discriminative feature space defined by class labels, and we simply consider all source examples with the same label as a cluster, i.e., the clusters are semantically defined.

**A Soft Selection of Source Examples.** The class-wise closeness assumed in the structural similarity across domains also suggests that, depending on the distances of different source instances to the target domain, their regularization effects may vary. This motivates us to weight different examples in $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ based on their respective distances to the corresponding target clusters, i.e., a strategy of soft sample selection. Similar ideas are adopted to address the issue of sample selection bias in [59, 60]. To this end, we first compute the $K$ cluster centroids $\{p_i^t \in Z\}_{i=1}^{K}$ of target instances in the feature space $Z$ via $k$-means clustering, where initial cluster assignments of target features $\{z_i^t\}_{i=1}^{n_t}$ are based on network predictions $\{p_i^t = f(z_i^t)\}_{i=1}^{n_t}$ — in other words, any $z_i^t$ is assigned to the prediction of cluster $\tilde{y}_i^t \in \{1, \ldots, K\}$ when $p_{\tilde{y}_i^t}^t$ is the largest element in $p_i^t$. For any source example $(x^s, y^s)$, we then compute its weight $w^s$ of soft selection based on the cosine similarity between the feature $z^s$ and the target centroids $\mu_{y^s}$ of cluster $y^s$, i.e.,

$$w^s = \frac{1}{2} \left(1 + \mu_{y^s}^\top z^s / (||\mu_{y^s}|| ||z^s||)\right) \in [0, 1],$$  \hspace{1cm} (5)

where we sum the cosine similarity with a constant 1 in parenthesis to shift the weight in $[0, 1]$. Eq. (5) gives the weights $\{w_i^s\}_{i=1}^{n_s}$ for source examples $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$. We use $\{w_i^s\}_{i=1}^{n_s}$ to weight the loss (4), giving rise to

$$\min_{\theta_s, \theta_f} L^s_{f \circ \varphi} = -\frac{1}{n_s} \sum_{i=1}^{n_s} w_i^s \log p_{i,y_i^s}^s.$$  \hspace{1cm} (6)

By doing so, we potentially improve the structural source regularization by better adapting to the intrinsic structures of target data. Note that we update the target cluster centroids $\{\mu_k^t\}_{k=1}^{K}$, and consequently the weights $\{w_i^s\}_{i=1}^{n_s}$ computed by Eq. (5), iteratively during network training (practically per training epoch), which makes the soft selection evolve with feature learning.

Combining the target clustering loss (1) with the source regularization (6) gives our objective of structurally Regularized deep Discriminative Clustering (SRDisC)

$$\min_{Q^t, \theta_s, \theta_f} L_{\text{SRDisC}} = L^s_{f \circ \varphi} + \lambda L^s_{f \circ \varphi},$$  \hspace{1cm} (7)

where $\lambda$ is a penalty parameter.

### 3.2 Modulating the Intrinsic Target Structures via Generative Learning with Self-Attentive Feature Interactions

Features learned by the objective (7) of regularized discriminative clustering tend to be amenable to domain adaptation, since the objective ideally pushes source and target instances
of the same classes/clusters into the respective same regions in the feature space \( \mathcal{Z} \), where generative modeling of data distributions has not been taken in account yet. In this section, we aim to further modulate the feature space learning from a generative perspective, by learning from the embedding \( \phi : \mathcal{X} \to \mathcal{Z} \) a set of \( K \) cluster centroids common to the two domains \([1]\) Learning cluster centroids enables probabilistic assignments of data instances to clusters, which potentially enhances the uncovering of intrinsic data structures in a manner complementary to the discriminative one in Section 3.1. To enable feed-forward learning of cluster centroids, we technically rely on self-attentive feature interactions of data instances. Details are presented as follows.

### 3.2.1 Deep Generative Target Clustering

Assume the availability of a parametric mapping \( \phi : \mathcal{Z}^n \to \mathbb{R}^{d \times K} \), which learns from \( n \) \( d \)-dimensional instance features in \( \mathcal{Z} \) to produce a set of \( K \) cluster centroids, denoted as \( \{c_k \in \mathbb{R}^d \}_{k=1}^K \). We implement \( \phi \) as a trainable feed-forward subnetwork parameterized by \( \theta_\phi \), as illustrated in Fig. 2 whose details are presented shortly in Section 3.2.2. To enable generative modeling of target data, we compute the following probability that softly assigns any \( z^t = \varphi(x^t) \) of target instance to the cluster \( k \)

\[
\hat{p}_k^t = \frac{\exp((1 + ||z^t - c_k||^2)^{-1})}{\sum_{k'=1}^{K} \exp((1 + ||z^t - c_{k'}||^2)^{-1})}. \tag{8}
\]

Eq. \( 8 \) is a variant of Student t-distribution \([58], [61]\) and we have \( \hat{p}_k^t \to \epsilon/(\epsilon + K - 1) < 1 \), which scales down with the increase of the cluster number \( K > 1 \)— values of \( \hat{p}_k^t \) are in fact very small for typical domain adaptation tasks that have more than a few dozens of classes. For ease of presentation, we compactly write the soft assignment probabilities of \( z^t \) to all the \( K \) clusters as \( \hat{p}^t \in [0, 1]^K \), and write \( \hat{P}^t = \{\hat{p}_k^t\}_{k=1}^n \) for all the target instances.

