Subtype-Aware Dynamic Unsupervised Domain Adaptation
Xiaofeng Liu, Fangxu Xing, Jane You, Jun Lu, C.-C. Jay Kuo, Fellow, IEEE, Georges El Fakhri, Fellow, IEEE, and Jonghye Woo, Senior Member, IEEE

Abstract—Unsupervised domain adaptation (UDA) has been successfully applied to transfer knowledge from a labeled source domain to target domains without their labels. Recently introduced transferable prototypical networks (TPNs) further address class-wise conditional alignment. In TPN, while the closeness of class centers between source and target domains is explicitly enforced in a latent space, the underlying fine-grained subtype structure and the cross-domain within-class compactness have not been fully investigated. To counter this, we propose a new approach to adaptively perform a fine-grained subtype-aware alignment to improve the performance in the target domain without the subtype label in both domains. The insight of our approach is that the unlabeled subtypes in a class have the local proximity within a subtype while exhibiting disparate characteristics because of different conditional and label shifts. Specifically, we propose to simultaneously enforce subtype-wise compactness and class-wise separation, by utilizing intermediate pseudo-labels. In addition, we systematically investigate various scenarios with and without prior knowledge of subtype numbers and propose to exploit the underlying subtype structure. Furthermore, a dynamic queue framework is developed to evolve the subtype cluster centroids steadily using an alternative processing scheme. Experimental results, carried out with multiview congenital heart disease data and VisDA and DomainNet, show the effectiveness and validity of our subtype-aware UDA, compared with state-of-the-art UDA methods.

Index Terms—Conditional shift, label shift, medical image diagnosis, subtype, unsupervised domain adaptation (UDA).

I. INTRODUCTION

RECENT advances in deep learning (DL) have shown promising performance in many supervised learning tasks [1]. In supervised learning, large-scale labeled training data are the key requirement for accurate model fitting at the training stage [2]. Although sufficient labels are available for some applications, such as ImageNet, acquiring large-scale labeled manual data is still challenging in many applications, including medical imaging, partly because of expert-driven annotation costs [3]. Besides, conventional empirical risk minimization tasks rely on independently and identically distributed (i.i.d.) training and testing data [1], [4]. In many real-world tasks, however, a variety of disparate data distributions, i.e., domain shift, between different domains exist, which hinders the application of supervised learning on new domains [5], [6]. For instance, in medical imaging, domain shift arises due to differences in scanners, protocols, and modalities, which also causes the high cost of collecting large-scale labeled training data annotated by clinicians [7].

Unsupervised domain adaptation (UDA) is an emerging technique, aiming at transferring domain knowledge learned from a labeled source domain to target domains with the help of unlabeled target data [2], [8], [9]. To counter the discrepancy in data distributions between two domains, the class-wise conditional alignment with respect to \( p(x|y) \) is typically employed [10]–[12]. The widely used UDA methods make use of adversarial learning and maximum mean discrepancy (MMD) to align the latent space distribution of sample data \( x \), i.e., \( p(f(x)) \), where \( f() \) is a feature extractor. Using Bayes’ theorem \( p(f(x)|y) = (p(y)f(x))p(f(x))/p(y) \) and assuming no concept shift with respect to \( p(y|f(x)) \) and label shift with respect to \( p(y) \), the alignment of conditional distribution \( p(f(x)|y) \) can be achieved by matching \( p(f(x)) \). Another challenge in UDA is the difference in the proportion of each class between two domains, thus leading to the label shift with respect to \( p(y) \) [13], [14].

Recently, transferable prototypical networks (TPNs) [10], [15] are proposed to promote the class separation in the source domain using a cross-entropy (CE) loss, by minimizing the distance of the class centers between two domains for the class-level conditional alignment with respect to \( p(x|y) \). The CE loss used in TPN, however, cannot minimize the inner class variation in the source domain [16], [17]. In addition, the optimization objective of the centroid closeness cannot guarantee the interclass separation in the target domain. This problem is well shown in Fig. 1 (left), where while the class centroids are perfectly aligned, there is a possibility that the sparsely distributed target samples are easily misclassified. One way to solve this would be by simply encouraging the cross-domain inner class feature distribution compactness. In many real-world applications, however, the unlabeled subtypes in a class can be highly diverse, and thus, an underlying local distribution forms an ensemble distribution. For example, the VisDA17 dataset [18] has the car, truck,
from winter to summer, the male/female of the pedestrian class can have different dress changes (shorts/skirt), which results in subtype conditional shift. Furthermore, the incidence of disease subtypes or the ownership rate of the sedan/SUV is usually varied across areas, which results in subtype label shift. The proportion difference on the subtype level can generally be more significant than the class level [22]. In the implementation, some of the subtypes may be undersampled with the limited batch size. These observations motivate us: 1) to expand the definition of class conditional and label shifts [5] to the fine-grained subtype level, i.e., \( p(f(x)|k) \), and 2) to reformulate that \( p(k) \) could differ across domains for subtype \( k \). As such, a more practical assumption would be the conditional and label shifts on both the class and subtype levels.

To address the aforementioned challenges, we propose to adaptively achieve both class-wise separation and cross-domain subtype-wise compactness. Specifically, we resort to feature space metric learning with intermediate pseudo-labels. In our subtype-aware UDA, we do not have access to the class label in the target domain and the subtype label in both the source and target domains. An online clustering scheme is proposed to explore the underlying subtype structure in an unsupervised manner. With the prior knowledge of subtype numbers, a straightforward choice would be the widely used K-means clustering, which defines \( k \) to be the subtype numbers. Due to taxonomies present in the data, however, it is challenging to identify the exact subtype number. We thereby further extend our method to the case of uncertain subtype numbers, by using the reliability path to capture the underlying subtype structure with an adaptive subgraph scheme. The subgraph scheme can be scalable over several classes and subtypes, with a few meta hyperparameters shared between clusters. The involved samples of subtypes are required to be adjacent to their subtype centroid in the feature space to directly impose the subtype-wise distribution compactness. The presence of the subtype is a general problem for any kind of DA method. In our method, assigning a pseudo-subtype label in each iteration requires the pseudo-class label to be assigned first. Then, the subtypes within each class are explored. Notably, the pseudo-class label assignment is a typical operation in the prototype-based methods. Thus, the prototype-based methods can be used as our baseline.

In addition, given the possible unstable learning caused by the undersampling of subtypes and the subtype label shift, we base our training on a dynamic memory framework to steadily evolve the cluster centroids to keep the network stably updated. Specifically, we design and maintain feature queue and centroid memory modules to store the sample features of several batches and the current centroids. These two modules are updated alternatively, and the features are refined with a momentum scheme, thereby allowing more representative sampling and clustering with low space cost. We break down the abrupt off-line entire dataset clustering [23] into two steps: 1) steady memory updates and 2) per batch pseudo label reassigning. Moreover, our subgraph and subtype centroid constructions are robust against the undersampling or label shift of any subtype.

This work extends our previous work [24] in the following significant ways.

1) An end-to-end dynamic queue framework with steady memory updates and per batch pseudo label reassigning.
which renders our subtype-aware UDA to be a practical and scalable solution.

2) The dynamic queue framework is able to process the imbalanced and undersampled subtypes, by involving the subtype cluster centroids steadily along with the iterations with an alternative processing scheme.

3) We demonstrate the generality of our subtype-aware UDA scheme in general large-scale UDA benchmarks, i.e., VisDA17, DomainNet, and GTA5toCityscapes.

4) Moreover, we carry out all experiments using a novel dynamic queue framework and provide a systematical subtype analysis with the two-level label structure of DomainNet.

The main contributions of this article are summarized as follows.

1) We propose to adaptively explore the subtype-wise conditional and label shifts in UDA without the subtype labels and explicitly enforce the subtype-aware compactness.

2) We systematically investigate cases with or without the prior information of subtype numbers. Our reliability-path-based subgraph scheme can effectively explore the underlying subtype local distributions with only a few meta hyperparameters in an online fashion.

3) Our alternatively updated dynamic memory queue and centroid module can be a practical and scalable solution to process the imbalanced and undersampled subtypes. It is possible to steadily evolve the cluster centroids to keep the network stably updated.

Taken together, this is the first attempt at exploring the underlying subtype structure, which does not have a subtype label in both the source and target domains. We evaluate the effectiveness of our framework on multiple benchmarks with different backbone models.

II. RELATED WORK

Recent advances in DL have achieved tremendous milestones in both computer vision and medical imaging areas [25]. DL has drastically converted the manner in which various features are extracted and fed into a prediction model into jointly learning both features and the prediction model from raw data in an end-to-end fashion [26]–[29]. In order for DL to be successfully applied to a variety of applications, massive annotated samples are typically demanded, where an i.i.d. assumption of training and testing data is required to ensure a seamless application of the trained model [1], [30].

A few challenges arise in applying DL to real-world applications. For example, a DL model that is trained on an annotated source domain does not generalize well on different target domains, because of the domain shift [31]–[33]. In addition, collecting the large-scale annotated training data in a novel environment can be costly or even prohibitive. Therefore, it is required to migrate domain knowledge learned from an annotated source domain with UDA to unseen target domains [34], [35].

