HG-SFDA: HyperGraph Learning Meets Source-Free Unsupervised Domain Adaptation

Jinkun Jiang, Qingxuan Lv, Yuezun Li, *Member, IEEE*, Yong Du, *Senior Member, IEEE*, Junyu Dong, *Member, IEEE*, Sheng Chen, *Life Fellow, IEEE*, and Hui Yu, *Senior Member, IEEE*

***Abstract*—Source-Free unsupervised Domain Adaptation (SFDA) aims to classify target samples by only accessing a pre-trained source model and unlabelled target samples. Since no source data is available, transferring the knowledge from the source domain to the target domain is challenging. Existing methods normally exploit the pair-wise relation among target samples and attempt to discover their correlations by clustering these samples based on semantic features. The drawbacks of these methods include: 1) the pair-wise relation is limited to exposing the underlying correlations of two more samples, hindering the exploration of the structural information embedded in the target domain; and 2) the clustering process only relies on the semantic feature, while overlooking the critical effect of domain shift, *i.e.*, the distribution differences between the source and target domains. To address these issues, we propose a new SFDA method that exploits the high-order neighborhood relation and explicitly takes the domain shift effect into account. Specifically, we formulate the SFDA as a hypergraph learning problem and construct hyperedges to explore the deep structural and context information among multiple samples. Moreover, we integrate a self-loop strategy into the constructed hypergraph to elegantly introduce the domain uncertainty of each sample. By clustering these samples based on hyperedges, both the semantic feature and domain shift effects are considered. We then describe an adaptive relation-based objective to tune the model with soft attention levels for all samples. Extensive experiments are conducted on Office-31, Office-Home, VisDA, DomainNet-126 and PointDA-10 datasets. The results demonstrate the superiority of our method over state-of-the-art counterparts. Our code is avaliable at https://github.com/OUC-POVA/HG-SFDA**

Received 12 January 2025; revised 28 April 2025 and 10 August 2025; accepted 27 October 2025. Date of publication 17 November 2025; date of current version 21 November 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2022ZD0117201; in part by the Key Research and Development Program of Shandong Province, China, under Grant 2024ZLGX06; and in part by Sanya Science and Technology Special Fund under Grant 2022KJCX92. The work of Yuezun Li was supported in part by the National Natural Science Foundation of China under Grant 62402464, in part by Shandong Natural Science Foundation under Grant ZR2024QF035, and in part by China Postdoctoral Science Foundation under Grant 2021TQ0314 and Grant 2021M703036. The associate editor coordinating the review of this article and approving it for publication was Dr. Tongliang Liu. *(Corresponding authors: Yuezun Li; Junyu Dong.)*

Jinkun Jiang, Qingxuan Lv, Yuezun Li, Yong Du, and Junyu Dong are with the School of Computer Science and Technology, Ocean University of China, Qingdao 266100, China (e-mail: jiangjinkun@stu.ouc.edu.cn; lvqingxuan@stu.ouc.edu.cn; liyuezun@ouc.edu.cn; csyongdu@ouc.edu.cn; dongjunyu@ouc.edu.cn).

Sheng Chen is with the School of Electronics and Computer Science, University of Southampton, SO17 1BJ Southampton, U.K. (e-mail: sqc@ecs.soton.ac.uk).

Hui Yu is with the School of Psychology and Neuroscience, University of Glasgow, G12 8QQ Glasgow, U.K. (e-mail: Hui.Yu@glasgow.ac.uk).

Digital Object Identifier 10.1109/TIP.2025.3631461

***Index Terms*—Source-free domain adaptation, unsupervised learning, hypergraph learning.**

I. Introduction

DEEP learning methods for vision tasks (*e.g.*, image classification), trained with a large number of training samples, can generalize well on the testing set with a similar data distribution [1], [2], [3], [4]. However, their performance notably degrades when applied to an unseen data distribution due to the phenomenon of domain shift, *i.e.*, differences in the data distribution between the source and target domains. Unsupervised Domain Adaptation (UDA) is a typical solution to this issue by transferring knowledge from the fully labeled source domain to the unlabeled target domain [5], [6], [7], [8], [9], [10].[1] However, traditional UDA methods require to access to the data of the source domain during training, which may be infeasible in real-world applications due to data privacy or intellectual property concerns [11], [12].

One emerging research direction, Source-Free unsupervised Domain Adaptation (SFDA), has recently been explored to address the above concerns and attracted increasing attention [13], [14], [15], [16], [17], [18], [19], [20], [21]. The setting of SFDA is stricter and more challenging than UDA because the source data is unavailable, and only a pre-trained source model and target data are available. Under this setting, obtaining more domain knowledge depends on how to effectively exploit the underlying relation of these target samples. One typical solution for this involves using the spirit of neighborhood clustering so that domain adaptation can be accomplished by exploring the neighborhood relation of target samples in feature space, *e.g.*, G-SFDA [13], NRC [22], AaD [23], SF(DA)^2 [20]. The intuition behind these methods is that similar target samples likely belong to the same semantic class and vice versa, and the sample relations in clusters can help the model to learn domain invariant knowledge. Despite promising results shown by these methods, they still suffer the following limitations: **1) Only pair-wise relations are considered in clustering.** Since no prior knowledge about the source data is available, only considering the pair-wise sample relations may not adequately capture the underlying relations hidden in the target domain, as illustrated in Fig. 1 (Left). This limitation results in failing to capture deeper structural

[1]General UDA methods are typically explored on standard image classification tasks, which fall under the scope of image processing.

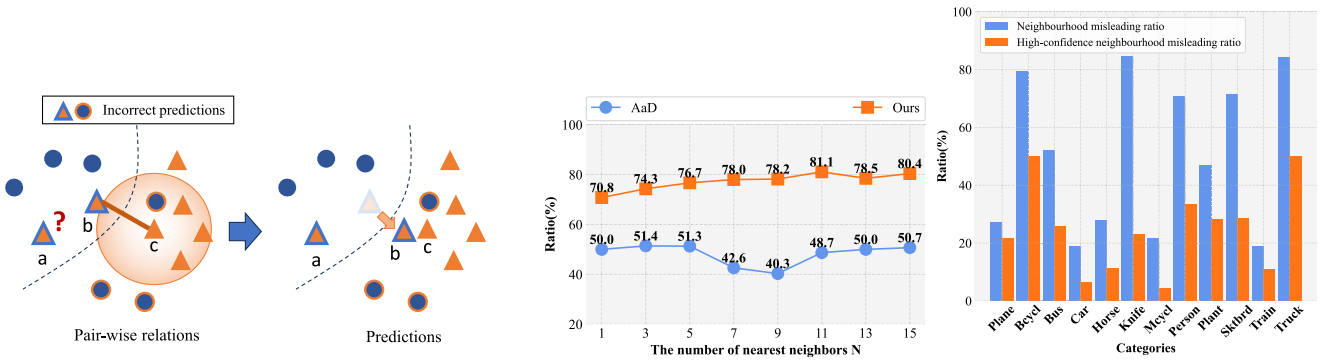

Fig. 1. (Left) The pair-wise relation for sample c only considers the affinity to sample b in its neighborhood, but it fails to consider the high-order relation between sample a and c, resulting in inaccurate predictions. (Middle) Comparison of the pair-wise relation based method AaD [23] and our method on the accuracy of target samples' nearest neighbors having the correct predicted labels. The accuracy of nearest neighbors is computed as follows: For each target sample, we calculate the ratio of its correctly classified neighbors with the same label. A higher ratio of correct neighbours means that truly similar neighbour samples can be found more accurately, thereby resulting in better pseudo labels for adaptation. (Right) "Neighborhood misleading ratio" and "High-confidence" denote the mismatch between the predicted label and ground truth label of neighbors, and neighbors with high prediction confidence [24]. Without involving the domain shift in optimization, the misleading ratio fluctuates among different categories, indicating that the domain shift is not generally solved. These figures are validated on the VisDA dataset [25].

information and makes the model easily distracted by outliers (*i.e.*, the sample wrongly predicted), which directly hinders the model from learning domain invariant knowledge, as seen in Fig. 1 (Middle). **2) Domain shift is not explicitly involved in clustering.** Existing works focus on seeking the semantic relation of target samples and assume that the domain shift can be reduced implicitly by only considering the semantic relation. This strategy cannot effectively address the domain shift problem, as it is not explicitly involved in the clustering process, thereby hindering clustering effectiveness, as shown in Fig. 1 (Right).

In this paper, we present a new clustering-based method called HG-SFDA to overcome the above limitations. Fig. 2 shows the overview of the proposed HG-SFDA method. Differing from the existing methods that only consider pair-wise relations, *e.g.*, NRC++ [26], AaD [23], SF(DA)^2 [20], our method explores the high-order neighborhood relations among multiple target samples while considering the domain shift phenomenon explicitly. Since high-order neighborhood relations can encapsulate the complex interplay among two or more target samples and little prior knowledge is used in the SFDA setting, this high-order neighborhood is the most valuable and handy resource that can aggregate more deep structural information and context. To capture the high-order relation, we propose a hypergraph learning method, which formulates the target samples as graph nodes and conducts hyperedges over the graph (see Subsection III-B).

To attach importance to the domain shift effect, we propose a novel self-loop strategy on the constructed hypergraph. This strategy involves creating self-loops on nodes to represent the domain uncertainty of corresponding samples. Domain uncertainty indicates the likelihood of samples belonging to the source or target domain. By involving the self-loops in clustering, the samples with high domain uncertainty are drawn more attention, which leads to a comprehensive consideration of both semantic relations and domain shift re-calibration, ultimately improving the effectiveness of clusters (see Subsection III-C).

Furthermore, we describe a new adaptive learning scheme that can be incorporated into mainstream objective functions. In particular, we assign "soft" attention levels for different samples, *i.e.*, paying more attention to hard samples and vice versus. For example, the samples having large differences in the same cluster should be concerned more than others. This also holds for samples from different clusters. Therefore, we dynamically assign different weights to samples according to the semantic distance between the target sample and its nearest neighbors (see Subsection III-D).

Extensive experiments are conducted on several image datasets (Office-31 [27], Office-Home [28], VisDA [25], DomainNet-126 [29]) and a 3D point cloud dataset (PointDA-10 [30]), to compare our method to the recent counterparts with the best results currently available. The results obtained show the superiority of our method on the SFDA problem.

The contributions of this paper can be recapped as follows. **1)** Different from the existing pair-wise relation-based methods, *e.g.*, NRC [22], AaD [23], SF(DA)^2 [20], we formulate SFDA as a hypergraph learning problem and explore the high-order neighborhood relations to excavate the underlying structural information. **2)** With the constructed hypergraph, we design a novel self-loop strategy to elegantly involve the domain shift into optimization. **3)** We describe an adaptive learning scheme to enhance the mainstream objectives by considering different attention levels.

