Adaptive Betweenness Clustering for Semi-Supervised Domain Adaptation

Jichang Li, Guanbin Li, *Member, IEEE*, and Yizhou Yu, *Fellow, IEEE*

Abstract— Compared to unsupervised domain adaptation, semi-supervised domain adaptation (SSDA) aims to significantly improve the classification performance and generalization capability of the model by leveraging the presence of a small amount of labeled data from the target domain. Several SSDA approaches have been developed to enable semantic-aligned feature confusion between labeled (or pseudo labeled) samples across domains; nevertheless, owing to the scarcity of semantic label information of the target domain, they were arduous to fully realize their potential. In this study, we propose a novel SSDA approach named Graph-based Adaptive Betweenness Clustering (G-ABC) for achieving categorical domain alignment, which enables cross-domain semantic alignment by mandating semantic transfer from labeled data of both the source and target domains to unlabeled target samples. In particular, a heterogeneous graph is initially constructed to reflect the pairwise relationships between labeled samples from both domains and unlabeled ones of the target domain. Then, to degrade the noisy connectivity in the graph, connectivity refinement is conducted by introducing two strategies, namely Confidence Uncertainty based Node Removal and Prediction Dissimilarity based Edge Pruning. Once the graph has been refined, Adaptive Betweenness Clustering is introduced to facilitate semantic transfer by using across-domain betweenness clustering and within-domain betweenness clustering, thereby propagating semantic label information from labeled samples across domains to unlabeled target data. Extensive experiments on three standard benchmark datasets, namely DomainNet, Office-Home, and Office-31, indicated that our method outperforms previous state-of-the-art SSDA approaches, demonstrating the superiority of the proposed G-ABC algorithm.

Index Terms— Semi-supervised domain adaptation, adaptive betweenness clustering, categorical domain alignment.

I. Introduction

DEEP neural network (DNN) has led to a series of breakthroughs in computer vision tasks such as Image Classification [1], [2], [3], [4], [5], Semantic Segmentation [6], [7], [8], [9], Object Detection [10], [11], [12], [13], Medical

Analysis [14], [15], etc. However, the impressive effectiveness of the training of deep network models remarkably depends on a large number of sample labels, necessitating laborious work in data annotation. An alternative solution comprises boosting the model for the domain of interest (a.k.a., target domain) by employing off-the-shelf labeled training samples from a relevant domain (a.k.a., source domain). Nonetheless, due to the distribution/domain gap, such a solution often cannot generalize well from the source domain to the target domain to deal with variant circumstances of domain gaps. Unsupervised domain adaptation (UDA), which aims to tackle the distribution gap and decrease the influence of domain shift, has thus gained significant attention for a long time [16], [17], [18], [19], [20], [21], [22]. Recently, semi-supervised domain adaptation (SSDA), a variant of the UDA task, has received wider attention [23], [24], [25]. With a few labeled target samples, SSDA can significantly enhance the adaptation model's performance w.r.t the target domain, compared to unsupervised domain adaptation. In this way, a small amount of annotated data in the target domain can be used to expand the semantic space, allowing a large number of samples of the same category from different domains to be clustered together at the feature level, so as to achieve partial categorical alignment.

Despite the advantages of SSDA over UDA, SSDA also presents its own specific challenges. At first, in the SSDA scenario, a supervised model trained on a small number of labeled target samples and a large amount of labeled source data can only achieve partial cross-domain feature alignment, as it only aligns features of labeled target samples and their correlated nearby samples with the corresponding feature clusters in the source domain [24]. In addition, the trained SSDA model is incapable of producing highly discriminative feature representations for the target domain since the massive labels of the source domain dominate the supervision and causes the learned feature representations to be biased towards the discrimination of the source domain [23]. Preliminary SSDA works, such as [23], [24], [25], and [26], have each proposed their individual solutions to tackle these challenges, and significant performance improvement has been witnessed. However, as with previous UDA studies focusing on global feature confusion at the domain level [20], [27], [28], [29], existing works are still unable to reach globally categorical domain alignment due to the scarcity of semantic label information for each category in the target domain. In other words, despite the presence of perfect domain-level alignment in feature confusion, it leads to label mismatch between massive unlabeled target
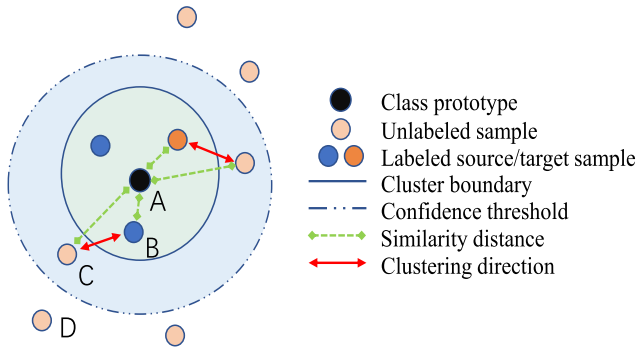
Fig. 1. An example to illustrate Adaptive Betweenness Clustering (ABC). The proposed G-ABC algorithm conducts sample clustering between a labeled point (e.g., "B") and an unlabeled point (e.g., "C"), when they have similarity distances within a confidence threshold to the same class prototype (e.g., "A") and they are with similar prediction distributions from the model. Herein, Point "A" guides the clustering process by serving as an intermediary (betweenness) point for Points "B" and "C". Point "D" is outside of the clustering range.

data and the data of the source domain, hence compromising the model's performance. Recent SSDA algorithms, such as [30] and [31], have demonstrated that the semantic-aligned feature confusion across domains appears to work properly in semi-supervised domain adaptation, since during domain alignment, sample features from both domains with the same class will likely be aggregated into a same cluster. However, these methods achieve categorical domain alignment mostly through promoting semantic alignment between labeled samples across domains, with far less emphasis on employing a vast amount of unlabeled target samples.

In this paper, we present a novel SSDA approach, named **G**raph-based **A**daptive **B**etweenness **C**lustering (**G-ABC**), to tackle the challenges of the SSDA tasks. To achieve globally categorical domain alignment, we propose to enforce semantic transfer from labeled samples across domains to unlabeled target samples in order to promote cross-domain semantic alignment. Using the ground-truth sample labels from both domains as references, the trained SSDA model may thereby propagate semantic label information to the unlabeled target samples. In this way, a substantial amount of target label information is augmented through semantic propagation, thus significantly enhancing the generalization of the model to the target domain. Specifically, we first construct a heterogeneous graph to capture the pairwise associations between unlabeled target samples and labeled samples from either the source or target domain, based on the pairwise label similarity of those paired samples. Then, we provide two connectivity refinement strategies, namely Confidence Uncertainty based Node Removal and Prediction Dissimilarity based Edge Pruning, to eliminate the noisy connectivity in the graph. In detail, the former degrades the connectivity towards unreliable unlabeled samples by removing unlabeled target instances with low predicted confidence, whereas the latter prunes the graph connections between samples with divergent probabilistic prediction distributions.

With the refined graph structure, we design a new clustering algorithm, namley Adaptive Bewteenness Clustering (ABC) as shown in Fig. 1. To achieve semantic transfer,

in this algorithm, we model the task of sample clustering between a labeled and an unlabeled sample as a binary pairwise-classification problem. The fundamental premise behind such an algorithm is to aggregate the feature representations of the paired samples that share the same class in the graph while separating those of different classes. In particular, this algorithm involves two strategies to achieve semantic propagation, namely across-domain betweenness clustering (ADBC) and within-domain betweenness clustering (WDBC), by clustering the unlabeled target samples towards the labeled source or target domains. As a result, the ADBC strategy encourages alignment between unlabeled target samples and the source domain, whereas the WDBC scheme strengthens linkages between labeled and unlabeled target samples. Ultimately, semantic label information can be gradually transferred into unlabeled target instances as model training iterates. In this way, the rising balance of semantic label information of the source and target domains eliminates model bias toward the source domain and achieves globally categorical domain alignment, driving the model to generate more domain-invariant yet discriminative target features.

To sum up, our main contributions can be shown as follows:

- We propose a novel SSDA framework called **G**raph-based **A**daptive **B**etweenness **C**lustering (**G-ABC**) to tackle semi-supervised domain adaptation. To achieve globally categorical domain alignment, the proposed G-ABC conducts cross-domain semantic alignment with semantic transfer from labeled samples of both domains to unlabeled target data.
- We construct a heterogeneous graph to characterize the associations between unlabeled target examples and labeled data of both domains. Two connectivity refinement strategies, namely Confidence Uncertainty based Node Removal and Prediction Dissimilarity based Edge Pruning, are further provided to decrease the noisy connectivity in the graph.
- Given the above-refined graph structure, we propose Adaptive Betweenness Clustering to impose semantic transfer across domains; in particular, we design across-domain betweenness clustering and within-domain betweenness clustering, respectively, to propagate semantic label information from labeled source and target domains to unlabeled target samples.
- We perform extensive experiments on three standard benchmark datasets, including DomainNet [32], Office-Home [33] and Office-31 [34], to verify the effectiveness of our proposed method, and the results show that our method outperforms all previous state-of-the-art SSDA methods by clear margins.

This paper is organized as follows. In Sec. II, we overview the prior research related to our work. In Sec. III, we present and describe the proposed algorithm for semi-supervised domain adaptation. Furthermore, we conduct comparative experiments to evaluate the performance of the proposed approach in Sec. IV, while the conclusions are illustrated in Sec. V.

II. RELATED WORK

A. Domain Adaptation

Domain adaptation (DA) addresses the problem of generalizing a model trained on a large number of labeled samples from the source domain to the target domain [35], [36], [37], [38], [38]. With the goal of decreasing the distribution gap across various domains, the most challenging issue of the DA problem is assisting the model in learning domain-invariant features. To accomplish domain adaptation, early classic algorithms to handle the domain adaptation tasks involve reducing the distribution discrepancies across domains assessed by Maximum Mean Discrepancy (MMD) [27], [28], sharing the identical cross-domain statistics (e.g., mean value and covariance) [39], and so on. Tzeng et al. [40], for instance, presented a method for leveraging MMD to drive the model to generate domain-invariant features by assessing the discrepancies between the model outputs of both domains. Long et al. optimized the adaptation model from a different perspective [41], i.e., employing multi-kernel MMD to evaluate the output differences of samples across domains at multiple model levels. In addition, recent advances trends favor adversarial domain alignment to expedite feature alignment across domains so that knowledge from classifiers trained on labeled source samples can be efficiently transferred to the target domain [20], [29], [42], [43], [44], [45], [46]. For example, Saito et al. conducted minimax training over unlabeled target samples to cluster these target features around domain-invariant class prototypes, thus imposing cross-domain feature alignment [23]. The aforementioned DA algorithms aimed at aligning source and target features at the domain level. However, many related advances, such as [30], [47], [48], and [49], demonstrated that decreasing the discrepancies of conditional distributions towards categorical domain alignments is preferred, resulting in improved adaptation between domains. To this end, it should be natural to incorporate semantic label information into adaptation. For instance, semantic alignment was proposed in [50], [51], and [52] to achieve this. Motivated by it, we also emphasize semantic-level domain alignment, leveraging source and target labels as references, and encouraging semantic transfer from labeled source and target domains to unlabeled target samples.