Based on the shift content, domain shifts [5], [36] can be categorized into four groups: 1) covariate shift with respect to \( p(x) \); 2) conditional shift with respect to \( p(x|y) \); 3) label shift with respect to \( p(y) \); and 4) concept shift with respect to \( p(y|x) \). To date, each shift has been studied independently, with an assumption that the other shift conditions are invariant across domains [5]. In many applications, the conditional shift [37], [38] is more likely to occur than the covariate shift [5]. The label shift [39] (also known as the target shift) occurs when the proportion of each class is different between source and target domains. We note that the partial UDA can also be considered a label shift [40]. The concept shift [5] arises when classifying a tomato into a vegetable or fruit in different countries; however, the concept shift is usually not a common problem in popular object classification or semantic segmentation tasks. The conditional and target shifts are investigated in a casual system \( (y \rightarrow x) \), with a simple linearity assumption [11], [41]. However, these approaches could be sensitive to a dataset scale.

The objective of traditional UDA is to simply align the covariate shift (i.e., only \( p(x) \) shift). Discrepancy-based techniques, such as minimizing MMD, have been widely used to address the covariate shift in source and target domains, by minimizing discrepancies defined on different layers of a shared model between source and target domains [42].

In recent years, adversarial learning makes use of an extra domain discriminator to promote domain confusion [43]. These approaches are based on the assumption that the proportion of the labels remains invariant between source and target domains [44]. However, the label shift (i.e., only \( p(y) \) shift) is common in many real-world applications [5]. Recently, Zhang et al. [36] proposed a margin disparity discrepancy, a novel measurement with rigorous generalization bounds. The attention scheme is used in [33] for UDA. Learning the domain invariant and class discriminative features is investigated in [38].

As stated above, in addition, the subtype conditional and label shifts are a realistic assumption, which occurs in many applications, but those shifts pose more challenges, especially when the subtypes are undersampled. As shown in Fig. 3, we adapt the class label shift concept in [22] to the subtype. According to [22], if the source distribution contains 50% car and 50% bus, while the target distribution contains 70% car and 30% bus, then successfully aligning these distributions in a representation space requires a classifier to predict the same fraction of car and bus on the source and target data. If one achieves 100% accuracy on the source data, then target accuracy will be at most 80%. Considering that the van class can be confused with the bus class, it is highly likely that the van class is misclassified as the bus class, as denoted with the blue rectangle in Fig. 3. Similarly, there also has a mismatch at a subtype level, as denoted by the red rectangles. In Fig. 3(c), we show the expected subtype-wise alignment.
Another line of research in UDA is to use pseudo-labels in the target domain [2], [30]. For example, the use of pseudo-labels for target class centroids calculation is proposed to match source class centroids [45]. Contrastive adaptation network (CAN) [46] is proposed to use the pseudo-labels in the target domain for contrastive domain discrepancy estimation. Since the pseudo-labels are likely to be noisy compared with manual labels, a new approach [47] is presented to measure its correctness by the Gaussian-uniform mixture model. In the present work, we instead propose a noise-robust subgraph scheme with a simple online semihard mining method.

The “subdomain” adaptation approach proposed in [48] also focuses on the alignment of class-wise matching, and each subdomain corresponds to a class. Our proposed subtype, however, is more fine-grained than the class for which we do not have the subtype label in either source or target domain.

Based on self-labeling, TPN [10] is proposed to achieve the class-wise conditional alignment, by matching the source and target class centers [15], [45]. In addition, it is possible to set source centers as class protocols for classification. While its CE loss can enforce the source class-wise separation [16], aligning the centers does not enforce the compactness of both source and target samples since the target samples can distribute sparsely with considerable within-class variations. In addition, the prototypes in [10] can be sensitive to the label shift.

The center loss [17] has been developed to enforce the compact distributed features for face recognition. Then, new approaches [49], [50] are presented to adapt the center loss to metric learning and optimal transport-based methods. Similarly, the large margin softmax, ArcFace, and CosFace losses are proposed to simultaneously enforce interclass separation and inner class closeness [16]. Recently, Tang et al. [51] encouraged class-wise closeness across domains. Similarly, Toldo et al. [52] proposed to cluster the representation of each pixel. However, those approaches only rely on the class label supervision, and the underlying subtype distributions and shifts are largely ignored. When the discrepancy within a class is large, it is unreasonable to strictly extract the shared pattern.

In addition, in [51] and [52], the samples in each cluster are known, and we only need to calculate the distance. Instead, without the subtype label in both domains, the unsupervised clustering (e.g., K-means or subgraph) can be more challenging to identify the examples in each cluster.

Our approach exploring the subtype in an online fashion is closely related to unsupervised clustering [23], [53]. Along this line of research, K-means is used as an unsupervised initialization for convolutional layers sequentially, following a bottom-up fashion [54]. Note that the fine-grained labeling of a subtype can be costly and challenging, especially in the medical area. As such, several efforts have been made for subtype discovery with unsupervised clustering [19], [21]. A deep clustering approach [23] is proposed to train a network with K-means clustering, following an end-to-end fashion. Moreover, K-means for class-level clustering is proposed in open-set DA [55], which is a quite common operation, since classification always requires inner class compactness. This work, however, does not consider the subtype-level clustering. In the present work, we investigate K-means [54] with the prior information on subtype numbers. In addition, without the need for the number of subtypes, we propose a scalable subgraph scheme. Furthermore, the global clustering used in [23] is used, by simply applying K-means on all of the training data of ImageNet, which, however, can be highly costly. In addition, their training of AlexNet takes about 12 days. To address this, efficient dimension reduction and dynamic memory schemes are proposed to simplify the online clustering. Notably, the dynamic memory architecture in our work can be robust to pseudo-label noise and subtype undersampling.

Generally, since collecting large-scale labeled data in the target domain poses a challenge, labeling partial and scarce target data can be a practical solution [56]. In addition, the pseudo-label can be utilized, which provides a good connection between semisupervised learning and UDA [2]. For example, recent work [57], [58] explores the scarce labeled target data in UDA with pseudo-label. In our framework, reliability-path-based subgraph construction can be facilitated with some known samples. In the present work, we only consider the case when the number of categories between source and target domains is consistent. Recently, Jing et al. [15] and Li et al. [40] proposed to achieve partial UDA with adaptively accumulated knowledge transfer and a deep residual correction network, respectively. Jin et al. [59] proposed a flexible interface between the closed and partial set UDA. The open set UDA has been proposed to explore the case of nonoverlapping class [60], [61]. These works are orthogonal to our contributions to explore the subtype structure. Exploring the subtype structure in semisupervised UDA or open set UDA can be a promising future direction.

III. METHODOLOGY

In the scenario of UDA, there are a labeled source domain $p^s(x, y)$ and a different target domain $p^t(x, y)$, and the learning system has access to a labeled source set $(x^s_1, y^s_1, ..., x^s_n, y^s_n)$ drawn from $p^s(x, y)$ and an unlabeled target set $(x^t_1)$ drawn from $p^t(x, y)$. The class label space of $y_i$ $\in \{1, 2, ..., N\}$ is shared for both domains. For indexing $N$ classes, we use $n$. We assume that there are $K_n$ underlying subtypes indexed with $k \in \{1, 2, ..., K_n\}$ in class $n$, and we do not have access to the fine-grained subtype label in both domains. UDA attempts to learn a discriminative model in the target domain $p^t(x, y)$ following the theorem.

**Theorem 1:** For a hypothesis $h$ drawn from $\mathcal{H}$, $\epsilon^t(h) \leq \epsilon^s(h) + (1/2)d_{\mathcal{H},\mathcal{M}}[s, t] + \min_{h \in \mathcal{H}_t}[\epsilon^s(h, l) + \epsilon^t(h, l)]$.

Here, $\epsilon^s(h)$ and $\epsilon^t(h)$ denote the expected losses with hypothesis $h$ in the source and target domains, respectively. The right side can be the upper bound of the target loss. Since the disagreement between the labeling functions $l^t$ and $l$, i.e., $\min_{h \in \mathcal{H}_t}[\epsilon^s(h, l) + \epsilon^t(h, l)]$, can be small, by optimizing $h$ with the source data [62], UDA aims to minimize the cross-domain divergence $d_{\mathcal{H},\mathcal{M}}[s, t]$ in the feature space of $f(x^s)$ and $f(x^t)$.

Instead of matching $p^s(f(x^s))$ and $p^t(f(x^t))$ [5], the prototypical networks attempt to align the class centroids [10]. However, it poses a challenge to define the decision boundary of the low-density distributed target sample, and thus, the inherent subtype structure is not fully investigated.

In this work, we aim to achieve the class separation in the target domain on the embedding space when the source domain classes are well-separated, and compactness of the class-wise source-target distribution is encouraged. In addition, under the fine-grained subtype local structures and their conditional and label shifts, the subtype-wise compact clustering can be a good alternative to achieve the class-wise alignment and compactness. Therefore, we have the following proposition.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Proposition 1: The class-level compactness can be a particular case of subtype-wise compactness by assigning the subtype number of this class to 1.