II. Related Work

A. Unsupervised Domain Adaptation

UDA methods transfer the knowledge from the fully labeled source domain to the unlabeled target domain. Generally, the UDA methods can be divided into several categories, ranging from minimizing distributional differences [31], [32], [33], [34], [35], [36], [37], [38], adversarial training [39], [40], [41], [42] to clustering [8], [43], [44], [45], [46], [47], [48]. The distributional differences are usually minimized using maximum mean discrepancy [49] and contrastive domain discrepancy [50]. In addition to minimizing distributional differences,

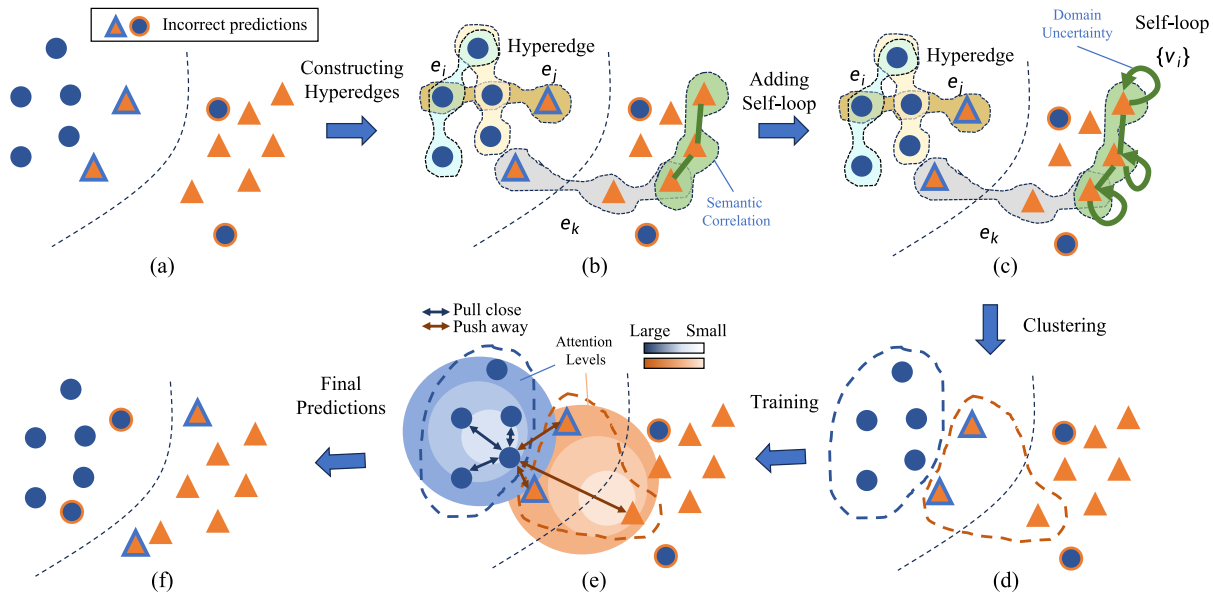

Fig. 2. Overview of the proposed method HG-SFDA. (a) Initial results. (b) The hyperedges are constructed on the target domain to capture complicated high-order neighborhood relations among multiple samples. (c) A self-loop strategy is proposed to consider the domain shift effect. (d) Clustering results by considering both hyperedges and self-loops. (e) After clustering, the model is trained using the objective in AaD [23] with the proposed adaptive learning scheme. (f) Final results.

domain adaptation can also be accomplished through domain adversarial methods. DANN [40] and VRADA [51] effectively confuse domain classifiers by countering their gradients using a gradient inversion layer. More recently, Clustering-based methods have gained popularity, which can discover the correlation of samples between source and target domain and extract the domain invariant knowledge. For example, CoDT [43] captures pseudo-labels to guide feature clustering by exploiting the complementary domain-shared features and target-specific features. CAT [44] achieves domain alignment and class-conditional alignment via a discriminative clustering loss and a clustering-based alignment loss.

B. Source-Free Unsupervised Domain Adaptation

SFDA is a more challenging category of UDA, which requires accomplishing domain adaptation only with a pre-trained source model and unlabeled target data [12], [17], [21], [24], [52], [53], [54], [55], [56], [57], [58], [59]. Most of the SFDA methods [56], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69] focus on learning domain invariant representations to facilitate cross-domain adaptation. Specifically, the work [63] introduces an image generator to update target images to resemble source images, and the study [62] employs a GAN-based generator to simulate source data. ASM [21] proposes an adversarial generation method for source styles based on a style generator. DIPE [65], VMP [64], and CAF [47] explore the transferability of source model parameters to generate better domain-invariant representations. In recent years, following the clustering spirit in general UDA, many clustering-based strategies are proposed to solve the SFDA problem [13], [16], [18], [20], [22], [23], [26], [70], [71], [72], [73]. For example, SHOT [15] and SHOT++ [72] generate single-feature prototypes by weighted k-means clustering and refine pseudo-labels based on the prototypes. BMD [16] generates balanced feature prototypes through inter-class balanced sampling and improves pseudo-label accuracy by using an intra-class multicentre clustering strategy. To alleviate the problem of prototype noise in prototype-based clustering methods, NRC++ [26] introduces a local structural clustering strategy to encourage prediction consistency among nearest neighbors with high affinity. AaD [23] further exploits underlying information from different samples and achieves SFDA by pulling together the predictions of nearest-neighbour features while dispersing predictions of dissimilar features. SF(DA)^2 [20] designs a spectral neighborhood clustering loss based on AaD [23] to identify partitions in the prediction space by spectral clustering. However, these methods only focused on pair-wise relations between samples and overlooked the domain shift issue in optimization, failing to extract the underlying structural information of target data.

C. Hypergraph Learning Methods

Hypergraph learning models high-order relationships by constructing hyperedges that connect more than two nodes, thereby capturing richer and more informative features, *e.g.*, [74], [75], [76], [77]. Early work formulated hypergraph embedding as a spectral optimisation problem [78]. Subsequently, hypergraph neural networks (HGNN) have introduced spectral convolution into hypergraphs, effectively exploiting high-order structures [75]. Based on this, further developments have led to Dynamic Hypergraph Neural Network (DHGNN) [79], which can dynamically update hypergraph structures. Recently, HGAT [74] combines a self-attention mechanism with hypergraphs to automatically learn the importance of nodes and hyperedges. Despite these advances, most existing works have been applied primarily to basic tasks such as social

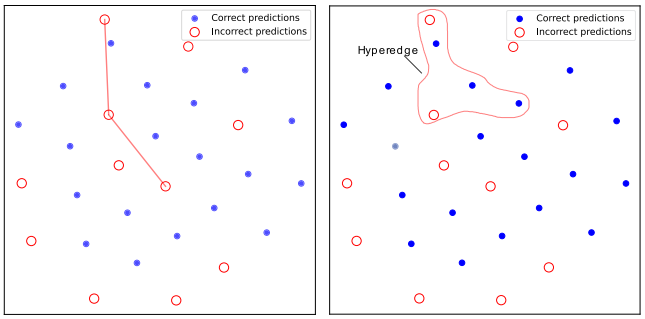

Fig. 3. Illustration of the result difference between the pairwise method AaD [23] (Left) and our method HG-SFDA (Right) using the VisDA dataset. By constructing high-order hyperedges to connect multiple samples, we can utilize more correctly classified neighborhood information to assist in correcting the prediction results of target samples.

network analysis [80], [81] and drug-target interactions [82], [83], leaving the potential of hypergraph learning in computer vision largely underexplored. In this paper, we reveal that the main bottleneck in the SFDA problem can be effectively addressed by leveraging high-order relationships among target domain samples.

III. Methodology

A. Problem Setting of SFDA

Denote the source domain as $\mathcal{D}_s = \left\{\left(x_i^s, y_i^s\right)\right\}_{i=1}^{N_s}$, where x_i^s and y_i^s represent a source sample and its corresponding label, and N_s is the number of samples. Denote the target domain as $\mathcal{D}_t = \left\{x_i^t\right\}_{i=1}^{N_t}$ containing N_t unlabeled samples. Let the model network be $\mathcal{O}$, which consists of a feature extractor f and a classifier g. Given an input sample x, the output of the feature extractor is denoted as $z = f(x)$, and the prediction vector (after softmax) of the classifier is denoted as $p = g(z)$. The objective of SFDA is to transfer the knowledge from the source domain to the target domain, by adapting a pre-trained source model $\mathcal{O}$ to the target domain $\mathcal{D}_t$, without accessing to the source domain data $\mathcal{D}_s$. Following the previous works [17], [20], [22], [23], [59], [84], we explore this task mainly on the close-set setting, *i.e.*, the source domain and target domain share the same label set.[2]

B. Exploring High-Order Neighborhood Relation

1) Motivation: To obtain the pseudo-labels for target samples, existing neighborhood clustering-based methods [13], [20], [23], [26] always explore pairwise relationships, encouraging the prediction of target samples similar to their nearest neighbors. However, as shown in Fig. 3 (Left), the pairwise nearest neighbors of a misclassified target sample are all misclassified. This indicates that the pairwise relation is limited to see more context, failing to exploit more complex relationships to adjust the prediction of target samples. To address this limitation, we formulate SFDA as a hypergraph learning problem to capture the high-order neighborhood relation. Fig. 3 (Right) illustrates the advantage of our method, where the misclassified target sample can be corrected by leveraging high-order neighborhood relation that captures more correctly classified neighbors.

2) Hypergraph Meets SFDA: Hypergraph is a graph structure with special edges that cover two more nodes. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ represent a hypergraph, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of nodes, $\mathcal{E} = \{e_1, \ldots, e_m\}$ is the set of hyperedges, and $\mathcal{W}$ is the affinity set corresponding to hyperedges. Each hyperedge $e \in \mathcal{E}$ consists of $k (k > 2)$ nodes, and the degree of each hyperedge is $k = \sum_{v \in e} \mathbf{1}$. $\mathcal{W}(e)$ denotes the affinity associated with hyperedge e. Formally, a hypergraph $\mathcal{G}$ can be represented by a relation matrix H with size $n \times m$, where each element $H(v, e) = 1$ if node v exists in hyperedge e; otherwise $H(v, e) = 0$. For node v, we use $\mathcal{N}(v)$ to denote its neighborhood which is a set containing nodes connected to v.

Given the target domain $\mathcal{D}_t$, we formulate a hypergraph structure based on the target samples. Specifically, we set each target sample x^t as a node v, *i.e.*, $v = x^t$, and form the hyperedge to effectively expose the underlying relation from a high-order perspective. We then seek the clusters for each target sample based on hyperedges and use these clusters to drive the fine-tuning of model $\mathcal{O}$ on target domain $\mathcal{D}_t$.

3) Hyperedge Generation: How to precisely select the nodes and measure the affinity of these nodes are fundamental in generating hyperedges. Specifically, we aim to generate hyperedges corresponding to every node, *i.e.*, $n = m$. For each node v_i, we set it as an anchor node and find its $k-1$ nearest neighbors. These $k-1$ neighbors and the node v_i together form a hyperedge e_i with degree k, denoted as $e_i = \{v_i\} \cup \{v_{i_1}, \ldots, v_{i_{k-1}}\}$. To obtain the nearest neighbors, we measure the similarity between samples using their cosine similarity of features z and then employ the KNN algorithm [23] to select $k-1$ neighbors based on the calculated similarity.

Given a hyperedge, we formulate its affinity by measuring the coherence of this hyperedge. Inspired by the work [85], we describe the coherence using a linear combination relation between its anchor node and other nodes. Assume that the feature of node v_i can be reconstructed by a linear combination of its $k-1$ neighbors, and the coefficient of each neighbor in this combination indicates the relation of node v_i to each neighbor. To obtain the affinity of a hyperedge e_i, we optimize the following objective function for node v_i

$$\begin{aligned} &\arg\min_{\mathbf{a}_i} \left\|\mathcal{F}(\mathcal{N}_{k-1}(v_i), \mathbf{a}_i) - f(v_i)\right\|_2^2 + \alpha \left\|\mathbf{a}_i\right\|_2, \\ &\text{s.t.} \quad \forall j, a_{i,j} \geq 0, a_{i,j} \in \mathbf{a}_i, \end{aligned} \tag{1}$$

where $\mathbf{a}_i = \{a_{i,1}, \ldots, a_{i,k-1}\}$ is the coefficient vector for node v_i and each element $a_{i,j}$ corresponds to the coefficient for the j-th neighbor of node v_i, while $\mathcal{N}_{k-1}(v_i) = \{v_{i_1}, \ldots, v_{i_{k-1}}\}$ is the set of $k-1$ neighbors of v_i obtained by KNN. In Eq. (1), we employ L2 norm of $\mathbf{a}_i$ for regularization, *i.e.*, $\|\mathbf{a}_i\|_2$ is the regularization term to make the coefficient vector sparse and α is a balancing factor, while $\mathcal{F}(\mathcal{N}_{k-1}(v_i), \mathbf{a}_i)$ denotes the linear combination operation as

$$\mathcal{F}(\mathcal{N}_{k-1}(v_i), \mathbf{a}_i) = \sum_{j=1}^{k-1} a_{i,j} \cdot f(v_{i_j}). \tag{2}$$

[2]Since the visualization projects high-dimensional features into 2-dimensional space, the Euclidean distance between nodes does not reflect their real similarity in the original feature space.