B. Domain Adaptation Related to Graphs

The graphs employed in domain adaptation capture latent topological structures among the training data across domains, such that the learned relationships between domain samples can then be leveraged to encourage the model to better adapt reliable data structures from both domains. In general, samples from the source and target domains are represented by distinct graph structures. Thus, many previous DA algorithms relating to graphs, such as [53], [54], and [55], first attempted to transfer knowledge learned on a labeled source graph to an unlabeled target graph. Based on the labeled graph in the source domain, these algorithms engaged source labels as the supervision signals, and the model therefore received training on both the source graph and the target graph respectively. For example, Pilanci et al. employed frequency analysis to align two data graphs through which information

can be transferred or shared [56], [57]. On the other hand, Ding et al. [58] proposed constructing a cross-domain graph based on samples from both domains to capture the intrinsic structure in the shared space among the training data in order to concurrently enforce domain transfer and label propagation. In this way, domain-invariant feature learning and target discriminative feature learning are unified into the same framework, thus benefiting each other for more effective knowledge transfer. Based on such an observation, we also propose to hire the cross-domain graph to achieve categorical domain alignment. Nevertheless, earlier related works usually model graph Laplacian regularization to push graph nodes closer, but this unsupervised technique disregards the usage of sample labels to assist semantics alignment [59], [60], [61]. In our work, we design a novel clustering algorithm called Adaptive Betweenness Clustering to take full advantage of both source and target labels, thereby contributing to enhanced performance for the model to classify target samples.

C. Semi-Supervised Domain Adaptation

Due to the availability of a few target labels, semi-supervised domain adaptation has the potential to significantly improve the classification performance of the model on the target domain in comparison to unsupervised domain adaptation [24], [62], [63], [64], [65]. Recent progress in SSDA, such as [23], [24], [26], [66], [67], and [25] have primarily focused on adversarial training to align cross-domain feature distributions. Here we mainly describe some related SSDA approaches that do not involve adversarial learning. For example, Mishra et al. [68] demonstrated that without the need for conventional adversarial domain alignment, self-supervision based pre-training and consistency regularization might be relied upon to produce a stronger classifier in the target domain, while Luo et al. [31] developed "Relaxed cGAN" to transfer image styles from source samples to unlabeled target samples in order to help achieve domain-level distribution alignment. Besides, Fang et al. [65] also focus on style transferring for achieving better adaptation, generating assistant features by transferring intermediate styles between labeled and unlabeled samples.

In addition to bridging the gap and exchanging knowledge between the source and target domains, Yang et al. [64] proposed decomposing the SSDA task into a semi-supervised learning (SSL) problem and an unsupervised domain adaptation problem. Specifically, the former is used to improve discrimination in the target domain, whereas the latter facilitates domain alignment. Such an algorithm trains two distinct classifiers utilizing Mixup and Co-training, respectively, in order to learn complementary features to each other, resulting in better domain adaptation. Similarly, [49] also performed adaptation by learning two classifier networks, trained for contradictory purposes. The first classifier groups target features to enhance intra-class density and increase categorical cluster gaps for robust learning, while the second, as a regularizer, disperses source features for a smoother decision boundary.

In this paper, we extend the motivation from [31] and [30] and propose G-ABC, which achieves categorical domain
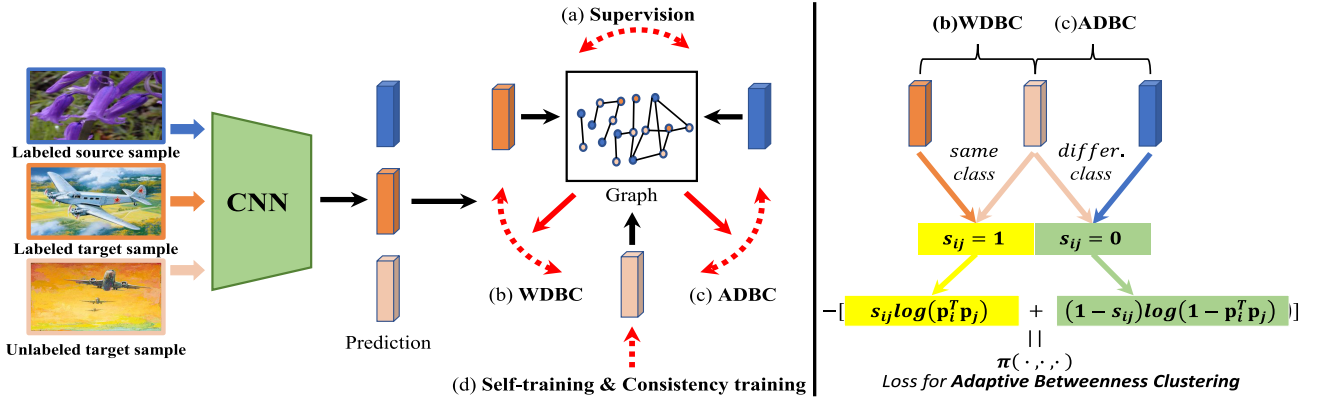
Fig. 2. An overview of the proposed framework and the training loss for Adaptive Betweenness Clustering. **Left:** (a) Supervision of labeled data from both source and target domains is applied to guarantee partially categorical domain alignment. (b) Within-domain betweenness clustering (WDBC) is used to determine the relationship between labeled and unlabeled target data. (c) Across-domain betweenness clustering (ADBC) is used to effectively align unlabeled target samples with the source domain. D) Auxiliary techniques for model optimization, including self-training, consistency training, etc. These four components together enable globally categorical domain alignment, progressively enhancing the model's performance, with (b) and (c) establishing reliable sample connectivity among training samples, represented by a heterogeneous graph. **Right:** Given a pairwise label s_{ij} between samples, the training loss of Adaptive Betweenness Clustering aims to bring samples from the same class closer together in the feature space when $s_{ij} = 1$, or to separate samples from different classes when $s_{ij} = 0$. This allows for semantic transfer from labeled source or target domains to unlabeled target samples. Note that the orange and light-orange samples belong to the same category, namely "plane", while the blue sample is from a different category, i.e. "flower".

alignment by providing increased access to unlabeled target samples during adaptation.

While there appears to be a superficial similarity between [69] and our proposed method in encouraging consistent predictions between features, we would like to clarify their differences as follows.

1) Reference [69] merely encourages prediction consistency or similarity between a feature and its few neighbors, all of which are from the unlabeled target data. In contrast, our method, G-ABC, employs the proposed clustering technique called Adaptive Betweenness Clustering to group unlabeled samples toward labeled source or target instances. This is achieved by enforcing consistent probabilistic prediction distributions between two similar samples while forcing inconsistency otherwise. More specifically, using the ground-truth sample labels from both domains as references, G-ABC is more effective in propagating the semantic label information from the labeled source and target domains to the unlabeled target examples. Conversely, [69] cannot achieve this, as it only makes the unlabeled target features more compact in an unsupervised manner.

2) The pairwise label similarity we propose can be viewed as a more credible measure of pairwise relationships than the pairwise feature similarity used in [69]. This is especially true when the label information of labeled examples across domains can be trusted. Building upon this premise, we have taken two connectivity refinement strategies to build a reliable graph structure of pairwise relationships, which effectively mitigates the potential harm of noisy connectivity on model performance improvement. In contrast, the pairwise feature similarity introduced in [69] involves unsupervised sample matching, which makes it more challenging to accurately pair samples of the same class.

III. THE PROPOSED METHOD

In SSDA, we are given labeled samples from the source and target domains, denoted by $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ and $\mathcal{D}_l = \{(x_i^l, y_i^l)\}_{i=1}^{N_l}$, as well as unlabeled samples from the target domain, denoted as $\mathcal{D}_u = \{(x_i^u,)\}_{i=1}^{N_u}$, where N_s, N_l and N_u are the sizes of $\mathcal{D}_s$, $\mathcal{D}_l$ and $\mathcal{D}_u$, respectively. Our goal is to train an SSDA model using $\mathcal{D}_s$, $\mathcal{D}_l$ and $\mathcal{D}_u$, followed by evaluating the trained model on the target domain.

Like existing SSDA works, such as [23], [24], and [66], we first parameterize the SSDA model by θ, made up of two components, namely a feature extractor $\mathcal{F}$ and a classifier $\mathcal{G}$. The classifier $\mathcal{G}$ is an unbiased linear network with a normalization layer that maps the extracted features from the feature extractor $\mathcal{F}$ into a spherical feature space. Here, the weight vectors of the classifier are denoted as $W = [w_1, w_2, \cdots, w_K]$, and these vectors can be regarded as the prototypes that represent K classes [23], [70]. Accordingly, samples with the same class label from the source or target domains are mapped nearby to the same class prototype in the feature space. As demonstrated in [23] and [24], this has a considerable impact on minimizing the cross-domain feature variance of samples with the same class label. In a short, the normalized feature with temperature T of an input image x, $\mathbf{f} = \frac{1}{T}\frac{\mathcal{F}(x)}{\|\mathcal{F}(x)\|}$, is fed into the classifier $\mathcal{G}$ to obtain the probabilistic prediction as follows:

$$p(x) = \sigma(\mathcal{G}(\mathbf{f})) = \sigma(W^{\mathsf{T}}\mathbf{f}), \quad (1)$$

where $\sigma(\cdot)$ is a softmax function. $p(x)$ reflects the similarity scores, achieved by calculating the cosine distances, between the point x and the prototypes of distinct classes. For convenience, we often abbreviate $p(x)$ as $\mathbf{p}$, i.e., $\mathbf{p} = p(x)$.

A. Overview

In this paper, we propose Graph-based Adaptive Betweenness Clustering (G-ABC) to tackle semi-supervised domain

adaptation. In detail, we first construct a heterogeneous graph to depict the pairwise associations between labeled samples from both domains and unlabeled examples from the target domain. Then, to degrade the noisy sample connectivity, we refine the original heterogeneous graph using two strategies: Confidence Uncertainty based Node Removal (CUNR) and Prediction Dissimilarity based Edge Pruning (PDEP). Afterwards, to associate complementary characteristics of the source and target labels with unlabeled target samples, we present Adaptive Betweenness Clustering to conduct semantic propagation, facilitating semantic alignment between domains. Finally, we leverage off-the-shelf and well-established techniques, such as pseudo-label selection, self-training [71], [72] and consistency training [73], [74], [75], to further optimize the model in order to achieve globally categorical domain alignment. An overview of the proposed method has been summarized in Fig. 2.

B. Graph Construction

The goal of graph construction is to discover the sample connectivity of the training data with a heterogeneous graph $G = \langle V, E, A \rangle$. In particular, $V = \{v_i\}_{i=1}^{N_s+N_l+N_u}$ represents the collection of graph nodes consisting of labeled source instances from $\mathcal{D}_s$, labeled target samples from $\mathcal{D}_l$, and unlabeled target samples from $\mathcal{D}_u$, whereas E collects pairwise associations between a graph node of the unlabeled samples and the other node of the labeled examples. Then, the relationships between graph nodes given by the non-negative affinity matrix A can be calculated as follows,

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,N_l} & a_{1,N_l+1} & \cdots & a_{1,N_l+N_s} \\ a_{2,1} & \cdots & a_{2,N_l} & a_{2,N_l+1} & \cdots & a_{2,N_l+N_s} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{N_u,1} & \cdots & a_{N_u,N_l} & a_{N_u,N_l+1} & \cdots & a_{N_u,N_l+N_s} \end{bmatrix},$$

where rows of this matrix denote the unlabeled samples of the target domain, while the first N_l columns and the last N_s columns of A refer to the labeled samples from the target and source domains, respectively. In addition, a_{ij} (i.e., the ij-th entry of A) is the weight of the edge connecting between graph nodes v_i and v_j, which encodes the mutual relationship of the sample pair.

Generally, the weight of a connectivity in a graph can be determined using cosine similarity between sample features [76], [77]. We propose in this paper that the pairwise associations can be established by comparing the given ground-truth labels of labeled samples to the model's predicted class labels of unlabeled samples; we refer to this as pairwise label similarity. Compared to pairwise feature similarity, pairwise label similarity can be viewed as a more credible measure of pairwise relationships, providing that the label information of labeled examples across domains is trustworthy. In addition, for the predicted label of the unlabeled target sample, we take measures to mitigate the noisy connectivity of the established graph (See Sec. III-C). Afterwards, we can obtain the pairwise label similarity a_{ij} of an edge between graph nodes consisting of an unlabeled sample x_i (a.k.a, v_i), and a labeled sample

x_j (a.k.a, v_j) affiliated with its ground-truth class label y_j, as follows,

$$a_{ij} = \mathbb{1}\{\hat{y}_i = y_j\}, \tag{2}$$

where $\hat{y}_i = \arg\max_k(p(x_i)[k])$ is the predicted class label of the unlabeled target sample x_i, while $\mathbb{1}\{\cdot\}$ is a binary indicator function. Notice that we illustrate the initial sample connectivity of the built graph in Fig. 3(a).