To tackle the conditional and label shifts at both class-level ($p'(f(x)|y) \neq p'(f(x)|y)$, $p'(y) \neq p'(y)$) and subtype-level ($p'(f(x)|k) \neq p'(f(x)|k)$, $p'(k) \neq p'(k)$), we aim to develop a subtype-aware alignment framework with a dynamic clustering scheme, as shown in Fig. 1.

We organize this section as follows.

We first describe our approach to tackle the class-wise separation and alignment in Section III-A. Then, we explore subtype-wise compactness for the following scenarios: 1) if we know the subtype number, we use the subtype-aware alignment framework in Section III-B and 2) if we do not know the subtype number, we use the reliability-path-based subgraph construction in Section III-C. To speed up the clustering (either $K$-means or subgraph), we propose queued dynamic clustering in Section III-D. Section III-E, i.e., optimization and implementation, provides detailed processing.

Our framework has two promising properties. The adaptive property is achieved in Section III-C (see Fig. 4). If we do not know how many subtypes we have, it is possible to cluster the samples adaptively. The dynamic property is achieved in Section III-D (see Fig. 5), which largely alleviates the memory cost and evolves the centroids smoothly.

A. Class-Wise Source Separation and Matching

Forcing the separation of classes in the source domain $p'(x, y)$, i.e., minimizing $c'(h)$, can be achieved by the conventional CE loss [16]. We carry out the classification with the extracted features through a remold of the distance to each class cluster centroid [10], [45].

For the labeled source data $\{(x_s^i, y_s^i)\}$, we represent the feature distribution of class $n$ with a centroid

$$c_n^s = \frac{1}{M_n^s} \sum_{i=1}^{M_n^s} f(x_s^i)$$  \hspace{1cm} (1)

where $M_n^s$ is the number of source samples involved. We can directly create a probability histogram for an input source sample $x_s^i$ using the softmax normalized distance between $x_s^i$ and the centroids $c_n^s$. Specifically, it is possible to formulate the likelihood of $x_s^i$ belonging to class $n$ as

$$p(y_s^i = n|x_s^i) = \frac{e^{-\|f(x_s^i) - c_n^s\|_2^2}}{\sum_{n=1}^{N} e^{-\|f(x_s^i) - c_n^s\|_2^2}}.$$  \hspace{1cm} (2)

With the one-hot encoding of true class $n$, we define the class-wise CE loss with respect to source domain samples as

$$L_{CE}^{\text{class}} = -\log p(y_s^i = n|x_s^i).$$  \hspace{1cm} (3)

The minimization of $L_{CE}^{\text{class}}$ encourages the separation of classes in the source domain. Therefore, the visually similar van and small truck in Fig. 2 are enforced to be different from each other, according to their class, which can be seen as a challenging case to facilitate the fine-grained feature exploration [63]. The more explicit interclass separation objective as in CAN [46] can potentially achieve further separation, leading to better performance.

Following the self-labeling scheme [2], [10], each target sample $x_t^i$ is assigned with a pseudo class label $\hat{y}_t^i$ to its nearest source centroids, i.e.,

$$\hat{y}_t^i = n, \text{ if } \min_{\forall n} \|f(x_t^i) - c_n^t\|^2_2.$$  \hspace{1cm} (4)

With the pseudo class label $\hat{y}_t^i$, we calculate the target domain class-level centroids

$$c_n^t = \frac{1}{M_n^t} \sum_{i=1}^{M_n^t} f(x_t^i)$$  \hspace{1cm} (5)

where $M_n^t$ is the number of target samples involved. We expect the close proximity of $c_n^s$ and $c_n^t$ with

$$L_{\text{class}} = \frac{1}{N} \sum_{n=1}^{N} \|c_n^t - c_n^s\|^2_2.$$  \hspace{1cm} (6)

which is not sensitive to label change, because it only selects the representative source and target distribution centroids.\footnote{The source and target center $c_n^s = (\sum_{M_n^s} M_n^t f(x_s^i)/M_n^t + M_n^t)$ used in [10] and its objective of close proximity of $c_n^s \leftrightarrow c_n^t$ or $c_n^s \leftrightarrow c_n^t$ are not robust to label shift. Note that if we double the source/target samples involved, $c_n^s$ would shift.}

Neither $L_{CE}^{\text{class}}$ nor $L_{\text{class}}$, however, considers the fine-grained subtype structure and inner class compactness [17].

We note that in class-wise separation and matching, the target class can be a subset of the source domains. We use the label to define the class center $c_n^t$ in the source domain and assign the label to the target domain data with the nearest center. If there is no new class in the target domain, then the target sample should correspond to a proper class in the source domain. It is fine to have target samples that are labeled as some of the source classes (i.e., in the case where the target classes are a subset of source classes). In the case of no target domain sample being assigned to class $n$, we simply do not calculate the class matching loss $L_{\text{class}}$ for class $n$ [see (6)], similar to TPN.

B. Subtype-Aware Alignment With $K_n$ Prior

With the prior knowledge of subtype numbers $K_n$ of class $n$, we can achieve feature space class-independent clustering with the concise $K$-means [54], by defining $K$ to be $K_n$.

We index $K_n$ clustered subtypes with $k \in \{1, 2, \ldots, K_n\}$ and calculate the source subtype centroids $x_k^n$. After the $K_n$ subtypes in the source domain are explored and the corresponding centroids $c_k^n$ are calculated, we assign each target sample $x_t^i$ using the softmax normalized distance between $x_t^i$ and the centroids $c_k^n$. Specifically, it is possible to formulate the likelihood of $x_t^i$ belonging to class $n$ as

$$p(y_t^i = n|x_t^i) = \frac{e^{-\|f(x_t^i) - c_k^n\|_2^2}}{\sum_{k=1}^{K_n} e^{-\|f(x_t^i) - c_k^n\|_2^2}}.$$  \hspace{1cm} (2)

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
sample with pseudo-label of class \( n \) to the subtype with the most adjacent centroid, i.e.,

\[
x^i_t \in \text{subtype } k \text{ if } \min_k \| f(x^i_t) - \mu_k \|^2_2
\]

which is used to calculate the target subtype centroids \( \mu_k^t \).

Considering the imbalance distribution of subtypes and the possible label shift, we assign the subtype centroids of both source and target samples with

\[
\mu_k^{xt} = \frac{\mu_k^s + \mu_k^t}{2}
\]

instead of averaging all of the source and target samples in subtype \( k \). By doing so, each subtype in both domains contributes equally to \( \mu_k^{xt} \). Then, we enforce all of the samples in subtype \( k \) to be close to the subtype centroid \( \mu_k^{xt} \). We omit the class notation for simplicity, and the subtype compactness objective \( L_{k}^{\text{sub}} \) is given by

\[
L_{k}^{\text{sub}} = \frac{1}{M_k^s} \sum_{i=1}^{M_k^s} \| f(x^i_t) - \mu_k^{xt} \|^2_2 + \frac{1}{M_k^t} \sum_{i=1}^{M_k^t} \| f(x^i_t) - \mu_k^{xt} \|^2_2
\]

where \( M_k^s \) and \( M_k^t \) are the source and target sample numbers of subtype \( k \) to balance the subtype label shift, respectively. \( L_{k}^{\text{sub}} \) is traversed for \( N \) classes and their \( K_n \) subtypes to calculate the sum of normalized subtype compactness losses

\[
L^{\text{sub}} = \frac{1}{N} \sum_{n}^{N} \left( \frac{1}{K_n^s} \sum_{k=1}^{K_n} L_{k}^{\text{sub}} \right).
\]

Of note, an image that does not belong to class \( n \) does not belong either to any of its \( K_n \) subtypes. The class-wise matching and subtype-wise compactness objectives can be aggregated as a hierarchical alignment loss as

\[
\frac{1}{N} \sum_{n}^{N} \left( \alpha L^{\text{class}} + \beta \frac{1}{K_n} \sum_{k=1}^{K_n} L_{k}^{\text{sub}} \right)
\]

where \( \alpha \) and \( \beta \) are balancing parameters. In the feature space, the learned representations are expected to form \( K_n \) compact clusters for class \( n \), although each cluster does not need to be far apart from one another.

We note that in our subtype clustering, the target subtypes can also be a subset of the source domains. In both \( K \)-means and reliability-path-based subgraph, we cluster the source domain data first. Then, we assign the subtype of the target domain samples to the nearest source domain subtype center [see (7)] just like the pseudo-class label assignment as discussed above. Therefore, there can be no target data for a subtype in the source domain, and we have \( \mu_k^{xt} \). Consequently, (8) enforces the compactness of the source domain data in this subtype only. In our reliability-path-based subgraph construction, we do not even define the number of subtypes.

C. Reliability-Path-Based Subgraph Construction

In practice, extensive trials are required to accurately estimate \( K_n \) for \( n \) classes, because of the varying value ranges of different classes, which is impractical especially for large-scale tasks. Notably, however, with a deterministic encoder \( f \), the distribution protocol can be similar among different subtypes and classes [64].