Similar to the objective \( 1 \) of deep discriminative clustering, we introduce an auxiliary distribution \( \hat{Q}^t = \{q_i^t\}_{i=1}^n \) to match \( \hat{P}^t \), giving rise to

\[
\min_{\hat{Q}^t, \{\theta_\phi, \theta_\o\}} \mathcal{L}_{\hat{Q}^t, \phi} = \text{KL}(\hat{Q}^t \parallel \hat{P}^t) + \sum_{k=1}^K \sum_{i=1}^n \hat{q}_{ki} \log \hat{q}_{ki}, \tag{9}
\]

where \( \hat{q}_{ki} \) is defined similarly as \( q^t_{ki} \) in \( 1 \). Optimization of \( 9 \) is again conducted by alternating in updating \( Q^t \), and using the updated \( Q^t \) as labels to train the network \( \phi \circ \varphi \) to update the parameters \( \{\theta_\phi, \theta_\o\} \), similar to the self-training strategies popularly used in recent methods \([35], [45], [62]\). It is obvious that the choice of auxiliary \( \hat{Q}^t \) is crucial for the learning success of \( 1 \); given that the cluster centroids \( \{c_k\}_{k=1}^K \) are also involved in the learning, without any constraints, learning could end with less sensible solutions that would not model intrinsic structures of target data generatively. To remedy, we again rely on labeled source data to impose structural regularization, as presented shortly.

### 3.2.2 Structural Source Regularization by Learning a Common Set of Cluster Centroids

Based on the class-wise closeness assumed in Section 2 we propose to regularize the generative target clustering \( 9 \) by making the learning of cluster centroids \( \{c_k\}_{k=1}^K \) common to the labeled source data \( \{(z^s_i, y^s_i)\}_{i=1}^n \). Similar to \( 8 \), the probability of softly assigning any \( z^s \) to its ground-truth centroid \( c_{y^s} \) is written as

\[
\hat{p}_{y^s}^s = \frac{\exp((1 + ||z^s - c_{y^s}||^2)^{-1})}{\sum_{k=1}^K \exp((1 + ||z^s - c_k||^2)^{-1})}. \tag{10}
\]

Given the definition in \( 10 \), a cross-entropy loss can be imposed on \( \{(z^s_i, y^s_i)\}_{i=1}^n \), which is equivalent to maximizing the probability of assigning any \( z^s \) to its true cluster \( y^s \). We also use the weights \( \{w_i^s\}_{i=1}^n \) computed by \( 5 \) to re-weight the loss contributions from individual source instances, resulting in

\[
\min_{\theta_\phi, \theta_\o} \mathcal{L}_{\phi \circ \o} = -\frac{1}{n_s} \sum_{i=1}^{n_s} w_i^s \log \hat{p}_{y_i^s}^s. \tag{11}
\]

Regularizing the generative learning of \( \{c_k\}_{k=1}^K \) via \( 11 \) makes the optimization well conditioned, since cluster assignments of \( \{z^s_i\}_{i=1}^n \) have been determined by their ground-truth labels \( \{y^s_i\}_{i=1}^n \).

Combining \( 9 \) with the source regularization \( 11 \) gives our objective of Structurally Regularized deep Generative Clustering (SRGenC)

\[
\min_{\hat{Q}^t, \theta_\phi, \theta_\o} \mathcal{L}_{SRGenC} = \mathcal{L}_{\phi \circ \o}^t + \lambda \mathcal{L}_{\phi \circ \o}^s, \tag{12}
\]

where \( \lambda \) is the penalty parameter that we take the same value as for \( 7 \).

**Remarks.** Equations \( 8 \) and \( 10 \) are based on a variant of Student t-distribution. The original Student t-distribution used in \([58], [61]\) has the effect of preserving the distances between instances and cluster centroids when they are moderately dissimilar; our variant by converting it as an exponential function tends to magnify the effect. We empirically observe that for \( z^t \) with \( \text{arg max}_{k} \hat{p}_k^t = k \) or \( z^s \) with its true label \( y^s = k \), they are learned to drift together with its cluster centroid \( c_k \) in the feature space, rather than to collapse to \( c_k \). In some cases, the class-wise distances between the source and target domains are even getting larger by applying \( 8 \) and \( 10 \), suggesting that the SRGenC objective \( 12 \) is indeed modulating the feature space learning via generative clustering, which is in contrast to existing methods \([4], [5], [6], [8], [38], [48]\) that explicitly align the features across the two domains. Results of these empirical studies are presented in Fig. 5.

### 3.2.3 Learning Cluster Centroids via Self-Attentive Feature Interactions

We have so far assumed that the parametric mapping \( \phi : \mathcal{Z}^n \to \mathbb{R}^{d \times K} \) has been given, which learns the \( K \) cluster centroids \( \{c_k \in \mathbb{R}^d\}_{k=1}^K \) in a feed-forward manner. Let \( C = \{c_1, \ldots, c_K\} \in \mathbb{R}^{d \times K} \). Considering that the
input to the learning of $\phi$ contains the set of features $Z = [Z^1, Z^2] \in \mathbb{R}^{d \times n}$, with $Z^i = [z_i^1, \ldots, z_i^n] \in \mathbb{R}^{d \times n}$, and $n = n_a + n_t$, we choose to adopt a framework of set transformer [18], [63] for this purpose (cf. [18] for its time complexity analysis during training). It uses a self-attention mechanism to learn from instance features in $Z$ to produce $C$. By encoding pairwise and even higher-order interactions among element features in $Z$, the framework is beneficial to iterative clustering in a feed-forward deep network. Fig. 2 gives the illustration.