C. Connectivity Refinement for Graph Unreliability

Once we obtain the initial heterogeneous graph, we introduce connectivity refinement to alleviate the unreliability of the graph. Due to high uncertainties caused by low predicted confidence, the graph nodes corresponding to unlabeled target samples are susceptible to receiving erroneous pseudo-labels, thereby resulting in noisy connectivity of the graph. In addition, as demonstrated in [78], [79], and [80], labels and features should vary smoothly over the edges of the graph so as to well conduct semantic propagation. To this end, features between nodes v_i and v_j of an edge should have a high degree of similarity when $a_{ij} = 1$, i.e., they should be neighbors in the spherical feature space. In this situation, the probabilistic prediction distributions of these two node samples predicted by the classifier ought to be similar. Consequently, there would be additional edges with noise in the graph, for which the probabilistic prediction distributions of both node samples from the model have a very low similarity. To eliminate the connectivity with noise and unreliability in the graph G, we present two connectivity refinement strategies, namely Confidence Uncertainty based Node Removal and Prediction Dissimilarity based Edge Pruning, resulting in a refined graph that represents the pairwise relationship of the data structure with high reliability. An illustration is shown in Fig. 3(b), (c), and (d).

1) Confidence Uncertainty Based Node Removal (CUNR): This strategy degrades the connectivity towards unreliable unlabeled samples through the removal of unlabeled target instances with low-confidence predictions. We employ a sufficiently high confidence threshold $\tau \in [0, 1]$ to choose reliable candidate nodes from unlabeled target samples:

$$g_i = \mathbb{1}\{\max_k(p(x_i)[k]) > \tau\}, \tag{3}$$

where g_i is a binary indicator to preserve the node $v_i \in V$ corresponding to unlabeled samples on the target domain when $g_i = 1$ and to remove the node v_i when $g_i = 0$.

2) Prediction Dissimilarity Based Edge Pruning (PDEP): As stated above, when $a_{ij} = 1$, the dissimilar prediction distributions between nodes of an edge in the graph might also lead to noisy connectivity in the graph, making a negative effect on semantic propagation. To remedy this, we first calculate the similarity score between the predicted label distributions of two nodes over an edge using dot product operation, and then threshold the similarity scores with a scalar κ so as to obtain more credible graph connectivity. Therefore, we can formularize it as follows,

$$\tilde{g}_{ij} = \neg a_{ij} \vee \mathbb{1}\{\mathbf{p}_i^\top \mathbf{p}_j > \kappa\}, \tag{4}$$
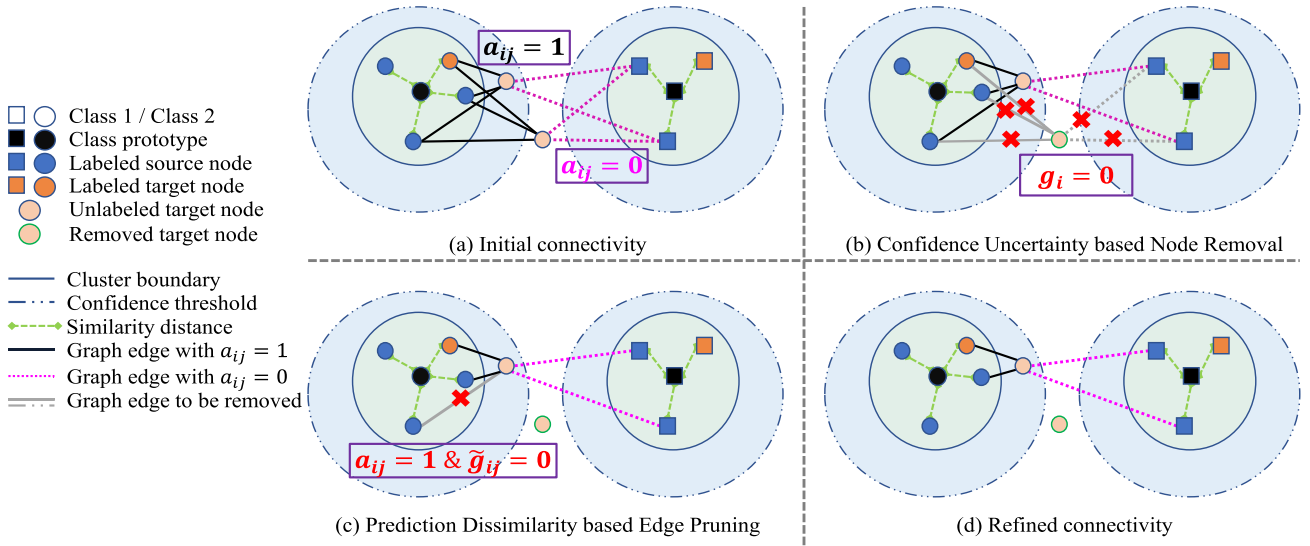
Fig. 3. A diagram depicting graph construction and connectivity refinement. (a) demonstrates the initially constructed sample connectivity of the graph, while (d) presents the refined graph after the connectivity refinement process is performed. The connectivity refinement process is further illustrated by (b) and (c), which effectively eliminate noisy connectivity through the CUNR and PDEP strategies, respectively, resulting in a more reliable graph structure to represent the relationships between samples. The technical details of these four subdiagrams are as follows: (a) Using pairwise label similarities between samples, the initial connectivity between training examples in a heterogeneous graph is constructed; (b) Confidence Uncertainty based Node Removal (CUNR) reduces the connectivity towards unreliable unlabeled samples by removing nodes with low predicted confidence; (c) Prediction Dissimilarity based Edge Pruning (PDEP) further removes the connections between graph samples whose probabilistic prediction distributions are dissimilar; (d) A refined graph is obtained to properly capture the pairwise associations between samples.

where $\mathbf{p}_i = p(x_i)$ and $\mathbf{p}_j = p(x_j)$. Besides, the binary indicator $\tilde{g}_{ij}$ serves to prune the graph edge connecting between nodes v_i (i.e., unlabeled target node x_i) and v_j (i.e., the sample x_j from labeled source or target data) when $\tilde{g}_{ij} = 0$; otherwise, the edge is preserved when $\tilde{g}_{ij} = 1$. In this manner, PDEP can effectively control the removal and preservation of the graph edge only when $a_{ij} = 1$. However, when $a_{ij} = 0$, PDEP becomes invalid, ensuring that the edge connecting nodes v_i and v_j in the graph will be consistently preserved.

Upon executing CUNR and PDEP, we can integrate Eq. (4) into Eq. (3) and then revise Eq. (3) for the rebuilt graph connectivity as follows,

$$g_i^j = g_i \cdot \tilde{g}_{ij}. \tag{5}$$

Here, the subscript i of g_i^j denotes the unlabeled target node x_i, and the superscript j indicates the other node x_j (from labeled source or target data) on the same edge as x_i.

Once we have the indicator g_i^j, we will be able to achieve more reliable connectivity between nodes in the graph G, hence improving the performance of semantic propagation.

D. Adaptive Betweenness Clustering

In this section, we conduct semantic transfer for categorical domain alignment using the updated graph that represents the reliable structure of the training data. Here, we propose a newly devised clustering algorithm for semantic propagation, called Adaptive Betweenness Clustering (ABC). On the basis of the rebuilt graph, such an algorithm propagates and aggregates labels along graph edges. This allows the semantic label information of a labeled sample to be transferred to an unlabeled sample with clustering between samples. Using this approach in the spherical feature space provided by the

prototypical classifier, the sample's predicted probability distributions indicate the cosine similarities between the feature and the prototypes for each category. Hence, this algorithm enforces a pair of samples with the same ground-truth (labeled) or predicted (unlabeled) class labels to have highly similar probability distributions. When the probability distribution of the latter is brought closer to that of the former, semantic propagation from labeled to unlabeled instances is achieved.

In specific, we first generate a sample pair from a labeled sample x_i and an unlabeled sample x_j by setting $s_{ij} = 1$ as a pairwise label if x_i and x_j belong to the same class, otherwise $s_{ij} = 0$ for different classes. Then, we adopt a binary cross-entropy loss to draw samples from the same class closer together in the feature space while separating samples from other classes. Utilizing a pairwise label as a target, adaptive betweenness clustering thus can be computed with the following loss:

$$\pi(x_i, x_j, s_{ij}) = -[s_{ij} \log(\mathbf{p}_i^\top \mathbf{p}_j) + (1 - s_{ij}) \log(1 - \mathbf{p}_i^\top \mathbf{p}_j)], \tag{6}$$

where $\mathbf{p}_i = p(x_i)$ and $\mathbf{p}_j = p(x_j)$. As demonstrated in [81] and [25], perturbations integrated into unlabeled target data can significantly improve the performance of the model; hence, we here augment unlabeled examples from the target domain for better propagation.

Due to domain shift in SSDA, the semantic information of labeled source samples, though is large in volume, is less correlated with unlabeled target examples, whereas the target label information of labeled samples has a greater correlation with unlabeled samples on the target domain, but is relatively scarce. Hence, to enable cross-domain semantic alignment, we propose across-domain betweenness clustering (ADBC)

and within-domain betweenness clustering (WDBC) to propagate semantic label information from labeled instances on the source and target domains to unlabeled target data. With the distinct but complimentary characteristics of semantic information from source and target labels, semantic transfer can be conducted with the following losses with regard to unlabeled target samples:

$$\mathcal{L}^{abc} = \mathcal{L}^{wdbc} + \mathcal{L}^{adbc}, \tag{7}$$

$$\mathcal{L}^{wdbc} = \frac{1}{N_u} \sum_{i \in I} \frac{1}{N_l} \sum_{j \in P} g_i^j \cdot \pi(x_i, x_j, a_{ij}), \tag{8}$$

$$\mathcal{L}^{adbc} = \frac{1}{N_u} \sum_{i \in I} \frac{1}{N_s} \sum_{j' \in Q} g_i^{j'} \cdot \pi(x_i, x_{j'}, a_{ij'}), \tag{9}$$

where I, P and Q denote the collections of unlabeled samples from the target domain, and labeled ones from the target and source domains, respectively, whose sample indexes are denoted by $i \in \{1, 2, \cdots, N_u\}$, $j \in \{1, 2, \cdots, N_l\}$ and $j' \in \{N_l + 1, N_l + 2, \cdots, N_l + N_s\}$.

Here, to gain a better understanding of how Eqs. (2)-(6) work well in Eq. (8) (Eq. (9) follows the same rule as Eq. (8)), we provide the following detailed clarifications.

- According to Eq. (3), when $g_i > 0$ (i.e. $g_i = 1$), the model prediction remains confident. At this time, if $a_{ij} = 1$ (i.e. $s_{ij} = 1$ in Eq. (6)) and $\mathbf{p}_i^\top \mathbf{p}_j > \kappa$ (in Eq. (4)), then $g_i^j > 0$ (i.e. $g_i^j = 1$ in Eq. (5)), and "$s_{ij} \log(\mathbf{p}_i^\top \mathbf{p}_j)$" (i.e., the first positive term of Eq. (6)) contributes to the loss in Eq. (8). In this case, x_i and x_j should be close to each other in the feature space.
- Still under the condition that $g_i > 0$ (i.e. $g_i = 1$) in Eq. (3), when $a_{ij} = 0$, $\tilde{g}_{ij} = 1$ in Eq. (4)) always holds true, so $g_i^j > 0$ (i.e. $g_i^j = 1$). In this case, "$(1 - s_{ij}) \log(1 - \mathbf{p}_i^\top \mathbf{p}_j)$", namely the second negative term in Eq. (6), contributes to the loss of Eq. (8), where x_i and x_j should be separated from each other.
- Finally, when $g_i < 0$ (i.e. $g_i = 0$), or when $a_{ij} = 0$ and $\mathbf{p}_i^\top \mathbf{p}_j < \kappa$, then $g_j^i < 0$ (i.e. $g_i^j = 0$), and thus in Eq. (6), neither "$s_{ij} \log(\mathbf{p}_i^\top \mathbf{p}_j)$" nor "$(1 - s_{ij}) \log(1 - \mathbf{p}_i^\top \mathbf{p}_j)$" contributes to the loss for Eq. (8).