In order to achieve the online source domain subtype clustering, we propose to form a subgraph with a reliability path. We presume that with a deterministic encoder \( f \), similar samples are likely to be closely distributed in the latent space and form a high-density region [65]. Given \( M^s \) samples from class \( n \) in the source domain, there are \((M^s)^2\) possible edges in a graph. To mine the local structure inherent in the feature space, two nodes \( \{f(x^j), f(x^j')\} \) are connected by the reliability path if \( \| f(x^j) - f(x^j') \|_2 \leq \epsilon \). To form a subgraph, the directly or indirectly linked nodes are combined.

To further eliminate the effect of noise and undersampled subtypes on a batch, we only select the subgraphs with more than \( m \) nodes as valid subtype clusters. After \( K_n \) subtypes in the source domain are explored and the corresponding centroids \( \mu_k^s \) are calculated, we assign each target sample with pseudo-label of class \( n \) to the subtype with the most adjacent centroid as in (7).

Since the confidence or reliability of pseudo target labels is relatively low [2], [47], we propose to apply a simple online semihard mining scheme to choose the target sample in a subtype. The cross-domain margin \( r \) is used to define a circle at the center of \( \mu_k^s \).

Given the target samples with the initial pseudo subtype label \( k \), we only select the samples that are distributed within the circle. Considering that some target samples may be densely distributed around the circle boundary, it is not reasonable to simply cut them apart. Therefore, we also resort to the reliability path to incorporate the closely distributed neighboring target samples. The subgraph construction can be robust to missing subtypes in the source or target domain caused by undersampling because \( m \) filters the unreliable source cluster out, and the self-labeling with semihard mining of \( f(x^j) \) rejects the additional subtypes in the sampled target domain. We illustrate the operations in Fig. 4.

With the reliability path connected \( M_k^t \) source samples and the refined \( M_k^t \) target samples in subtype \( k \), we calculate \( \mu_k^{xt} = (\mu_k^s + \mu_k^t)/2 \) and enforce the subtype-wise compactness with \( L_{k}^{\text{sub}} \) as in (9).

In summary, there are three hyperparameters in our online subgraph construction, including \( \epsilon \), \( r \), and \( m \). They are shared for all classes and their subtypes, which can be regarded as meta-knowledge across clusters. Furthermore, \( \epsilon \) can be simplified to constant 1. We can even change it to any positive number via multiplying the corresponding factors [49]. The range of \( m \) can also be narrow and similar among different subtypes/classes.

D. Queued Dynamic Clustering

There is a possibility that there could be many underlying subtypes, i.e., \( K_n \) is large, and some of the subtypes may be undersampled in both domains of a mini-batch. As such, it is challenging to obtain the representative subtype-wise sampling with a small batch. Therefore, instead of extracting the feature of the entire training set at each iteration via deep clustering [23], which requires extensive computation for unsupervised clustering and large GPU memory (if the dataset is very large, e.g., DomainNet), we propose to incorporate more representative samples with a dynamic feature queue module that evolves smoothly along with the network parameter update.

As shown in Fig. 5, the dynamic queue maintains the features and their label of \( I \) iterations, where \( I \) is usually set to
5 in our implementation. With more sampled data in multiple batches, the calculated center can be more representative. Moreover, the centroids memory module stores the class and subtype centers with the current \( T \) batches. The “subtype” here represents the temporary cluster that evolves continuously along with the training iterations. Our centroid-based classifiers and networks are updated simultaneously during uninterrupted iterations. The detailed processing is given in Algorithm 1.

In our dynamic memory scheme, we store five batches samples and update the batches along with the training. The batch size is set to 64 in all of our tasks. With Float32 in Pytorch, each dimension takes 4 bytes. Therefore, the memory cost of the 256-D feature of five batches samples is \( 64 \times 5 \times 256 \times 4 = 327,680 \) bytes = 327.68 kB. Compared with the memory of our GPU device, i.e., 32 GB, the memory cost of the feature storage is relatively low. The I/O can also be compared with the unsupervised clustering. In fact, the unsupervised clustering, e.g., \( K \)-means, limits the scale of batch in our SubUDA. Our dynamic memory does follow the first-in-first-out (FIFO) protocol to keep five recent batches along with the iterations. As shown in Fig. 5, the early batches are dropped. Thus, the queue is a more natural data structure choice.\(^2\)

With a batch of training samples \( \{ x_i^t \} \) and \( \{ x_i^s \} \) in an iteration, we first encode these samples into feature vectors \( \{ f(x_i^t) \}, \{ f(x_i^s) \} \), which are added to the queue. Then, we read the label of source samples \( \{ x_i^s \} \) in this batch to calculate the class-level CE loss \( L_{CE}^{class} \). The pseudo-label of target samples is self-labeled with the centroids of source features in the \( I \) batches, and \( L_{CE}^{class} \) is also applied to the source and target class centroids among \( I \) batches.

After the class-wise pseudo-label is inferred, we can explore the subtype structure of paired class-wise source and target distributions on \( I \) batches. Based on the availability of the subtype numbers, \( K \)-means or online reliability-path-based sub-graph construction is applied to adaptively cluster \( I \) batches; therefore, \( L_{sub} \) can be calculated. The overall optimization objective can be expressed as

\[
L = L_{CE}^{class} + \alpha L_{class} + \beta L_{sub}
\]  

(12)

and we update \( f \) with stochastic gradient descent.

With the updated parameters, we recompute the feature representation of this batch \( f(x_i^t) \). The feature memory in queue is dynamically rewritten with a momentum strategy

\[
f''(x_i^t) \gets \lambda f'(x_i^t) + (1 - \lambda) f(x_i^t)
\]  

(13)

where \( \lambda \) is an empirical momentum weight and \( f(x_i^t) \) and \( f'(x_i^t) \) are the feature representations extracted by the original and updated extractor in this iteration, respectively. Note that \( f''(x_i^t) \) of this batch is stored in our dynamic queue.

\section{E. Optimization and Implementation}

Many backbone networks extract high-dimensional representations, such as 4096-D or 2048-D features, from image datasets, thereby demanding high time cost in subsequent clustering and memory cost with respect to memory. To alleviate this, deep clustering \[23\] performs principal component analysis (PCA) on whole datasets for dimension reduction. However, in our online SubUDA, PCA is no longer applicable, as there are different timestamps of differently sampled features, leading to incompatible sample statistics.

Applying PCA in each iteration is also computationally expensive. Therefore, to compact high-dimensional features to 256 dimensions, we simply add a nonlinear head layer of \( fc \rightarrow bn \rightarrow relu \rightarrow dropout \rightarrow fc \rightarrow relu \) jointly tuned during online SubUDA iterations. For downstream tasks, such as calculating the L2 distance between features, the head layer is removed.

Caron \textit{et al.} \[23\] conducted uniform sampling in all of the training epochs to prevent the training from collapsing into a few large subtype clusters. However, without the subtype and target class labels, it is difficult to apply the same strategy in our online UDA environment. Thus, we develop a concise method for SubUDA to reweight the loss function with

\[
\alpha_k \propto \frac{1}{\sqrt{M_k^t + M_k^s}}
\]  

(14)

based on the sample numbers in the \( k \)th subtype. Samples from smaller clusters tend to contribute more to backpropagation. Therefore, in order to include more potential samples, the

\(^2\)We used “collections.deque” to construct the queue in Python, which can efficiently add or remove the elements on both sides of the queue.
Fig. 6. Prepossessing of the CHD task.

TABLE I
DETAILED STRUCTURE OF OUR FIVE-CHANNEL DSC NETWORK.
THE CONV DW/PW INDICATE THE DEPTH AND POINT WISE
CONVOLUTIONS, RESPECTIVELY. THE FIRST LAYER
ADOPTS THE TRADITIONAL CONVOLUTIONAL
KERNEL (THE STRIDE IS SET TO 2) [66].

| Input Size | Type / Stride | Filter Shape |
|------------|--------------|--------------|
| 128 × 128 × 5 | Conv / s2 | 32 kernels of 3 × 3 × 3 |
| 64 × 64 × 32 | Conv dw / s1 | 32 kernels of 3 × 3 dw |
| 64 × 64 × 32 | Conv pw / s1 | 64 kernels of 1 × 1 × 32 pw |
| 64 × 64 × 64 | Conv dw / s2 | 64 kernels of 3 × 3 dw |
| 32 × 32 × 64 | Conv dw / s2 | 128 kernels of 1 × 1 × 64 pw |
| 32 × 32 × 128 | Conv pw / s1 | 128 kernels of 3 × 3 dw |
| 16 × 16 × 128 | Conv pw / s1 | 128 kernels of 1 × 1 × 128 pw |
| 8 × 8 × 128 | Conv dw / s2 | 128 kernels of 3 × 3 dw |
| 8 × 8 × 128 | Conv pw / s1 | 128 kernels of 1 × 1 × 128 pw |

Fig. 7. Multichannel DSC backbone for our five-view CHD analysis.

decision boundary tends to move farther. We can summarize
the optimization objective as

\[
\mathcal{L} = \mathcal{L}_{\text{CE}} + \frac{1}{N} \sum \left( \alpha \mathcal{L}_{\text{class}} + \beta \frac{\omega}{K_n} \sum \mathcal{L}_{\text{sub}} \right). \quad (15)
\]

As stated above, the subtype degenerates to class alignment,
if we set the subtype number of each class as 1. If we know the
subtype number \(K_n\) and use \(K\)-means, there would be no such
kind of problem. In some evaluations, we just set \(K_n = 1\) to
illustrate the necessity for exploring subtypes. For the case
of unknown \(K_n\), using relatively small \(\tau\) and \(m\) can also be
helpful to yield many subclusters.

