Progressive Graph Learning for Open-Set Domain Adaptation

Yadan Luo [* 1]   Zijian Wang [* 1]   Zi Huang [1]   Mahsa Baktashmotlagh [1]

Abstract

Domain shift is a fundamental problem in visual recognition which typically arises when the source and target data follow different distributions. The existing domain adaptation approaches which tackle this problem work in the *closed-set* setting with the assumption that the source and the target data share exactly the same classes of objects. In this paper, we tackle a more realistic problem of *open-set* domain shift where the target data contains additional classes that are not present in the source data. More specifically, we introduce an end-to-end Progressive Graph Learning (PGL) framework where a graph neural network with episodic training is integrated to suppress underlying conditional shift and adversarial learning is adopted to close the gap between the source and target distributions. Compared to the existing open-set adaptation approaches, our approach guarantees to achieve a tighter upper bound of the target error. Extensive experiments on three standard open-set benchmarks evidence that our approach significantly outperforms the state-of-the-arts in open-set domain adaptation.

1. Introduction

While deep learning has made remarkable advances across a wide variety of machine-learning tasks and applications, it is commonly assumed that the training and test data are drawn from the same distribution. In practice, however, this assumption can be violated due to a number of factors, such as the change of lighting conditions, background, environment, or data modalities, which is referred to as the *domain shift* problem.

Unsupervised Domain Adaptation (UDA) approaches tackle the domain shift problem by aligning the training (source)

---

*Equal contribution [1]School of Information Technology and Electrical Engineering, The University of Queensland, Australia. Correspondence to: Yadan Luo <lyadanluol@gmail.com>.

and test (target) distributions, and can be roughly divided into statistical matching (Baktashmotlagh et al., 2013; 2014; 2016; Tzeng et al., 2014; Long et al., 2013; 2015; Ganin et al., 2016; Tzeng et al., 2017; Li et al., 2019) or adversarial learning (Ganin et al., 2016; Tzeng et al., 2017; Long et al., 2017; Ghifary et al., 2016) methods. Theoretical analysis of UDA approaches has been widely studied (Ben-David et al., 2006; Mansour et al., 2009; Zhang et al., 2019), which provides rigorous error bounds on the target data.

Existing UDA algorithms are developed under the assumption that the source and target domains share an identical group of classes. Such a scenario typically refers to a *closed-set* setting, which could be hardly guaranteed in real-world applications. Therefore, a more realistic Open-set adaptation setting has been introduced recently (Saito et al., 2018) which allows the target data to contain an additional "unknown" category, covering all irrelevant classes not present in the source domain.

The core idea of unsupervised open-set domain adaptation (OUDA) approaches (Busto & Gall, 2017; Baktashmotlagh et al., 2019; Saito et al., 2018; Liu et al., 2019; Feng et al., 2019) is to learn a classifier from a larger hypothesis space for both shared and unknown classes in the source and target domains. According to (Ben-David et al., 2006; Mansour et al., 2009), the target error is bounded by the source risk, discrepancy distance across the domains, the shared error coming from the conditional shift (Zhao et al., 2019), and the open-set risk. Open-set risk contributes the most to the error bound when a large percentage of data is unknown.

While promising, existing OUDA approaches (Busto & Gall, 2017; Saito et al., 2018; Baktashmotlagh et al., 2019; Liu et al., 2019; Feng et al., 2019) lack an essential theoretical analysis of the aforementioned partial risks and the upper bound for the target risk, thus omitting potential solutions for improvement and leading to a biased solution. With the aim of minimising the aforementioned partial risks and achieving a tighter error bound for open-set adaptation, we combine the following four strategies in an end-to-end progressive learning framework:

1. To suppress the source risk, we decompose the original hypothesis space $\mathcal{H}$ into two subspaces $\mathcal{H}_1$ and $\mathcal{H}_2$, where $\mathcal{H}_1$ includes classifiers for the shared classes of the source and target domains and $\mathcal{H}_2$ is specific

to classifying unknowns in the target domain. With a restricted size of the subspace $\mathcal{H}_1$, the possibility of misclassifying source data as unknowns will be reduced.

2. To control the open-set risk, we adopt the progressive learning paradigm (Bengio et al., 2009), where the target samples with low classification confidence are gradually rejected from the target domain and inserted as the pseudo-labeled unknown set in the source domain. This mechanism suppresses the potential negative transfer where the private representations across domains are falsely aligned.

3. We address conditional shift (Zhao et al., 2019) at the both sample- and manifold-level in a transductive setting. Specifically, we design an episodic training scheme and align conditional distributions across domains by gradually replacing the source data with the pseudo-labeled known data in each episode. We learn class-specific representations by aggregating the source and target features and passing episodes through deep graph neural networks.

4. Our algorithm is seamlessly equipped with an adversarial domain discriminator, which effectively closes the gap between the source and target marginal distributions for the known categories.

We applied our method on three challenging open-set object recognition benchmarks, *i.e.*, the *Office-Home*, *VisDA-17*, and *Syn2Real-O*, to confirm its superiority to the existing state-of-the-art open-set domain adaptation approaches.

2. Preliminaries

In this section, we introduce the notations, problem settings and the theoretical definitions and analysis for the tasks of closed-set and open-set unsupervised domain adaptation.

Definition 1. *Closed-set Unsupervised Domain Adaptation (UDA).* *Let $\mathbb{P}^s$ and $\mathbb{Q}^t$ be the distributions of the source domain and the target domain, respectively. The corresponding label spaces for both domains are equal, i.e., $\mathcal{Y}_s = \mathcal{Y}_t = \{1, \ldots, C\}$, where C is the number of classes. The ultimate goal is to learn an optimal classifier $h \in \mathcal{H}$ for the target domain $h : \mathcal{X}_t \to \mathcal{Y}_t$, based on the labeled source data and the unlabeled target data, where $\mathcal{H}$ is the hypothesis space of classifiers.*

Definition 2. *Open-set Unsupervised Domain Adaptation (OUDA) (Saito et al., 2018).* *Assume that we have the labeled source data $D_s = \{(x_{s_i}, y_{s_i})\}_{i=1}^{n_s} \sim \mathbb{P}^s$ and unlabeled target data $D_t = \{x_{t_j}\}_{j=1}^{n_t} \sim \mathbb{Q}_X^t$, where $\mathbb{P}^s$ being the joint probability distribution of the source domain, $\mathbb{Q}_X^t$ being the marginal distribution of the target domain,*

with n_s and n_t indicating the size of source and target dataset respectively. With training samples drawn i.i.d from both domains, the goal is to learn an optimal target classifier $h : \mathcal{X}_t \to \mathcal{Y}_t$. Here the target label space $\mathcal{Y}_t = \{\mathcal{Y}_s, unk\} = \{1, \ldots, C + 1\}$ includes the additional unknown class $C + 1$, which is not present in the source label space $\mathcal{Y}_s$.