Technically, the framework relies on the following parametric mapping of Multihead Attention (MA) [63]

$$
\phi^{MA}(Z^1, Z^2, Z^3) = W^{4T} [Z^1; \ldots; Z^m] 
$$

s.t. $Z^i = W_i^{3T} Z^3 \sigma \left((W_i^{1T} Z^1)^T (W_i^{2T} Z^2)/\sqrt{d}\right)$, $i = 1, \ldots, m$.

Let $Z^1 = Z^2 = Z^3 = Z$. $\{W_1^1, W_2^1, W_3^1 \in \mathbb{R}^{d \times \frac{d}{m}}\}$ and $W^4 \in \mathbb{R}^{d \times d}$ are the trainable parameters, and we collectively write $\theta_{\phi^{MA}} = \{\{W_1^1, W_2^1, W_3^1\}_{i=1}^{m}, W^4\}$. We have the intermediate $Z_i^1 \in \mathbb{R}^{\frac{d}{m} \times n}$. Eq. (13) computes, after projecting the $d$-dimensional element features in $Z$ into $d/m$ ones, the self similarities among the elements in $Z$, and then uses the obtained similarities to re-mix the elements, where $\sigma(\cdot)$ is the softmax function that applies to the matrix-formed argument in a row-wise manner. The following parametric mapping of Multihead Attention (MAB) can be built upon the mapping (13)

$$
\phi^{MAB}(Z^1, Z^2) = Z^1 + \phi^{MLP}(Z^2) 
$$

s.t. $Z^2 = Z^1 = \phi^{MA}(Z^1, Z^2, Z^3)$, where $\phi^{MLP}(\cdot)$ is a parametric mapping function implemented as a Multi-layer Perceptron (MLP), whose parameters are denoted as $\theta_{\phi^{MLP}}$. We write parameters of the MAB collectively as $\theta_{\phi^{MAB}} = \{\theta_{\phi^{MA}}, \theta_{\phi^{MLP}}\}$. Let $Z^1 = Z^2 = Z$.

Eq. (14) outputs a feature matrix of equal size, which contains the information of pairwise interactions among the elements in $Z$. By stacking multiple such blocks, information about higher-order interactions can be also encoded.

We finally have the feed-forward function $\phi : Z^n \to \mathbb{R}^{d \times K}$ for learning cluster centroids, defined as

$$
\phi(Z, C^0) := \phi^{MAB_2}(C', C') 
$$

s.t. $C' = \phi^{MAB_2}(C', \hat{\phi}^{MLP}(\phi^{MAB_1}(Z, Z)))$, where $\hat{\phi}^{MLP}(\cdot)$ denotes another MLP parameterized by $\theta_{\phi^{MLP}}$, and $C^0 \in \mathbb{R}^{d \times K}$ contains $K$ initial learning seeds, which we initialize by sampling their $d$ entries from a normal distribution. Intuitively, individual seeds aggregate instance features by self-attentive interactions, and cluster centroids would be finally obtained with the learning. The intermediate $C' \in \mathbb{R}^{d \times K}$ in (15) is already in the form of cluster centroids, and the final $C = \phi(Z, C^0)$ by an additional MAB mapping may further improve the clustering purity, as discussed in [18].

Overall, our subnetwork $\phi$ of learning cluster centroids via self-attentive feature interactions is parameterized by $\theta_\phi = \{\theta_{\phi^{MAB}_1}, \theta_{\phi^{MAB}_2}, \theta_{\phi^{MLP}}, \theta_{\phi^{MA}}, \theta_{\phi^{MLP}}\}$. Note that this auxiliary subnetwork is only used during training, which helps to modulate the learning of feature space $Z$ via the SRGenC objective (12), during testing, it is discarded and only the network $\phi$ used for inference.

### 3.3 Overall Training and Inference
The overall training objective combines (7) for structurally regularized discriminative clustering and (12) for the generative counterpart, resulting in a hybrid model of Structurally Regularized Deep Clustering. We term the method as SRDC++ to emphasize both its hybrid nature of regularized discriminative and generative clusterings, and the extension to the method of SRDC proposed in our preliminary version [19]. By writing the network parameters collectively as $\theta = \{\theta_\phi, \theta_\theta, \theta_\phi\}$, we have

$$
\min_{Q', Q, \theta} \mathcal{L}_{SRDC++} = \mathcal{L}_{SRDCsC} + \mathcal{L}_{SRGenC}. 
$$

Note that the objective (16) is applied to the whole network (cf. Fig. 1) in an end-to-end fashion, which learns all the parameters in $\theta$ simultaneously. We summarize the training process of SRDC++ using stochastic gradient descent in Algorithm 1. During inference, the trained network is used to classify any testing instance on the target domain via a simple forward pass of $f \circ \varphi$.