We note that the online clustering is performed in every
batch iteration, and we only calculate with the samples only
in a batch and compare their feature with the previous center
vector, rather than all of the previous features. For the classi-
fication in testing, the centroids of training features are used
as prototypical [10].

IV. EXPERIMENTS

Our experiments were performed both on medical subtype
data and the popular VisDA and DomainNet benchmarks to
evaluate the effectiveness of our framework. We implemented
our framework using PyTorch and the reported training times
were based on a V100 GPU (32-GB memory).

For the class-level prototypes used in testing, we did not see
a significant difference. After training, the network was able to
embed these three prototypes closely as TPN [10]. As shown in
Table II, the difference is consistently smaller than 0.002. The
similar performance also implicitly demonstrates the domain
invariant property of our extracted feature representations. The
reported results in the subsequent evaluations are based on the
prototypes of \(c_{ni}^n\).

For a fair comparison, we follow the previous standard
evaluation protocol to validate the hyperparameters with grid
searching. The sensitivity analysis of these hyperparameters
(e.g., \(K_n\), \(m\), \(\tau\), and \(\epsilon\)) is also provided in Figs. 9–13.

A. Congenital Heart Defect Transfer Task

Congenital heart defects (CHDs) are the most common
congenital disability, which leads to the neonates’ death.
Usually, the clinical diagnosis of the disease relies on chosen
echocardiograms from multiviews. A multicenter dataset of
five-view echocardiograms of 1608 labeled source subjects

Fig. 9. Comparison with the other STOAs with respect to $A$-distance (left) and the sensitive analysis of $\tau$ (right) in SubUDA for CHD task. We use the discriminator as [69].

Fig. 10. Sensitive analysis of $K_n$ (right) and $m$ of patient class in CHD dataset with SubUDA. The red rectangle indicates the accuracy and the blue triangle indicates the AUC of CDF.

Fig. 11. Sensitive analysis of $\epsilon$ in SubUDA for CHD and VisDA17 datasets.

Fig. 12. Sensitive analysis of $K_n$ (right) and $m$ of car class in the VisDA17 dataset of dynamic SubUDA with the ResNet152 backbone. The red rectangle indicates the accuracy and the blue triangle indicates the AUC of CDF.

Fig. 13. Sensitive analysis of $m$ in DomainNet with the ResNet101 backbone.

The four subtypes of CHD, including atrial septal defect (ASD), ventricular septal defect (VSD), patent ductus artery (PDA), and tetralogy of Fallot (TOF), were confirmed by at least two senior ultrasound clinicians or by a final intraoperative diagnosis to analyze the influence of the subtype structure. We note that the fine-grained subtype label was not used in training, as large-scale labeling was expensive, whereas primary clinicians can obtain the normal/patient label relatively easily. In addition, we attempted to tackle the subtype-aware alignment for the conventional class-wise discriminative model. Fig. 8 (left) shows that subtypes source CHD samples often appear to be tightly distributed, indicating the underlying similarity of the inner subtype. With our subtype-wise compactness objective, both the source and target samples were grouped into high-density regions with respect to subtypes.

1) Data Collection and Prepossessing: We placed each subject in the supine position and exposed the chest area to the echocardiogram. In total, five standard 2-D views were acquired, including parasternal long axis (PSLAX), parasternal short axis (PSSAX), apical four chambers (A4C), subxiphoid long axis (SXLAX), and suprasternal long axis (SSLAX). The keyframe was manually selected by a skilled clinician. In particular, we marked the isovolumic relaxation process as a keyframe, if the contraction of ventricles is stopped. The preprocessing flow is shown in Fig. 6.

A total of 1608 subjects in the BCH source domain (823 normal, 209 VSD, 276 ASD, 124 TOF, and 176 PDA) were used in this work. For the acquisition, we used PHILIPS iE 33. The frequency of the transducer ranged between 3 and 8 MHz. A total of 800 subjects were collected in the HMS target domain (300 normal, 150 VSD, 150 ASD, 100 TOF, and 100 PDA). For the acquisition, we used PHILIPS EPIQ 7C. The frequency of the transducer ranged between 3 and 8 MHz.

In sum, imaging instruments (PHILIPS iE 33 versus EPIQ 7C), patient demographics, and the echocardiogram imaging background of healthcare clinicians were the major differences between these two medical centers, which caused the data shifts between two domains. We used 75%/5%/20% split of the target subjects for training, validation, and testing, respectively.

2) Backbone Network Structure for the CHD Task: The effectiveness of depthwise separable convolution (DSC)-based [66] multichannel network for single-domain five-view echocardiogram processing has been successfully demonstrated [70]. Instead of merely concatenating in a fully connected (FC) layer as in multibranch networks, the multichannel scheme was able to adaptively fuse the knowledge in all layers [71]. The keyframe was manually selected by a skilled clinician. In particular, we marked the isovolumic relaxation process as a keyframe, if the contraction of ventricles is stopped. The preprocessing flow is shown in Fig. 6.

A total of 1608 subjects in the BCH source domain (823 normal, 209 VSD, 276 ASD, 124 TOF, and 176 PDA) were used in this work. For the acquisition, we used PHILIPS iE 33. The frequency of the transducer ranged between 3 and 8 MHz. A total of 800 subjects were collected in the HMS target domain (300 normal, 150 VSD, 150 ASD, 100 TOF, and 100 PDA). For the acquisition, we used PHILIPS EPIQ 7C. The frequency of the transducer ranged between 3 and 8 MHz.

In sum, imaging instruments (PHILIPS iE 33 versus EPIQ 7C), patient demographics, and the echocardiogram imaging background of healthcare clinicians were the major differences between these two medical centers, which caused the data shifts between two domains. We used 75%/5%/20% split of the target subjects for training, validation, and testing, respectively.

2) Backbone Network Structure for the CHD Task: The effectiveness of depthwise separable convolution (DSC)-based [66] multichannel network for single-domain five-view echocardiogram processing has been successfully demonstrated [70]. Instead of merely concatenating in a fully connected (FC) layer as in multibranch networks, the multichannel scheme was able to adaptively fuse the knowledge in all layers [71]. The keyframe was manually selected by a skilled clinician. In particular, we marked the isovolumic relaxation process as a keyframe, if the contraction of ventricles is stopped. The preprocessing flow is shown in Fig. 6.

A total of 1608 subjects in the BCH source domain (823 normal, 209 VSD, 276 ASD, 124 TOF, and 176 PDA) were used in this work. For the acquisition, we used PHILIPS iE 33. The frequency of the transducer ranged between 3 and 8 MHz. A total of 800 subjects were collected in the HMS target domain (300 normal, 150 VSD, 150 ASD, 100 TOF, and 100 PDA). For the acquisition, we used PHILIPS EPIQ 7C. The frequency of the transducer ranged between 3 and 8 MHz.

In sum, imaging instruments (PHILIPS iE 33 versus EPIQ 7C), patient demographics, and the echocardiogram imaging background of healthcare clinicians were the major differences between these two medical centers, which caused the data shifts between two domains. We used 75%/5%/20% split of the target subjects for training, validation, and testing, respectively.

2) Backbone Network Structure for the CHD Task: The effectiveness of depthwise separable convolution (DSC)-based [66] multichannel network for single-domain five-view echocardiogram processing has been successfully demonstrated [70]. Instead of merely concatenating in a fully connected (FC) layer as in multibranch networks, the multichannel scheme was able to adaptively fuse the knowledge in all layers [71]. The keyframe was manually selected by a skilled clinician. In particular, we marked the isovolumic relaxation process as a keyframe, if the contraction of ventricles is stopped. The preprocessing flow is shown in Fig. 6.

A total of 1608 subjects in the BCH source domain (823 normal, 209 VSD, 276 ASD, 124 TOF, and 176 PDA) were used in this work. For the acquisition, we used PHILIPS iE 33. The frequency of the transducer ranged between 3 and 8 MHz. A total of 800 subjects were collected in the HMS target domain (300 normal, 150 VSD, 150 ASD, 100 TOF, and 100 PDA). For the acquisition, we used PHILIPS EPIQ 7C. The frequency of the transducer ranged between 3 and 8 MHz.

In sum, imaging instruments (PHILIPS iE 33 versus EPIQ 7C), patient demographics, and the echocardiogram imaging background of healthcare clinicians were the major differences between these two medical centers, which caused the data shifts between two domains. We used 75%/5%/20% split of the target subjects for training, validation, and testing, respectively.