The source risk $R_s(h)$ and target risk $R_t(h)$ of a classifier $h \in \mathcal{H}$ with respect to the source distribution $\mathbb{P}^s$ and the target distribution $\mathbb{Q}^t$ are given by,

$$R_s(h) = \mathbb{E}_{(x,y)\sim\mathbb{P}^s}\mathcal{L}(h(x), y) = \sum_{i=1}^{C} \pi_i^s R_{s,i}(h),$$

$$R_t(h) = \mathbb{E}_{(x,y)\sim\mathbb{Q}^t}\mathcal{L}(h(x), y) = \sum_{i=1}^{C+1} \pi_i^t R_{t,i}(h), \quad (1)$$

where $\pi_i^s = \mathbb{P}^s(y = i)$ and $\pi_i^t = \mathbb{Q}^t(y = i)$ are class-prior probabilities of the source and target distributions, respectively. The bounded loss function $\mathcal{L} : \mathcal{Y}_t \times \mathcal{Y}_t \to \mathbb{R}$ satisfies symmetry and triangle inequality. Particularly, the partial risk $R_{s,i}(h)$ and $R_{t,i}(h)$ can be defined as,

$$R_{s,i}(h) = \mathbb{E}_{x\sim\mathbb{P}^s(x|i)}\mathcal{L}(h(x), i),$$
$$R_{t,i}(h) = \mathbb{E}_{x\sim\mathbb{Q}^t(x|i)}\mathcal{L}(h(x), i). \quad (2)$$

Before introducing the generalization bound for open-set domain adaptation, it is crucial to define a discrepancy measure between the source and target domains:

Definition 3. *Discrepancy Distance (Mansour et al., 2009).* *For any $h, h' \in \mathcal{H}$, the discrepancy between the distributions of the source and target domains can be formulated as:*

$$disc(\mathbb{P}^s, \mathbb{Q}^t) = \sup_{h,h'\in\mathcal{H}} |\mathbb{E}_{\mathbb{P}^s}\mathcal{L}(h, h') - \mathbb{E}_{\mathbb{Q}^t}\mathcal{L}(h, h')|. \quad (3)$$

Theorem 2.1. *Open-set Domain Adaptation Upper Bound (Fang et al., 2019).* *Given the hypothesis space $\mathcal{H}$ with a mild condition that constant function $C + 1 \in \mathcal{H}$, for $\forall h \in \mathcal{H}$, the expected error on target samples $R_t(h)$ is bounded as,*

$$\frac{R_t(h)}{1 - \pi_{C+1}^t} \leq R_s(h) + disc(\mathbb{Q}_{X|Y\leq C}^t, \mathbb{P}_X^s) + \lambda$$

$$+ \underbrace{\frac{\pi_{C+1}^t}{1 - \pi_{C+1}^t} R_{t,C+1}(h)}_{\text{open set risk } \Delta_o}, \quad (4)$$

where the shared error $\lambda = \min_{h\in\mathcal{H}} \frac{R_t^*(h)}{1 - \pi_{C+1}^t} + R_s(h)$. The proof can be founded in the supplementary material.

Remark. *To compute the error upper bound for the closed-set unsupervised domain adaptation, Theorem 2.1 can be*

reduced to:

$$R_t(h) \le R_s(h) + disc(\mathbb{Q}_X^t, \mathbb{P}_X^s) + \lambda', \quad (5)$$

where $\pi_{C+1}^t = 0$ *and* $\lambda' = \min_{h \in \mathcal{H}} R_t(h) + R_s(h)$.

According to Equation (4), the target error is bounded by four terms, which opens four directions for improvement:

- Source risk $R_s(h)$. Assuming that source domain does not include any unknown samples, a part of the source risk can be avoided, which in turn minimizes the error upper bound. This direction is rarely investigated in the existing literature of open-set domain adaptation.

- Discrepancy distance $disc(\mathbb{Q}_{X|Y \le C}^t, \mathbb{P}_X^s)$. Minimizing the discrepancy distance between the source and the target domains has been well investigated in recent years in statistics-based (Gretton et al., 2006) or adversarial-based approaches (Ganin et al., 2016).

- Shared error λ of the joint ideal hypothesis h^*. λ tends to be large when the conditional shift encountered, where the class-wise conditional distributions are not aligned even with marginal distribution aligned.

- Open set risk Δ_o. When a large percentage of data is unknown ($\pi_{C+1}^t \to 1$), this term contributes the most to the error bound. As shown in Equation (4), it can be interpreted as the mis-classification rate for the unknown samples.

3. Progressive Open-Set UDA

Aiming to minimise the four partial risks mentioned above, we reformulate the open-set unsupervised domain adaptation in a progressive way, and as such, we redefine the task at hand as follows.

Definition 4. *Progressive Open-Set Unsupervised Domain Adaptation (POUDA). Given the labeled source data* $D_s = \{(x_{s_i}, y_{s_i})\}_{i=1}^{n_s} \sim \mathbb{P}^s$ *and unlabeled target data* $D_t = \{x_{t_j}\}_{j=1}^{n_t} \sim \mathbb{Q}_X^t$, *the main goal is to learn an optimal target classifier* $\tilde{h} \in \mathcal{H}_1 \subset \mathcal{H}$ *for the shared classes* $Y_s = Y_t^* = \{1, \dots, C\}$ *and a pseudo-labeling function* $h_b \in \mathcal{H}_2 \subset \mathcal{H}$ *for the unknown class* $C + 1$.

Assume the target set will be pseudo-labeled through M *steps, thereby the enlarging factor for each step is defined as* $\alpha = \frac{1}{M}$. *As long as the hypothesis* $\tilde{h}$ *and* h_b *share the same feature extraction part, we can decompose the shared hypothesis* $\tilde{h}$ *into* $\tilde{h}(x) = \arg\max_{i \in Y_s} p(i|x)$ *and define the pseudo-labeling function* h_b *at the* m-*th step in line with* $\tilde{h}$'s *prediction:*

$$h_b^{(m)} = \begin{cases} C+1, & \text{if } \text{rank}(\max_{i \in Y_s} p(i|x)) \le \alpha_u^{(m)}, \\ \tilde{y}, & \text{if } \text{rank}(\max_{i \in Y_s} p(i|x)) \ge \alpha_k^{(m)}, \end{cases} \quad (6)$$

with $\alpha_u^{(m)} = \beta.\alpha.m.n_t$ *and* $\alpha_k^{(m)} = n_t - (1 - \beta).\alpha.m.n_t$ *being the index-based threshold to classify the unknown and known samples. The hyperparameter* $\beta \in (0,1)$ *measures the openness of the given target set as the ratio of unknown samples.* rank$(\cdot)$ *is a global ranking function which ranks predicted probabilities in ascending order and returns the sorted index list as an output. The pseudo-labeling function gives* $\tilde{y} = \tilde{h}(x)$ *for the possible known samples, and* $C + 1$ *for the unknown ones.*

In our case, the upper bound of expected target risk is formulated in the following theorem,

Theorem 3.1. *POUDA Error Bound. Given the hypothesis space* $\mathcal{H}_1, \mathcal{H}_2 \subset \mathcal{H}$, $\exists \alpha^* \in (0,1)$, *for* $\forall \tilde{h} \in \mathcal{H}_1$ *and* $\forall h_b \in \mathcal{H}_2$, *with a condition that the openness* β *of the target set is fixed, the expected error* $R_t(\tilde{h}, h_b)$ *on the target samples is bounded as:*

$$\frac{R_t(\tilde{h}, h_b)}{1 - \pi_{C+1}^t} \le (1 - \pi_\alpha)(R_s(\tilde{h}) + disc(\mathbb{Q}_{X|Y \le C}^t, \mathbb{P}_X^s)) + \lambda$$

$$+ \underbrace{\frac{\pi_\alpha \pi_{C+1}^t}{1 - \pi_{C+1}^t} R_{t,C+1}(h_b)}_{\text{progressive open set risk } \tilde{\Delta}_o} + const,$$