We perform the optimization Eq. (1) over all nodes. Each node v_i corresponds to a hyperedge $e_i = \{v_i\} \cup \{v_{i_1}, \ldots, v_{i_{k-1}}\}$, and its affinity can be represented by the vector $\mathbf{a}_i$ as $\mathcal{W}(e_i) = \{1, a_{i,1}, \ldots, a_{i,k-1}\} = \{1\} \cup \mathbf{a}_i$, where 1 is the fixed coefficient for the anchor node v_i.[3]

4) Refining Node Representations: Given the hypergraph, we utilize the hypergraph convolution layer [76], which adopts the vertex-hyperedge-vertex information propagation to efficiently exploit the high-order correlations, and to learn the structure-aware knowledge. This process can be written as

$$X^{l+1} = \sigma\left(D_v^{-1} H W D_e^{-1} H^{\mathrm{T}} X^l \Theta^l\right), \tag{3}$$

where X^l is the feature matrix of the nodes in the l layer learned, D_v and D_e are the diagonal degree matrices of vertices and hyperedges, respectively, while W is a diagonal matrix whose ith diagonal element is the weight of the ith hyperedge, and $\sigma(\cdot)$ denotes the nonlinear activation function. The parameter to be learned during training is Θ, which acts as a filter applied to the nodes for refinement. After the convolution operation, the high-order structure-aware feature X^{l+1} is obtained.

5) Why Hypergraph Is Better: In general, the relationships among target samples are complex, as more than two samples may share common properties. Consequently, relying solely on pair-wise relationships may result in a loss of information compared to utilizing high-order relations. For a complete mathematical proof regarding the advantages of hypergraphs, the reader is referred to [77].

6) Time Complexity Consideration: Generating hyperedges theoretically demands much more running time than pair-wise relation-based methods. However, each hyperedge can be generated independently by solving the optimization (1). This allows for parallelization of the process, which greatly reduces the overall time cost, making it comparable to pair-wise relation-based methods. Experimental demonstrations of this will be provided in Subsection IV-C.

C. Handling Domain Shift by Self-Loop

1) Motivation: Existing methods [13], [22], [23], [26] consider the neighborhood relations without explicitly taking the domain shift effect into account. These methods assume that all samples receive the same level of attention regarding domain assignment. However, samples likely require varying attention levels. For instance, those near the boundary between the source and target domain need more attention during optimization compared to samples far away from this boundary. Domain uncertainty reflects the likelihood of samples belonging to either the source or target domain, corresponding to these varying attention levels. We develop a self-loop for each target sample to indicate its domain uncertainty and incorporate the domain uncertainty into semantic relations to alleviate the domain shift problem.

2) Self-Loops Generation: Following [15], [19], [71], and [72], we estimate this domain uncertainty using the entropy value based on the pseudo-labels from the classifier g. Samples with high entropy indicate that the prediction probabilities across all categories are similar. It means that the classifier is uncertain about the category of the input sample. In other words, the classifier, trained on the source domain, has limited knowledge about this sample. As shown in Fig. 4, the entropy of samples at the classification boundary before domain adaptation is high, indicating high domain uncertainty (The transition from yellow to purple indicates an increase in entropy.).

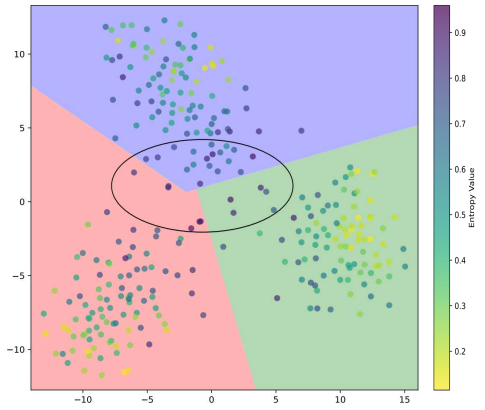

Fig. 4. T-SNE feature visualizations before adaptation on the VisDA dataset. The point colors indicate the entropy values of the samples. The transition from yellow to purple indicates an increase in entropy, reflecting higher domain uncertainty.

Denote $\mathcal{E}_s = \{\{v_1\}, \ldots, \{v_n\}\}$ as the self-loops for nodes. By adding self-loops, the hypergraph can be updated as $\mathcal{G} = (\mathcal{V}, \mathcal{E} \cup \mathcal{E}_s, \mathcal{W} \cup \mathcal{W}_s)$, where $\mathcal{W}_s$ is the affinity set of self-loops. To obtain the affinity of a self-loop, the most straightforward way is to calculate its entropy using the prediction vector p. However, solely relying on one node may suffer from the inner deviation of this node. Therefore, we consider its neighbors for representation to mitigate the errors. Specifically, given a node v_i, we find its corresponding hyperedge e_i and average the prediction vectors of the other $k-1$ nodes in this hyperedge. Then we calculate the entropy based on the averaged prediction vector and employ the exponential function for normalization. The rationale for using the exponential function is that it can assign larger weights to samples with higher entropy, allowing us to prioritize the samples likely to have been shifted. The affinity of the self-loop is calculated as

$$\mathcal{W}_s(\{v_i\}) = \exp(\phi(\bar{p}_i)), \tag{4}$$

where $\bar{p}_i$ is the averaged prediction vector given by

$$\bar{p}_i = \frac{1}{k-1} \sum_{v_j \in e_i / v_i} g(f(v_j)), \tag{5}$$

in which e_i / v_i denotes the nodes in e_i excluding v_i, and $\phi(\cdot)$ denotes the calculation of entropy:

$$\phi(\bar{p}_i) = \frac{1}{\log |C|} \sum_{c \in C} -\bar{p}_i^c \log \bar{p}_i^c. \tag{6}$$

In Eq. (6), C is the set of categories, and $\bar{p}_i^c$ denotes the average averaged prediction vector for category c. Based

[3] In the experiments, we consistently observe that the optimized coefficients are always less than 1, which aligns with our understanding that the highest affinity is typically assigned to the node itself.

TABLE I
ADAPTIVE LEARNING SCHEME

Method	Objective
AaD [23]	$-\sum_{j\in\mathcal{A}_i} p_i^{\mathrm{T}} p_j + \lambda \sum_{k\in\mathcal{B}_i} p_i^{\mathrm{T}} p_k$
AaD + ALS	$-\sum_{j\in\mathcal{A}_i} (1-d_{ij}^{\gamma}) p_i^{\mathrm{T}} p_j + \lambda \sum_{k\in\mathcal{B}_i} (1-d_{ik}^{\gamma}) p_i^{\mathrm{T}} p_k$

on each hyperedge obtained above, we add self-loops into every node to form a new hyperedge. For the nodes in hyperedge e_i, the affinity of their self-loops can be defined as $\mathbf{b}_i = \{\mathcal{W}_s(\{v_i\})\} \cup \{\mathcal{W}_s\{v_{i_1}\}, \ldots, \mathcal{W}_s\{v_{i_{k-1}}\}\}$. After adding self-loops, the affinity of hyperedge e_i can be defined as $\mathcal{W}(e_i) = \mathcal{W}(e_i) + \mathbf{b}_i$. Thus the affinity of all hyperedges is a set of $\{\mathcal{W}(e_1), \ldots, \mathcal{W}(e_m)\}$.

D. Clustering and Training

1) Clustering Based on High-Order Relations: To find the cluster for each node, we update the relation matrix H with representations calculated in Eq. (1) instead of the fixed value 1 or 0. Mathematically, the elements of H are updated as

$$H(v_i, e_j) = \begin{cases} \mathcal{W}\left(e_j\right)|_{v_i}, & v_i \in e_j, \\ 0, & v_i \notin e_j, \end{cases} \tag{7}$$

where $\mathcal{W}\left(e_j\right)|_{v_i}$ picks the element in $\mathcal{W}(e_i)$ corresponding to node v_i. Given the relation matrix H, we can obtain the relations of all nodes to all hyperedges with each row represents the relations of this node to all hyperedges, which reflects the knowledge of this node correlating with the hypergraph, *i.e.*, the target domain. Thus it can be viewed as a compact $(1 \times m)$ representation for this node. Then we perform KNN based on the representation of this node and find the top-h neighbors to form a cluster. To reduce the clustering cost, we compact the representation for each node by projecting H from $n \times m$ to $n \times m'(m' < m)$ (*e.g.*, by PCA). Therefore, for each node v_i, we obtain a set of neighbors as $\mathcal{A}_i$ and regard the rest nodes as a background set $\mathcal{B}_i$.

2) Adaptive Learning Scheme: Based on the above cluster results, we design an adaptive learning scheme that assigns "soft" attention levels to different samples. This scheme can be adapted into objectives considering both inter- and intra-cluster relations. We use AaD [23] as an example for illustration. Table I shows the objectives of AaD and AaD + Adaptive Learning Scheme (ALS). Note that $\mathcal{A}_i$ and $\mathcal{B}_i$ denote the same cluster and different cluster with sample x_i^t. By minimizing this objective, the first term **increases** the similarity of samples within the same cluster while the second term **decreases** the similarity in different clusters. Considering that samples within the same cluster are expected to be consistent, so those with notable discrepancies should obtain more attention. In contrast, samples in different clusters should differ, meaning those with small distances should be prioritized. To facilitate this, we calculate d_{ij}, which is the normalized Euclidean distance between the p_i and p_j, and we use γ to regulate the magnitude of distance. In the first term, $(1-d_{ij}^{\gamma})$ decreases when samples x_i^t and x_j^t are not similar. As a result, to minimize this term, p_i and p_j are pulled closer. Similarly, in the second term, $\left(1-d_{ik}^{\gamma}\right)$ increases when samples x_i^t and x_k^t are similar. Thus p_i and p_k are pushed further apart. More adaptations are studied in Subsection IV-C.

Algorithm 1 Overall Procedure of HG-SFDA

Input: Target domain $\mathcal{D}_t$, Source model $\mathcal{O}$, Total training iterations T, Interval of updating hypergraph T_{in}
Output: Fine-tuned model $\mathcal{O}$
1: **for** $t = 0 \rightarrow T$ **do**
2: **if** $t \,\%\, T_{in} = 0$ **then**
3: Constructing hypergraph $\mathcal{G}$: Generating hyperedges $\mathcal{E}$ and calculating affinity set $\mathcal{W} \cup \mathcal{W}_s$
4: Utilizing the hypergraph convolution layer to refine node representations X^{l+1}
5: **end if**
6: Training batch $\mathcal{V}_b \sim$ Target domain $\mathcal{D}_t$
7: **for** node $v_i \sim$ Training batch $\mathcal{V}_b$ **do**
8: Performing clustering based on node v_i
9: **end for**
10: Training model $\mathcal{O}$ on $\mathcal{V}_b$ using objectives
11: **end for**

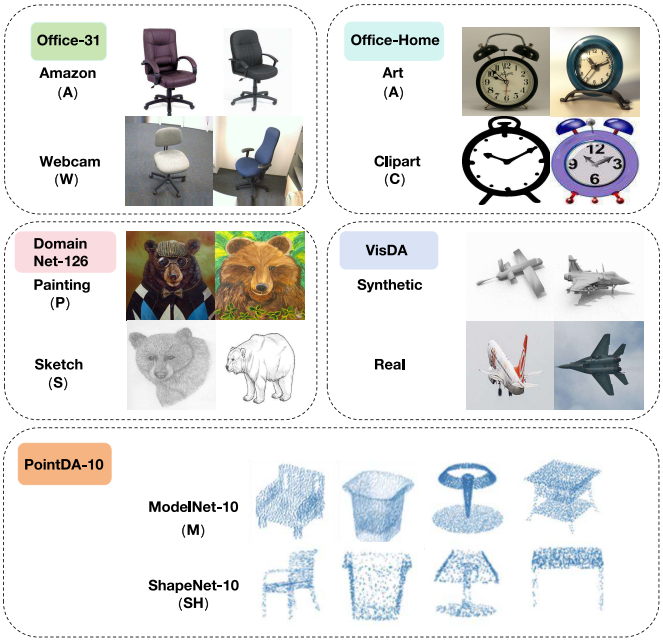

Fig. 5. Examples of Office-31, Office-Home, DomainNet-126, VisDA and PointDA-10 dataset images.

3) Overall Procedure: As summarized in Algorithm 1, we alternatively perform hyperedge generation (line 2-line 4), clustering process (line 8), and model training process (line 10). After each iteration, the model parameters are updated and the hypergraph can be reconstructed based on the refined features. To further reduce the time cost, we set a time interval T_{in} for constructing the hypergraph. These steps are executed until the maximum number of iterations T is reached.