2) Backbone Network Structure for the CHD Task: The effectiveness of depthwise separable convolution (DSC)-based [66] multichannel network for single-domain five-view echocardiogram processing has been successfully demonstrated [70]. Instead of merely concatenating in a fully connected (FC) layer as in multibranch networks, the multichannel scheme was able to adaptively fuse the knowledge in all layers [71]. The keyframe was manually selected by a skilled clinician. In particular, we marked the isovolumic relaxation process as a keyframe, if the contraction of ventricles is stopped. The preprocessing flow is shown in Fig. 6.

A total of 1608 subjects in the BCH source domain (823 normal, 209 VSD, 276 ASD, 124 TOF, and 176 PDA) were used in this work. For the acquisition, we used PHILIPS iE 33. The frequency of the transducer ranged between 3 and 8 MHz. A total of 800 subjects were collected in the HMS target domain (300 normal, 150 VSD, 150 ASD, 100 TOF, and 100 PDA). For the acquisition, we used PHILIPS EPIQ 7C. The frequency of the transducer ranged between 3 and 8 MHz.
| Compared Methods       | Accuracy (%) ↑ | AUC ↑ |
|------------------------|----------------|-------|
| Source only            | 76.4 ± 0.12   | 0.721 ± 0.005 |
| MCD [67]               | 86.6 ± 0.15   | 0.856 ± 0.003 |
| GTA [68]               | 90.9 ± 0.17   | 0.873 ± 0.005 |
| CRST [2]               | 93.2 ± 0.09   | 0.882 ± 0.006 |
| Prototypical Network (TPN) [10] | 93.4 ± 0.14     | 0.885 ± 0.004 |

| Proposed Methods       | Accuracy (%) ↑ | AUC ↑ |
|------------------------|----------------|-------|
| SubUDA-Dyn (Kn = 4)    | 96.8 ± 0.16   | 0.921 ± 0.002 |
| SubUDA (Kn = 4)        | 96.2 ± 0.13   | 0.910 ± 0.003 |
| SubUDA (Kn = 1):class-wise compactness | 94.7 ± 0.11   | 0.902 ± 0.004 |
| SubUDA-μk1 (Kn = 4)    | 95.4 ± 0.10   | 0.903 ± 0.005 |
| SubUDA-μk2 (Kn = 4)    | 96.0 ± 0.13   | 0.908 ± 0.004 |
| SubUDA-DR (Kn = 4)     | 96.2 ± 0.11   | 0.911 ± 0.002 |
| SubUDA-Dyn (m = 8)     | 96.6 ± 0.14   | 0.914 ± 0.005 |
| SubUDA (m = 8)[24]     | 96.0 ± 0.12   | 0.907 ± 0.004 |
| SubUDA+ (m = 8)        | 95.5 ± 0.14   | 0.902 ± 0.003 |

We note that the SubUDA was proposed in our prior work [24], which did not include the dynamic memory scheme.

In addition, the suffix -μk1 denotes using \( \mu_k^{(t)} = \left( \sum_{i=1}^{M_k^{(t)}} f(x_i^{(t)})/M_k^{(t)} + M_k^{(t)} \right) \) as the subtype centroid, which is not robust against the subtype label shift. Among them, the dynamic queue scheme played the most important role in addressing the difficulty of subtype undersampling and subtype label shifts. SubUDA-DR took 4× more clustering time, while the improvement was marginal. Therefore, we would recommend using a dimension reduction head to achieve more efficient processing.

As shown in Table III, the SubUDA \((m = 8)\) adopted the reliability-path-based online subgraph as an alternative to \(K\)-means. With an appropriate \( m \), e.g., \( m = 8 \), the adaptively learned clustering achieved the performance on par with the \(K\)-means with \( K_n = 4 \). We can see from Fig. 9 (right) that the semihard mining scheme in SubUDA \((m = 8)\) is not sensitive to \( \tau \) for a relatively large range. This may be attributed to the fact that the network can flexibly learn to adjust the ratio of \( \epsilon \) and \( \tau \) in mapping space [49]. We note that the too strict semihard mining scheme, i.e., too small \( \tau \), can degenerate our SubUDA to conventional class centroids matching since no target samples are selected to form a subtype cluster.

A hyperparameter-sensitive analysis with respect to \( K_n \) and \( m \) for two kinds of online clustering schemes is shown in Fig. 10. Consensus clustering is widely used to assess the clustering stability [81], and the optimal clustering is usually achieved in the elbow position of the area under CDF changes, where the CDF is for the consensus matrices.\(^3\) We can see that the accuracy peak typically coincides with the best metric of clustering consensus, which clearly indicates that the SubUDA performance relied on the good subtype clustering. The choice of \( K_n = 4 \) also matched our prior knowledge of CHD patient subtype numbers.

The sensitivity analysis of \( \epsilon \) in CHD and VisDA datasets is shown in Fig. 11. We can see that the accuracy is not sensitive to a relatively large range of \( \epsilon \), as expected.

Since there were clearly four subtypes in our data, concise \(K\)-means can be used as our solution. However, Fig. 10 (right) shows that the accuracy curve is robust for a relatively large range of \( m \), which is promising for the hyperparameter tuning of the case without the prior information of subtype numbers. Specifically, when we do not know \( K_n \), we propose to use a subgraph for clustering with \( \epsilon = 1 \), \( \tau \), and \( m \) to decide the cluster numbers. Therefore, we only needed two hyperparameters \((\tau \text{ and } m)\) rather than \(N\).

From Fig. 8, we can clearly see that samples from the same subtype are closely distributed, but not clearly separated. We only enforced subtype compactness but did not enforce subtype separation since classification is our goal in this work.

### B. Synthetic-to-Real ImageUDA

VisDA17 [18] is well suited to evaluate our framework for a large-scale case, where VisDA17 contains a synthetic source domain and a real image target domain. It contains a total of 280k images in 12 classes, in which a total of 152409 synthetic 2-D images were used as the source domain and a total of 55400 real images were used as the target domain. This scale, however, poses a challenge to domain adaptation. Some examples are shown in Fig. 2. For the car and truck classes, we can see that the inner class variation

---

\(^3\)https://github.com/ZigaSajovic/Consensus_Clustering
is large, while the interclass variation is relatively small. For fair comparisons with the other methods, we used the standard ResNet101 as the backbone and set the batch size to 64 and $T = 5$. We set $\alpha = 1$, $\lambda = 0.5$, and $\beta = 0.5$ and used Adam with $lr = 1e^{-4}$.

We did not know the subtype number in this task, and therefore, the reliability-path-based subgraph was utilized as our default solution. In fact, there is no clear definition of subtypes in VisDA, e.g., the car class can be grouped according to the maker, type, and color based on different taxonomy. Validating the best $K_n$ for all of the classes is tedious, and the optimal $K_n$ can be different when we change the backbone.

In Fig. 12, we investigate the effect of $K_n$ and $m$ on the car class. Given the ambiguous subtype grouping taxonomy, the accuracy curve with respect to $K_n$ is smoother than the CHD transfer task. Based on the accuracy and elbow position of consensus clustering analysis, we selected $m = 5$ for our online subgraph clustering. The performance was stable for a wide range of $m$, and the selected $m$ was shared for all subtypes and classes. Based on our experiments, the car class was more inclined to be clustered with respect to the vehicle type. In Table V, we provide the validated $K_n$ in the VisDA dataset. Since the two settings of $K_n$ had similar performance, we selected the smaller $K_n$ as the subtype number.

Table IV tabulates the accuracy using VisDA17. We used suffixes +Dyn, -$\mu_k$, and -$\tau$ to denote SubUDA with queue memory, without subtype balance weight, and without semihard target mining, respectively. Besides, the suffix -$\eta_k$ denotes using $\mu_{st} = \frac{\sum_{i=1}^{N_k} f(x_{st}^i)/M_k}{\sum_{i=1}^{N_k} M_k + M_k'}$ as the subtype centroid, which was not robust against the subtype label shift. For each proposed approach, we independently ran the test five times, and the average and standard deviation (SD) of the accuracy metrics were reported.

All SubUDAs, including the $K$-means and subgraph-based methods, outperformed TPN [10] by around 2% with respect to mean accuracy, demonstrating the effectiveness of subtype-wise compactness. SubUDA achieved even superior performance over the methods with the stronger ResNet-152 backbone [34], [68]. We also adopted ResNet-152 as our backbone to demonstrate the generalizability of SubUDA. The improvement was consistent with the ResNet101 backbone. CAN [46] has been a powerful baseline for image classification UDA. Notably, CAN introduced the interclass domain discrepancy measure, which improves the mean accuracy in VisDA-17 over the CAN with only interclass term by 3.3%. In addition, there are several training tricks used in CAN to boost the performance, e.g., alternative optimization and class-aware sampling. We also used CAN as our baseline and added the subtype compactness objectives to further boost the performance by 1.9%.