(7)

where the shared error $\lambda = \min_{\tilde{h} \in \mathcal{H}_1} \frac{R_t^*(\tilde{h})}{1 - \pi_{C+1}^t} + (1 - \pi_\alpha) R_s(\tilde{h})$ *and* π_α *indicates the probability that target samples being pseudo-labeled by* h_b *(refer to the supplementary material for proof).*

Remark. *For* $\tilde{h} \in \mathcal{H}_1 \subset \mathcal{H}$ *and* $h_b \in \mathcal{H}_2 \subset \mathcal{H}$, *the following inequality holds,*

$$\sup_{\tilde{h} \in \mathcal{H}_1} R_s(\tilde{h}) \le \sup_{h \in \mathcal{H}} R_s(h),$$

$$\sup_{h_b \in \mathcal{H}_2} \frac{\pi_{C+1}^t}{1 - \pi_{C+1}^t} R_{t,C+1}(h_b) \le \sup_{h \in \mathcal{H}} \frac{\pi_{C+1}^t}{1 - \pi_{C+1}^t} R_{t,C+1}(h).$$

(8)

We can observe that our progressive learning framework can achieve a tighter upper bound compared with conventional open-set domain adaptation framework.

4. Methodology

In this section, we go through the details of the proposed Progressive Graph Learning (PGL) framework as illustrated in Figure 1. Our approach is mainly motivated by the two aspects of alleviating the shared error λ and effectively controlling the progressive open-set risk $\tilde{\Delta}_o$.

Minimizing the shared error λ. Conditional shift (Zhao et al., 2019) is the most significant obstacle for finding a
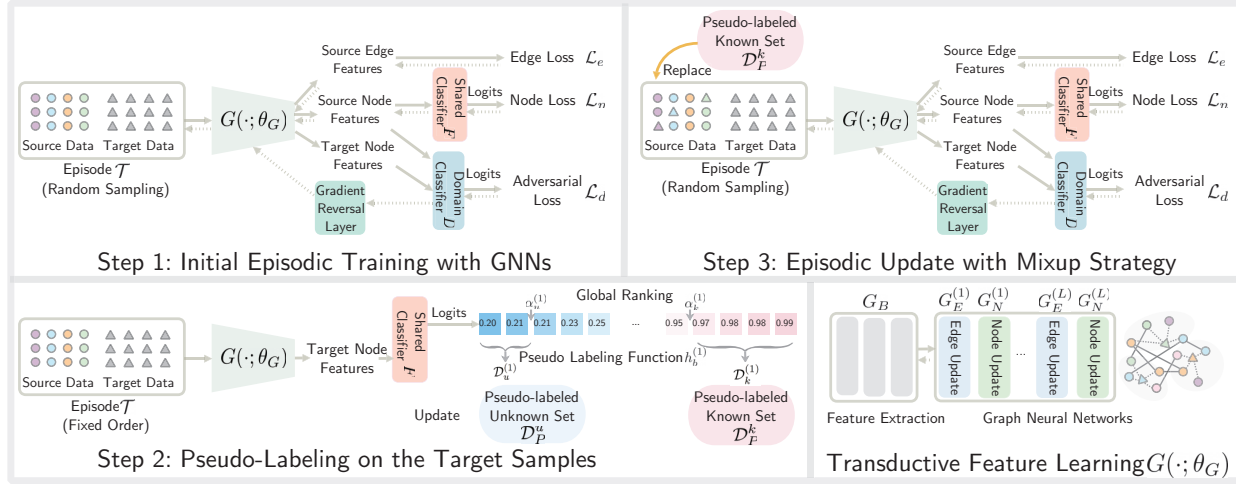
Figure 1. Proposed PGL framework. Circles indicate the source data and triangles are the target data. Different colors indicate different classes. By alternating between *Steps 2* and *3*, we progressively achieve the optimal classification model $F \circ G$ for the shared classes and pseudo-labeling function h_b for rejecting the unknowns.

joint ideal classifier for the source and target data, which arises when the class-conditional distributions of the input features substantially differ across the domains. That means, with unaligned distributions of the source distribution $\mathbb{P}^s_{X|Y}$ and target distribution $\mathbb{Q}^t_{X|Y \leq C}$, there is no guarantee to find an ideal shared classifier for both domains. Therefore, we address the conditional shift in a transductive setting from two perspectives:

- *Sample-level*: Motivated by (Vinyals et al., 2016; Snell et al., 2017), we adopt the episodic training scheme (Section 4.1), and leverage the source samples from each class to "support" predictions on unlabeled data in each episode. With an enlarging labeled set through pseudo-labeling (Section 4.2), we progressively update training episodes by replacing the source samples with pseudo-labeled target samples (Section 4.3).

- *manifold-level*: To regularize the class-specific manifold, we construct L-layer Graph Neural Networks (GNNs) on top of the backbone network $G_B(\cdot; \theta_B)$ (*e.g.*, ResNet), which consists of paired node update networks $G_N(\cdot; \theta_N)$ and edge update networks $G_E(\cdot; \theta_E)$. The source nodes and pseudo-labeled target nodes from the same class are densely connected, aggregating information though multiple layers.

Controlling progressive open-set risk $\tilde{\Delta}_o$. As discussed in Section 4.2, we iteratively squeeze the index-based thresholds $\alpha_u^{(m)}$ and $\alpha_k^{(m)}$ to approximate the optimal threshold α^* as illustrated in Figure 2. Since the thresholds are mainly determined by the enlarging factor α, we can always seek a proper value of α to alleviate the mis-classification error and the subsequent negative transfer. Our experimental

results characterize the trade-off between computational complexity and performance improvement.

4.1. Initial Episodic Training with GNNs

Firstly, we denote the initial episodic formulation of a batch input as $\mathcal{T}^{(0)} = \{\mathcal{T}_s^{(0)}, \mathcal{T}_t^{(0)}\} = \{\tau_{s,i}^{(0)}, \tau_{t,i}^{(0)}\}_{i=1}^B$, with B as the batch size. Each episode in the batch consists of two parts, *i.e.*, the source episode $\tau_{s,i}^{(0)} = \{(x_i, y)\}_{i=1}^C \sim \mathbb{P}^s_{X|Y}$ randomly sampled from each class $c \in Y_s$ and the target episode $\tau_{t,i}^{(0)} = \{x_j\}_{j=C+1}^{2C} \sim \mathbb{Q}^t_X$ randomly sampled from the target set. All instances in a mini-batch can form an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each vertex $v_i \in \mathcal{V}$ is associated with a source or a target feature, and the edge $e_{ij} \in \mathcal{E}$ between nodes v_i and v_j measures the node affinity. The integrated GNNs are naturally able to perform a transductive inference taking advantage of labeled source data and unlabeled target data. The propagation rule for edge update and node update is elaborated in the following subsections.