IV. EXPERIMENTS

A. Experimental Setup

1) Datasets: Note that the SFDA task is typically validated using image classification, a general task in image processing. Therefore, we follow [20], [21], [47], [48], [87] and evaluate our method on four commonly used 2D image benchmark

datasets, Office-31 [27], Office-Home [28], DomainNet-126 [29] and VisDA [25]. The Office-31 dataset contains 3 domains of Amazon, Webcam and DSLR with 31 classes and 4652 images. The Office-Home dataset contains 15,500 images, covering 4 domains of Real, Clipart, Art and Product with 65 classes. DomainNet is a large-scale dataset. Following [17], [20], and [59], we use a subset of it that contains 126 classes from 4 domains Clipart (C), Painting (P), Real (R) and Sketch (S), and we refer to it as DomainNet-126 containing over 500,000 images. VisDA is a large-scale dataset with 12 classes for both synthetic and real object recognition tasks, containing 152,000 synthetic images in the source domain and 55,000 real object images in the target domain. Moreover, we also use a challenging 3D point cloud recognition dataset PointDA-10 [30] to fully evaluate our method. PointDA-10 is a 3D point cloud benchmark dataset designed for a domain adaptation, with 3 domains and 10 classes, denoted as ModelNet-10, ShapeNet-10, and ScanNet-10, respectively. It contains 27,700 training images and 5,100 test images. Examples of these dataset images are shown in Fig. 5.

2) Compared Methods: Following previous works [20], [21], [23], [59], we compare our method with general DA and SFDA methods. Specifically, we involves 30 SFDA methods, including pair-wise clustering-based methods (*e.g.*, SHOT [15], SHOT++ [72], BMD [16], NRC [22], NRC++ [26], AaD [23], SF(DA)^2 [20]) and other methods using various strategies such as pseduo-label filtering (*e.g.*, DIPE [65], CoWA-JMDS [71], C-SFDA [17], D-MCD [66], LLN [24], Co-learn [59], HCL [56], GPUE [19]), uncertainty-guided (*e.g.*, U-SFAN [55], CAF [47]), semi-supervised learning (*e.g.*, RLD [86]), domain distribution generation (*e.g.*, Sub-Sup [67], Mixup [57], ASM [21], DSiT-B [89], SFDA-DE [68], VMP [64], VDM [69]) and domain alignment (*e.g.*, TPDS [58], Improved SFDA [48], DPC [87], PSAT-GDA [90], A^{2}Net [14]).

3) Implementation Details: Our method is implemented using PyTorch 1.13.1 [93] with a Nvidia RTX 3090 GPU. In the training stage, we employ SGD optimizer with a momentum 0.9 for the 2D image datasets and use Adam optimizer for the PointDA-10 dataset. The batch size for all datasets is set to 64. The starting learning rate for 2D image datasets is set to 10^{-3}, and the one for PointDA-10 dataset is set to 10^{-6}. We train 50 epochs for Office-31 and train 40 epochs for Office-Home while 35 epochs for VisDA, and 100 epochs for PointDA-10. To construct hyperedges, we set the degree $k=6$ and update the hypergraph structure every $T_{in}=50$ iterations. We set $\alpha = 2$ in (1). To cluster the samples, we consider $h=3$ nearest neighbors. In the main experiment, we employ the objective of AaD (same λ) with our ALS and a regularization term $\mathcal{L}_{reg}$ following [24] and [94]. Note that the regularization term can prevent the model from overly focusing on incorrect predictions by accumulating knowledge of the predictions at previous training time stamps. For t-th time stamp (iteration), the target prediction is defined as $q_i^{(t)} = \delta q_i^{(t-1)} + (1-\delta)\, p_i^{(t)}$. The initial state of $q_i^{(0)}$ is set to 0 and δ is the weight factor. Slightly different from [24], we use KL divergence instead of inner product to measure their difference, as $\mathcal{L}_{reg} = \mathrm{KL}(q_i \| p_i)$. This term combines with the objective using a factor η as $\mathcal{L}_{AaD+ALS} + \eta\mathcal{L}_{reg}$. We set δ and η to 0.8 and 2, respectively.

B. Results

The accuracy results of different methods on Office-Home, Office-31 and VisDA, DomainNet-126, and PointDA-10 datasets are shown in Tables II to V, respectively. In each table, we show the accuracy results for each task attained by various methods and their average accuracy (Avg) over all tasks. The best results are marked in bold. SF denotes Source-Free unsupervised domain adaptation, and a mark $\times$ in this column means that the method requires access to source domain data during domain adaptation, while a mark ✓ means that the method belongs to the SFDA. Base denotes the base network of each method. Following the previous works [17], [21], [87], [89], [90], we use ResNet (RN) [1] and ViT-B (ViT) [95] as the backbone networks on 2D image datasets. Note that ResNet-50 [1] is used on Office-31, DomainNet-126 and Office-Home datasets, while ResNet-101 is employed on VisDA dataset for a fair comparison. For PointDA-10 dataset, we use PointNet as in [96].

1) Office-Home: Following previous works [22], [23], [24], [26], our method is compared to several recent state-of-the-art DA methods and SFDA methods, published in 2020 to 2024. As shown in Table II, our method achieves the best average accuracy result of 76.3% by ResNet backbone, which outperforms all the DA and SFDA methods compared. In particular, it surpasses the most recent SFDA methods ASM [21], DPC [87] and Improved SFDA [48] by 3.5%, 3.2% and 2.9%, respectively. And it exceeds the clustering-based methods BMD [16], AaD [23] by 2.7% and 3.6%, respectively. This is because either BMD or AaD explores the pairwise relation among prototype or neighbor samples, which limits their ability to exploit complex high-order relationships, making them more susceptible to outliers. When using ViT backbone, our method further improves the performance and exceeds the current best method PSAT-GDA [90] by 3.3%, demonstrating the effectiveness of our method on this dataset.

2) Office-31 and VisDA: Table III shows the performance of several methods on Office-31 and VisDA datasets. We evaluate more counterparts that are specifically designed for these two datasets. The results reveal that our method outperforms all the DA and SFDA benchmark methods on these two datasets. On Office-31 dataset, our method achieves the best performance using both RN and ViT backbone, surpassing the leading SFDA methods SF(DA)^2 [20], DPC [87] and DSiT-B [89] by 2.3%, 1.7% and 2.2%, respectively. This highlights the effectiveness of leveraging high-order relationships. Compared to Office-31, the VisDA dataset is more challenging due to significant distributional differences between the source domain (synthetic images) and the target domain (real images). Due to the limited capacity of RN, our method does not achieve notable performance gain. But after switching to the ViT backbone, our method improves significantly and surpasses the current best method DSiT-B [89] by 3.6%.

3) DomainNet-126: The DomainNet dataset is more challenging due to its large size, significant differences in

TABLE II
ACCURACY (%) OF DIFFERENT METHODS ON OFFICE-HOME DATASET

Method	SF	Base	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
CoVi (ECCV'22) [37]	×	RN	58.5	78.1	80.0	68.1	80.0	77.0	66.4	60.2	82.1	76.6	63.6	86.5	73.1
RAIN (IJCAI'23) [38]	×	RN	57.0	79.7	82.8	67.9	79.5	81.2	67.7	53.2	84.6	73.3	59.6	85.6	73.0
COT (CVPR'23) [46]	×	RN	57.6	75.2	83.2	67.8	76.2	75.7	65.4	56.2	82.4	75.1	60.7	84.7	71.7
TCPL (TIP'24) [10]	×	RN	61.2	80.5	82.8	68.8	75.1	76.5	71.7	59.8	83.5	78.1	66.2	87.6	74.3
SHOT (ICML'20) [15]	✓	RN	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
NRC (NeurIPS'21) [22]	✓	RN	57.7	80.3	82.0	68.1	79.8	78.6	65.3	56.4	83.0	71.0	58.6	85.6	72.2
DIPE (CVPR'22) [65]	✓	RN	56.5	79.2	80.7	70.1	79.8	78.8	67.9	55.1	83.5	74.1	59.3	84.8	72.5
CoWA-JMDS (ICML'22) [71]	✓	RN	56.9	78.4	81.0	69.1	80.0	79.9	67.7	57.2	82.4	72.8	60.5	84.5	72.5
VMP (NeurIPS'22) [64]	✓	RN	57.9	77.6	82.5	68.6	79.4	80.6	68.4	55.6	83.1	75.2	59.6	84.7	72.8
D-MCD (AAAI'22) [66]	✓	RN	59.4	78.9	80.2	67.2	79.3	78.6	65.3	55.6	82.2	73.3	62.8	83.9	72.2
AaD (NeurIPS'22) [23]	✓	RN	59.3	79.3	82.1	68.9	79.8	79.5	67.2	57.4	83.1	72.1	58.5	85.4	72.7
Sub-Sup (ECCV'22) [67]	✓	RN	61.0	80.4	82.5	69.1	79.9	79.5	69.1	57.8	82.7	74.5	**65.1**	86.4	74.0
BMD (ECCV'22) [16]	✓	RN	58.1	79.7	82.6	69.3	81.0	80.7	70.8	57.6	83.6	74.0	60.0	85.9	73.6
U-SFAN (ECCV'22) [55]	✓	RN	57.8	77.8	81.6	67.9	77.3	79.2	67.2	54.7	81.2	73.3	60.3	83.9	71.9
TPDS (IJCV'24) [58]	✓	RN	59.3	80.3	82.1	70.6	79.4	80.9	69.8	56.8	82.1	74.5	61.2	85.3	73.5
NRC++ (TPAMI'23) [26]	✓	RN	57.8	80.4	81.6	69.0	80.3	79.5	65.6	57.0	83.2	72.3	59.6	85.7	72.5
C-SFDA (CVPR'23) [17]	✓	RN	60.3	80.2	82.9	69.3	80.1	78.8	67.3	58.1	83.4	73.6	61.3	86.3	73.5
CAF (TIP'23) [47]	✓	RN	59.8	81.2	83.2	67.2	79.2	80.1	68.4	56.4	83.0	73.7	61.2	85.9	73.2
LLN (ICLR'23) [24]	✓	RN	58.4	78.7	81.5	69.2	79.5	79.3	66.3	58.0	82.6	73.4	59.8	85.1	72.6
Co-learn (ICCV'23) [59]	✓	RN	57.7	80.4	83.3	70.1	80.1	80.6	66.6	55.5	84.1	72.1	57.6	84.3	72.7
RLD (ECCV'24) [86]	✓	RN	62.2	81.0	79.7	68.8	**85.4**	78.6	67.7	**61.7**	79.5	69.0	64.1	**88.2**	73.8
Improved SFDA (CVPR'24) [48]	✓	RN	60.7	78.9	82.0	69.9	79.5	79.7	67.1	58.8	82.3	74.2	61.3	86.4	73.4
DPC (CVPR'24) [87]	✓	RN	59.5	80.6	82.9	69.4	79.3	80.1	67.3	57.2	83.7	73.1	58.9	84.9	73.1
ASM (TIP'24) [21]	✓	RN	56.9	79.1	82.9	69.5	79.6	79.7	67.9	55.1	82.6	74.7	60.5	84.8	72.8
Ours	✓	RN	62.4	**82.2**	83.4	**73.3**	83.7	**82.8**	**73.4**	61.5	**85.5**	**76.8**	62.6	87.8	**76.3**
CDTrans (ICLR'22) [88]	×	ViT	68.8	85.0	86.9	81.5	87.1	87.3	79.6	63.3	88.2	82.0	66.0	90.6	80.5
Mixup (ICML'22) [57]	✓	ViT	65.3	82.1	86.5	77.3	81.7	82.4	77.1	65.7	84.6	81.2	70.1	88.3	78.5
DIPE (CVPR'22) [65]	✓	ViT	66.0	80.6	85.6	77.1	83.5	83.4	75.3	63.3	85.1	81.6	67.7	89.6	78.2
DSiT-B (ICCV'23) [89]	✓	ViT	69.2	83.5	87.3	80.7	86.1	86.2	77.9	67.9	86.6	82.4	68.3	89.8	80.5
PSAT-GDA (TMM'23) [90]	✓	ViT	73.1	88.1	89.2	82.1	88.8	88.9	83.0	72.0	89.6	83.3	73.7	91.3	83.6
Ours	✓	ViT	**76.1**	**91.7**	**91.4**	**87.5**	**91.7**	**92.2**	**87.0**	**75.6**	**91.7**	**88.4**	**77.1**	**92.8**	**86.9**

cross-domain distributions, and category complexity. We evaluate 7 domain shifts built from the 4 domains and we report the top-1 accuracy under each domain shift as well as the average accuracy (Avg) over all domain shifts. The results, presented in Table IV, demonstrate the superiority of our method, which outperforms the second-best approach (GPUE [19]) by 2.7% using RN and the second-best approach (RLD [86]) by 2.1% using ViT.