Similar to the CHD transfer task, the dynamic quench scheme substantially boosted adaptation performance. Without a clear subtype definition, the semihard target mining may also be helpful to reject confusing samples. Considering that there can be many underlying subtypes, balancing the cluster size with $\omega_k$ is also necessary. Besides, as more subtypes lead to fewer samples in each subtype cluster, the more severe the subtype label shift is likely to occur. Therefore, calculating $\mu_{st}$ in a subtype label shift robust manner is crucial. We did not test SubUDA-DR since its clustering speed was not scalable to large-scale datasets.

Converting clustering into deep clustering [23] is highly challenging and costly. With our proposed dynamic online clustering and dimension reduction scheme, the training of VisDA took about 10 h, while CRST [2] took about 8 h. We note that the speed of testing was the same as conventional alternatives. The memory modules used in the dynamic online
TABLE VI

DETAILED STATISTICS OF EACH DIVISION IN THE DOMAINNET DATASET. WE REGARD THE DIVISION AND SUBCLASS LABEL AS "CLASS" AND "SUBTYPE" LABEL, RESPECTIVELY

| Divisions | Furniture | Mammal | Tool | Cloth | Electricity | Building | Office | Human | Road Trans | Food | Nature | Cold Blooded |
|-----------|-----------|--------|------|-------|-------------|----------|--------|-------|------------|------|---------|-------------|
| Subtype   | 35        | 25     | 28   | 23    | 25          | 21       | 21     | 20    | 15         | 14   | 14      | 12          |

| Divisions | Music | Fruit | Sport | Tree | Bird | Vegetable | Shape | Kitchen | Water Trans | Sky Trans | Insect | Others |
|-----------|-------|-------|-------|------|------|-----------|-------|---------|-------------|-----------|--------|--------|
| Subtype   | 10    | 9     | 10    | 8    | 7    | 8         | 7     | 5       | 6           | 4         | 4      | 13     |

scheme only stored the extracted feature vector, which required 1.4× more memory than vanilla TPN. With an acceptable and practical training cost, the improvement was substantial.

C. Large-Scale Two-Level UDA in DomainNet

DomainNet [82] is the largest domain adaptation dataset to date, which consists of ~0.6M images from 24 divisions (meta-class). There are the second-level labels of 345 subclasses for 24 meta-classes, as listed in Table VI. DomainNet has six distinct domains, i.e., Clipart (clp), Infograph (inf), Painting (pnt), Quickdraw (qdr),4 Real (rel), and Sketch (skt). For each domain, the standard evaluation protocol split the dataset, following a 70%/30% scheme for training and testing sets, respectively. The number of instances for both 24 meta-classes and 345 subclasses is different,5 which introduces the label shifts at both class and subtype levels.

To utilize the two-level label system in DomainNet to investigate the subtype-aware UDA, we combined the subclasses with corresponding meta-classes in a division into corresponding meta-classes and assumed that the subclasses are not accessible. Therefore, we only have the meta-class label in the source domain, and the classifier works for meta-class prediction. The subclass within a meta-class can be greatly different from one another. For example, sport meta-class includes baseball, yoga, hockey puck, and so on. It is difficult to extract the shared pattern across these subclasses. With this setting, it is possible to investigate the effect of subtype on large-scale datasets.

We reimplemented several recent methods for the meta-class classification, including TPN [10], PANDA [76], and DMRL [77]. The superscript “meta” means that we only used meta-class label, and the subtype label was not accessible. We set α = 1, λ = 0.5, and β = 0.5 and used Adam with lr = 1e−4.

To compare our framework with the subtype label supervision method, we used subclass and grouped the prediction of subclasses as the meta-class prediction. We note that the subtype label supervised setting can empirically be an “upper bound” of our subtype label unsupervised setting. We denote this setting with the superscript “sub.”

We used suffixes -ωk and -τ to denote the subtype balance weight and semihard target mining, respectively. We note that we cannot efficiently or practically implement the clustering in DomainNet without either the dynamic memory scheme or dimension reduction. We also assumed that Kω was unknown and thus used the adaptive subgraph construction. We validated the choice of m as shown in Fig. 13 and consistently chose m = 10.

By explicitly considering the underlying subtypes, our SubUDA+Dyn outperformed the baseline TPN [10] and the recent SOTA methods [76], [77]. Since we knew the number of subclass in DomainNet, we simply assigned Kn in K-maps, according to the statistics in Table VI, which is denoted by SubUDA+Dynkmeta. With the prior knowledge of subtype number, the performance is better than the adaptive subgraph construction-based SubUDA+Dynkmetak. The higher accuracy of SubUDA+Dynkmetakm=10 over SubUDA+Dynkmetakm=0−ωk and SubUDA+Dynkmetakm=10−τ clearly indicates the effectiveness of subtype balance weight and the semihard target mining.

The variation within a subclass in DomainNet can be large. For instance, the subclass of the airplane can involve jet, turboprop, single-engine aerobatic, and so on. In Table VII, we provide the validated Kn in the VisDA dataset. For the two settings where Kn had similar performance, we selected the smaller Kn as the subtype number. We can find that the validated Kn is consistently larger than the subclass number, which indicates that some of the subclasses can have large variations and should be further split.

In Table VIII, with the validated Kn, SubUDA+DynVah−kωk achieved a state-of-the-art performance. More appealingly, it outperformed the subtype label supervised TPNsub. We note that fine-grained labeling is costly and noisy in real-world applications. Moreover, validating Kn is costly in large-scale datasets. Considering the relatively small improvement of SubUDA+DynVah−kωk over SubUDA+Dynkmeta, we may not need to know the number of subclass.

Our SubUDA framework does not require that the classes or subtypes are the same in the source and target domains. To investigate the robustness of our framework in the case where only a subset of subtypes exist in the target domain, we investigated the performance of removing 0%, 25%, 50%, and 75% subtypes in the training set of the target domain. We used the shared 25% subtypes in each class in our testing set at the testing stage. The subtypes in testing existed in all training settings. The result is shown in Fig. 14. We can see that the performance of both TPN and our SubUDA drops due to the label shift [79]. However, our subtype clustering framework was less affected by the label shift, compared with TPN [10].

Though our method does not require the number of categories to be the same between two domains, it is not

---

4https://quickdraw.withgoogle.com/data
5http://ai.bu.edu/M3SDA/
TABLE VIII

| Single-Source Meta-Class Evaluations on the DomainNet [82]. Several Single-Source Adaptation Baselines are Validated on the DomainNet-MetaClass Dataset, including TPN [10], PANDA [76], and DMRL [77]. The TPN* metadata utilized the subtype label as a supervision signal, which can be regarded as an upper bound. The left column denotes the source domain, and the top row denotes the target domain. The average in the most right column represents the average accuracy of each row. The all-ave. in the bottom denotes the average performance of all domain combinations.

| PANDA*meta | TPN*meta | DMRL*meta |
|------------|----------|-----------|
| clp        | N/A      | N/A       | N/A       |
| inf        | N/A      | N/A       | N/A       |
| cnt        | N/A      | N/A       | N/A       |
| qdr        | rel      | rel       | rel       |
| skt        | rel      | rel       | rel       |

TABLE IX

| Experimental Results for GTA5 to Cityscapes |
|--------------------------------------------|
| Fig. 14. Sensitive analysis of subtype difference in source and target domains of TPN (gray line) [10], SubUDA+Dynmeta=10 (red line), and SubUDA+Dynmeta=10 (blue line). |

specifically designed for the partial UDA. We are exploring the subtype, which is a more fine-grained level of the category. Therefore, it is more reasonable to investigate the label shift of subtype rather than category (e.g., partial UDA), which is evidenced in Fig. 14. In addition, for partial UDA tasks [79], the performance of our SubUDA is similar to the TPN.

D. UDA for Segmentation

We then investigated the effectiveness of the subtype compactness in segmentation, which is equivalent to pixel-wise classification in segmentation. We evaluated our framework on the GTA5 [85] to Cityscapes [84] task, which has 19 shared segmentation classes. There are more than 24000 labeled game engine rendered images in the GTA5 dataset. We followed the evaluation protocol and backbones as in [52] and adopted the cropping strategy in ROAD [86] to enlarge the batch size to 10 as in [86]. We note that representing each pixel with a feature vector and clustering them can be quite costly. For example, the batch size in [84] is set to 1. We simply set \( m = 20 \) for all classes in our SubUDA+Dyn. We note that, similar to DomainNet, SubUDA without Dyn was not scalable to the segmentation task.

We compare our SubUDA+Dyn with the other methods in Table IX. Based on the class-wise compactness method of [84], SubUDA outperformed [84], by 3.7% with respect to the mean Intersection-Over-Unions (IoUs). In comparison, Cords et al. [84] focused on the class-wise compactness, which is different from our subtype-wise structure. The point-wise feature similarity within a class can be problematic, e.g., simply enforcing the similarity between the spatial features related to the car’s window and car’s wheel. Actually, in [84], the samples in each cluster were known, and we only needed to calculate the distance. We note that this is the case when we have the label in the source domain and perform supervised clustering. However, it would be more challenging for unsupervised clustering (e.g., subgraph) to identify the examples in each cluster without the subtype label in both domains.