4.1.1. EDGE UPDATE

The generic propagation rule for normalized edge features at the l-th layer can be defined as,

$$
\begin{aligned}
A_{ij}^{(l)} &= \sigma\big(G_E^{(l)}(\|v_i^{(l-1)} - v_j^{(l-1)}\|; \theta_E^{(l)})\big), \\
\mathcal{E}^{(l)} &= D^{-\frac{1}{2}}(A^{(l)} + I)D^{-\frac{1}{2}},
\end{aligned}
\tag{9}
$$

with σ being the sigmoid function, D the degree matrix of $A^{(l)} + I$, I the identity matrix, and $G_E^{(l)}(\cdot; \theta_E^{(l)})$ the non-linear network parameterized by θ_E.

4.1.2. NODE UPDATE

Similarly, the propagation rule for node features at the l-layer is defined as,

$$\hat{v}_i^{(l-1)} = \sum_{j \in \mathcal{N}(i)} (v_i^{(l-1)} \mathcal{E}_{ij}^{(l-1)}),$$
$$v_i^{(l)} = G_N^{(l)}([v_i^{(l-1)}; \hat{v}_i^{(l-1)}]; \theta_N^{(l)}), \qquad (10)$$

with $\mathcal{N}(i)$ being the neighbor set of the node v_i, $[\cdot;\cdot]$ the concatenation operation and $G_N^{(l)}(\cdot; \theta_N^{(l)})$ the network consisting of two convolutional layers, LeakyReLU activations and dropout layers. The node embedding is initialized with the extracted representations from the backbone embedding model, *i.e.*, $v_i^{(0)} = G_B(x_i)$.

4.1.3. JOINT OPTIMIZATION

Adaptive Learning. We exploit adversarial loss to align the distributions of the source and target features extracted from the backbone network $G_B(\cdot; \theta_B)$. Specifically, a domain classifier $D(\cdot; \theta_D)$ is trained to discriminate between the features coming from the source or target domains, along with a generator G_B to fool the discriminator D. The two-player minimax game shown in Eq.(11) is expected to reach an equilibrium resulting in the domain invariant features:

$$\mathcal{L}_d = \mathbb{E}_{x \sim \mathcal{T}_s} \log[D(G_B(x))] + \mathbb{E}_{x \sim \mathcal{T}_t} \log[1 - D(G_B(x))].$$

Node Classification. By decomposing the shared hypothesis $\tilde{h}$ into a feature learning module $G(\cdot, \theta_G)$ and a shared classifier $F(\cdot, \theta_F)$, we train the both networks to classify the source node embedding. To alleviate the inherent class imbalance issue, we adopt the focal loss to down-weigh the loss assigned to correctly-classified examples:

$$\mathcal{L}_n = -\mathbb{E}_{(x,y) \sim \mathcal{T}_s} \sum_{l=1}^{L} (1 - F(G(x)^{(l)}))^\rho \log[F(G(x)^{(l)})],$$

with the hyperparameter $\rho = 2$ and $G(x)^{(l)}$ being the node embedding from the l-th node update layer. The total loss combines all losses from L layers to improve the gradient flow in the lower layers.

Edge Classification. Based on the given labels of the source data, we construct the ground-truth of edge map $\widehat{Y}$, where $\widehat{Y}_{ij} = 1$ if x_i and x_j belong to the same class, and $\widehat{Y}_{ij} = 0$ otherwise. The networks are trained by minimizing the following binary cross-entropy loss:

$$\mathcal{L}_e = -\mathbb{E}_{(x,y) \sim \mathcal{T}_s} \sum_{l=1}^{L} \widehat{Y} \log \mathcal{E}^{(l)} + (1 - \widehat{Y}) \log[1 - \mathcal{E}^{(l)}].$$

Final Objective Function. Formally, our ultimate goal is to learn the optimal parameters for the proposed model,

$$(\theta_N^*, \theta_E^*, \theta_F^*, \theta_D^*) = \arg\min \mathcal{L}_n + \mu \mathcal{L}_e + \gamma \mathcal{L}_d,$$
$$(\theta_B^*) = \arg\min \mathcal{L}_n + \mu \mathcal{L}_e - \gamma \mathcal{L}_d, \quad (11)$$
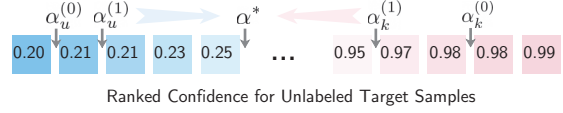

Figure 2. An illustration of the progressive learning to construct the pseudo-labeled target set. α^* indicates the ideal threshold for classifying known and unknown samples.

with μ and γ the coefficients of the edge loss and adversarial loss, respectively.

4.2. Pseudo-Labeling in Progressive Paradigm

With the optimal model parameters obtained at the m-th step, we freeze the model and feed all the target samples in the forward direction, as shown in the *Step 2* of Figure 1. Then, we rank the maximum likelihood $\max_{i \in Y_s} p(i|x)$ produced from the shared classifier $F(G(x)^{(L)})$ in an ascending order. Giving priority to the "easier" samples with relatively high/low confidence scores, we select $\alpha.m.n_t$ samples to enlarge the pseudo-labeled known set $\mathcal{D}_P^k$ and known set $\mathcal{D}_P^u$ (Refer to Eq. (6)):

$$\mathcal{D}_P^k \leftarrow \mathcal{D}_k^{(0)} \cup \mathcal{D}_k^{(1)} \ldots \cup \mathcal{D}_k^{(m)},$$
$$\mathcal{D}_P^u \leftarrow \mathcal{D}_u^{(0)} \cup \mathcal{D}_u^{(1)} \ldots \cup \mathcal{D}_u^{(m)},$$
$$\mathcal{D}_k^{(m)} = \{(x_i, \tilde{y}_i)\}_{i=1}^{(1-\beta).\alpha.m.n_t},$$
$$\mathcal{D}_u^{(m)} = \{(x_j, C+1)\}_{j=1}^{\beta.\alpha.m.n_t}. \quad (12)$$

Note that $\mathcal{D}_k^{(m)}$ and $\mathcal{D}_u^{(m)}$ are newly annotated known set and unknown set, respectively and the pseudo-label is given by $\tilde{y}_i = \arg\max_{y \in Y_s} p(y|x)$. To find a proper value of enlarging factor α, we have two options: by aggressively setting a large value to α, the progressive paradigm can be accomplished in fewer steps resulting in potentially noisy and unreliable pseudo-labeled candidates; on the contrary, choosing a small value of α can result in a steady increase of the model performance and the computational cost.