4) PointDA-10: Table V shows the performance of our method compared with other methods on PointDA-10 dataset. Since this dataset is for 3D point cloud recognition, only a few methods have reported their performance on it. It can be seen that our method achieves the best average accuracy of 60.7%, surpassing the current best methods SF(DA)^2 [20], BMD [16] by 3.6% and 3.7%, respectively. This experiment corroborates that our method is not only effective in general 2D images but also in 3D point cloud recognition tasks.

C. Ablation Study

1) Effect of Each Component: We investigate the effects of the three components in our proposed HG-SFDA, namely, high-order neighborhood relation, self-loop strategy, and adaptive relation-based loss, on Office-31 and Office-Home datasets. As shown in Table VI, the first row is the baseline performance without using any of these components, which is degraded to the AaD method [23]. The second row represents using $\mathcal{L}_{reg}$, which slightly improves the baseline by 0.6% and 0.4%. The third row highlights the effectiveness of ALS, leading to improvements of 0.8% and 1%. Rows four to six illustrate the effectiveness of our proposed hypergraph learning strategy. The fifth row shows the results of using the hypergraph learning strategy on top of second row (*i.e.*, with $\mathcal{L}_{reg}$, without ALS), boosting performance by 1.2% and 2.8%. The sixth row corresponds to the setting of using the hypergraph learning strategy on top of third row (*i.e.*, with $\mathcal{L}_{reg}$, with ALS), which further improves results by 1.5% and 2.6%. Specifically, by exploring the high-order information, the performance is improved by 0.1% and 2%. By adding the self-loop to hyperedge, the performance is improved by 1.1% and 0.8%, respectively. After adding the adaptive relation-based loss, additional 0.5% and 0.4% of performance enhancement is achieved.

2) Sensitivity of Batch Size: Table VII shows the performance of our method compared to the pair-wise methods using different batch sizes on Office-31 (A→W). The results of Table VII indicate that as the batch size increases, the performance of our method improves, while the performance of NRC [22], AaD [23] and LLN [24] remain stable. This is because more samples result in more complex correlations, which highlights the effectiveness of high-order relations. To ensure a fair comparison, we use a batch size of 64 for all datasets.

TABLE III
ACCURACY (%) OF DIFFERENT METHODS ON OFFICE-31 AND VISDA DATASET

Method	SF	Base	Office-31							VisDA
			A→D	A→W	D→W	W→D	D→A	W→A	Avg	S→R
CoVi (ECCV'22) [37]	×	RN	98.0	97.6	**99.3**	100.0	77.5	78.4	91.8	88.5
RAIN (IJCAI'23) [38]	×	RN	93.8	88.8	96.8	99.5	75.5	76.7	88.5	82.7
COT (CVPR'23) [46]	×	RN	96.1	96.5	99.1	100.0	76.7	77.4	91.0	87.1
TCPL (TIP'24) [10]	×	RN	-	-	-	-	-	-	-	87.8
SHOT (ICML'20) [15]	✓	RN	94.0	90.1	98.4	99.9	74.7	74.3	88.6	82.9
HCL (NeurIPS'21) [56]	✓	RN	90.8	91.3	98.2	100.0	72.7	72.7	87.6	83.5
A^2Net (ICCV'21) [14]	✓	RN	94.5	94.0	99.2	100.0	76.7	76.1	90.1	84.3
SHOT++ (TPAMI'22) [72]	✓	RN	94.3	90.4	98.7	99.9	76.2	75.8	89.2	87.3
NRC (NeurIPS'21) [22]	✓	RN	96.0	90.8	99.0	100.0	75.3	75.0	89.4	85.9
D-MCD (AAAI'22) [66]	✓	RN	94.1	93.5	98.8	100.0	76.4	76.4	89.9	87.5
DIPE (CVPR'22) [65]	✓	RN	96.6	93.1	98.4	99.6	75.5	77.2	90.1	83.1
CoWA-JMDS (ICML'22) [71]	✓	RN	94.4	95.2	98.5	100.0	76.2	77.6	90.3	86.9
Mixup (ICML'22) [57]	✓	RN	94.6	93.2	98.9	100.0	78.3	**78.9**	90.7	87.8
SFDA-DE (CVPR'22) [68]	✓	RN	96.0	94.2	98.5	99.8	76.6	75.5	90.1	86.5
BMD (ECCV'22) [16]	✓	RN	96.2	94.2	98.0	100.0	76.0	76.0	90.1	88.7
Sub-Sup (ECCV'22) [67]	✓	RN	95.6	94.6	99.2	99.8	77.0	77.7	90.7	88.2
U-SFAN (ECCV'22) [55]	✓	RN	94.2	92.8	98.0	99.0	74.6	74.4	88.8	82.7
AaD (NeurIPS'22) [23]	✓	RN	96.4	92.1	99.1	100.0	75.0	76.5	89.9	88.0
LLN (ICLR'23) [24]	✓	RN	-	-	-	-	-	-	-	86.4
TPDS (IJCV'24) [58]	✓	RN	97.1	94.5	98.7	99.8	75.7	75.5	90.2	87.6
NRC++ (TPAMI'23) [26]	✓	RN	95.9	91.2	99.1	100.0	75.5	75.0	89.5	88.1
Co-learn (ICCV'23) [59]	✓	RN	96.6	92.5	98.9	99.8	77.3	76.6	90.3	88.2
C-SFDA (CVPR'23) [17]	✓	RN	96.2	93.9	98.8	99.7	77.3	77.9	90.5	87.8
CAF (TIP'23) [47]	✓	RN	95.0	93.5	99.1	99.9	76.3	78.4	90.3	87.3
Improved SFDA (CVPR'24) [48]	✓	RN	95.3	94.2	98.3	99.9	76.4	77.5	90.3	88.4
DPC (CVPR'24) [87]	✓	RN	95.8	94.5	98.9	100.0	76.5	76.8	90.5	88.8
$SF(DA)^2$ (ICLR'24) [20]	✓	RN	95.8	92.1	99.0	99.8	75.7	76.8	89.9	88.1
ASM (TIP'24) [21]	✓	RN	96.0	95.1	98.7	100.0	75.3	77.2	90.4	84.1
Ours	✓	RN	**98.4**	**98.2**	99.1	**100.0**	**78.6**	78.7	**92.2**	**89.6**
CDTrans (ICLR'22) [88]	×	ViT	97.0	96.7	99.0	100.0	81.1	81.9	92.6	88.4
SSRT-B (CVPR'22) [91]	×	ViT	98.6	97.7	99.2	100.0	83.5	82.2	93.5	88.7
Mixup (ICML'22) [57]	✓	ViT	95.4	96.1	98.6	100.0	80.2	80.1	91.7	86.3
DIPE (CVPR'22) [65]	✓	ViT	94.8	95.5	98.5	100.0	77.5	77.1	90.5	82.8
DSiT-B (ICCV'23) [89]	✓	ViT	98.0	97.2	99.1	100.0	81.7	81.8	93.0	87.6
Ours	✓	ViT	**98.4**	**99.0**	**99.9**	99.8	**86.3**	**87.6**	**95.2**	**91.2**

TABLE IV
ACCURACY (%) OF DIFFERENT METHODS ON DOMAINNET-126 DATASET

Method	SF	Base	S→P	C→S	P→C	P→R	R→S	R→C	R→P	Avg
MCC (ECCV'20) [92]	×	RN	47.3	34.9	41.9	72.4	35.3	44.8	65.7	48.9
SHOT (ICML'20) [15]	✓	RN	66.1	60.1	66.9	80.8	59.9	67.7	68.4	67.1
NRC (NeurIPS'21) [22]	✓	RN	65.7	58.6	64.5	82.3	58.4	65.2	68.2	66.1
AaD (NeurIPS'22) [23]	✓	RN	65.4	54.2	59.8	81.8	54.6	60.3	68.5	63.5
Co-learn (ICCV'23) [59]	✓	RN	65.7	60.1	63.8	79.2	58.2	68.0	67.6	66.1
C-SFDA (CVPR'23) [17]	✓	RN	67.4	62.1	68.5	80.4	62.7	70.8	71.1	69.0
GPUE (CVPR'23) [19]	✓	RN	67.5	64.0	68.8	76.5	65.7	74.2	70.4	69.6
$SF(DA)^2$ (ICLR'24) [20]	✓	RN	67.7	59.6	67.8	**83.5**	60.2	68.8	70.5	68.3
Ours	✓	RN	**71.3**	**66.3**	**71.5**	82.6	**65.8**	**75.2**	**73.1**	**72.3**
RLD (ECCV'24) [86]	✓	ViT	76.6	68.5	**77.8**	**85.1**	68.3	76.9	77.8	75.9
Ours	✓	ViT	**80.7**	**75.9**	68.5	83.5	**76.8**	**81.1**	**79.6**	**78.0**

3) Interval T_{in} in Updating Hypergraph: The constructed hypergraph is periodically updated during training to maintain its effectiveness. To investigate the impact of the interval T_{in} on the achievable accuracy, we experiment on Rw→Pr track on Office-Home dataset. Fig. 6 (Left) illustrates the effect of using different update intervals on our method, showing that the accuracy of our method is stable around 86% as the interval number T_{in} increases from 50 to 100. This indicates that the performance of our method is insensitive to the update interval. In the main experiment, we select 50 as the final interval number. Note that Fig. 6 (Left) reflects the optimization results for the "hypergraph update interval T_{in}" parameter alone, while keeping other parameters at their initial values.

4) Degree k in Hyperedges: Fig. 6 (Right) is based on the optimal update interval (50) determined in Fig. 6 (Left), and further shows the effect of different degree k in hyperedges to

TABLE V
ACCURACY (%) OF DIFFERENT METHODS ON POINTDA-10 DATASET

Method	SF	M→SC	M→SH	SC→M	SC→SH	SH→M	SH→SC	Avg
ADDA (CVPR'17) [9]	×	30.5	61.0	48.9	51.1	40.4	29.3	43.5
MCD (CVPR'18) [97]	×	31.0	62.0	46.8	59.3	41.4	31.3	45.3
PointDAN (NeurIPS'19) [30]	×	33.0	64.2	49.1	64.1	47.6	33.9	48.7
SHOT (ICML'20) [15]	✓	31.8	62.1	67.6	56.9	75.8	24.3	53.1
VDM (TCSVT'22) [69]	✓	30.9	58.4	45.3	61.8	61.0	**40.8**	49.7
NRC (NeurIPS'21) [22]	✓	25.8	64.8	70.1	68.1	59.8	26.9	52.6
AaD (NeurIPS'22) [23]	✓	34.6	69.6	68.0	66.6	67.7	28.8	55.9
BMD (ECCV'22) [16]	✓	32.8	66.1	75.0	62.0	**81.5**	24.4	57.0
NRC++ (TPAMI'23) [26]	✓	27.6	67.2	74.5	71.2	60.2	30.4	55.1
SF(DA)^2 (ICLR'24) [20]	✓	**35.5**	70.3	70.4	69.2	68.3	29.0	57.1
Ours	✓	33.4	**71.4**	**80.2**	**73.1**	77.3	28.6	**60.7**

TABLE VI
EFFECT OF EACH COMPONENT

Hypergraph High-order	Hypergraph Self-loop	Adaptive Relation	$\mathcal{L}_{reg}$	Office-31	Office-Home
×	×	×	×	89.9	72.7
×	×	×	✓	90.5 (+0.6)	73.1 (+0.4)
×	×	✓	✓	90.7 (+0.8)	73.7 (+1.0)
✓	×	×	✓	90.6 (+0.7)	75.1 (+2.4)
✓	✓	×	✓	91.7 (+1.8)	75.9 (+3.2)
✓	✓	✓	✓	**92.2** (+2.3)	**76.3** (+3.6)

TABLE VII
IMPACT OF BATCH SIZE ON ACCURACY (%)

Method	Batch Size		
	32	64	128
NRC [22]	90.2	90.8	90.1
AaD [23]	93.3	92.6	91.5
LLN [24]	91.8	92.2	91.8
Ours	**94.3**	**98.1**	**98.6**

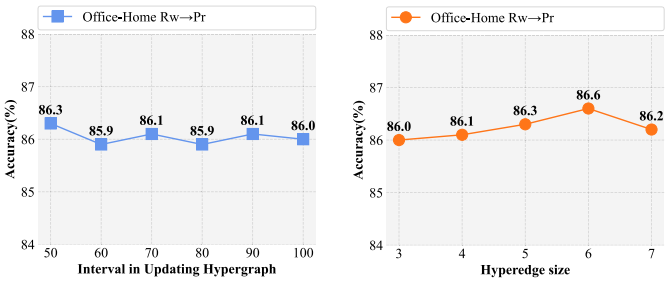

Fig. 6. (Left) Effect of different intervals in updating hypergraph. (Right) Effect of different hyperedge degrees.

the achievable accuracy. The experiment setting is the same as above. We change the hyperedge degree k from 3 to 7 and find that our method is only slightly affected, which indicates that the performance is also insensitive to hyperedge degree.