### 4 An Extension for Domain-Adapted Semantic Segmentation
In this section, we present an extension of our proposed SRDC++ for the task of semantic segmentation of images
in a UDA setting. Fig. 3 illustrates the modified network architecture. The task is by nature to classify each pixel in an input image into one of multiple semantic classes. To facilitate the discussion, we inherit most of the math notations used in the previous sections, and override some of them when the context requires. Assume the input image is of the size $h \times w$, we now denote the labeled source data as $\{(x_{ij}^s)_{j=1}^{h_w}, (y_{ij}^s)_{j=1}^{h_w}\}_{i=1}^{n_s}$, and the unlabeled target ones as $\{(x_{ij}^t)_{j=1}^{h_w}, (y_{ij}^t)_{j=1}^{h_w}\}_{i=1}^{n_t}$, in other words, each instance $x$ now represents the observations at a pixel, and a set $\{x_{ij}\}_{j=1}^{a}$ represents an image. As illustrated in Fig. 3, our feature extractor $\varphi$ reduces the input resolution of $h \times w$ via multiple layers of convolution and pooling, resulting in $d$ feature maps of the size $h/a \times w/a$ for each input image, where $a$ is the ratio of subsampling, and we denote $h^a = h/a$ and $w^a = w/a$; for simplicity, we assume that they have no rounding issue. The feature maps are passed through the classifier $f$ and upsampled via bilinear interpolation to have the network output of the size $h \times w \times K$, i.e., one segmentation map per class of the total $K$ classes. We correspondingly write $\{(p_{i,j}^s = f \circ \varphi(x_{ij}^s) \in [0, 1]^h_w)_{j=1}^{h_w}\}_{i=1}^{n_s}$ for the source data, and the same applies to the target ones; we also write $p_{i,j,k}$ for the $k^{th}$ element of $p_{i,j}$.

Given these definitions, the component of SRDisC objective (7) in SRDC++ can be readily applied. To apply the component of SRGenC (12) that operates in the feature space $Z$, a subtle issue is that $Z$ is now defined on the feature maps of reduced resolution, which is incompatible with the original resolution of segmentation maps on which the ground-truth labels of source data are defined. To address this issue, we note that during back-propagation of network training, each pixel located at the feature maps receives supervision signals from a field of the size $a \times a$ in the output segmentation maps; in the extreme case, all the $K$ classes may appear in such an $a \times a$ local field. We thus propose a weighted combination scheme that enables receiving supervision from any class that appears in such a local receptive field, by overriding the structural source regularization term $L_{\phi o \varphi}^s$ in the SRGenC objective (12) as follows

$$\min_{\theta_o, \theta_o} L_{\phi o \varphi}^s = \frac{1}{n_s h^a w^a} \sum_{i=1}^{n_s} \sum_{j=1}^{h^a w^a} \sum_{k=1}^{K} \sum_{k=1}^{K} \frac{\tau_{i,j}(k)}{a^2} \log \hat{p}_{i,j,k}, \tag{17}$$

where $\tau_{i,j}(k)$ counts the pixel number of the $k^{th}$ class in a local $a \times a$ field in the ground-truth segmentation map, which corresponds to the $j^{th}$ pixel location in feature maps of the $i^{th}$ training source image. Fig. 4 illustrates the scheme.

Our proposed SRDC++ is motivated from the assumption of structural similarity between the source and target domains, which includes the notions of domain-wise discrimination and class-wise closeness specified in Section 2.1. For the task of adapting semantic segmentation across domains, a basic strategy exists that accounts for the spatial segmentation layout of the structured output maps [20], [21]. We thus add a third notion into our structural similarity assumption, as follows.

- **Layout-wise consistency** assumes that the spatial layout of semantic segmentation maps is consistent between the source and target domains.

To implement this notion into our learning objective, we first define a weighted self-information map $O = \{o_j \in \mathbb{R}^{h_w}_j\}_{j=1}^{K}$ for any input image $\{x_j\}_{j=1}^{h_w}$, which follows [21]; the $k^{th}$ element of any $o_j$ in an $i^{th}$ image is computed as

$$o_{i,j,k} = -p_{i,j,k} \log p_{i,j,k}. \tag{18}$$

Thus $o_{i,j,k}$ represents the entropy of assigning the pixel $x_{i,j}$ to the $K$ classes. We implement the notion of layout-wise consistency by statistically enforcing the consistency between the distributions of $\{O_{i,j}^s\}_{i=1}^{n_s}$ and $\{O_{i,j}^t\}_{i=1}^{n_t}$. Technically, there exist many quantities that measure the first- or high-order statistics between the distributions, such as maximum mean discrepancy (MMD) [64] and central moment discrepancy (CMD) [65]; minimizing such quantities would promote the distribution consistency. In this work, we follow the adversarial training strategy used in the state-of-the-art methods [20], [21]. Define $g : \mathbb{R}^{K \times h_w} \rightarrow [0, 1]$, parameterized by $\theta_g$, as the domain discriminator that classifies the source
\( \{O^t_i\}_{i=1}^n \) as positive and \( \{O^t_i\}_{i=1}^n \) otherwise. We implement \( g \) as a convolutional subnetwork that applies to any map \( O \), which is followed by a sigmoid layer for binary classification. Since the spatial layout of the source \( \{O^s_i\}_{i=1}^n \) is determined by the ground-truth segmentation maps, we use the following objective to enforce consistency of the target layout to the source one

\[
\min_{\theta_s, \theta_y} L_{Layout} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \log g(O^s_i),
\]

where elements of any \( O^s_i \) are computed by \( \phi^s \), which involves network predictions from the function \( f_{\phi^s} \). We use the following objective to account for an adversarial training

\[
\min_{\theta_s} L_{Layout}^{Adv} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \log g(O^s_i) - \frac{1}{n_t} \sum_{i=1}^{n_t} \log(1-g(O^t_i)).
\]