4.3. Episodic Update with Mix-up Strategy

We mix the source data with the samples from the updated pseudo-labeled known-set $\mathcal{D}_P^k$ at the m-th step, and construct new episodes $\mathcal{T}^{(m+1)}$ at the $(m+1)$-th step, as depicted in the *Step 3* of Figure 1. In particular, We randomly replace the source samples with pseudo-labeled known data with a probability $\mathbb{P}_r^{(m)} = m\alpha$. Each episode in the new batch consists of three parts,

$$\tau_{s,i}^{(m+1)} = \{(x_i, y_i)\}_{i=1}^{C \times (1-\mathbb{P}_r^{(m)})} \sim \mathbb{P}^s(x|y),$$
$$\tilde{\tau}_{t,i}^{(m+1)} = \{(x_k, \tilde{y}_k)\}_{k=1}^{C \times \mathbb{P}_r^{(m)}} \sim \mathbb{Q}^t(x|\tilde{y}),$$
$$\tau_{t,i}^{(m+1)} = \{x_j\}_{j=C+1}^{2C} \sim \mathbb{Q}_X^t, \quad (13)$$

with $\mathbb{Q}^t(x|\tilde{y})$ being the conditional distribution of the pseudo-labeled known set at the m-th step. Then, we update the model parameters according to Equation (11) and repeat pseudo-labeling with the newly constructed episodes until convergence.

5. Experiments

In this section, we quantitatively compare our proposed model against various domain adaptation baselines on the *Office-Home*, *VisDA-17* and *Syn2Real-O* datasets. The baselines include four open-set domain adaptation methods of **ATI-λ** (Busto & Gall, 2017), **OSBP** (Saito et al., 2018), **STA** (Liu et al., 2019), **DAOD** (Feng et al., 2019); two closed-set domain adaptation methods of **MMD** (Gretton et al., 2006), **DANN** (Ganin et al., 2016), and a basic **ResNet-50** (He et al., 2016) deep classification model. To be able to apply the closed-set baseline methods (ATI-λ, MMD, DANN, ResNet-50) in the open-set setting, we follow the previous baselines (Liu et al., 2019; Feng et al., 2019) and reject unknown outliers from the target data using **OSVM** (Jain et al., 2014) and **OSNN** (Júnior et al., 2017).

Evaluation Metrics: To evaluate the proposed method and the baselines, we utilize three widely used measures (Saito et al., 2018; Liu et al., 2019), *i.e.*, accuracy (**ALL**), normalized accuracy for all classes (**OS**) and normalized accuracy for the known classes only (**OS***):

$$\text{ALL} = \frac{|x : x \in \mathcal{D}_t \wedge h(x) = y|}{|(x, y) : (x, y) \in \mathcal{D}_t|},$$

$$\text{OS} = \frac{1}{C+1} \sum_{i=1}^{C+1} \frac{|x : x \in \mathcal{D}_t^i \wedge h(x) = i|}{|x : x \in \mathcal{D}_t^i|},$$

$$\text{OS}^* = \frac{1}{C} \sum_{i=1}^{C} \frac{|x : x \in \mathcal{D}_t^i \wedge h(x) = i|}{|x : x \in \mathcal{D}_t^i|}, \tag{14}$$

with $\mathcal{D}_t^i$ being the set of target samples in the i-th class, and $h(\cdot)$ the classifier. In our case, we use the shared classifier $\tilde{h}$ for the known classes and pseudo-labeling function h_b for the unknown one.

Implementation Details: PyTorch implementation of our approach is avaibale in an annonymized repository[1]. In our experiments, we employ ResNet-50 (He et al., 2016) or VGGNet (Simonyan & Zisserman, 2015) pre-trained on ImageNet as the backbone network. For VGGNet, we only fine-tune the parameters in FC layers. The networks are trained with the ADAM optimizer with a weight decay of 5×10^{-5}. The learning rate is initialized as 1×10^{-4} and 1×10^{-5} for the GNNs and the backbone module respectively, and then decayed by a factor of 0.5 every 4 epochs. The

[1]https://github.com/BUserName/PGL

Table 1. The detailed settings for the three datasets used in our experiments.

Dataset	B	Node Features	Edge Features	α	β
Office-Home	2	512-D	512-D	0.05	0.6
VisDA-17	8	1,024-D	1,024-D	0.05	0.85
Syn2Real-O	6	1,024-D	1,024-D	0.05	0.9

dropout rate is fixed to 0.2 and the depth of GNN L is set to 1 for all experiments. The loss coefficients γ and μ are empirically set to 0.4 and 0.3, respectively. The detailed experimental settings for the three datasets are summarized in Table 1, where B being the batch size, α the enlarging factor, β the hyperparameter for balancing the openness. The image feature extracted by the fc7 layer of VGGNet backbone is a 4096-D vector, and the deep feature extracted from the ResNet-50 is a 2048-D vector.

5.1. Datasets

Office-Home (Venkateswara et al., 2017) is a challenging domain adaptation benchmark, which comprises 15,500 images from 65 categories of everyday objects. The dataset consists of 4 domains: Art (**Ar**), Clipart (**Cp**), Product (**Pr**), and Real-World (**Rw**). Following the same splits used in (Liu et al., 2019), we select the first 25 classes in alphabetical order as the known classes, and group the rest of the classes as the unknown.

VisDA-17 (Peng et al., 2017) is a cross-domain dataset with 12 categories in two distinct domains. The **Synthetic** domain consists of 152,397 synthetic images generated by 3D rendering and the **Real** domain contains 55,388 real-world images from MSCOCO (Lin et al., 2014) dataset. Following the same protocol used in (Saito et al., 2018; Liu et al., 2019), we construct the known set with 6 categories and group the remaining 6 categories as the unknown set.

Syn2Real-O (Peng et al., 2018) is the most challenging synthetic-to-real testbed, which is constructed from the *VisDA-17*. The Syn2Real-O dataset significantly increases the openness to 0.9 by introducing additional unknown samples in the target domain. According to the official setting, the **Synthetic** source domain contains training data from the *VisDA-17* as the known set, and the target domain **Real** includes the test data from the *VisDA-17* (known set) plus 50k images from irrelevant categories of MSCOCO dataset (unknown set).

5.2. Results and Analysis

As reported in Table 2, Table 3, and Table 4, we clearly observe that our method **PGL** consistently outperforms the state-of-the-art results, improving mean accuracy (OS*) by 8.1%, 28.0% and 29.6% on the benchmark datasets of

Table 2. Recognition accuracies (%) on 12 pairs of source/target domains from *Office-Home* benchmark using ResNet-50 as the backbone. **Ar**: Art, **Cp**: Clipart, **Pr**: Product, **Rw**: Real-World. * indicates our re-implementation with the officially released code.