5) Scale Factor γ and Hyperparameter λ in Objective: As shown in Table I, γ is the scale factor controlling the attention levels. Fig. 7 (Left) shows the effect of different γ on Rw→Pr track on Office-Home dataset, showing that our method is not sensitive to γ either. We experimentally prove that taking $\gamma=7$ is valid for all datasets. We then study the effect of different λ

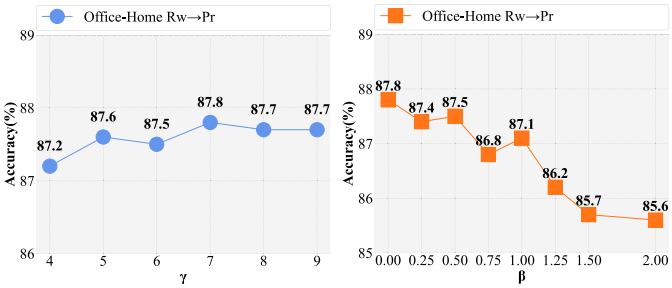

Fig. 7. (Left) Effect of different scale factor γ. (Right) Effect of different balancing factor β.

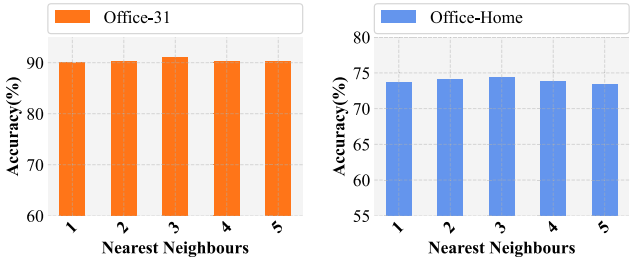

Fig. 8. Effect of different numbers of nearest neighbors on Office-31 (Left) and Office-Home (Right).

on the second loss term. Since the rate of decay is controlled by a balancing factor β as $\lambda=\left(1+10\cdot\frac{\text{iter}}{\text{maxiter}}\right)^{-\beta}$, where maxiter is the maximum number of iterations, we study the effect of β instead. Fig. 7 (Right) shows the effect of different β on Rw→Pr track on Office-Home dataset, which indicates that a proper β is important, *e.g.*, $\beta=0$ is the best for Office-Home dataset.

6) Number of Nearest Neighbors in Clustering: This part studies the effect of using different numbers of nearest neighbors in clustering. Fig. 8 shows the corresponding performance on Office-31 (Left) and Office-Home (Right) using different nearest neighbors, from 1 to 5. From Fig. 8, we observe that using the median value 3 achieves the best performance on these two datasets. This is due to the fact that a small number of neighbors is prone to be affected by outliers and a mass of neighbors may lose the locality of group information.

7) Adapting ALS on Various Objectives: As shown in Table VIII, we add the ALS to the objectives in G-SFDA,

TABLE VIII
ADAPTING ALS ON VARIOUS OBJECTIVES

Method	Loss	Avg accuracy (%)
G-SFDA [13]	$-g(W_{ij}p_i^{\mathrm{T}}p_j)+\sum_{c=1}^{C}KL(\overline{p}_c\|\|q_c)$	76.9
G-SFDA + ALS	$-g((1-d_{ij}^{\gamma})W_{ij}p_i^{\mathrm{T}}Tp_j)+\sum_{c=1}^{C}KL(\overline{p}_c\|\|q_c)$	**77.2**
NRC [22]	$-g(W_{ij}p_i^{\mathrm{T}}p_j)+\sum_{c=1}^{C}KL(\overline{p}_c\|\|q_c)$	79.8
NRC + ALS	$-g((1-d_{ij}^{\gamma})W_{ij}p_i^{\mathrm{T}}p_j)+\sum_{c=1}^{C}KL(\overline{p}_c\|\|q_c)$	**80.1**
AaD [23]	$-\sum_{j\in C_i}p_i^{\mathrm{T}}p_j+\lambda\sum_{m\in\mathcal{B}_i}p_i^{\mathrm{T}}p_m$	79.8
AaD + ALS	$-\sum_{j\in C_i}(1-d_{ij}^{\gamma})p_i^{\mathrm{T}}p_j+\lambda\sum_{m\in\mathcal{B}_i}(1-d_{ik}^{\gamma})p_i^{\mathrm{T}}p_m$	**80.4**
SF(DA)^2 [20]	$-\frac{2}{K}\sum_{j\in\mathcal{N}_i^K}p_i^{\mathrm{T}}p_j+\sum_{k\in\mathcal{B}}(p_i^{\mathrm{T}}p_k)^2+\mathcal{L}_{IFA}+\mathcal{L}_{FD}$	77.3
SF(DA)^2 + ALS	$-\frac{2}{K}\sum_{j\in\mathcal{N}_i^K}(1-d_{ij}^{\gamma})p_i^{\mathrm{T}}p_j+\sum_{k\in\mathcal{B}}(1-d_{ik}^{\gamma})(p_i^{\mathrm{T}}p_k)^2+\mathcal{L}_{IFA}+\mathcal{L}_{FD}$	**78.1**

TABLE IX
ADAPTING HYPERGRAPH ON VARIOUS OBJECTIVES

Method	Avg accuracy (%)
G-SFDA [13]	76.9
G-SFDA + HG	**79.4**
NRC [22]	79.8
NRC + HG	**81.3**
AaD [23]	79.8
AaD + HG	**82.8**
SF(DA)^2 [20]	77.3
SF(DA)^2 + HG	**80.3**

NRC, AaD and SF(DA)^2 on Office-Home (C→P). G-SFDA and NRC only consider intra-cluster relations while AaD and SF(DA)^2 consider intra- and inter-cluster relations. It can be seen that the performance of all the methods are further improved after using ALS. In our main experiments, AaD is employed as the base objective.

8) Adapting Hypergraph on Various Objectives: To demonstrate our method can serve as a plug-and-play strategy, we follow the setting in Table VIII to apply our method to various objectives. The results, shown in Table IX, demonstrate that using hypergraph learning strategy can notably improve the accuracy on all different objectives. This corroborates that our method is not limited to AaD but can be used as a plug-and-play strategy for other objective functions.

9) Feasibility of Using Domain Uncertainty to Solve Domain Shifts: For feasibility validation, we split VisDA dataset to ensure a sufficient number of samples for each class and conduct experiments on it. Specifically, we validate three variants: 1) only using the source model, 2) our method without the self-loop strategy, and 3) our method using the self-loop strategy. As shown in Table X, for the tasks of bcycl, bus and truck, the performance are very poor when only using the source model, which indicates that these tasks have large domain uncertainty. For these tasks, especially for truck, the self-loop strategy can further improve the performance. This is because the domain shifts of difficult samples can be mitigated by adding large self-loop weights (domain uncertainty). Thus the clustering of such samples can be focused.

10) Complexity Analysis: Table XI shows the training time, the number of parameters, the floating point operations, memory usage, and accuracy of different methods on Office-Home under the same setting. It can be seen that our method maintains comparable training time and GPU memory usage, with a moderate increase in the number of parameters (Params) and the floating point operations (FLOPs), but achieves notable performance improvement. It is worth noting that the process of building a hypergraph is executed in RAM rather than on the GPU. While our method requires more RAM than other methods, it is unlikely to pose a barrier in real-world application, given that RAM is relatively inexpensive and widely available. The results indicate that our method properly balances complexity and performance, demonstrating the efficacy of hypergraph learning. Note that the existing neighborhood clustering-based methods [20], [22], [23], [24] are all improved based on the same baseline method [13], and they have comparable number of parameters (Params) and the floating point operations (FLOPs).

D. Further Analysis

1) Feature Visualization: We visualize feature distribution using t-SNE [98] on Office-Home (C→A). Meanwhile, we choose 4 comparisons, the Source model, NRC [22], AaD [23] and SF(DA)^2 [20]. As can be seen from Fig. 9 (Top), from Source to Ours, we can observe that the target samples of the same class become more coherent, and the margins between different classes become clearer and larger, demonstrating the effectiveness of our proposed method. Fig. 9 (Bottom) shows the comparison of feature distribution between our method and the general pairwise method AaD [23] during the training process. The results indicate that, as training epochs progress, our method consistently achieves better clustering performance than the pairwise method AaD, highlighting the effectiveness of hypergraph learning in SFDA.

2) Compared to CLIP-Based Multimodal Methods: Several recent methods employ CLIP [99] to solve the SFDA problem. Since CLIP contains powerful prior knowledge, directly comparing these methods to ours would be unfair. For a compromise, we compare our method with the methods that use ResNet and ViT as the backbone of CLIP, and average the adaptation results grouped by target domain name. As shown in Table XII. Our method outperforms CLIP-related methods on all tasks for Office-31 and VisDA. Specifically, for ResNet, it outperforms CLIP-RN [99] and DIFO [84] by 20.8% and 5.9%, as well as 2.9% and 0.8%, respectively. For ViT, the improvement over CLIP-RN [99] and DIFO [84] is 15.4% and

TABLE X

FEASIBILITY OF USING DOMAIN UNCERTAINTY TO SOLVE DOMAIN SHIFTS (ACCURACY %)

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg
Source	71.1	18.2	46.0	70.0	57.7	4.4	82.3	24.0	57.8	35.0	87.3	7.7	46.8
Ours w/o self-loop	96.1	79.5	80.3	**79.5**	93.3	**98.9**	90.9	**85.9**	93.7	87.4	89.1	27.5	83.5
Ours w/ self-loop	**96.4**	**80.6**	**81.7**	79.4	**93.3**	98.1	**91.8**	83.5	**95.1**	**90.4**	**90.5**	**36.6**	**84.8**

TABLE XI

COMPLEXITY AND PERFORMANCE ANALYSIS

Method	A→R		R→A		R→C		R→P		GPU (MB)	RAM (MB)	FLOPs (G)	Params (M)
	Time (min)	Acc(%)	Time (min)	Acc(%)	Time (min)	Acc(%)	Time (min)	Acc(%)				
NRC (NeurIPS'21) [22]	26.1	81.5	10	71.7	16	58.6	16.4	85.9	323.2	5853	263.04	24.05
LLN (ICLR'23) [24]	25.7	80.7	10	74	16.1	59.6	16.6	83.7	323.2	6003	263.04	24.05
AaD (NeurIPS'22) [23]	26.2	81.4	9.8	72.8	15.9	59.4	16.2	85.1	323.1	5975	263.04	24.05
SF(DA)^2 (ICLR'24) [20]	29.9	79.6	8.1	69.6	12.4	57.5	12.6	84.8	361	9708	263.04	24.05
Ours	24.4	83.4	12.6	76.8	21	62.6	21	87.8	313.1	16138	268.26	24.50