Combining the objective \( L_{SRGenC} \) of SRDC++ with the above \( L_{Layout} \) and \( L_{Layout}^{Adv} \), gives our overall objective when applying the SRDC++ method to semantic segmentation

\[
\min_{\theta_s} \max_{Q', Q, \theta} L_{SRGenC}^{SRGenC} = L_{SRDIsC} + L_{SRGenC} + \beta L_{Layout}^{Adv} - \beta L_{Layout}^{Adv}.
\]

where \( \beta \) is a penalty parameter, and note that the regularization term \( L_{\phi^s} \) in \( L_{SRGenC} \) has been overcoded by \( \theta \).

5 Experiments of Domain Adaptation in an Inductive Setting

In this work, we verify the efficacy of our proposed SRDC++ on two UDA tasks of image classification and semantic segmentation. In the UDA literature, the former task is mainly conducted in a transductive setting, where classification of the unlabeled target data is achieved together with the model learning, and the latter one is mainly conducted in an inductive setting, where the learned model is to be applied to a held-out test set of target instances sampled from the same target domain. In this section, we first present experiments in the inductive setting for both the tasks of image classification and semantic segmentation; to the best of our knowledge, we are the first to report comprehensive experiments in the inductive setting on benchmark UDA datasets of image classification. Our experiments of image classification in the transductive setting are presented in Section 6.

Datasets. We use the following five UDA benchmarks for our experiments of image classification. Office-31 \([22]\) contains 4,110 images of 31 classes shared by three distinct domains, namely, Amazon (A), Webcam (W), and DSLR (D); they define six adaptation tasks by pair-wise domain combination. ImageCLEF-DA \([23]\) is a benchmark containing 600 images of 12 classes per data domain of the total three domains, namely, Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P); they again define six adaptation tasks by pair-wise domain combination. Office-Home \([24]\) is a more challenging benchmark, containing around 13,500 images of 65 classes shared by four distinct domains, namely, Art (Ar), Clipart (Cl), Product (Pr), and Real-World (Rw); they define 12 adaptation tasks by pair-wise domain combination. VisDA-2017 \([25]\) is a dataset for the difficult task of synthetic-to-real transfer (Synthetic→Real); it has images of 12 classes, including 152,397 synthetic ones and 55,388 natural ones. The benchmark of Digits includes three 10-class digit datasets of SVHN (S) \([26]\), MNIST (M) \([27]\), and USPS (U) \([28]\); SVHN has colored, extremely blurred images of real-world digits, MNIST has grayscale images of digits on clean background, and USPS has grayscale images with digits written in unconstrained styles; they define three conventional tasks of \( S \rightarrow M, M \rightarrow U, \) and \( U \rightarrow M. \)

We use the following three datasets for our experiments of semantic segmentation. GTA5 \([29]\) includes 24,966 synthetic images from the computer game of Grand Theft Auto V. SYNTHIA \([30]\) produces 9,400 synthetic images by rendering a virtual city using Unity engine. Cityscapes \([31]\) captures real street scenes, including images of 2,975 training ones, 500 validation ones, and 1,525 test ones. We follow the literature and evaluate two common adaptation tasks of GTA5 \( \rightarrow \) Cityscapes and SYNTHIA \( \rightarrow \) Cityscapes.