V. CONCLUSION

In this work, we presented a novel approach to adaptively carry out the fine-grained subtype-aware alignment in UDA, based on the observations that subtype-wise conditional and label shifts widenly exist and those shifts can be adaptively...
aligned without the subtype label. We proposed a flexible yet principled approach for the case with or without prior knowledge on the number of subtypes. In addition to concise K-means, we further extended our approach with an online subgraph scheme using a reliability path, which can be scalable to many classes and subtypes with only a few meta hyperparameters. We also demonstrated that our alternatively updated dynamic memory queue and centroids modules could effectively process the imbalanced and undersampled subtypes. Our thorough evaluations on CHD transfer and large-scale general UDA tasks clearly demonstrated its validity and superiority to the recent SOTA UDA approaches.

REFERENCES

[1] T. Che et al., “Deep verifier networks: Verification of deep discriminative models with deep generative models,” 2019, arXiv:1911.07421.
[2] Y. Zou, Z. Yu, X. Liu, B. V. K. V. Kumar, and J. Wang, “Confidence regression,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 5982–5991.
[3] X. Liu, Y. Zou, Y. Song, C. Yang, J. You, and B. V. K. Kumar, “Ordinal regression with neuron stick-breaking for medical diagnosis,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 1–11.
[4] X. Liu et al., “Feature-level frankenstein: Eliminating variations for discriminative recognition,” in Proc. Int. Conf. Comput. Vis. (ICCV), Jun. 2019, pp. 637–646.
[5] W. M. Kouw and M. Loog, “An introduction to domain adaptation and transfer learning,” 2018, arXiv:1812.11806.
[6] X. Liu, S. Li, Y. Ge, P. Ye, J. You, and J. Lu, “Recursive conditional Gaussian for ordinal unsupervised domain adaptation,” in Proc. ICCV, Oct. 2021, pp. 764–773.
[7] D. Shen, G. Wu, and H.-I. Suk, “Deep learning in medical image analysis,” Annu. Rev. Biomed. Eng., vol. 19, pp. 221–248, Jun. 2017.
[8] L. Shao, F. Zhu, and X. Li, “Transfer learning for visual categorization: A survey,” IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 5, pp. 1019–1034, May 2015.
[9] L. Zhang, J. Fu, S. Wang, D. Zhang, Z. Dong, and C. L. P. Chen, “Guide profiling,” outcome in pediatric acute lymphoblastic leukemia by gene expression classification,” in Proc. AAAI, 2021, vol. 35, no. 3, pp. 2189–2197.
[10] J. Li et al., “Relaxed asymmetric deep hashing learning: Point-to-angle matching,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 11, pp. 4791–4805, Nov. 2020.
[11] X. Liu, W. Ji, J. You, G. E. Fakih, and J. Woo, “Severity-aware semantic segmentation with reinforced Wasserstein training,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 12566–12575.
[12] X. Liu, T. Che, Y. Lu, and C. Yang, “AUTO3D: Novel view synthesis through unsupervised learned variational viewpoint and global 3D representation,” in Proc. ECCV, 2020, pp. 52–71.
[13] X. Liu, X. Han, Y. Qiao, Y. Ge, and L. Jun, “Unimodal-uniform constrained Wasserstein training for medical diagnosis,” 2019, arXiv:1911.02475.
[14] X. Liu et al., “Unimodal regularized neuron stick-breaking for ordinal classification,” Neurocomputing, vol. 388, pp. 34–44, 2020.
[15] X. Liu, B. Hu, X. Liu, J. Lu, J. You, and L. Kong, “Energy-constrained self-training for unsupervised domain adaptation,” in Proc. ICPR, Jan. 2021, pp. 7515–7520.
[16] X. Liu, F. Xing, C. Yang, G. E. Fakih, and J. Woo, “Adapting off-the-shelf source segmenter for target medical image segmentation,” in Proc. MICCAI, 2021, pp. 549–557, 2020, pp.1019–1034, May 2015.
[17] X. Liu et al., “Generative self-training for cross-domain unsupervised tagged-to-cine MRI synthesis,” in Proc. MICCAI, 2021, pp. 138–148.
[18] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, “Transferable attention for domain adaptation,” in Proc. AAAI Conf. Artif. Intell., 2019, vol. 33, no. 1, pp. 5345–5352.
[19] P. O. Pinheiro, “Unsupervised domain adaptation with similarity learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8004–8013.
[20] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, “Simultaneous deep transfer across domains and tasks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 4068–4076.
[21] Y. Zhang, T. Liu, M. Long, and M. Jordan, “Bridging theory and algorithm for domain adaptation,” in Proc. Int. Conf. Mach. Learn., Jun. 2019, pp. 7404–7413.
[22] S. Magliacane, T. van Ommen, T. Claassen, and S. Bongers, “Domain adaptation by using causal inference to predict invariant conditional distributions,” in Proc. NIPS, 2018, pp. 1–11.
[23] S. Li, S. Song, G. Huang, Z. Ding, and C. Wu, “Domain invariant and class discriminative feature learning for visual domain adaptation,” IEEE Trans. Image Process., vol. 27, no. 9, pp. 4260–4273, Sep. 2018.
[24] Y. S. Chan and H. T. Ng, “Word sense disambiguation with distributional semantics,” in Proc. Int. Conf. Mach. Learn., Jun. 2019, pp. 7404–7413.
[25] S. Li et al., “Deep residual correction network for partial domain adaptation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 7, pp. 2329–2344, Jul. 2021.
[26] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, “Domain adaptation with conditional transferable components,” in Proc. ICML, 2016, pp. 2839–2848.
[27] M. Long, Y. Cao, J. Wang, and M. I. Jordan, “Learning transferable features with deep adaptation networks,” in Proc. ICML, 2015, pp. 97–105.
[28] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in Proc. CVPR, Jul. 2017, pp. 7167–7176.
[29] J. G. Moreno-Torres, T. Raeder, R. Aliax-Rodríguez, N. V. Chawa, and F. Herrera, “A unifying view on dataset shift in classification,” Pattern Recognit., vol. 45, no. 1, pp. 521–530, Jan. 2012.
[30] C. Chen et al., “Progressive feature alignment for unsupervised domain adaptation,” in Proc. CVPR, Jun. 2019, pp. 627–636.
[31] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, “Contrastive adaptation network for unsupervised domain adaptation,” in Proc. CVPR, Jun. 2019, pp. 4893–4902.
[32] X. Gu, J. Sun, and Z. Xu, “Spherical space domain adaptation with robust pseudo-label loss,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9101–9110.
[33] Y. Zhu et al., “Deep subspace domain adaptation network for image classification,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 4, pp. 1713–1722, Apr. 2021.
A. Coates and A. Y. Ng, “Learning feature representations with
Y. Pan, T. Yao, Y. Li, C.-W. Ngo, and T. Mei, “Exploring
G. He, X. Liu, F. Fan, and J. You, “Classification-aware semi-supervised
domain adaptation,” in Proc. CVPR Workshops, Jun. 2020, pp. 964–965.
Z. Wang, B. Du, and Y. Guo, “Domain adaptation with neural embedding
matching,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 7, pp. 2387–2397, Jul. 2020.
Q. Wang and T. Breckon, “Unsupervised domain adaptation via
structured prediction based selective pseudo-labeling,” in Proc. AAAI Conf.
Artif. Intell., 2020, vol. 34, no. 4, pp. 6243–6250.
Y. Jin, X. Wang, M. Long, and J. Wang, “Minimum class confusion for
versatile domain adaptation,” in Proc. Eur. Conf. Comput. Vis. Springer, 2020, pp. 464–480.
H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang, “Separate to
adapt: Open set domain adaptation via progressive separation,” in Proc.
IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019,
pp. 2927–2936.
B. Fu, Z. Cao, M. Long, and J. Wang, “Learning to detect open classes
for universal domain adaptation,” in Proc. Eur. Conf. Comput. Vis. Springer, 2020, pp. 567–583.
S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, “Analysis of
representations for domain adaptation,” in Proc. NIPS, 2007, pp. 1–8.
X. Liu, B. V. K. V. Kumar, P. Jia, and J. You, “Hard negative
generation for identity-disentangled facial expression recognition,” Pattern
Recognit., vol. 88, pp. 1–12, Apr. 2019.
A. Fahad et al., “A survey of clustering algorithms for big data:
Taxonomy and empirical analysis,” IEEE Trans. Emerg. Topics Comput.,
vol. 2, no. 3, pp. 267–279, Sep. 2014.
F. M. Carlucci, A. D’Innocente, S. Bucci, B. Caputo, and T. Tommasi,
“Domain generalization by solving jigsaw puzzles,” in Proc. CVPR,
Jun. 2019, pp. 2229–2238.
A. G. Howard et al., “MobileNets: Efficient convolutional neural
networks for mobile vision applications,” 2017, arXiv:1704.04861.
K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, “Maximum classi-
fier discrepancy for unsupervised domain adaptation,” in Proc. CVPR,
Jun. 2018, pp. 3723–3732.