Method	Ar→Cl		Ar→Pr		Ar→Rw		Cl→Rw		Cl→Pr		Cl→Ar		Pr→Ar		Pr→Cl		Pr→Rw		Rw→Ar		Rw→Cl		Rw→Pr		Avg.	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
ResNet+OSNN	33.7	32.1	40.6	39.4	57.0	56.6	47.7	46.9	40.3	39.1	34.0	32.3	39.7	38.5	36.3	35.0	59.7	59.6	52.1	51.4	39.2	38.0	59.2	59.2	45.0	44.0
ResNet+OSVM	37.5	38.7	42.2	42.6	49.2	51.4	53.8	55.5	48.5	50.0	39.2	40.3	53.4	55.1	43.5	44.8	70.6	72.9	65.6	67.4	49.5	50.8	72.7	75.1	52.1	53.7
DANN+OSVM	52.3	52.1	71.3	72.4	82.3	83.8	73.2	74.5	62.8	64.1	61.4	62.3	63.5	64.5	46.0	46.3	77.2	78.3	70.5	71.3	55.5	56.2	79.1	80.7	66.2	67.2
ATI-λ+OSNN	53.1	54.2	68.6	70.4	77.3	78.1	74.3	75.3	66.7	68.3	57.8	59.1	61.2	62.6	53.9	54.1	79.9	81.1	70.0	70.8	55.2	55.4	78.3	79.4	66.4	67.4
OSBP	56.1	57.2	75.8	77.8	83.0	85.4	75.5	77.2	69.2	71.3	64.6	65.9	64.6	65.3	48.3	48.7	79.5	81.6	72.1	73.5	54.3	55.3	80.2	81.9	68.6	70.1
STA*	58.1	-	71.6	-	85.0	-	75.8	-	69.3	-	63.4	-	65.2	-	53.1	-	80.8	-	74.9	-	54.4	-	81.9	-	69.5	-
STA*	46.6	45.9	67.0	67.2	76.2	76.6	64.9	65.2	57.7	57.6	50.2	49.3	49.5	48.4	42.9	40.8	76.6	77.3	68.7	68.6	46.0	45.4	73.9	74.5	60.0	59.8
DAOD	56.1	55.5	69.1	69.2	78.7	79.3	77.3	78.2	69.6	70.2	62.6	62.9	66.8	67.7	**59.7**	**60.3**	83.3	85.0	72.3	73.2	59.9	60.4	81.8	82.8	69.8	70.4
PGL	**61.6**	**63.3**	**77.1**	**78.9**	**85.9**	**87.7**	**82.8**	**85.9**	**72.0**	**73.9**	**68.8**	**70.2**	**72.2**	**73.7**	58.4	59.2	82.6	84.8	**78.6**	**81.5**	**65.0**	**68.8**	**83.0**	**84.8**	**74.0**	**76.1**

Table 3. Recognition accuracies (%) for open-set domain adaptation experiments on the *Syn2Real-O* (ResNet-50).

Method	Aer	Bic	Bus	Car	Hor	Kni	Mot	Per	Pla	Ska	Tra	Tru	UNK	OS	OS*
ResNet (He et al., 2016)+OSVM	29.7	39.2	49.9	54.0	76.8	22.2	71.2	32.6	75.1	21.5	65.2	0.6	45.2	44.9	44.8
DANN (Ganin et al., 2016)+OSVM	50.8	44.1	19.0	58.5	76.8	26.6	68.7	**50.5**	82.4	21.1	69.7	1.1	33.6	46.3	47.4
OSBP (Saito et al., 2018)	75.5	67.7	68.4	**66.2**	71.4	0.0	86.0	3.2	39.4	23.2	68.1	3.7	**79.3**	50.1	47.7
STA (Liu et al., 2019)	64.1	**70.3**	53.7	59.4	80.8	20.8	90.0	12.5	63.2	30.2	78.2	2.7	59.1	52.7	52.2
PGL	**81.5**	68.3	**74.2**	60.6	**91.9**	**45.4**	**92.2**	41.0	**87.9**	**67.5**	**79.2**	**6.4**	49.6	**65.5**	**66.8**

Table 4. Performance comparisons on the *VisDA-17* (VGGNet).

Method	Bic	Bus	Car	Mot	Tra	Tru	UNK	OS	OS*
MMD+OSVM	39.0	50.1	64.2	79.9	86.6	16.3	44.8	54.4	56.0
DANN+OSVM	31.8	56.6	71.7	77.4	87.0	22.3	41.9	55.5	57.8
ATI-λ+OSVM	46.2	57.5	56.9	79.1	81.6	32.7	65.0	59.9	59.0
OSBP	51.1	67.1	42.8	84.2	81.8	28.0	**85.1**	62.9	59.2
STA	52.4	69.6	59.9	87.8	86.5	27.2	84.1	66.8	63.9
PGL	**93.5**	**93.8**	**75.7**	**98.8**	**96.2**	**38.5**	68.6	**80.7**	**82.8**

Table 6. Performance comparisons *w.r.t.* varying enlarge factor α on the *VisDA-17* (ResNet-50).

Enlarging Factor	Syn2Real-O		Office-Home (Ar-Cl)	
	OS	OS*	OS	OS*
$\alpha = 0.20$	63.0	63.3	59.9	61.1
$\alpha = 0.10$	64.5	65.7	60.7	61.6
$\alpha = 0.05$	**65.6**	**66.5**	**61.8**	**63.1**

Table 5. Ablation performance on the *Syn2Real-O* (ResNet-50). "w" indicates with and "w/o" indicates without.

Model	UNK	ALL	OS	OS*
PGL w/o Progressive	43.6	44.8	54.4	55.3
PGL w NLL	48.6	49.7	56.9	57.6
PGL w/o GNNs	49.2	50.3	57.8	58.5
PGL w/o Mix-up	**49.8**	51.3	62.5	63.6
PGL	49.6	**51.5**	**65.5**	**66.8**

Office-Home, *Syn2Real-O* and *VisDA-17* datasets respectively. Note that our proposed approach provides significant performance gains for the more challenging datasets of *Syn2Real-O* and *VisDA-17* which require knowledge transfer across different modalities. This phenomenon can be also observed in the transfer sub-tasks with a large domain shift *e.g.*, **Rw→Cl** and **Pr→Ar** in *Office-Home*, which demonstrates the strong adaptation ability of the proposed framework. To study the validity of the progressive paradigm and early stopping strategy, we provide detailed graphs of our test performance per training step (OS, OS* and ALL scores) in the supplementary material.

Ablation Study: To investigate the impact of the derived progressive paradigm, GNNs, node classification loss, and mix-up strategy, we compare four variants of the PGL model on the *Syn2Real-O* dataset shown in Table 5. Except for **PGL w/o Progressive** that takes $\alpha = 1$ and $\beta = 0.6$, all experiments are conducted under the default setting of hyperparameters. **PGL w/o Progressive** corresponds to the model directly trained with one step, followed by pseudo-labeling function for classifying the unknown samples. As shown in Table 5, without applying the progressive learning strategy, the OS result of **PGL w/o Progressive** significantly drops by 16.9% because **PGL w/o Progressive** does not leverage the pseudo-labeled target samples leading to the failure in minimizing the shared error at the sample-level. In **PGL w NLL**, the focal loss of the node classification objective is replaced with the Negative log-likelihood (NLL) loss, resulting in OS performance dropping from 65.5% to 56.9%. Due to the absence of the focal loss re-weighting module, the model tends to assign more pseudo-labels to easy-to-classify samples, which consequently hinders effective graph learning in the episodic training process. In **PGL w/o GNNs**, we used ResNet-50 as the backbone for feature learning, which triggers 12.5% OS performance drops comparing to the graph learning model. The inferior results reveal that the GNN module can learn the class-wise
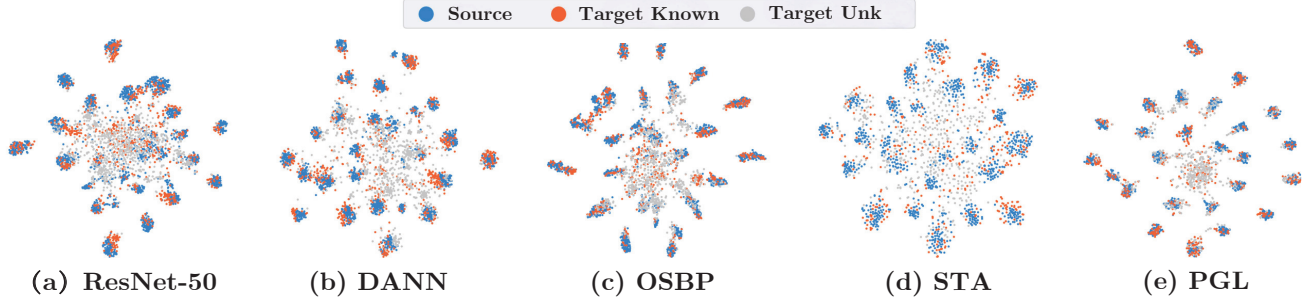
Figure 3. The t-SNE visualization of feature distributions on the Rw→Ar task (*Office-Home*) with the ResNet-50 backbone.