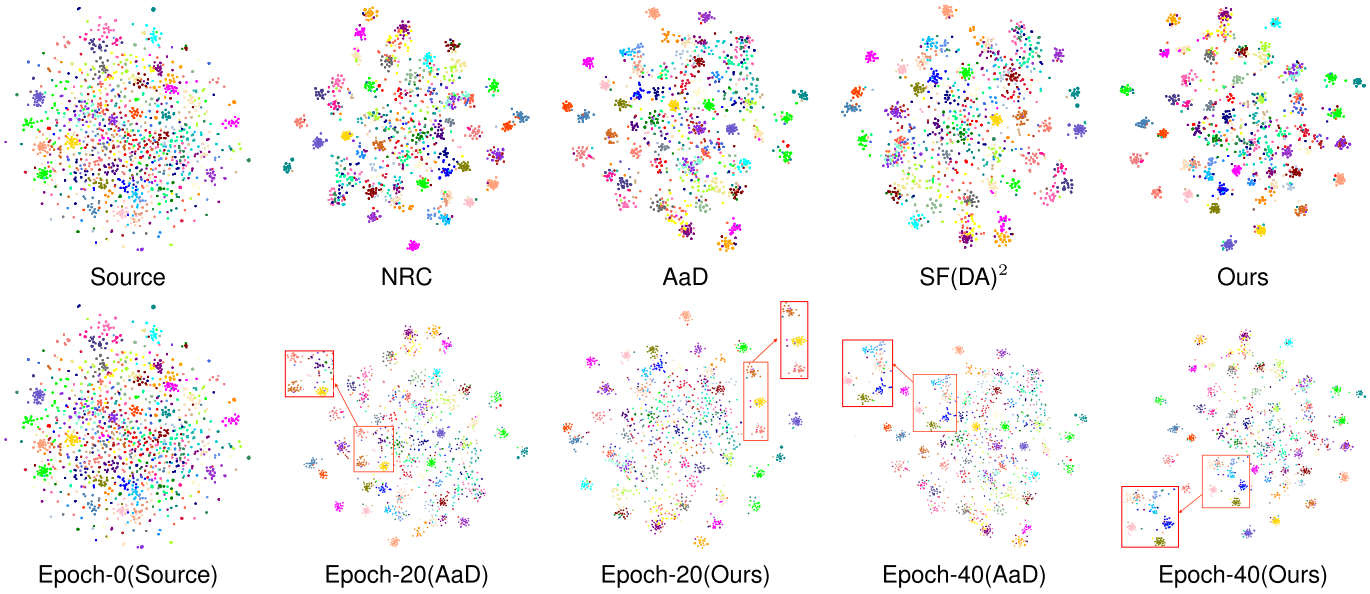

Fig. 9. Visualization of feature distribution on the C→A in Office-Home. T-SNE feature distribution over 65 categories, where different colors stand for different categories. (Top) Note that NRC [22] (NeurIPS'21), AaD [23] (NeurIPS'22), and SF(DA)^2 [20] (ICLR'24) are pair-wise based methods. Source indicates the source model before domain adaptation. (Bottom) Comparison of feature changes during training between our method and the pairwise method AaD [23].

TABLE XII

ACCURACY (%) OF CLIP-BASED METHODS. THE BACKBONE OF IMAGE ENCODER IS RESNET AND VIT, RESPECTIVELY

Method	Office-31				VisDA
	→A	→D	→W	Avg	S→R
CLIP-RN (ICML'21) [99]	73.1	73.9	67.0	71.4	83.7
Source+CLIP-RN	76.3	90.4	84.0	83.6	82.0
DIFO-CLIP-RN (CVPR'24) [84]	78.6	95.3	93.9	89.3	88.8
Ours-RN	**78.7**	**99.2**	**98.7**	**92.2**	**89.6**
CLIP-ViT (ICML'21) [99]	76.0	82.7	80.6	79.8	82.9
Source+CLIP-ViT	78.5	93.0	89.6	87.0	82.0
DIFO-CLIP-ViT (CVPR'24) [84]	83.1	98.0	96.4	92.5	90.3
Ours-ViT	**87.0**	**99.1**	**99.5**	**95.2**	**91.2**

8.3%, and 2.7% and 0.9%, respectively. This demonstrates the competitive capacity of our method compared to CLIP-based methods.

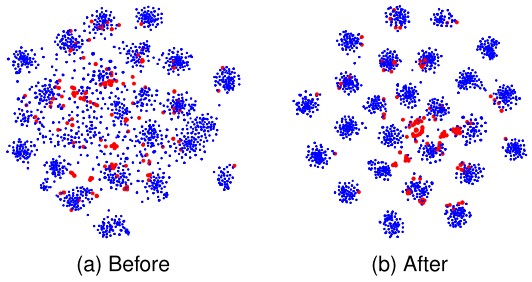

Fig. 10. T-SNE feature visualizations before and after adaptation. Blue and red colors correspond to known and unknown categories.

3) Source-Free Open-Set Domain Adaptation (Soda): We provide additional experiments in the open-set DA setting on Office-Home. In the open-set scenario, the target domain includes unseen classes that are not contained in the source domain. For open-set detection, we follow the same protocol

TABLE XIII
SOURCE-FREE OPEN-SET DOMAIN ADAPTATION ON OFFICE-HOME

Method	SF	Avg accuracy (%)
ResNet (CVPR'2016) [1]	×	65.3
OSBP (ECCV'18) [100]	×	65.7
STA (CVPR'19) [101]	×	69.5
GLC (CVPR'23) [102]	×	69.8
SHOT (ICML'20) [15]	✓	72.8
SHOT-IM (ICML'20) [15]	✓	71.5
SHOT+HCL (NeurIPS'21) [56]	✓	73.2
CoWA-JMDS (ICML'22) [71]	✓	73.2
AaD (NeurIPS'22) [23]	✓	71.8
U-SFAN (ECCV'22) [55]	✓	73.5
CREL (CVPR'23) [18]	✓	73.3
Ours	✓	**77.3**

TABLE XIV
NUMERICAL ABLATION ON REGULARIZATION TERM IN EQ.(1)

Regularization norm l	α	Office-31 (A→W)	Office-Home (P→A)
L1	1.0	97.9	72.9
	2.0	95.0	72.3
L2	1.0	96.4	72.5
	2.0	98.1	73.1

for the detection of unseen classes as in SHOT [15]. We sort the entropy of the samples and perform two-class k-means clustering. The high entropy clusters are then classified as unknown samples and the low entropy clusters are classified as known samples. The known samples are used to train the model. As can be seen from the results in Table XIII, our method outperforms the current state-of-the-art method [55] with an improvement of 3.8%. This scenario further highlights the benefit of high-order relations in uncovering the underlying correlations, especially the semantic difference between known and unknown categories (see the illustration of Fig. 10).

4) Norm l and Factor α in Eq (1): Table XIV shows the study regarding the employed norm l and factor α in the regularization term on Office-31 (A→W) and Office-Home (P→A). We employ the L2 norm with $\alpha = 2$ in the main experiments. The grouping effect is improved by grouping more relevant neighbors to construct a hyperedge and controlling the hypergraph sparsity more effectively.

5) Limitations and Future Works: One limitation of our method is that it solely relies on visual features to construct the hypergraph, neglecting the potential benefits of linguistic information. With the emergence of powerful vision-language models (VLMs), future work could explore enriching sample representations using multi-modal semantic embeddings derived from VLMs. Integrating such multi-modal information may enable the construction of more expressive hypergraphs, facilitating the capture of more complex and robust relationships among samples.

V. CONCLUSION

This paper has introduced a new SFDA method, called HG-SFDA, that exploits high-order neighborhood relations and explicitly takes the domain shift effect into account. Specifically, we have constructed hyperedges over the target samples by considering their semantic similarity and have developed a self-loop strategy to involve the domain uncertainty of target samples in hypergraph optimization. Then we have further proposed an adaptive relation-based objective that pushes close samples in a cluster and pulls away samples in different clusters with soft attention levels. Extensive experiments conducted on mainstream datasets have demonstrated the efficacy of our method on the SFDA problem.

REFERENCES

[1] 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.

[2] J. Liang, N. Homayounfar, W.-C. Ma, Y. Xiong, R. Hu, and R. Urtasun, "PolyTransform: Deep polygon transformer for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9131–9140.

[3] S. Liu, K. Liu, W. Zhu, Y. Shen, and C. Fernandez-Granda, "Adaptive early-learning correction for segmentation from noisy annotations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2606–2616.

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 213–229.

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

[6] 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.

[7] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3071–3085, Dec. 2019.

[8] H. Tang, K. Chen, and K. Jia, "Unsupervised domain adaptation via structurally regularized deep clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8725–8735.

[9] 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.

[10] J. Gao, X. Ma, and C. Xu, "Learning transferable conceptual prototypes for interpretable unsupervised domain adaptation," *IEEE Trans. Image Process.*, vol. 33, pp. 5284–5297, 2024.

[11] Y. Fang, P.-T. Yap, W. Lin, H. Zhu, and M. Liu, "Source-free unsupervised domain adaptation: A survey," *Neural Netw.*, vol. 174, Jun. 2024, Art. no. 106230.

[12] J. Li, Z. Yu, Z. Du, L. Zhu, and H. T. Shen, "A comprehensive survey on source-free domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5743–5762, Aug. 2024.

[13] 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. 8978–8987.

[14] H. Xia, H. Zhao, and Z. Ding, "Adaptive adversarial network for source-free domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9010–9019.

[15] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6028–6039.

[16] S. Qu, G. Chen, J. Zhang, Z. Li, W. He, and D. Tao, "BMD: A general class-balanced multicentric dynamic prototype strategy for source-free domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 165–182.

[17] N. Karim, N. C. Mithun, A. Rajvanshi, H.-P. Chiu, S. Samarasekera, and N. Rahnavard, "C-SFDA: A curriculum learning aided self-training framework for efficient source free domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 24120–24131.

[18] Y. Zhang, Z. Wang, and W. He, "Class relationship embedded learning for source-free unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7619–7629.

[19] M. Litrico, A. Del Bue, and P. Morerio, "Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7640–7650.

[20] U. Hwang, J. Lee, J. Shin, and S. Yoon, "SF(DA)^2: Source-free domain adaptation through the lens of data augmentation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024, pp. 1793–1811.

[21] M. Jing, J. Li, K. Lu, L. Zhu, and H. T. Shen, "Visually source-free domain adaptation via adversarial style matching," *IEEE Trans. Image Process.*, vol. 33, pp. 1032–1044, 2024.

[22] S. Yang, J. Van de Weijer, L. Herranz, and S. Jui, "Exploiting the intrinsic neighborhood structure for source-free domain adaptation," in *Proc. Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 29393–29405.

[23] S. Yang, Y. Wang, K. Wang, and S. Jui, "Attracting and dispersing: A simple approach for source-free domain adaptation," in *Proc. NIPS*, 2022, pp. 5802–5815.

[24] L. Yi et al., "When source-free domain adaptation meets learning with noisy labels," 2023, *arXiv:2301.13381*.

[25] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "VisDA: The visual domain adaptation challenge," 2017, *arXiv:1710.06924*.

[26] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, S. Jui, and J. Yang, "Trust your good friends: Source-free domain adaptation by reciprocal neighborhood clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 15883–15895, Dec. 2023.

[27] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.

[28] 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.

[29] 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.

[30] C. Qin, H. You, L. Wang, C. J. Kuo, and Y. Fu, "PointDAN: A multi-scale 3D domain adaption network for point cloud representation," in *Proc. Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 32, 2022, pp. 7190–7201.

[31] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[32] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 97–105.

[33] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[34] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 801–814, Apr. 2019.

[35] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2016, vol. 30, no. 1, pp. 2058–2065.

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

[37] J. Na, D. Han, H. J. Chang, and W. Hwang, "Contrastive vicinal space for unsupervised domain adaptation," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 92–110.

[38] Q. Peng, Z. Ding, L. Lyu, L. Sun, and C. Chen, "RAIN: RegulArization on input and network for black-box domain adaptation," 2022, *arXiv:2208.10531*.

[39] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 2, Jul. 2015, pp. 1180–1189.

[40] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res. (JMLR)*, vol. 17, no. 59, pp. 1–35, 2016.

[41] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, vol. 32, no. 1, pp. 4058–4065.

[42] X. Li, C. Lan, G. Wei, and Z. Chen, "Semantic-aware message broadcasting for efficient unsupervised domain adaptation," *IEEE Trans. Image Process.*, vol. 33, pp. 5340–5353, 2024.

[43] Q. Liu and Z. Wang, "Collaborating domain-shared and target-specific feature clustering for cross-domain 3D action recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 137–155.

[44] Z. Deng, Y. Luo, and J. Zhu, "Cluster alignment with a teacher for unsupervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9943–9952.

[45] K.-Y. Wei and C.-T. Hsu, "Generative adversarial guided learning for domain adaptation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, p. 100.

[46] Y. Liu, Z. Zhou, and B. Sun, "COT: Unsupervised domain adaptation with clustering and optimal transport," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 19998–20007.