To provide a more intuitive demonstration, we conducted empirical analysis on the proportion of unlabeled target samples that meet the conditions specified in Eqs. (2) and (3) proposed in the CUNR and PDEP strategies during the model training process. The results are visualized in Fig. 4. Fig. 4(a) shows that as the model trains, the number of unlabeled samples satisfying the condition "$g_{ij} = 1$" gradually increases, indicating an increasing level of predicted confidence in the unlabeled target samples. Moreover, in Fig. 4(b), it can be observed that as the training progresses, the ratio of unlabeled target domain samples satisfying the condition "$a_{ij} = 1$ and $\mathbf{p}_i^\top \mathbf{p}_j > \kappa$" approaches the ratio of that satisfying the condition "$a_{ij} = 1$". This demonstrates that the similarity between the prediction distributions of the unlabeled target samples and the labeled samples from the source or target domains increases gradually during the training process. On the contrary, the number of unlabeled target samples satisfying the condition "$a_{ij} = 0$ and $\mathbf{p}_i^\top \mathbf{p}_j > \kappa$" decreases as the training progresses
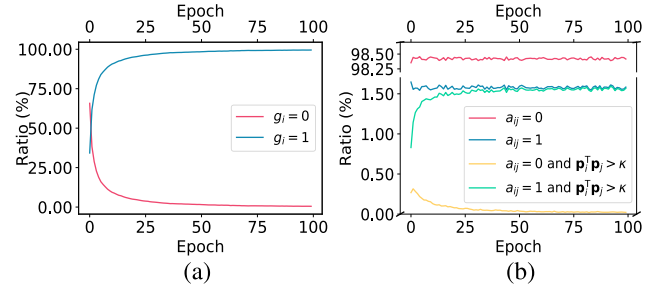

Fig. 4. Empirical analysis on the proportion of unlabeled target samples that meet the conditions proposed in the CUNR and PDEP strategies during the model training process. (a) and (b) are specified in the conditions introduced by Eqs. (2) and (3). For (b), it is important to note that due to the large number of classes in the dataset, only a small fraction of the sample pairs randomly generated in each epoch consist of an unlabeled target sample and a labeled source or target sample with the same label. Therefore, the proportion of unlabeled target domain samples satisfying "$a_{ij} = 1$" will be much smaller compared to the proportion of that satisfying "$a_{ij} = 0$". The experiment is performed with "R → P" on Office-Home using ResNet-34, under the 3-shot setup.

and may eventually approach zero. These findings support our assumption that when two samples, one from the unlabeled target samples and the other from the labeled samples of the source or target domain, do not belong to the same class, their prediction distributions will be dissimilar.

E. Further Optimization of G-ABC Based SSDA

After achieving cross-domain semantic alignment, we apply auxiliary techniques, such as pseudo-label selection, self-training, and consistency training, to further enhance the model training.

1) Pseudo-Label Selection: Due to the scarcity of target labels, overfitting is likely to occur when the $\mathcal{L}^{wdbc}$ loss is applied. To mitigate this issue, we apply a pseudo-label selection strategy to unlabeled target samples and preserve pseudo-labels with high confidence to increase the number of target labels, hence enhancing the semantic label diversity on the target domain. In this work, pseudo-label selection employs the prediction capability of a model to generate artificial hard labels for a subset of unlabeled target samples; hence, a collection of pseudo-labeled target samples, denoted by $\mathcal{D}_{pu}$, can be obtained as follows,

$$\mathcal{D}_{pu} \leftarrow \{(x, \hat{y}) | \hat{y} = \arg\max_k (p(x)[k]), \\ \max(p(x)) > \tau', \forall x \in \mathcal{D}_u\}, \tag{10}$$

where τ' denotes another higher confidence threshold than τ in Eq. (3). Noted that $\mathcal{D}_{pu}$ is only used in the $\mathcal{L}^{wdbc}$ loss, namely Eq. (8).

2) Self-Training: According to [68] and [72], we adopt self-training to boost the model's robustness against the selected samples in $\mathcal{D}_{pu}$. Specifically, we employ the progressive self-training technique described in [72], termed label consistency, in which the model is constrained to generate the same output when the selected images are augmented with slight perturbations. In practice, label consistency can be

implemented through the following loss:

$$\mathcal{L}^{lab} = -\frac{1}{N_{pu}} \sum_{(x_i, y_i) \in \mathcal{D}_{pu}} p_y(y_i) \log(p(\mathbf{Aug}(x_i))), \quad (11)$$

where $N_{pu} = |\mathcal{D}_{pu}|$ indicates the sample size of $\mathcal{D}_{pu}$, $p_y(\cdot)$ represents the function to create a one-hot probability vector for a pseudo-label, and $\mathbf{Aug}(\cdot)$ denotes the function to perturb the input images.

3) Consistency Training: The pseudo-label selection mechanism fails to assign pseudo-labels to all unlabeled target samples. In accordance with [73], we can adequately leverage unlabeled target samples through consistency training, hence increasing the smoothness of the model. According to [73], we can achieve this by preserving the consistency of the model's output distributions between unlabeled target samples and their perturbed counterparts using KullbacLeibler (KL) divergence as follows,

$$\mathcal{L}^{con} = \frac{1}{N_u} \sum_{x_i \in \mathcal{D}_u} \tilde{p}(x_i) \log(\frac{\tilde{p}(x_i)}{p(\mathbf{Aug}(x_i))}), \quad (12)$$

$$\tilde{p}(x_i) = \text{Sharpen}(p(x_i)) = \frac{(p(x_i))^{\frac{1}{T'}}}{\sum_{k=1}^{K} (p(x_i)_k)^{\frac{1}{T'}}}, \quad (13)$$

where k indicates the k-th element of the target distribution vector $p(x_i^u)$ and T' is the temperature factor. It should be noticed that different from [73], we here use a "soft" distribution with a sharpening function $\text{Sharpen}(\cdot)$ proposed in [82] and [83] to sharpen the observed probability distribution $\tilde{p}(x_i)$, thereby driving the model to generate lower-entropy predictions.

4) Training Objectives: The loss function for optimizing the model can be expressed as a combination of the cross-entropy loss $\mathcal{L}^{ce}$ over all labeled samples accessible across domains and additional losses as previously described, i.e.,

$$\mathcal{L}^{Overall} = \mathcal{L}^{ce} + \mathcal{L}^{lab} + \alpha \mathcal{L}^{con} + \beta \mathcal{L}^{abc}, \quad (14)$$

where α and β are scalar hyper-parameters for loss weights. The model is updated using stochastic gradient descent (SGD) for backpropagation.

IV. EXPERIMENTS

A. Datasets

The proposed G-ABC approach is evaluated on three widely used benchmark datasets, including **DomainNet** [32], **Office-Home** [33] and **Office-31** [34]. To conduct a fair comparison, we follow all configurations of adaptation scenarios on different datasets as considered in [23], [24], [25], [26], while each category has one-shot or three-shot samples with labels available in the target domain during training.

DomainNet, consisting of 345 classes and six domains, is a large-scale benchmark dataset designed to evaluate multi-source domain adaption approaches. Following [23], we use a subset of the dataset proposed in [32] as one of our evaluation benchmarks. Similar to MME [23], we only chose 4 domains: Real (**R**), Clipart (**C**), Painting (**P**), and Sketch (**S**) (each comprises 126 categories of images), as other domains and categories may contain samples with excessive noise.

In accordance with [23], [24], [25], and [26], we construct experiments on seven adaptation scenarios employing these four domains.

Office-Home is a notable SSDA benchmark dataset containing numerous challenging adaptation scenarios. There are 65 classes in this collection, and the available domains are Real (**R**), Clipart (**C**), Art (**A**), and Product (**P**). To achieve a fair comparison, we apply 12 adaptation scenarios to this dataset compared to previous SSDA methods, including [23], [24], [25], [26], and [64].

Office-31 consists of 31 object categories organized into three domains: Amazon (**A**), DSLR (**D**), and Webcam (**W**). These categories contain objects frequently seen in offices, such as keyboards, file cabinets, and laptops. Following previous SSDA efforts [23], [24], [26], we select Amazon as the source domain since only Amazon is a large domain with sufficient samples for each class, whereas Webcam and DSLR do not. Therefore, we only consider two adaptation scenarios on this relatively smaller SSDA dataset benchmark, i.e., "W → A" and "D → A".

B. Implementation

To be fair, we adhere to the conventional SSDA task configurations from earlier research [23], [24], [26]. Specifically, we first select AlexNet [89] and ResNet-34 [90] with pre-trained weights on ImageNet [89] as the backbone networks for all our experiments. However, the last layer of each backbone is replaced with a prototypical classifier based on cosine similarity, followed by an unbiased linear neural network that takes normalized features from the feature extractor as inputs. Here, we optimize the entire model using mini-batch stochastic gradient descent (SGD) with momentum. In addition, throughout each iteration, we first train the model under supervision on all labeled samples from both domains to generate representative prototypes of each class, followed by the proposed ADBC and WDBC stages to further enhance the model. Moreover, we use RandAugment [91] and Cutout [92] to generate perturbations for unlabeled target data used in Eq. (8), (9), (11), and (12).

For fair comparisons, the majority of the remaining experimental settings in our proposed method are identical to previous SSDA efforts like [23], [24], and [26]. Similar to [23] and [24], we implement all experiments on the PyTorch[1] platform. Besides, during each iteration, we randomly select four mini-batches from $\mathcal{D}_s$, $\mathcal{D}_l$, $\mathcal{D}_{pu}$, and $\mathcal{D}_u$, with batch sizes of 32, 32, 32, and 64 for AlexNet or 24, 24, 24, and 48 for ResNet-34. In addition, we employ the same learning rate schedule as [93], with the learning rate ξ_t at the t-th iteration set as follows:

$$\xi_t = \frac{\xi_0}{(1 + 0.0001 \times t)^{0.75}}, \quad (15)$$

where ξ_0 represents the initial learning rate. To balance numerous loss terms, we set α and β in Eq. (14) to 0.03 and 25.0. Then, based on [25] and [72], we set the confidence threshold to $\tau = 0.95$ and $\tau' = 0.975$. In addition, we set the similarity

[1]https://pytorch.org/

TABLE I

RESULTS (%) ON DOMAINNET UNDER THE SETTINGS OF 1-SHOT AND 3-SHOT WITH BOTH ALEXNET (ANET) AND RESNET-34(RN-34) BACKBONES. THE TOP BEST METHODS ARE IN **BOLD**. (MEAN ACCURACY AND STANDARD VARIANCE OVER 3 TRIALS)