Settings and Implementation Details. We first present our settings and implementation details for experiments of image classification. For all adaptation tasks in each benchmark, we use all the data on the source domain as the training ones, and make a random, half-half splitting of training and test data for samples of each class on the target domain; the data settings are fixed once prepared. Note that for UDA in the inductive setting, each model is trained on the labeled source data and the training set of unlabeled target data, and is then evaluated on the target test set. To implement our SRDC++ for image classification, we use ImageNet \([66]\) pre-trained ResNet-50 \([67]\), with two newly added fully-connected (FC) layers respectively of 512 and \( K \) (i.e., the number of classes) neurons; the lower layers of the network are used as the feature extractor \( \varphi(\cdot) \), and the two upper FC layers are used as the classifier \( f(\cdot) \). The cluster centroid learner \( \varphi(\cdot) \) has three MABs (cf. Fig. 2), where we set the head number \( m = 4 \) and each MLP has three FC layers with ReLU non-linearity. Setting of the penalty \( \lambda \) in \( (7) \) and \( (12) \) follows the rule \( \lambda_i = 2(1 + \exp(-\gamma i))^{-1} - 1 \), where \( i \) is the epoch index normalized to \([0, 1] \) and \( \gamma = 10 \); the rule scales the values of \( \lambda \) up from 0 to 1 during the training and gradually switches the learning from the labeled source data to the unlabeled target ones, i.e., \( \lambda \) is empirically multiplied to losses on target data to suppress noises. We perform regularized deep clustering in a 2048-dimensional feature space (i.e., \( d = 2048 \)). Initializations of the weights \( \{w^s_i\}_{i=1}^n \) for soft selection of source samples and those of the auxiliary distributions \( q' \) and \( q' \) are given in Algorithm 1. When learning cluster centroids via the feed-forward function \( \phi \), we apply Batch Whitening \([45]\) to the respective feature batches of the two domains, which projects them into the same spherical distribution. We use SGD to train the feature extractor \( \varphi \) and classifier \( f \); where the pre-trained layers are fine-tuned and the newly added FC layers are trained from scratch; learning rates for the FC layers follow \( \eta_i = \eta_0 (1 + \alpha i)^{-\gamma} \), where \( i \) is the epoch index normalized to \([0, 1] \) and \( \eta_0 = 0.01, \alpha = 10, \gamma = 0.75 \), and those for the pre-trained layers are set as one tenth of the above schedule; we set the momentum, weight decay, batch size, and number of training epochs respectively as 0.9, 0.0001, 64, and 200. The cluster centroid learner \( \phi \) is trained
by Adam optimizer [68] with default parameters. We use data augmentations of random region crop and horizontal flip and perform three random trials during training; during testing, we report the averaged classification result of center region crops on all test images. For VisDA-2017, we follow the settings used in [6]; we set $\lambda = 1$, the initial learning rate $\eta_0 = 0.001$, and train for 20 epochs; we report per-class, mean accuracy on 12 classes of the dataset. For Digits, we follow the settings used in [6]; we adopt its same base network of a LeNet [27], and use Adam optimizer [68] with learning rate 0.0002 for all network parameters of $\phi$, $f$, and $\phi$, where we set the batch size as 128 and train for 200 epochs; we use the standard train/test split of each dataset, and do not use data augmentations; we report the averaged classification result over five random trials on test data.

For experiments of semantic segmentation, we follow the standard protocol where the validation set of Cityscapes is used for evaluation [20], [21], [53]. We report per-class IoU and mIoU on 19 and 16 classes respectively shared by GTA5 and SYNTHIA with Cityscapes. We use ResNet-101 based DeepLab-v2 [69] as the base network, where we set the subsampling ratio $a = 8$. We use the same domain discriminator $g(\cdot)$ as in [21]. Following [21], we set $\lambda = \beta = 0.001$. For this task of semantic segmentation, we do not do soft selection of source samples, i.e., $\{w^s_i = 1\}_{i=1}^m$. We use SGD to optimize the segmentation network $f \circ \phi$, where we set the learning rate, momentum, weight decay, batch size, and number of training iterations respectively as $2.5 \times 10^{-4}$, 0.9, $5 \times 10^{-4}$, 1, and 250,000. For $\phi$ and $g$, we use Adam [68] optimizer with learning rates $2.5 \times 10^{-4}$ and $10^{-4}$ respectively. The learning rate is scheduled by a polynomial decay of power 0.9. Other implementation details are the same as those for image classification.

Fig. 5. Learning diagnosis on the effect of the SRGenC objective (12) used in SRDC++. Three types of distances for the source and target data are plotted against the training epochs. Comparisons between our SRDC++ and MCD [6] are made in the two figures of last column. The experiments are conducted on the adaptation tasks of $A \rightarrow D$ and $D \rightarrow A$ on the Office-31 benchmark [22]. Refer to the main text for how these distances are defined and computed.

TABLE 1

| Component         | $A \rightarrow W$ | $A \rightarrow D$ | $D \rightarrow A$ | $W \rightarrow A$ | mean |
|-------------------|--------------------|--------------------|--------------------|--------------------|------|
| Source Only       | 79.3±1.1           | 81.6±0.6           | 63.1±0.7           | 65.7±0.8           | 72.4 |
| SRDisC            | 90.2±0.1           | 89.2±0.2           | 73.5±0.4           | 74.1±0.3           | 81.8 |
| SRDisC+SRGenC    | 91.8±0.3           | 92.3±0.2           | 76.0±0.5           | 76.0±0.3           | 84.0 |
| SRDisC+SRGenC+S  | 92.9±0.1           | 93.7±0.6           | 77.8±0.2           | 76.7±0.1           | 85.1 |

5.1 Ablation Studies and Learning Analyses

In this section, we present ablation studies and analyses of learning on our proposed SRDC++. All experiments are conducted on the Office-31 benchmark [22].

Ablation Studies. We first examine the effects of three key components in SRDC++, namely, Structurally Regularized deep Discriminative Clustering (SRDisC), Structurally Regularized deep Generative Clustering (SRGenC), and the scheme for Soft Selection of Source Samples ($S^3$). Experiments are conducted on the Office-31 benchmark [22] in an inductive setting.