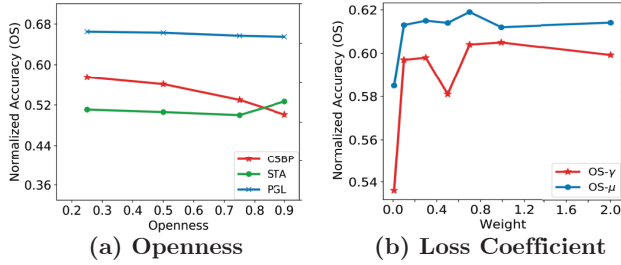*Figure 4.* Performance Comparisons *w.r.t.* varying (a) openness of the *Syn2Real-o* (ResNet-50); (b) loss coefficients μ and γ on the Ar→Cl task (*Office-Home*) with the ResNet-50 backbone.

manifold, which mitigates the potential noise and permutation by aggregating the neighboring information. **PGL w/o Mix-up** refers to the model that constructs episodes without taking any pseudo-labeled target data. We observe that the OS performance of **PGL w/o Mix-up** is 4.6% lower than the proposed model, confirming that replacing the source samples with pseudo-labeled target samples progressively can alleviate the side effect of conditional shift.

Robustness Analysis to Varying Openness: To verify the robustness of the proposed PGL, we conduct experiments on the *Syn2Real-O* with the openness varying in $\{0.25, 0.5, 0.75, 0.9\}$. The openness is defined as the ratio of unknown samples to all samples in the entire target set, which explicitly implies the level of challenge. The results of **OSBP**, **STA** and the proposed **PGL** are depicted in Figure 7(a). Note that **OSBP** and our **PGL** approach empirically sets a hyperparameter (β in our case) to control the openness, while **STA** automatically generates the soft weight in adversarial way and inevitably results in performance fluctuation. We observe that **PGL** consistently outperforms the counterparts by a large margin, which confirms its resistance to the change in openness.

Sensitivity to Loss Coefficients μ and γ: We show the sensitivity of our approach to varying the edge loss coefficient μ and adversarial loss coefficient γ in Figure 7(b). We vary the value of one loss coefficient from (0, 2] at each time,

while fixing the other parameter to the default setting. Two observations can be drawn from Figure 7(b): The OS score becomes stable when loss coefficients are within the interval of [0.7, 2]; When $\mu \to 0$, $\gamma \to 0$, the model performance drops by 4.6% and 10.2% respectively, which verifies the importance of the edge supervision and adversarial learning in our framework.

Sensitivity to Enlarging Factor α: We further study the effectiveness of the enlarging factor α, which controls the enlarging speed of the pseudo-labeled set, shown in Table 6. We note that the proposed model with a smaller value of α consistently performs better on both the *Syn2Real-O* and *Office-Home* datasets. This testifies our theoretical findings that the progressive open-set risk $\tilde{\Delta}_o$ can be controlled by consecutively classifying unknown samples. With a sacrifice on the training time, this strategy also provides more reliable pseudo-labeled candidates for the shared classifier learning preventing the potential error accumulation in the next several steps.

t-SNE Visualization. To intuitively showcase the effectiveness of OUDA approaches, we extract features from the baseline models (**ResNet-50**, **DANN**, **OSBP**, **STA**) and our proposed model **PGL** on the Rw→Ar task (*Office-Home*) with the ResNet-50 backbone. The feature distributions are visualized with t-SNE afterwards. As shown in Figure 5, compared with **ResNet-50** and **DANN**, open-set domain adaptation methods generally have a better separation between the known (in blue and red) and unknown (in grey) categories. **STA** achieves a better alignment between the source and target distributions in comparison with **OSBP**, while the **PGL** can obtain a clearer class-wise classification boundary benefiting from our graph neural networks and the mix-up strategy. We conduct additional experiments on the challenging *Syn2Real-O* dataset with a high openness. We randomly sample 200 episodes from the dataset including 2,400 source points and 2,400 target points and visualize the distributions of the learned representations from the compared baseline models and the proposed model in Figure 5. Figure 5(c)-(e) shows that the open-set domain adaptation methods are more robust to disturbance from unknowns
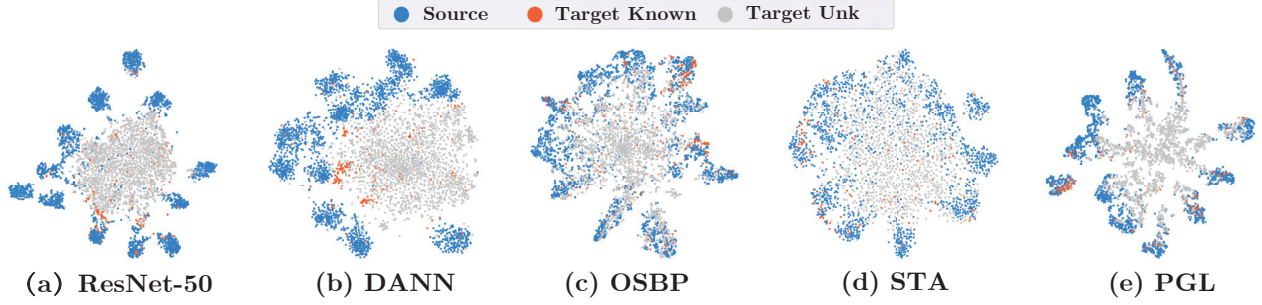
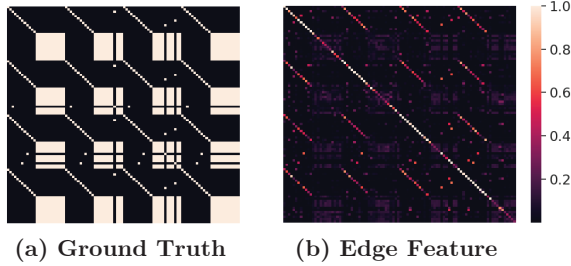Figure 5. The t-SNE visualization for the source and target data in the *Syn2Real-O* dataset.