Net	Method	R→C 1-shot	R→C 3-shot	R→P 1-shot	R→P 3-shot	P→C 1-shot	P→C 3-shot	C→S 1-shot	C→S 3-shot	S→P 1-shot	S→P 3-shot	R→S 1-shot	R→S 3-shot	P→R 1-shot	P→R 3-shot	Mean 1-shot	Mean 3-shot
ANet	S+T [23]	43.3	47.1	42.4	45.0	40.1	44.9	33.6	36.4	35.7	38.4	29.1	33.3	55.8	58.7	40.0	43.4
	DANN [23]	43.3	46.1	41.6	43.8	39.1	41.0	35.9	36.5	36.9	38.9	32.5	33.4	53.5	57.3	40.4	42.4
	MME [23]	48.9	55.6	48.0	49.0	46.7	51.7	36.3	39.4	39.4	43.0	33.3	37.9	56.8	60.7	44.2	48.2
	Meta-MME [62]	-	56.4	-	50.2	-	51.9	-	39.6	-	43.7	-	38.7	-	60.7	-	48.8
	BiAT [66]	54.2	58.6	49.2	50.6	44.0	52.0	37.7	41.9	39.6	42.1	37.2	42.0	56.9	58.8	45.5	49.4
	APE [24]	47.7	54.6	49.0	50.5	46.9	52.1	38.5	42.6	38.5	42.2	33.8	38.7	57.5	61.4	44.6	48.9
	PAC [68]	55.4	61.7	54.6	56.9	47.0	59.8	46.9	52.9	38.6	43.9	38.7	48.2	56.7	59.7	48.3	54.7
	Relaxd-cGAN [31]	-	56.8	-	51.8	-	52.0	-	44.1	-	44.2	-	42.8	-	61.1	-	50.5
	ECACL-T [30]	56.8	62.9	54.8	58.9	**56.3**	60.5	46.6	51.0	**54.6**	51.2	45.4	**48.9**	62.8	67.4	53.4	57.7
	ECACL-P [30]	55.8	62.6	54.0	59.0	56.1	60.5	46.1	50.6	**54.6**	50.3	45.0	48.4	62.3	67.4	52.8	57.6
	S³D [84]	53.5	56.5	51.8	52.2	49.1	53.9	40.1	44.4	44.9	48.7	39.9	39.2	61.7	65.4	48.7	51.5
	CLDA [85]	56.3	59.9	56.0	57.2	50.8	54.6	42.5	47.3	46.8	51.4	38.0	42.7	64.4	67.0	50.7	54.3
	CDAC [25]	56.9	61.4	55.9	57.5	51.6	58.9	44.8	50.7	48.1	51.7	44.1	46.7	63.8	66.8	52.1	56.2
	G-ABC (Ours)	**60.17±0.78**	**63.89±0.41**	**57.44±0.66**	**59.73±0.33**	55.98±0.93	**64.03±0.34**	**48.75±0.50**	**53.42±0.57**	54.11±0.61	**56.36±0.54**	**47.09±0.91**	48.17±0.35	**67.84±0.79**	**70.78±0.36**	**55.92**	**59.68**
RN-34	S+T [23]	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
	DANN [23]	58.2	59.8	61.4	62.8	56.3	59.6	52.8	55.4	57.4	59.9	52.2	54.9	70.3	72.2	58.4	60.7
	MME [23]	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
	UODA [26]	72.7	75.4	70.3	71.5	69.8	73.2	60.5	64.1	66.4	69.4	62.7	64.2	77.3	80.8	68.5	71.2
	Meta-MME [62]	-	73.5	-	70.3	-	72.8	-	62.8	-	68.0	-	63.8	-	79.2	-	70.1
	BiAT [66]	73.0	74.9	68.0	68.8	71.6	74.6	57.9	61.5	63.9	67.5	58.5	62.1	77.0	78.6	67.1	69.7
	APE [24]	70.4	76.6	70.8	72.1	72.9	76.7	56.7	63.1	64.5	66.1	63.0	67.8	76.6	79.4	67.6	71.7
	ELP [86]	72.8	74.9	70.8	72.1	72.0	74.4	59.6	63.3	66.7	69.7	63.3	64.9	77.8	81.0	69.0	71.6
	PAC [68]	74.9	78.6	73.0	74.3	72.6	76.0	65.8	69.6	67.9	69.4	68.7	70.2	76.7	79.3	71.4	73.9
	DECOTA [64]	79.1	80.4	74.9	75.2	76.9	78.7	65.1	68.6	72.0	72.7	69.7	71.9	79.6	81.5	73.9	75.6
	ECACL-T [30]	73.5	76.4	72.8	74.3	72.8	75.9	65.1	65.3	70.3	72.2	64.8	68.6	78.3	79.7	71.1	73.2
	ECACL-P [30]	75.3	79.0	74.1	77.3	75.3	79.4	65.0	70.6	72.1	74.6	68.1	71.6	79.7	82.4	72.8	76.4
	S³D [84]	73.3	75.9	68.9	72.1	73.4	75.1	60.8	64.4	68.2	70.0	65.1	66.7	79.5	80.3	69.9	72.1
	UODAv2 [87]	77.0	79.4	75.4	76.7	75.5	78.3	66.5	70.2	72.1	74.2	70.9	72.1	79.7	82.3	73.9	76.2
	MCL [88]	77.4	79.4	74.6	76.3	75.5	78.8	66.4	70.9	74.0	74.7	70.7	72.3	82.0	83.3	74.4	76.5
	CLDA [85]	76.1	77.7	75.1	75.7	71.0	76.4	63.7	69.7	70.2	73.7	67.1	71.1	80.1	82.9	71.9	75.3
	CDAC [25]	77.4	79.6	74.2	75.1	75.5	79.3	67.6	69.9	71.0	73.4	69.2	72.5	80.4	81.9	73.6	76.0
	G-ABC (Ours)	**80.74±0.41**	**82.07±0.21**	**76.84±0.63**	76.72±0.32	**79.26±0.19**	**81.57±0.41**	**71.95±0.47**	**73.68±0.38**	**75.04±0.54**	**76.27±0.20**	**73.21±0.32**	**74.28±0.09**	**83.42±0.61**	**83.87±0.28**	**77.47**	**78.23**

threshold κ to 0.20. The value of temperatures involved in the construction of the model architecture and the sharpening function in Eq. (13) are set to 0.05 and 0.85, respectively. Due to the distinctness of each dataset and adaptation scenario, we set the total number of training epochs $\mathcal{T}$ to varying values. Note that $\mathcal{T} = 100$ is a common value setting for a variety of application scenarios.

To choose the hyper-parameters, such as α, β, τ and κ, similar to **MME** [23], we selected three labeled examples as the validation set for the target domain and utilized these validation examples to choose the value choice of these hyper-parameters when the validation accuracy was at its highest. Also, during this process, we froze the other hyper-parameters while conducting experiments with a specific one.

1) Class-Wise Similarity Score: To assess the effectiveness of Adaptive Betweenness Clustering, we define a Class-wise Similarity Score (CSS) to measure the average prediction similarity between two classes, where one class c originates from the unlabeled target domain and the other class c' comes from labeled source and target domains. In this way, we use $s(c, c')$ to define the CSS between the class c and c', and more specifically, $s(c, c')$ can be formulated as follows,

$$s(c, c') = \frac{1}{N_u^c} \sum_{i \in I^c} \frac{1}{N_l^{c'} + N_s^{c'}} \sum_{j \in P^{c'} \cup Q^{c'}} \mathbf{p}_i^\top \mathbf{p}_j, \quad (16)$$

where I^c, $P^{c'}$ and $Q^{c'}$ denote the collections of unlabeled target samples of class c, labeled target samples of the class c', and labeled source samples of class c', respectively, each containing instances with sizes of N_u^c, $N_l^{c'}$ and $N_s^{c'}$. In general, the larger the average prediction similarity between classes c and c' is, the greater the CSS $s(c, c')$ is. At this point, a higher CSS score indicates that there is a greater similarity in predictions between unlabeled samples from class c in the target domain and the labeled source or target data from

class c'. This implicitly suggests that these samples are close to each other in the feature space.

C. Comparison With State-of-the-Arts

We compare the classification performance of our proposed G-ABC algorithm to that of previous state-of-the-art SSDA algorithms, including **S+T** [23], **DANN** [23], **MME** [23], **UODA** [26], **Meta-MME** [62], **BiAT** [66], **APE** [24], **ELP** [86], **PAC** [68], **Relaxed-cGAN** [31], **DECOTA** [64], **ECACL-T** [30], **ECACL-P** [30], **S³D** [84], **UODAv2** [87], **MCL** [88], **CLDA** [85] and **CDAC** [25]. Note that **S+T** refers to an approach that trains the adaptation model solely with supervision on labeled samples from both domains, whilst **DANN** refers to the method presented in [20], but additionally applies a standard cross-entropy loss on a few labeled samples in the target domain.

1) On DomainNet: In order to highlight the advantages of the proposed algorithm, we compare our G-ABC strategy to numerous existing alternatives on the DomainNet benchmark. Table I presents the results of this dataset benchmark utilizing 1-shot and 3-shot settings with AlexNet and ResNet-34 as the corresponding backbone networks. As demonstrated, our proposed G-ABC method achieves more average performance gains than all existing approaches in the majority of DomainNet adaptation cases. Specifically, G-ABC improves the prior best-performing SSDA algorithm, i.e., **ECACL-T** and **DECOTA**, by mean accuracy margins of 2.3% and 3.1% while employing AlexNet and ResNet-34 as the backbones, respectively, for all adaptation scenarios under the 1-shot setting. In addition, the proposed method outperforms competing approaches in most of the adaptation scenarios defined on DomainNet with a 3-shot setting by outperforming the best available results (accuracy of 57.7% and 76.5% in **ECACL-T** and **MCL**) by 1.88% and 2.08% on average, when AlexNet and ResNet-34 serve as the backbone networks, respectively. These results demonstrate the effectiveness of our algorithm in dealing with SSDA tasks on DomainNet.

TABLE II

RESULTS (%) ON OFFICE-HOME UNDER THE SETTING OF 3-SHOT WITH BOTH ALEXNET (ANET) AND RESNET-34 (RN-34) BACKBONES. THE TOP BEST METHODS ARE IN **BOLD**. (MEAN ACCURACY AND STANDARD VARIANCE OVER 3 TRIALS)

Net	Method	R→C	R→P	R→A	P→R	P→C	P→A	A→P	A→C	A→R	C→R	C→A	C→P	Mean
ANet	S+T [23]	44.6	66.7	47.7	57.8	44.4	36.1	57.6	38.8	57	54.3	37.5	57.9	50.0
	DANN [23]	47.2	66.7	46.6	58.1	44.4	36.1	57.2	39.8	56.6	54.3	38.6	57.9	50.3
	MME [23]	51.2	73.0	50.3	61.6	47.2	40.7	63.9	43.8	61.4	59.9	44.7	64.7	55.2
	Meta-MME [62]	50.3	-	-	-	48.3	40.3	-	44.5	-	-	44.5	-	-
	BiAT [66]	-	-	-	-	-	-	-	-	-	-	-	-	56.4
	APE [24]	51.9	74.6	51.2	61.6	47.9	42.1	65.5	44.5	60.9	58.1	44.3	64.8	55.6
	PAC [68]	**58.9**	72.4	47.5	61.9	**53.2**	39.6	63.8	**49.9**	60.0	54.5	36.3	64.8	55.2
	CLDA [85]	51.5	74.1	**54.3**	**67**	47.9	47	65.8	47.4	**66.6**	**64.1**	**46.8**	67.5	**58.3**
	CDAC [25]	54.9	75.8	51.8	64.3	51.3	**43.6**	65.1	47.5	63.1	63.0	44.9	65.6	56.8
	G-ABC (Ours)	55.12±0.71	**76.21±0.59**	53.20±0.45	64.59±0.43	50.45±0.70	41.76±0.51	**67.41±0.67**	47.51±0.87	62.07±0.93	63.52±0.97	42.72±0.81	**68.23±0.46**	57.73
RN-34	S+T [23]	55.7	80.8	67.8	73.1	53.8	63.5	73.1	54.0	74.2	68.3	57.6	72.3	66.2
	DANN [23]	57.3	75.5	65.2	69.2	51.8	56.6	68.3	54.7	73.8	67.1	55.1	67.5	63.5
	MME [23]	64.6	85.5	71.3	80.1	64.6	65.5	79	63.6	79.7	76.6	67.2	79.3	73.1
	Meta-MME [62]	65.2	-	-	-	64.5	66.7	-	63.3	-	-	67.5	-	-
	APE [24]	66.4	86.2	73.4	82.0	65.2	66.1	81.1	63.9	80.2	76.8	66.6	79.9	74.0
	Relaxed-cGAN [31]	68.4	85.5	73.8	81.2	68.1	67.9	80.1	64.3	80.1	77.5	66.3	78.3	74.2
	DECOTA [64]	**70.4**	87.7	74.0	82.1	68.0	69.9	81.8	64	80.5	79	68.0	83.2	75.7
	CLDA [85]	66.0	87.6	**76.7**	82.2	63.9	**72.4**	81.4	63.4	**81.3**	80.3	**70.5**	80.9	75.5
	CDAC [25]	67.8	85.6	72.2	81.9	67	67.5	80.3	65.9	80.6	80.2	67.4	81.4	74.2
	G-ABC (Ours)	70.02±0.18	**88.09±0.27**	75.96±0.48	**82.81±0.11**	**69.27±0.53**	70.54±0.42	**83.78±0.31**	**67.24±0.14**	80.37±0.10	**80.18±0.44**	69.22±0.25	**83.89±0.62**	**77.19**