Diagnoses of Learning. As discussed in Section 3.2.2, our SRGenC objective (12) is to modulate the feature space learning via generative clustering, rather than to explicitly align the features across domains, as in many of existing methods [4], [5], [6], [8], [38], [48]. To verify this empirically, we conduct experiments on the adaptation tasks of $A \rightarrow D$ and $D \rightarrow A$ on the Office-31 benchmark, and examine how the following three types of distances in the feature space evolve during training; they are the Euclidean distance
between each instance and its learned cluster centroid, averaged over all the instances (dubbed as Instance-to-Centroid), the distance between the class mean of instances and their centroid, averaged over all the classes (dubbed as SrcInsMean-to-TgtInsMean), where class assignments of target instances are based on their pseudo labels; the former two types of distances are computed separately for source and target data. In Fig. 5, we plot the evolving curves of these distances during training, by comparing with the representative domain-aligning method MCD [6].

We have the following observations from Fig. 5 (1) for both the source and target domains, Instance-to-Centroid and InsMean-to-Centroid are stabilized at certain levels of distances with the training, suggesting that our method does not enforce either source or target instances to collapse to the learned cluster centroids; (2) the SrcInsMean-to-TgtInsMean of MCD decreases linearly with the training, while that of our SRDC++ does not; in some cases (e.g., the D→A task of Fig. 5(f)), the SrcInsMean-to-TgtInsMean of SRDC++ even increases with the training, indicating that our method is indeed modulating the feature space learning towards uncovering the intrinsic discrimination of target data, rather than aligning the features across domains. Observations from experiments on other UDA tasks and benchmarks are of similar quality. These observations corroborate our discussions in Section 3.2.2.

Analysis of Convergence and Generalization. We use the adaptation tasks of A→D and D→A on the Office-31 benchmark to report the training convergence of SRDC++.

#### Table 2

Comparative results (%) in the inductive setting on the Office-31 benchmark [22]. All methods are based on the base model of ResNet-50.

| Method      | A→W | A→D | D→A | W→A            | mean       |
|-------------|-----|-----|-----|----------------|------------|
| Source Only | 79.3| 81.6| 63.1| 65.7           | 72.4       |
| DANN [9]    | 80.8| 82.4| 66.0| 64.6           | 73.5       |
| MCD [6]     | 86.5| 86.7| 72.4| 70.9           | 79.1       |
| SRDC [19]   | 91.9| 91.6| 75.6| 75.7           | 83.7       |
| SRDC++      | 92.9| 93.7| 77.0| 76.7           | 85.1       |
| Oracle      | 98.8| 97.6| 87.8| 87.8           | 93.0       |

#### Table 3

Comparative results (%) in the inductive setting on the ImageCLEF-DA benchmark [23]. All methods are based on the base model of ResNet-50.

| Method      | I→P | F→I | C→I | C→F | F→C | mean |
|-------------|-----|-----|-----|-----|-----|------|
| Source Only | 76.6| 90.7| 92.3| 88.9| 72.1| 92.2 |
| DANN [5]    | 77.2| 88.4| 92.9| 87.0| 72.9| 92.2 |
| MCD [6]     | 78.4| 88.9| 95.2| 88.3| 73.8| 92.1 |
| SRDC [19]   | 79.0| 91.0| 96.9| 90.2| 75.3| 93.9 |
| SRDC++      | 79.0| 92.3| 97.0| 92.6| 77.0| 94.7 |
| Oracle      | 79.3| 94.7| 97.2| 94.7| 79.5| 97.2 |

5.2 Image Classification

In this section, we report comparative results of image classification in the inductive setting on the benchmarks of Office-31 [22], ImageCLEF-DA [23], Office-Home [24], VisDA-2017 [25], and Digits. As most of existing methods do not report results in the inductive setting (except on the Digits benchmark), we choose the most representative methods of DANN [9] and MCD [6] to compare with; we also compare with our previous method of SRDC [19].

We apply these methods in the inductive setting to report results. We use the methods of Source Only (i.e., training the same classification model \( f \circ \varphi \) with labeled source data alone) and Oracle (i.e., training the model with labeled target data) as references of the lower and upper performance bounds, respectively. These results are reported in Tables 2, 3, 4, 5 and 6.

On all the five benchmarks, SRDC++ outperforms DANN and MCD, often with a large margin; it outperforms SRDC as well. On Digits, SRDC++ also outperforms existing methods of TPN, CyCADA, and GPDA. These results confirm the superiority of our proposed method for UDA in the inductive setting. There still exist large performance gaps between results of SRDC++ and those of the Oracle method, suggesting that inductive UDA is indeed a challenging setting; further innovations are expected to close the gaps. For qualitative analysis, we visualize the distributions of the learned features from SRDC++ and MCD on the W→A task of Office-31. Fig. 7 shows that SRDC++ learns a more discriminative feature space.

5.3 Semantic Segmentation

We use the objective (21) when applying SRDC++ to adaptation tasks of semantic segmentation, which combines the
terms of SRDisC and SRGenC, and also the two adversarial terms for promoting consistency of spatial layouts across domains (the two terms are dubbed as Layout). To investigate how these terms play roles in the learning, we conduct ablation studies by (1) removing the layout terms, which gives a method dubbed as SRDC++ w/o Layout, and (2) replacing the terms of SRDisC and SRGenC with a standard cross-entropy loss applied to the labeled source data, which we dub as SourceCE+Layout, and is exactly the method of AdvEnt [21]. We conduct experiments on the benchmark tasks of GTA5 → Cityscapes and SYNTHIA → Cityscapes. Results in Tables 7 and 8 show that compared with the SRDC++ objective [21], both ablations cause clear degradation of performance. The experiments confirm both the efficacy of our proposed hybrid model of structurally regularized deep clustering (i.e., SRDisC + SRGenC) for semantic segmentation, and its compatibility with the established practice of enforcing the layout-wise consistency.