Figure 6. Visualization of edge features on the *Syn2Real-O*. *Left*: the binary ground-truths label map. *Right*: the learned edge map from the proposed edge update networks. Best viewed in color.

compared with **ResNet-50** and **DANN**, as the source data (shown in blue) and the target data from the shared classes (shown in red) are aligned. A comparison between Figure 5(d) and Figure 5(e) reveals that the proposed graph learning and mix-up strategy have the ability to align class-specific conditional distributions across the domains, which means, the representations of the source and target data belonging to the same class are well mixed and less distinguishable.
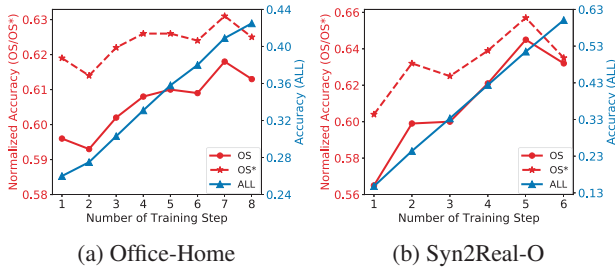

Figure 7. Recognition accuracies of the proposed PGL method on the (a) Ar→Cl task (*Office-Home*) and (b) *Syn2Real-O* datasets.

Edgemap Visualization. To further analyze the validity of the edge update networks, we extract the learned feature map from the PGL with a single-layer GNN on the *Syn2Real-O* dataset. As visualized in Figure 6(b), a large value of $\mathcal{E}_{ij}$ corresponds to a high degree of correlations between node v_i and v_j, which resembles the pattern of the ground-truth edge label $\widehat{Y}$ as displayed in Figure 6(a).

Quantitative Analysis over Training Steps. Figure 7 illustrates the recognition performance of PGL over training steps on the Ar→Cl task of the *Office-Home* dataset and *Syn2Real-O* dataset, respectively. Three evaluation metrics are used to testify performance, *i.e.*, the overall accuracy **ALL**, and normalized accuracies **OS** and **OS***. All metrics gain a performance over the first several steps as the pseudo-labeled target samples added in the source episodes can assist the classifier to make a more accurate prediction. Then, the normalized accuracy OS and OS* experience a downward because the enlarging pseudo-labeled set brings along noise and disturbance, which may degrade the model performance. In contrast, the accuracy ALL continuously increases as more unknown target samples are correctly classified, which occupy a large portion in the target domain. The results characterize a trade-off between normalized accuracy OS / OS* and the accuracy for unknowns. Considering the core value of domain adaptation is to correctly classify the classes of interest rather than irrelevant classes, we choose to stop the model updates at the training step 7 for the *Office-Home* and the step 5 for the *Syn2Real-O*.

6. Conclusion

We have addressed the open-set domain shift problem in both sample- and manifold-level by controlling the open-set risk. Experiments show that our proposed progressive graph learning framework performs consistently well on challenging object recognition benchmarks for open-set adaptation with significant domain discrepancy and conditional shifts.

Acknowledgements

References

Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. Unsupervised domain adaptation by domain invariant projection. In Proc. Int. Conference on Computer Vision (ICCV), 2013.

Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. Domain adaptation on the statistical manifold. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

Baktashmotlagh, M., Harandi, M. T., and Salzmann, M. Distribution-matching embedding for visual domain adaptation. Journal of Machine Learning Research, 17:108:1–108:30, 2016.

Baktashmotlagh, M., Faraki, M., Drummond, T., and Salzmann, M. Learning factorized representations for open-set domain adaptation. In Proc. Int. Conference on Learning Representations (ICLR), 2019.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In Proc. Advances in Neural Information Processing Systems (NeurIPS), 2006.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In Proc. Int. Conference on Machine Learningn (ICML), 2009.

Busto, P. P. and Gall, J. Open set domain adaptation. In Proc. Int. Conference on Computer Vision (ICCV), 2017.

Fang, Z., Lu, J., Liu, F., Xuan, J., and Zhang, G. Open set domain adaptation: Theoretical bound and algorithm. CoRR, arXiv preprint arXiv:1907.08375, 2019.

Feng, Q., Kang, G., Fan, H., and Yang, Y. Attract or distract: Exploit the margin of open set. In Proc. Int. Conference on Computer Vision (ICCV), 2019.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. S. Domain-adversarial training of neural networks. Journal of Machine Learning Research, pp. 59:1–59:35, 2016.

Ghifary, M., Kleijn, W. B., Zhang, M., Balduzzi, D., and Li, W. Deep reconstruction-classification networks for unsupervised domain adaptation. In Proc. European Conference on Computer Vision (ECCV), 2016.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. J. A kernel method for the two-sample-problem. In Proc. Advances in Neural Information Processing Systems (NeurIPS), 2006.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

Jain, L. P., Scheirer, W. J., and Boult, T. E. Multi-class open set recognition using probability of inclusion. In Proc. European Conference on Computer Vision (ECCV), 2014.

Júnior, P. R. M., de Souza, R. M., de Oliveira Werneck, R., Stein, B. V., Pazinato, D. V., de Almeida, W. R., Penatti, O. A. B., da Silva Torres, R., and Rocha, A. Nearest neighbors distance ratio open-set classifier. Machine Learning, 106(3):359–386, 2017.

Li, J., Lu, K., Huang, Z., Zhu, L., and Shen, H. T. Transfer independently together: A generalized framework for domain adaptation. IEEE Transactions on Cybernetics, 49(6):2144–2155, 2019.

Lin, T., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. In Proc. European Conference on Computer Vision (ECCV), 2014.

Liu, H., Cao, Z., Long, M., Wang, J., and Yang, Q. Separate to adapt: Open set domain adaptation via progressive separation. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. S. Transfer feature learning with joint distribution adaptation. In Proc. Int. Conference on Computer Vision (ICCV), 2013.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. In Proc. Int. Conference on Machine Learning (ICML), 2015.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. Deep transfer learning with joint adaptation networks. In Proc. Int. Conference on Machine Learning (ICML), 2017.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. In Proc. Conference on Learning Theory (COLT), 2009.

Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. Visda: The visual domain adaptation challenge. CoRR, arXiv preprint arXiv:1710.06924, 2017.

Peng, X., Usman, B., Saito, K., Kaushik, N., Hoffman, J., and Saenko, K. Syn2real: A new benchmark for synthetic-to-real visual domain adaptation. CoRR, arXiv preprint arXiv:1806.09755, 2018.

Saito, K., Yamamoto, S., Ushiku, Y., and Harada, T. Open set domain adaptation by backpropagation. In Proc. European Conference on Computer Vision (ECCV), 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proc. Int. Conference on Learning Representations (ICLR), 2015.

Snell, J., Swersky, K., and Zemel, R. S. Prototypical networks for few-shot learning. In Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. CoRR, arXiv preprint arXiv:1412.3474, 2014.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Pan-chanathan, S. Deep hashing network for unsupervised domain adaptation. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. In Proc. Advances in Neural Information Processing Systems (NeurIPS), 2016.

Zhang, Y., Liu, T., Long, M., and Jordan, M. I. Bridging theory and algorithm for domain adaptation. In Proc. Int. Conference on Machine Learning (ICML), 2019.

Zhao, H., des Combes, R. T., Zhang, K., and Gordon, G. J. On learning invariant representations for domain adaptation. In Proc. Int. Conference on Machine Learning (ICML), 2019.