TABLE III

RESULTS (%) ON OFFICE-31 UNDER THE SETTINGS OF 1-SHOT AND 3-SHOT WITH THE ALEXNET BACKBONE. THE TOP BEST METHODS ARE IN **BOLD**. (MEAN ACCURACY AND STANDARD VARIANCE OVER 3 TRIALS)

Method	W→A		D→A		Mean	
	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
S+T [23]	50.4	61.2	50.0	62.4	50.2	61.8
DANN [23]	57.0	64.4	54.5	65.2	55.8	64.8
MME [23]	57.2	67.3	55.8	67.8	56.5	67.6
BiAT [66]	57.9	68.2	54.6	68.5	56.3	68.4
APE [24]	-	67.6	-	69.0	-	68.3
PAC [68]	53.6	65.1	54.7	66.3	54.2	65.7
CLDA [85]	64.6	70.5	62.7	72.5	63.6	71.5
CDAC [25]	63.4	70.1	62.8	70.0	63.1	70.0
G-ABC (Ours)	**67.9±1.26**	**70.97±0.48**	**65.73±1.03**	**73.06±0.35**	**66.81**	**72.02**

2) On Office-Home: To validate the feasibility of the proposed G-ABC algorithm in SSDA, we also compare the results of our method to those of earlier methods on Office-Home. Similar to prior baselines [23], [24], [25], [26], [64], we conduct experiments on this dataset under the 3-shot setting, AlexNet and ResNet-34 as the backbones, and all 12 adaptation scenarios for Office-Home. Table II illustrates the classification accuracy of each adaptation scenario and the average performance of the proposed G-ABC algorithm on Office-Home, respectively. As demonstrated, the proposed method achieves the best average classification performance when utilizing the ResNet-34 backbone, and the accuracy surpasses the best baseline **DECOTA** by significant margins of 1.49%.

3) On Office-31: Aiming at further confirming the efficacy of the proposed G-ABC method, we conduct experiments comparing to the existing state-of-the-art approaches using the smallest benchmark dataset, namely Office-31. In order to retain consistency with prior approaches and to assure a fair comparison, we only use AlexNet as the backbone of this work. Table III shows that our method obtains the highest classification performance in both "W → A" and "D → A" cases, with an average accuracy of 66.81% (+3.21%) under the 1-shot setting and 72.02% (+0.52%) under the 3-shot setting. In other words, our strategy outperforms the existing best-performing baseline, **CLDA**, by significant average accuracy margins in all adaptation scenarios, indicating the improved effectiveness of the proposed method on this dataset.

4) Discussion: It appears that superior performance gains have been observed on the DomainNet dataset compared to Office-Home. This is because Office-Home is a relatively simpler SSDA benchmark dataset. As shown in Table II, it is evident that prior state-of-the-art (SOTA) methods have reached their performance limits on this dataset. Similar to these approaches, this saturation in performance makes it challenging for our method to achieve significant improvements compared to existing SOTA methods. In contrast, DomainNet has a larger domain shift, which poses challenges for domain adaptation methods and offers more room for improvement. As illustrated in Table I, our method demonstrates more notable gains on DomainNet, as it is designed to effectively handle such complex domain shifts.

D. Ablation Study

We conduct extensive experiments to individually confirm the efficacy of each component of our proposed G-ABC approach. Specifically, Table IV shows the main ablation study results, where all experiments are performed in four adaptation scenarios on DomainNet using ResNet-34 as the backbone under the 3-shot setup. Furthermore, Fig. 5 illustrates the evaluation of Adaptive Betweenness Clustering in both ADBC and WDBC, while Fig 6 demonstrates the impact of removing CUNR and PDEP on graph construction and their influence on the model performance.

1) Effectiveness of ADBC and WDBC: To determine the effectiveness of $\mathcal{L}^{adbc}$ and $\mathcal{L}^{wdbc}$ proposed in our method,
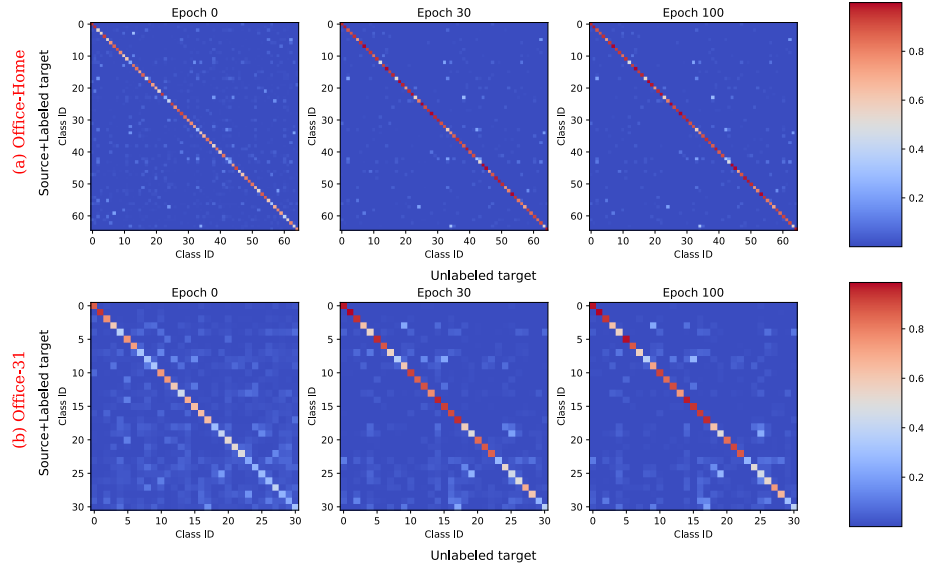
Fig. 5. The evaluation of Adaptive Betweenness Clustering involves analyzing the confusion matrices for each epoch on every dataset. Each element in these matrices is associated with a Class-wise Similarity Score, denoted as $s(c, c')$. This score, defined by Eq. (16) in Sec. IV-B, quantifies the similarity between two classes, c and c'. In this context, class c refers to the class whose samples are the unlabeled target data, while class c' includes classes from both the labeled source and target domains. A higher $s(c, c')$ score in each element suggests a greater similarity in predictions between the unlabeled samples from class c in the target domain and the labeled source or target data from class c'. The experiments are performed with (a): "R → P" on Office-Home using ResNet-34, and (b): "D→A" on Office-31 using AlexNet, respectively, both under the 3-shot setup.

TABLE IV

RESULTS (%) OF ABLATION STUDY ON DOMAINNET UNDER THE SETTING OF 3-SHOT WITH THE RESNET-34 BACKBONE. † DENOTES THAT THE ADAPTATION MODEL IS TRAINED BY $\mathcal{L}^{adbc}$ OR $\mathcal{L}^{wdbc}$ LOSS WITHOUT INCORPORATING PERTURBATIONS INTO THE SAMPLES. ‡ DENOTES THAT THE PSEUDO-LABELED TARGET SAMPLES ARE NOT USED IN THE $\mathcal{L}^{wdbc}$ LOSS. MOREOVER, ♣ REPRESENTS THE REMOVAL OF THE FIRST POSITIVE TERM, NAMELY "$s_{ij} \log(\mathbf{p}_i^\top \mathbf{p}_j)$", FROM EQ. (6) DURING TRAINING. SIMILARLY, ♠ DENOTES THE EXCLUSION OF THE SECOND NEGATIVE TERM, NAMELY "$(1 - s_{ij}) \log(1 - \mathbf{p}_i^\top \mathbf{p}_j)$", FROM EQ. (6) WHEN MODEL TRAINING GOES ON

M-(#)	$\mathcal{L}^{ce}$	$\mathcal{L}^{adbc}$	$\mathcal{L}^{wdbc}$	$\mathcal{L}^{lab}$	$\mathcal{L}^{con}$	R→C	C→S	S→P	R→S	Mean
1	✓	-	-	-	-	60.0	55.0	59.5	50.1	56.2
2	✓	✓	-	-	-	78.0	67.6	71.0	69.5	71.5
3	✓	-	✓	-	-	78.1	68.8	70.8	70.8	72.1
4	✓	✓	✓	-	-	80.3	70.0	73.2	72.5	74.0
5	✓	-	-	✓	-	77.2	70.2	74.0	69.9	72.8
6	✓	-	-	-	✓	72.6	66.1	69.1	65.5	68.4
7	✓	-	-	✓	✓	78.3	71.5	74.3	71.0	73.8
8	✓	✓	-	✓	-	80.5	72.4	75.3	72.7	75.2
9	✓	-	✓	✓	-	81.0	73.1	76.0	73.1	75.8
10	✓	✓	✓	✓	-	81.6	73.2	75.9	73.8	76.1
11	✓	✓	✓	✓	✓	82.2	73.4	76.3	74.3	76.5
12	✓	†	†	✓	✓	81.5	72.1	75.3	72.5	75.4
13	✓	†	†	-	-	78.6	68.5	72.9	70.4	72.6
14	✓	✓	‡	✓	✓	82.0	72.9	75.7	73.2	75.9
15	✓	♣	♣	✓	✓	81.3	72.8	75.0	73.1	75.6
16	✓	♠	♠	✓	✓	79.8	72.0	75.5	72.4	74.9

we first train the model using only labeled samples from both domains, serving as the baseline being depicted in row M-(1) of Table IV. According to Table IV, training the model with both ADBC and WDBC delivers greater classification performance gains than training the model with simply one of both. It can be observed that row M-(4) in Table IV increases the baseline by an average of 17.9%, while the accuracy rates in row M-(2) and row M-(3) can only exceed the baseline by 15.4% and 16.0%, respectively, thereby confirming the validity of the ADBC and WDBC stages. In addition, when row M-(5) of Table IV is considered as another baseline, a similar situation can be observed when contrasting among row M-(8), row M-(9) and row M-(10).