In Tables 7 and 8, we also compare with the existing results reported in the literature, which are all achieved in the inductive UDA setting. On both of the two tasks, our proposed SRDC++ outperforms all the existing methods, confirming its efficacy for tasks other than image classification. However, the results are still largely behind those from the Oracle method, suggesting that semantic segmentation in an inductive UDA setting is very challenging; better approaches are to be developed to close the gaps. Example segmentation maps on the task of GTA5 → Cityscapes are given in Fig. 8.

### 6 Domain Adaptation in a Transductive Setting

In this section, we present experiments of domain-adapted image classification in the transductive setting to verify the efficacy of our proposed SRDC++.

#### Settings

We use the same four benchmarks of Office-31, ImageCLEF-DA, Office-Home, and VisDA-2017 for our experiments. Different from the inductive setting, we follow the literature of transductive UDA for image classification and use all the labeled source data and unlabeled target data when training the UDA models; results are directly compared on the learned label assignments of target data. Following the standard protocol in the literature, we use ResNet-50 as the base network for all the benchmarks except VisDA-2017, for which we use ResNet-101 as the base network. Other settings and implementation details are the same as those used in the inductive setting (cf. Section 5).

#### 6.1 Comparative Results

We compare SRDC++ with the state-of-the-art methods in Tables 9, 10, 11, and 12 including our previous SRDC [19]. While the performance of existing methods may vary on the
four benchmarks, SRDC++ performs consistently well and outperforms all existing methods on the four benchmarks. By comparing the results in these tables with those in Tables 2, 3, and 4 we observe that UDA tasks in the transductive setting are easier than the inductive counterparts; nevertheless, efficacy of SRDC++ is still verified on these easier
In this work, we have proposed a hybrid model of structurally regularized deep clustering, termed SRDC++, for unsupervised domain adaptation. The objective of SRDC++ includes two key components respectively for discriminative and generative clusterings of unlabeled target data, which are constrained by structural regularization from the labeled source data. The discriminative component is based on a deep clustering framework that minimizes the KL divergence between predictive and auxiliary distributions of network outputs. To enable generative clustering, SRDC++ learns domain-shared cluster centroids in the feature space via self-attentive interactions of instance features. In this work, we also extend SRDC++ for the adaptation task of semantic segmentation, by incorporating into its objective additional terms for promoting consistency of spatial layouts between the segmentation maps on the source and target domains. We report comparative results on UDA benchmarks of image classification and semantic segmentation. In both the inductive and transductive settings; we expect that these results contribute as new benchmarks to the research community.

**7 Conclusion**

In this work, we have proposed a hybrid model of structurally regularized deep clustering, termed SRDC++, for unsupervised domain adaptation. The objective of SRDC++ includes two key components respectively for discriminative and generative clusterings of unlabeled target data, which are constrained by structural regularization from the labeled source data. The discriminative component is based on a deep clustering framework that minimizes the KL divergence between predictive and auxiliary distributions of network outputs. To enable generative clustering, SRDC++ learns domain-shared cluster centroids in the feature space via self-attentive interactions of instance features. In this work, we also extend SRDC++ for the adaptation task of semantic segmentation, by incorporating into its objective additional terms for promoting consistency of spatial layouts between the segmentation maps on the source and target domains. We report comparative results on UDA benchmarks of image classification and semantic segmentation. In both the inductive and transductive settings; we expect that these results contribute as new benchmarks to the research community.

**Acknowledgments**

This work is supported in part by the National Natural Science Foundation of China (Grant No.: 61771201), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No.: 2017ZT07X183), the Guangdong R&D key project of China (Grant No.: 2019B010155001), and Microsoft Research Asia.

**References**

[1] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 22, pp. 1345–1359, 2010.

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” Machine Learning, vol. 79, pp. 151–175, 2010.

[3] M. Wang and W. Deng, “Deep visual domain adaptation: A survey,” Neurocomputing, vol. 312, pp. 135–153, 2018.

[4] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, “Transferable representation learning with deep adaptation networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, pp. 3071–3085, 2019.

[5] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” J. Mach. Learn. Res., vol. 20, pp. 2096–2030, 2018.

[6] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, “Maximum classifier discrepancy for unsupervised domain adaptation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3723–3732.

[7] Y. Zhang, H. Tang, K. Jia, and M. Tan, “Domain-symmetric networks for adversarial domain adaptation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 5026–5035.

[8] C. Chen, W. Xiao, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang, “Progressive feature alignment for unsupervised domain adaptation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 627–636.

[9] Y. Pan, T. Yao, Y. Li, Y. Wang, C. Ngo, and T. Mei, “Transferable prototypical networks for unsupervised domain adaptation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2234–2242.

[10] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, and Q. Tian, “Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3941–3950.