[47] J. Pei, Z. Jiang, A. Men, L. Chen, Y. Liu, and Q. Chen, "Uncertainty-induced transferability representation for source-free unsupervised domain adaptation," *IEEE Trans. Image Process.*, vol. 32, pp. 2033–2048, 2023.

[48] Y. Mitsuzumi, A. Kimura, and H. Kashima, "Understanding and improving source-free domain adaptation from a theoretical perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 28515–28524.

[49] A. Gretton, K. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2007, pp. 513–520.

[50] G. Kang, L. Zheng, Y. Yan, and Y. Yang, "Deep adversarial attention alignment for unsupervised domain adaptation: The benefit of target expectation maximization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 401–416.

[51] S. Purushotham, W. Carvalho, T. Nilanon, and Y. Liu, "Variational recurrent adversarial deep domain adaptation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1367–1381.

[52] Y. Liu, W. Zhang, and J. Wang, "Source-free domain adaptation for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1215–1224.

[53] W. Chen, L. Lin, S. Yang, D. Xie, S. Pu, and Y. Zhuang, "Self-supervised noisy label learning for source-free unsupervised domain adaptation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 10185–10192.

[54] C. Saltori, S. Lathuilière, N. Sebe, E. Ricci, and F. Galasso, "SF-UDA^3D: Source-free unsupervised domain adaptation for LiDAR-based 3D object detection," in *Proc. Int. Conf. 3D Vis. (3DV)*, Nov. 2020, pp. 771–780.

[55] S. Roy et al., "Uncertainty-guided source-free domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 537–555.

[56] J. Huang, D. Guan, A. Xiao, and S. Lu, "Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data," in *Proc. Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 3635–3649.

[57] J. N. Kundu et al., "Balancing discriminability and transferability for source-free domain adaptation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 11710–11728.

[58] S. Tang, A. Chang, F. Zhang, X. Zhu, M. Ye, and C. Zhang, "Source-free domain adaptation via target prediction distribution searching," *Int. J. Comput. Vis.*, vol. 132, no. 3, pp. 654–672, Mar. 2024.

[59] W. Zhang, L. Shen, and C.-S. Foo, "Rethinking the role of pre-trained networks in source-free domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 18795–18805.

[60] C. Yang, X. Guo, Z. Chen, and Y. Yuan, "Source free domain adaptation for medical image segmentation with Fourier style mining," *Med. Image Anal.*, vol. 79, Jul. 2022, Art. no. 102457.

[61] J. Hong, Y.-D. Zhang, and W. Chen, "Source-free unsupervised domain adaptation for cross-modality abdominal multi-organ segmentation," *Knowl.-Based Syst.*, vol. 250, Aug. 2022, Art. no. 109155.

[62] V. K. Kurmi, V. K. Subramanian, and V. P. Namboodiri, "Domain impression: A source data free domain adaptation method," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 615–625.

[63] Y. Hou and L. Zheng, "Visualizing adapted knowledge in domain transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13824–13833.

[64] M. Jing, X. Zhen, J. Li, and C. G. M. Snoek, "Variational model perturbation for source-free domain adaptation," in *Proc. Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2022, pp. 17173–17187.

[65] F. Wang, Z. Han, Y. Gong, and Y. Yin, "Exploring domain-invariant parameters for source free domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7151–7160.

[66] T. Chu, Y. Liu, J. Deng, W. Li, and L. Duan, "Denoised maximum classifier discrepancy for source-free unsupervised domain adaptation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2022, vol. 36, no. 1, pp. 472–480.

[67] J. N. Kundu, S. Bhambri, A. Kulkarni, H. Sarkar, V. Jampani, and R. V. Babu, "Concurrent subsidiary supervision for unsupervised source-free domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 177–194.

[68] N. Ding, Y. Xu, Y. Tang, C. Xu, Y. Wang, and D. Tao, "Source-free domain adaptation via distribution estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7212–7222.

[69] J. Tian, J. Zhang, W. Li, and D. Xu, "VDM-DA: Virtual domain modeling for source data-free domain adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3749–3760, Jun. 2022.

[70] S. Tang et al., "Nearest neighborhood-based deep clustering for source data-absent unsupervised domain adaptation," 2021, *arXiv:2107.12585*.

[71] J. Lee, D. Jung, J. Yim, and S. Yoon, "Confidence score for source-free unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 12365–12377.

[72] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng, "Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8602–8617, Nov. 2022.

[73] W. Li, M. Cao, and S. Chen, "Jacobian norm for unsupervised source-free domain adaptation," 2022, *arXiv:2204.03467*.

[74] R. Zhang, Y. Zou, and J. Ma, "Hyper-SAGNN: A self-attention based graph neural network for hypergraphs," 2019, *arXiv:1911.02613*.

[75] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2019, vol. 33, no. 1, pp. 3558–3565.

[76] Y. Gao, Y. Feng, S. Ji, and R. Ji, "HGNN+: General hypergraph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3181–3199, Mar. 2023.

[77] Y. Feng, S. Ji, Y.-S. Liu, S. Du, Q. Dai, and Y. Gao, "Hypergraph-based multi-modal representation for open-set 3D object retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 4, pp. 2206–2223, Apr. 2024.

[78] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Proc. Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2007, pp. 1601–1608.

[79] J. Jiang, Y. Wei, Y. Feng, J. Cao, and Y. Gao, "Dynamic hypergraph neural networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2635–2641.

[80] J.-G. Young, G. Petri, and T. P. Peixoto, "Hypergraph reconstruction from network data," *Commun. Phys.*, vol. 4, no. 1, p. 135, Jun. 2021.

[81] D. Yang, B. Qu, J. Yang, and P. Cudré-Mauroux, "LBSN2 Vec++: Heterogeneous hypergraph embedding for location-based social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1843–1855, Apr. 2022.

[82] S. Jin et al., "A general hypergraph learning algorithm for drug multi-task predictions in micro-to-macro biomedical networks," *PLOS Comput. Biol.*, vol. 19, no. 11, Nov. 2023, Art. no. e1011597.

[83] R. Viñas et al., "Hypergraph factorization for multi-tissue gene expression imputation," *Nature Mach. Intell.*, vol. 5, no. 7, pp. 739–753, Jul. 2023.

[84] S. Tang, W. Su, M. Ye, and X. Zhu, "Source-free domain adaptation with frozen multimodal foundation model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 23711–23720.

[85] T. Jin, Z. Yu, Y. Gao, S. Gao, X. Sun, and C. Li, "Robust ℓ_2-hypergraph and its applications," *Inf. Sci.*, vol. 501, pp. 708–723, Oct. 2019.

[86] J. Song, T. S. Kim, J. Kim, G. Nam, T. Kooi, and J. Choo, "Is user feedback always informative? Retrieval latent defending for semi-supervised domain adaptation without source data," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024, pp. 72–92.

[87] H. Xia, S. Xia, and Z. Ding, "Discriminative pattern calibration mechanism for source-free domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 23648–23658.

[88] T. Xu, W. Chen, P. Wang, F. Wang, H. Li, and R. Jin, "CDTrans: Cross-domain transformer for unsupervised domain adaptation," 2021, *arXiv:2109.06165*.

[89] S. Sanyal, A. R. Asokan, S. Bhambri, A. Kulkarni, J. N. Kundu, and R. V. Babu, "Domain-specificity inducing transformers for source-free domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 18882–18891.

[90] S. Tang, Y. Shi, Z. Song, M. Ye, C. Zhang, and J. Zhang, "Progressive source-aware transformer for generalized source-free domain adaptation," *IEEE Trans. Multimedia*, vol. 26, pp. 4138–4152, 2024.

[91] T. Sun, C. Lu, T. Zhang, and H. Ling, "Safe self-refinement for transformer-based domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7191–7200.

[92] Y. Jin, X. Wang, M. Long, and J. Wang, "Minimum class confusion for versatile domain adaptation," in *Proc. ECCV*, 2020, pp. 464–480.

[93] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*.

[94] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 30, 2022, pp. 1195–1204.

[95] D. A. ID, M. Cord, M. Douze, M. Francisco, D. A. ID, and J. Y. Herve, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 10347–10357.

[96] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[97] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732.

[98] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res. (JMLR)*, vol. 9, no. 11, pp. 1–27, 2008.

[99] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.

[100] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada, "Open set domain adaptation by backpropagation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 153–168.

[101] H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang, "Separate to adapt: Open set domain adaptation via progressive separation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2927–2936.

[102] S. Qu et al., "Upcycling models under domain and category shift," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 20019–20028.

Jinkun Jiang received the bachelor's degree in computer science and technology from Shandong University of Science and Technology in 2021 and the master's degree in computer technology from the Ocean University of China in 2025, where he is currently pursuing the Ph.D. degree with the Marine Research Group, Visual Laboratory. His research interests include computer vision and transfer learning.

Qingxuan Lv was born in Shanxi, China, in 1996. He received the bachelor's degree in computer science and technology from Shanxi University of Finance and Economics in 2018 and the master's degree in computer science and technology from the Ocean University of China (OUC) in 2021, where he is currently pursuing the Ph.D. degree with the Ocean Group, Vision Laboratory. His research interests include computer vision and machine learning. Specifically, he is interested in universal domain adaptation and semantic segmentation.

Yuezun Li (Member, IEEE) received the B.S. degree in software engineering from Shandong University in 2012, the M.S. degree in computer science in 2015, and the Ph.D. degree in computer science from University at Albany–SUNY in 2020. He was a Senior Research Scientist with the Department of Computer Science and Engineering, University at Buffalo–SUNY. He is currently a Lecturer with the Ocean University of China. His research interests include computer vision and multimedia forensics. He has been recognized on Stanford's list of the Top 2% of Scientists Worldwide since 2023 and received the 2024 ACM Qingdao Rising Star Award.

Yong Du (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from Jiangnan University, Wuxi, China, in 2011 and 2014, respectively, and the Ph.D. degree from South China University of Technology, Guangzhou, China, in 2019. He is currently an Associate Professor with the School of Computer Science and Technology, Ocean University of China. He has published a number of papers in leading conferences and journals, including CVPR, ICCV, ECCV, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IJCV. His research interests include computer vision, image processing, machine learning, and generative modeling.

Junyu Dong (Member, IEEE) received the B.Sc. and M.Sc. degrees in applied mathematics from the Department of Applied Mathematics, Ocean University of China, Qingdao, China, in 1993 and 1999, respectively, and the Ph.D. degree in image processing from the Department of Computer Science, Heriot-Watt University, Edinburgh, U.K., in November 2003. He is currently a Professor with the Minister of Faculty of Information Science and Engineering, Ocean University of China. His research interests include machine learning, big data, computer vision, and underwater image processing.

Sheng Chen (Life Fellow, IEEE) received the B.Eng. degree in control engineering from East China Petroleum Institute, Dongying, China, in 1982, the Ph.D. degree in control engineering from City University, London, in 1986, and the D.Sc. degree from the University of Southampton, Southampton, U.K., in 2005. From 1986 to 1999, he held research and academic appointments with The University of Sheffield, U.K, The University of Edinburgh, U.K, and the University of Portsmouth, U.K. Since 1999, he has been with the School of Electronics and Computer Science, University of Southampton, where he is currently a Professor of intelligent systems and signal processing. He has authored more than 700 research articles. He has more than 22 000 Web of Science citations with H-index 64 and more than 42 000 Google Scholar citations with H-index 87. His research interests include adaptive signal processing, wireless communications, modeling and identification of nonlinear systems, neural network and machine learning, intelligent control system design, evolutionary computation methods, and optimization. He was elected as a fellow of the Royal Academy of Engineering, U.K., in 2014. He is a fellow of Asia–Pacific Artificial Intelligence Association (FAAIA) and IET. He was one of the 200 original ISI Highly Cited Researcher in engineering in March 2004.

Hui Yu (Senior Member, IEEE) received the Ph.D. degree from Brunel University London in 2009. He is currently a Professor of visual and cognitive computing and leads the Visual Computing and Social Robotics Group with the University of Glasgow, U.K. His research interests include visual and cognitive computing as well as machine learning with applications to image/video understanding, 4D facial analysis, social robotics, and intelligent vehicle. He has been awarded the Industrial Fellowship by the Royal Academy of Engineering. He serves as the Associate Vice President for the IEEE Systems, Man and Cybernetics Society, and an Associate Editor for IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, and IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS journal.