2) Effectiveness of Adaptive Betweenness Clustering: In order to further understand the role of the proposed clustering method, Adaptive Betweenness Clustering (ABC), in ADBC and WDBC, we employed the Class-wise Similarity Score (CSS), which is illustrated in detail in Sec. IV-B. The CSS is used to measure the similarity in predictions between unlabeled target samples and labeled source or target domains. As depicted in the heatmaps of Fig. 5, the CSS scores between the same classes progressively increase, while those between different classes decrease throughout the model training process. This observation indicates that unlabeled target samples have a tendency to cluster together with labeled source and target data from the same class in the feature space, while samples from different classes become more distant from each other. This demonstrates the efficacy of Adaptive Betweenness Clustering in facilitating semantic propagation, thereby raising the performance of ADBC and WDBC. To further validate
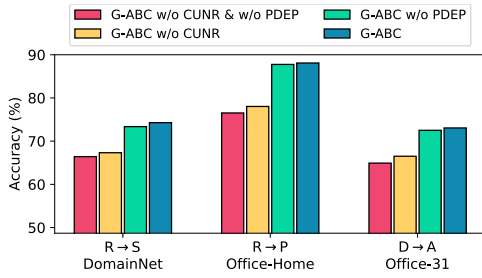
Fig. 6. The impact of removing CUNR and PDEP on performance during graph construction. The experiments are performed on adaptation scenarios of "R→S" on DomainNet, "R→P" on Office-Home, and "D→A" on Office-31, respectively. All of them are conducted under a 3-shot setup, with the first two using ResNet-34 as network backbones, while the latter uses AlexNet.

the effectiveness of the Adaptive Betweenness Clustering loss, we also performed validation experiments on the two terms of the loss presented in Eq. (6), namely the first positive term "$s_{ij} \log(\mathbf{p}_i^\mathsf{T} \mathbf{p}_j)$" and the second negative term "$(1 - s_{ij}) \log(1 - \mathbf{p}_i^\mathsf{T} \mathbf{p}_j)$". According to Table IV, it can be observed that row M-(15), M-(16), and M-(7) correspond to the removal of individual terms or both terms from Eq. (6). By comparing these results with the classification performance of the full model indicated by row M-(11), it can be seen that removing either one or both terms leads to a decrease in model performance, with a more pronounced effect when both terms are removed. This finding indirectly verifies the effectiveness of the adaptive betweenness clustering loss.

3) Effectiveness of CUNR and PDEP: To confirm the impact of CUNR and PDEP during graph construction, we present the results of removing each of them individually, as well as both, on three SSDA benchmarks. As shown in Fig. 6, the comparison between the full model "G-ABC" and its variants "G-ABC w/o CUNR" or "G-ABC w/o PDEP" demonstrates that removing either CUNR or PDEP leads to a significant decline in the model's overall performance. Moreover, when both are eliminated, referred to as "G-ABC w/o CUNR & w/o PDEP", the model's performance reaches its lowest point. This demonstrates that CUNR or PDEP effectively eliminates noisy connectivity while constructing a reliable graph structure, thereby enhancing the model's performance. In particular, CUNR provides a greater performance improvement than PDEP. This is because unlabeled target samples with lower confidence, which would be removed by CUNR during training, not only negatively impact samples of the same class but also affect samples from different classes, resulting in more substantial harm to the model's performance.

4) Effectiveness of Self-Training: Examining the necessity of $\mathcal{L}^{lab}$, we should use experiments in row M-(4) of Table IV in which the model is trained with $\mathcal{L}^{ce}$, $\mathcal{L}^{adbc}$ and $\mathcal{L}^{wdbc}$ as the baseline. As shown in Table IV, the average performance of row M-(10) with additional $\mathcal{L}^{lab}$ loss is 2.1% more than the baseline, indicating the necessity of this component for our proposed G-ABC approach.

5) Effectiveness of Consistency Training: Table IV indicates the effectiveness of consistency training as well. Comparing row M-(10) (or row M-(6)) with row M-(11) (or row M-(7)), it is evident that consistency training for all unlabeled target data is beneficial.

6) Effectiveness of Pseudo-Label Selection: In order to explore the impact of pseudo-label selection, we omit the pseudo-labeled target samples applying to Eq. (8). Table IV demonstrates that in the comparison between rows M-(11) and M-(14), the average accuracy of row M-(14) is 0.6% less than that of row M-(11), revealing that pseudo-label selection is also effective for enhancing the performance of the model.

7) Effectiveness of Sample Perturbation: We propose to introduce perturbations into unlabeled target samples on both ADBC and WDBC. According to Table IV, by comparing row M-(4) and row M-(11) with row M-(13) and row M-(12), respectively, we observe that the performance of the mean accuracy decreases by 1.4% and 1.1%, respectively, indicating that it is necessary to include perturbations in our model training.

E. Further Analysis

We also investigate the hyper-parameter sensitivity to the loss weights α and β, the similarity threshold κ, as well as the hyper-parameter reasonability with respect to the confidence threshold τ'. In addition, we visualize the feature distributions across domains using t-SNE [94].

1) Hyper-parameter Sensitivity to α and β: In Fig. 7(a) and (b), we highlight the influence of α and β. It can be observed that when $\alpha = 0.03$ and $\beta = 25.0$, the trained model achieves the highest performance in image classification. However, the accuracy decreases significantly when they are adjusted further from the optimal value. By introducing g_i^j in Eqs. (8) and (9) to remove graph nodes with noise sample connectivity, a larger proportion of unlabeled target samples do not actually participate in model updates but do contribute to the calculation of $\mathcal{L}_{abc}$, resulting in a smaller scale of $\mathcal{L}_{abc}$. Therefore, setting β to a higher value, i.e., 25.0, is advantageous for balancing the influence of $\mathcal{L}_{abc}$ and other loss items during model updates.

2) Hyper-parameter Sensitivity to τ and κ in CUNR and PDEP: We also conduct experiments to assess the sensitivity of our method to the hyperparameters τ and κ. Fig. 7(c) and (d) illustrate that the classification performance of the model achieves best when τ and κ are set to 0.95 and 0.20, respectively; however, changing either of these parameters, especially on τ to less than 0.90 and κ to greater than 0.50, results in a decline in accuracy. This is due to the fact that lighter CUNR causes a greater number of non-confident unlabeled target samples to be preserved in the graph, whereas higher PDEP causes an excessive number of node removals inside the graph, resulting in unreliable knowledge transfer between target samples.

3) Hyper-parameter Rationality to τ': We conduct additional experiments to prove the validity of setting τ' to 0.975 as opposed to 0.95 by plotting variations in the quantity and accuracy of pseudo-labels for target samples whose expected probability are greater than τ'. As depicted in Fig. 8, when τ' is set to 0.95, $\mathcal{D}_{pu}$ can collect significantly more pseudo-labeled target samples from $\mathcal{D}_u$. However, the large amount of noise contained in the pseudo-labels will also bring challenges to the model. This proves setting τ' to 0.975 is better than 0.95.
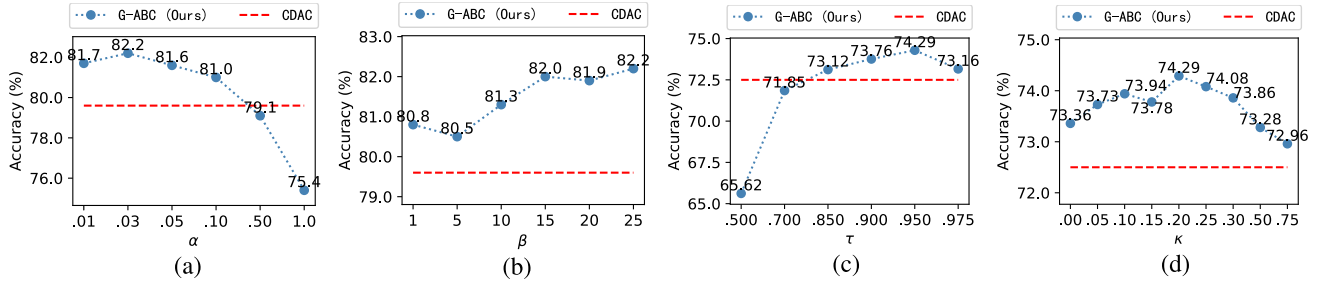
Fig. 7. Sensitivity with respect to hyper-parameters α, β, τ and κ. The experiments are performed on DomainNet under the setting of the 3-shot using the ResNet-34 backbone, where (a) and (b) are with "R→C", and (c) and (d) are with "R→S". "**CDAC**" [25] is the best-performing baseline method in both adaptation scenarios. As illustrated, our method is not highly sensitive to changes in the hyper-parameters α, β, τ and κ. This is because, over a wide range, our G-ABC approach outperforms the baseline method "**CDAC**" significantly for all four hyper-parameters.
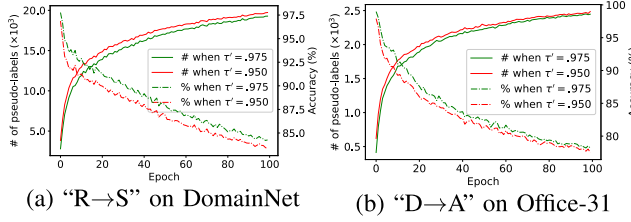

Fig. 8. The evolution of the numbers (#) and the accuracy (%) of the pseudo-labels over epochs while varying the confidence threshold τ'. The experiments are performed with (a): "R→S" on DomainNet using ResNet-34, and (b): "D→A" on Office-31 using AlexNet, both under the 3-shot setup.
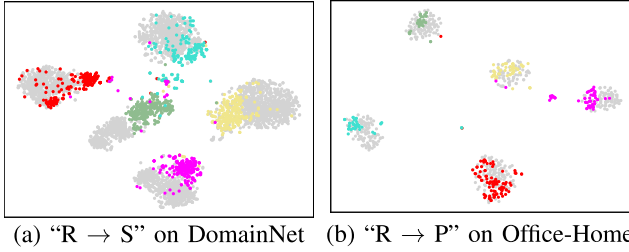

Fig. 9. Feature visualization using t-SNE. We randomly choose five classes with (a): "R → S" on DomainNet, and (b): "R → P" on Office-Home, respectively, with the ResNet-34 backbone and the 3-shot setup. Herein, data points in grey represent source samples, while brightly colored examples are from the target domain. The red, lightblue, purple, green, and yellow represent the categories of "Axe", "Bird", "Fence", "Shoe", and "Truck" on DomainNet, while these colors correspond to the categories of "Batteries", "Calendar", "Flowers", "Glasses", and "Monitor" within the Office-Home dataset.

4) Feature Visualization: Using t-SNE for feature visualization, we present the feature distributions obtained by the proposed method for both domains in Fig. 9. It can be observed that in the feature space, the learned features from different domains that belong to the same class are mapped nearby and clustered together, whereas those from distinct categories are significantly separated. This implies that the model trained using the proposed G-ABC approach is capable of producing domain-invariant and discriminative target features, thus contributing to the improved performance of the SSDA task.

V. CONCLUSION

This paper presents a novel SSDA method named Graph-based Adaptive Betweenness Clustering for achieving categorical domain alignment. It facilitates cross-domain

semantic alignment by enforcing semantic transfer from labeled source and target data to unlabeled target samples. In this approach, a heterogeneous graph is first constructed to represent pairwise relationships between labeled examples from both domains and unlabeled target samples. Then, two strategies including Confidence Uncertainty based Node Removal and Prediction Dissimilarity based Edge Pruning are proposed to refine the connectivity in the graph to alleviate the influence of noisy edges. Provided with the refined graph, we present adaptive betweenness clustering to accomplish semantic transfer across domains with semantic propagation from labeled source or target examples to unlabeled samples on the target domain. Extensive experimental results as well as comprehensive analysis performed well on three benchmark datasets demonstrate the superiority of our proposed method, achieving new state-of-the-art results.

REFERENCES

[1] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *Proc. Int. Interdiscipl. PhD Workshop (IIPhDW)*, May 2018, pp. 117–122.

[2] W. Boulila, M. Sellami, M. Driss, M. Al-Sarem, M. Safaei, and F. A. Ghaleb, "RS-DCNN: A novel distributed convolutional-neural-networks based-approach for big remote-sensing image classification," *Comput. Electron. Agricult.*, vol. 182, Mar. 2021, Art. no. 106014.

[3] L. Li, J. Wang, J. Li, Q. Ma, and J. Wei, "Relation classification via keyword-attentive sentence mechanism and synthetic stimulation loss," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 9, pp. 1392–1404, Sep. 2019.

[4] S. Wu, G. Deng, J. Li, R. Li, Z. Yu, and H.-S. Wong, "Enhancing TripleGAN for semi-supervised conditional instance synthesis and classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10091–10100.

[5] J. Li, G. Li, F. Liu, and Y. Yu, "Neighborhood collective estimation for noisy label identification and correction," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 128–145.

[6] B. Liu, J. Jiao, and Q. Ye, "Harmonic feature activation for few-shot semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 3142–3153, 2021.

[7] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Appl. Soft Comput.*, vol. 70, pp. 41–65, Sep. 2018.

[8] J. Yang et al., "An adversarial perturbation oriented domain adaptation approach for semantic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12613–12620.

[9] X. Xiong, S. Li, and G. Li, "Unpaired image-to-image translation based domain adaptation for polyp segmentation," in *Proc. IEEE 20th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2023, pp. 1–5.

[10] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[11] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5001–5009.

[12] G. Zhao, G. Li, R. Xu, and L. Lin, "Collaborative training between region proposal localization and classification for domain adaptive object detection," in *Computer Vision—ECCV 2020*. Glasgow, U.K.: Springer, Aug. 2020, pp. 86–102.

[13] P. Yan, Z. Wu, M. Liu, K. Zeng, L. Lin, and G. Li, "Unsupervised domain adaptive salient object detection through uncertainty-aware pseudo-label learning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 3, pp. 3000–3008.

[14] H.-Y. Zhou et al., "Generalized radiograph representation learning via cross-supervision between images and free-text radiology reports," *Nature Biomed. Eng.*, vol. 4, no. 1, pp. 32–40, 2023.

[15] H.-Y. Zhou et al., "A transformer-based representation-learning model with unified processing of multimodal input for clinical diagnostics," *Nature Biomed. Eng.*, vol. 7, pp. 743–755, Jun. 2023.

[16] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning for differing training and test distributions," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 81–88.

[17] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 692–699.

[18] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.

[19] S. Herath, M. Harandi, and F. Porikli, "Learning an invariant Hilbert space for domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3845–3854.

[20] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2017.

[21] A. Kumagai and T. Iwata, "Unsupervised domain adaptation by matching distributions based on the maximum mean discrepancy via unilateral transformations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4106–4113.

[22] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, pp. 1–46, Oct. 2020.

[23] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8050–8058.

[24] T. Kim and C. Kim, "Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 591–607.

[25] J. Li, G. Li, Y. Shi, and Y. Yu, "Cross-domain adaptive clustering for semi-supervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2505–2514.

[26] C. Qin, L. Wang, Q. Ma, Y. Yin, H. Wang, and Y. Fu, "Contradictory structure learning for semi-supervised domain adaptation," in *Proc. SIAM Int. Conf. Data Mining (SDM)*. Philadelphia, PA, USA: SIAM, 2021, pp. 576–584.

[27] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[28] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2272–2281.

[29] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.

[30] K. Li, C. Liu, H. Zhao, Y. Zhang, and Y. Fu, "ECACL: A holistic framework for semi-supervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8578–8587.

[31] Q. Luo et al., "Relaxed conditional image transfer for semi-supervised domain adaptation," 2021, *arXiv:2101.01400*.

[32] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1406–1415.

[33] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5018–5027.

[34] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2010, pp. 213–226.

[35] D. Huang, J. Li, W. Chen, J. Huang, Z. Chai, and G. Li, "Divide and adapt: Active domain adaptation via customized learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7651–7660.

[36] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1426–1435.

[37] J. Zhuang, Z. Chen, P. Wei, G. Li, and L. Lin, "Discovering implicit classes achieves open set domain adaptation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.

[38] Z. Zhang et al., "Divide and contrast: Source-free domain adaptation via adaptive contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 5137–5149.

[39] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 443–450.

[40] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*.

[41] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[42] J.-C. Su, Y.-H. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker, "Active adversarial domain adaptation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 739–748.

[43] M. Chen, S. Zhao, H. Liu, and D. Cai, "Adversarial-learned loss for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 3521–3528.

[44] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 135–150.

[45] M. Awais et al., "Adversarial robustness for unsupervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8568–8577.

[46] J. Yang et al., "Exploring robustness of unsupervised domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9194–9203.

[47] C. Chen et al., "Progressive feature alignment for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 627–636.

[48] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei, "Transferrable prototypical networks for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2239–2247.

[49] L. Zhong, Z. Fang, F. Liu, J. Lu, B. Yuan, and G. Zhang, "How does the combined risk affect the performance of unsupervised domain adaptation approaches?" in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 12, pp. 11079–11087.

[50] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6673–6683.

[51] S. Li et al., "Semantic concentration for domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9102–9111.

[52] W. Xiao, Z. Ding, and H. Liu, "Implicit semantic response alignment for partial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 13820–13833.

[53] Y. Zhang, G. Song, L. Du, S. Yang, and Y. Jin, "DANE: Domain adaptive network embedding," 2019, *arXiv:1906.00684*.

[54] M. Wu, S. Pan, C. Zhou, X. Chang, and X. Zhu, "Unsupervised domain adaptive graph convolutional networks," in *Proc. Web Conf.*, Apr. 2020, pp. 1457–1467.

[55] S. Yang, G. Song, Y. Jin, and L. Du, "Domain adaptive classification on heterogeneous information networks," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1410–1416.

[56] M. Pilancı and E. Vural, "Domain adaptation on graphs via frequency analysis," in *Proc. 27th Signal Process. Commun. Appl. Conf. (SIU)*, Apr. 2019, pp. 1–4.

[57] M. Pilanci and E. Vural, "Domain adaptation on graphs by learning aligned graph bases," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 587–600, Feb. 2022.

[58] Z. Ding, S. Li, M. Shao, and Y. Fu, "Graph adaptive knowledge transfer for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 37–52.

[59] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16, 2003, pp. 1–8.

[60] C. H. Nguyen and H. Mamitsuka, "Discriminative graph embedding for label propagation," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1395–1405, Sep. 2011.

[61] L. Wang, Z. Ding, and Y. Fu, "Adaptive graph guided embedding for multi-label annotation," in *Proc. IJCAI*, Jul. 2018, pp. 1–7.

[62] D. Li and T. Hospedales, "Online meta-learning for multi-source and semi-supervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 382–403.

[63] S. Chen, M. Harandi, X. Jin, and X. Yang, "Semi-supervised domain adaptation via asymmetric joint distribution matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5708–5722, Dec. 2021.

[64] L. Yang et al., "Deep co-training with task decomposition for semi-supervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8906–8916.

[65] Z. Fang, J. Lu, F. Liu, and G. Zhang, "Semi-supervised heterogeneous domain adaptation: Theory and algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 1087–1105, Jan. 2023.

[66] P. Jiang, A. Wu, Y. Han, Y. Shao, M. Qi, and B. Li, "Bidirectional adversarial training for semi-supervised domain adaptation," in *Proc. 29th Int. Joint Conf. Artif. Intell., 17th Pacific Rim Int. Conf. Artif. Intell.*, Jul. 2020, pp. 934–940.

[67] B. Li et al., "Learning invariant representations and risks for semi-supervised domain adaptation," 2020, *arXiv:2010.04647*.

[68] S. Mishra, K. Saenko, and V. Saligrama, "Surprisingly simple semi-supervised domain adaptation with pretraining and consistency," 2021, *arXiv:2101.12727*.

[69] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui, "Generalized source-free domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8958–8967.

[70] K. Saito, D. Kim, S. Sclaroff, and K. Saenko, "Universal domain adaptation through self supervision," 2020, *arXiv:2002.07953*.

[71] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[72] A. Kurakin et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 596–608.

[73] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," 2019, *arXiv:1904.12848*.

[74] J. Li, S. Wu, C. Liu, Z. Yu, and H.-S. Wong, "Semi-supervised deep coupled ensemble learning with classification landmark exploration," *IEEE Trans. Image Process.*, vol. 29, pp. 538–550, 2020.

[75] S. Wu, J. Li, C. Liu, Z. Yu, and H.-S. Wong, "Mutual learning of complementary networks via residual correction for improving semi-supervised classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6500–6509.

[76] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1381–1391, May 2019.

[77] Y. Bai, C. Wang, Y. Lou, J. Liu, and L.-Y. Duan, "Hierarchical connectivity-centered clustering for unsupervised domain adaptation on person re-identification," *IEEE Trans. Image Process.*, vol. 30, pp. 6715–6729, 2021.

[78] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 1, pp. 55–67, Jan. 2008.

[79] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang, "Label propagation via teaching-to-learn and learning-to-teach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1452–1465, Jun. 2017.

[80] Y. Liu, J. Lee, M. Park, S. Kim, and Y. Yang, "Transductive propagation network for few-shot learning," *CoRR*, vol. abs/1805.10002, pp. 1–14, May 2018.

[81] S.-A. Rebuffi, S. Ehrhardt, K. Han, A. Vedaldi, and A. Zisserman, "LSD-C: Linearly separable deep clusters," 2020, *arXiv:2006.10039*.

[82] J. Li, R. Socher, and S. C. H. Hoi, "DivideMix: Learning with noisy labels as semi-supervised learning," 2020, *arXiv:2002.07394*.

[83] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "MixMatch: A holistic approach to semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5049–5059.

[84] J. Yoon, D. Kang, and M. Cho, "Semi-supervised domain adaptation via sample-to-sample self-distillation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1978–1987.

[85] A. Singh, "CLDA: Contrastive learning for semi-supervised domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 5089–5101.

[86] Z. Huang et al., "Effective label propagation for discriminative semi-supervised domain adaptation," *CoRR*, vol. abs/2012.02621, Dec. 2020.

[87] C. Qin, L. Wang, Q. Ma, Y. Yin, H. Wang, and Y. Fu, "Semi-supervised domain adaptive structure learning," *IEEE Trans. Image Process.*, vol. 31, pp. 7179–7190, 2022.

[88] Z. Yan, Z. Wu, G. Li, Y. Qin, X. Han, and S. Cui, "Multi-level consistency learning for semi-supervised domain adaptation," in *Proc. 31st Int. Joint Conf. Artif. Intell. (IJCAI)*, L. D. Raedt, Ed. International Joint Conferences on Artificial Intelligence Organization, Jul. 2022, pp. 1530–1536.

[89] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1–9.

[90] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[91] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 3008–3017.

[92] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.

[93] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1180–1189.

[94] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Jichang Li received the M.Eng. degree from the School of Computer Science and Technology, South China University of Technology, in 2020. He is currently pursuing the Ph.D. degree with the Department of Computer Science, The University of Hong Kong. His current research interests include computer vision and deep learning.

Guanbin Li (Member, IEEE) received the Ph.D. degree from The University of Hong Kong in 2016. He is currently an Associate Professor with the School of Computer Science and Engineering, Sun Yat-sen University. He has authored and coauthored more than 100 papers in top-tier academic journals and conferences. His current research interests include computer vision, image processing, and deep learning. He was a recipient of the ICCV 2019 Best Paper Nomination Award. He serves as the Area Chair for the Conference of VISAPP. He has been serving as a Reviewer for numerous academic journals and conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IJCV, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, TCyb, CVPR, ICCV, ECCV, and NeurIPS.

Yizhou Yu (Fellow, IEEE) received the Ph.D. degree from the University of California at Berkeley in 2000. He is a Chair Professor with The University of Hong Kong and was a Faculty Member with the University of Illinois at Urbana–Champaign for 12 years. His current research interests include AI foundation models, AI-based multimedia content generation, AI for medicine, and computer vision. He has also served on the program committee of many leading international conferences, including CVPR, ICCV, and SIGGRAPH. He was a recipient of the 2002 U.S. National Science Foundation CAREER Award and the ACCV 2018 Best Application Paper Award. He has served on the Editorial Board of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS.