Reinforced Reweighting for Self-supervised Partial Domain Adaptation

Keyu Wu, Shengkai Chen, Min Wu, *Senior Member, IEEE*, Shili Xiang, Ruibing Jin, Yuecong Xu, Xiaoli Li, *Fellow, IEEE*, and Zhenghua Chen*, *Senior Member, IEEE*

***Abstract*—Domain adaptation enables the reduction of distribution differences across domains, allowing for effective knowledge transfer from one domain to a different domain. In recent years, partial domain adaptation (PDA) has attracted growing interest due to its focus on a more realistic scenario, where the target label space is a subset of the source label space. As the source and target domains do not possess the same label space in the PDA setting, it is challenging but crucial to mitigate the domain gap without incurring negative transfer. In this paper, we propose a Reinforced Reweighting united with Self-supervised Adaptation (R2SA) method to address the challenges in PDA by leveraging the merits of deep reinforcement learning (DRL) and self-supervised learning (SSL) simultaneously in a cooperative way. Reinforced reweighting aims to learn a source reweighting policy automatically based on information provided by the PDA model, while self-supervised adaptation aims to boost the adaptability of the PDA model through an additional self-supervised objective on the target domain. Extensive experiments on several cross-domain benchmarks demonstrate that our method achieves state-of-the-art results, with larger performance gains on more challenging tasks.**

***Impact Statement*—In this paper, we propose a novel Reinforced Reweighting united with Self-supervised Adaptation (R2SA) method, a pioneering approach in partial domain adaptation (PDA) that innovatively integrates reinforcement learning, self-supervised learning and PDA. The new framework addresses critical gaps in existing PDA strategies and sets a new benchmark. A significant innovation is the introduction of the reinforced reweighting technique, a first in PDA. Traditional methods relied on simpler source selection strategies, often leading to the exclusion of potentially valuable data due to hard selection biases. R2SA, however, leverages a novel source reweighting policy to enhance the utilization of shared information across domains and ensure better model performance and reliability. Furthermore, R2SA also introduces a unique self-supervised adaptation method to significantly improve the model's adaptability, making the PDA model more robust and efficient. Our extensive experiments on multiple benchmark datasets demonstrate that R2SA consistently outperforms existing methods, which significantly advances the field.**

***Index Terms*—Partial domain adaptation, reinforcement learning, self-supervised learning**

I. Introduction

DEEP neural networks have demonstrated remarkable performance in various supervised learning tasks [1], [2]. Nevertheless, the success of deep learning models often depends on the availability of large labeled datasets, while annotating data for every new task can be impractical, expensive, and time-consuming. Consequently, domain adaptation (DA) has gained significant attention in recent years as its objective is to transfer knowledge from a related labeled source domain to a target domain lacking labels. The essence of DA lies in aligning the distributions of distinct domains by identifying invariant feature representations, which allows building a predictive model for the target domain using the labeled source domain data [3], [4].

Conventional DA methods mainly concentrate on aligning source and target distributions by minimizing discrepancy losses based on high-order statistics [5]–[7] or learning domain-invariant features via adversarial training [8]–[11]. However, these methods often assume that the source and target domains share an identical label space, while in real-world scenarios, it can be challenging to find a relevant source domain with the same label space as the target domain. When this assumption is not met, the performance of these DA methods may be significantly hindered due to negative transfer [12], [13]. In the era of large-scale datasets like ImageNet [14], a more general situation arises where the source label space encompasses the target label space, known as partial domain adaptation (PDA).

PDA is a more practical and general setting than conventional DA, as it is often simpler to access a large-scale source dataset when working with a smaller-scale unlabeled target dataset. However, PDA also poses a greater challenge due to the uncertain target label space and potential negative transfer arising from outlier source data. Existing PDA methods typically reweight source domain data to minimize the impact of irrelevant categories [15]–[17]. In recent years, alternative advanced learning techniques, such as reinforcement learning (RL) and self-supervised learning (SSL), have also been explored to improve PDA approaches. Domain Adversarial Reinforcement Learning (DARL) [18], Reinforced Transfer Network (RTNet) [19], Deep Reinforcement Learning based Data Selector (DRL-DS) [20], and Reinforced Adaptation Network (RAN) [21] employ various deep reinforcement learning (DRL) methods to determine policies for selecting source data. Meanwhile, Self-Supervised Partial Domain Adaptation (SSPDA) [22] leverages the signal captured by a jigsaw puzzle task as an additional objective to the PDA problem formulation.

* Corresponding Author

Keyu Wu, Shengkai Chen, Min Wu, Shili Xiang, Ruibing Jin and Yuecong Xu are with Institute for Infocomm Research, A*STAR, Singapore (e-mail: wu_keyu@i2r.a-star.edu.sg, chen_shengkai@i2r.a-star.edu.sg, wumin@i2r.a-star.edu.sg, sxiang@i2r.a-star.edu.sg, jin_ruibing@i2r.a-star.edu.sg, xuyu0014@e.ntu.edu.sg).

Xiaoli Li and Zhenghua Chen are with with Institute for Infocomm Research and Centre for Frontier AI Research, A*STAR, Singapore (e-mail: xlli@i2r.a-star.edu.sg, chen0832@e.ntu.edu.sg).

This paragraph will include the Associate Editor who handled your paper.

Despite these recent advances, there are still gaps in improving PDA using both DRL and SSL simultaneously. Additionally, current DRL-based methods only focus on simpler scenarios of source selection, rather than on more complex reweighting scenarios. To address these limitations, we propose a Reinforced Reweighting united with Self-supervised Adaptation (R2SA) method to solve the PDA problem by merging the merits of DRL and SSL simultaneously. Our approach consists of three sub-models, namely, the PDA model, the DRL model, and the SSL model. Reinforced reweighting aims to learn a reweighting policy automatically for source domain samples based on their similarities to the target domain samples in the feature space. Both the state and the reward of this DRL model are derived from the PDA model, while in turn, the PDA model is trained based on the reweighted source domain data. It is worth noting that different from previous DRL-based PDA methods, the proposed reinforced reweighting method aims to learn a source reweighting policy rather than a source selection policy to overcome the limitations caused by hard selection [23]. Generally, the advantages of reweighting over selection include better leverage of shared information and more robustness to noise. Selection may discard potentially useful shared information between source and target domains, while reweighting better leverages shared information by learning from the full source domain data and emphasizing relevant samples. Additionally, reweighting can reduce the impact of small errors in estimating source-target similarity, while selection is less robust to noise since these errors can lead to the exclusion of useful data as well as the inclusion of irrelevant samples. Meanwhile, in contrast to conventional reweighting methods, the utilization of DRL-based reweighting also introduces the ability to dynamically adapt and optimize the weighting policy throughout the entire learning process, thereby enhancing the performance of reweighting by offering a more flexible and robust solution.

In the meantime, our approach, which is distinct from current SSL-based methods, introduces an additional cross-domain adjustment. This is achieved by conducting representation learning solely on target domain data, while explicitly excluding outlier classes from the source domain. This additional self-supervised objective enhances the adaptability of the PDA model by leveraging the target domain data more effectively. The self-supervised task network is optimized using the target domain data only, allowing the feature extractor to concentrate on learning target-domain-specific discriminative features. In an end-to-end manner, the three sub-models are simultaneously optimized, resulting in a more robust and effective PDA model.

In summary, our key contributions are as follows:

- We propose the Reinforced Reweighting united with Self-supervised Adaptation (R2SA) method by synergistically integrating DRL, SSL, and PDA. By creatively combining these techniques, R2SA formulates a cohesive framework that enhances the efficiency and efficacy of the PDA model, marking a significant step forward in tackling PDA problems.
- The proposed reinforced reweighting method enables the automatic learning of a source reweighting policy for the first time based on feature similarity, overcoming the limitations of hard selection and providing a more flexible approach for handling PDA problems.
- The proposed self-supervised adaptation method is capable of boosting the adaptability of the PDA model through performing additional representation learning on the target domain data, facilitating improved alignment between source and target domains while preventing interference from outlier source classes.
- Extensive experimental results demonstrate that R2SA substantially surpasses existing PDA methods and attains state-of-the-art performance, validating the effectiveness of our approach in addressing the challenges of PDA.

II. Related Work

Unsupervised Domain Adaptation In this paper, we focus on unsupervised domain adaptation (UDA) which aims to transfer knowledge from a labeled source domain to an unlabeled target domain via mitigating the distribution shift across domains. UDA methods can generally be divided into two categories, namely, discrepancy-based and adversarial-based approaches. Discrepancy-based methods typically add adaptation layers and mitigate domain distribution discrepancies through minimizing statistical differences between the source and target domains. For instance, Maximum Mean Discrepancy (MMD) based methods minimize the MMD across domains to align their kernel embeddings of distributions [5], while CORAL aligns the second-order statistics of the source and target distributions [6]. On the other hand, inspired by generative adversarial networks (GANs), adversarial-based methods learn transferable features in an adversarial manner by employing a domain classifier. For example, Domain-Adversarial Neural Network (DANN) [24] introduces a gradient reversal layer to effectively confuse the domain classifier in order to encourage the learning of domain-invariant features. In contrast, Adversarial Discriminative Domain Adaptation (ADDA) [9] begins with pre-training a source encoder and proceeds to train a separate target encoder through adversarial training so that a discriminator and the target encoder are optimized iteratively to minimize the domain gap.

Partial Domain Adaptation Different from conventional DA methods that assume source and target domains have identical label spaces, partial domain adaptation (PDA) adopts a more relaxed label space assumption, allowing the source label space to encompass additional categories not present in the target domain. That is, PDA deals with a more general and practical scenario in reality. Compared to DA, PDA becomes more challenging since the target label space is unknown, and outlier source data can result in negative transfer if they are aligned with the target domain. To mitigate negative transfer, most PDA methods concentrate on decreasing the significance of source data belonging to the outlier classes by up-weighting relevant source samples while down-weighting irrelevant ones.

To down-weight irrelevant samples during transfer, Selective Adversarial Network (SAN) [15] utilizes class-wise discriminators to estimate the source instance weights. Similarly, Importance Weighted Adversarial Nets (IWAN) [16] evaluates the

significance of source samples by utilizing an auxiliary domain classifier. Partial Adversarial Domain Adaptation (PADA) [17] adopts estimated target label distribution to assign class-level weights to both domain discriminator and source classifier. In contrast, Example Transfer Network (ETN) [25] focuses on quantifying transferability of source samples and reweights them based on their discriminative information. Deep Residual Correction Network (DRCN) [26] identifies the most relevant source classes with a plug-in residual block and align them with target data using a weighted class-wise matching strategy. BA^{3}US [27] presents an alternative weighting scheme which augments the target domain using source samples and combines it with adaptive uncertainty suppression. Dual Alignment for PDA (DAPDA) [28] designs a reweighting network that assigns instance-level weights to target data and class-level weights to source data. More recently, rather than adjusting the feature extractor based on reweighted distribution alignment, Adversarial Reweighting (AR) [29] addresses the domain gap by reweighting source data features. Critical Classes and Samples Discovering Network (CSDN) [30] proposes an adaptive source class weighting strategy for dynamically selecting the most relevant classes while placing greater emphasis on ambiguous target samples with inconsistent predictions. Select, Label, and Mix (SLM) [31] filters source outliers, refines classification boundaries through discriminative labeling, and applies mixup for domain invariance enhancement concurrently to mitigate negative transfer while elevating discriminability and fostering domain-invariant representations.

In recent years, more advanced learning techniques have significantly contributed to enhancing PDA solutions. Among these, Deep Reinforcement Learning (DRL), which merges Reinforcement Learning (RL) principles with deep learning's ability to process high-dimensional data, is recognized as a key driver of innovation. DRL operates on the principle of an agent making decisions through interaction with an environment to maximize cumulative rewards. This process involves continuous strategy refinement through trial and error, enabling the agent to adapt its approach based on environmental feedback. Unlike traditional machine learning paradigms that depend on labeled datasets, DRL agents learn from the outcomes of their actions, allowing them to effectively manage complex decision-making problems. DRL addresses a wide range of problem settings, from discrete to continuous action spaces. Algorithms like the Deep Q-Network (DQN) [32] have shown remarkable success in discrete action domains by approximating the optimal action-value function with deep neural networks. Meanwhile, for environments that require a continuous action approach, methods like Deep Deterministic Policy Gradient (DDPG) [33] provide fine-grained control and precision. Leveraging these DRL advancements, innovative PDA methodologies have been developed. Specifically, Domain Adversarial Reinforcement Learning (DARL) [18], Reinforced Transfer Network (RTNet) [19], Deep Reinforcement Learning based Data Selector (DRL-DS) [20], and Reinforced Adaptation Network (RAN) [21] eliminate outlier source data through learning source data selection policies through deep Q-network (DQN) [32], the actor-critic algorithm [34], Dueling Double DQN [35], and Double DQN [36], respectively.

Concurrently, self-supervised learning (SSL) has also significantly advanced DA. The effectiveness of self-supervision has been validated through various studies [37]–[39]. Particularly, the Self-Supervised Partial Domain Adaptation (SSPDA) approach [22] advocates for employing the Jigsaw puzzle as a secondary task, which significantly contributes to performance improvement in PDA scenarios.

In this work, our R2SA jointly trains three models to solve the PDA problem by leveraging the merits of DRL, PDA, and SSL simultaneously in a cooperative manner. Different from the existing DRL-based methods and conventional methods exemplified by SLM, we focus on learning a source reweighting policy rather than source selection policies since hard selection can suffer from high variance and lead to difficulties if useful source samples are discarded by mistake [23]. Meanwhile, the DRL-based reweighting mechanism in R2SA offers more adaptability compared to the static criteria used by conventional methods like SLM. That is, R2SA can exhibit better robustness and generalization capabilities. Moreover, we propose to exploit an SSL-based auxiliary task to promote the adaptability of the PDA model by performing additional representation learning only on the target domain data without interference from the outlier source samples. This comprehensive approach distinguishes our method from previous works and addresses the challenges of partial domain adaptation more effectively.

III. Method

A. Framework Overview

A PDA task consists of a labeled source domain $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and an unlabeled target domain $\mathcal{D}_t = \{x_j^t\}_{j=1}^{n_t}$, where n_s and n_t denote the total number of source and target samples, respectively. The source label space encompasses the target label space, that is, $\mathcal{Y}_t \subset \mathcal{Y}_s$, and the source and target distributions $\mathcal{P}_s \neq \mathcal{P}_t$ even within the shared label space due to domain shift. Therefore, it is critical for PDA methods to eliminate irrelevant source information while encouraging transfer of relevant source knowledge to the target domain.

To this end, a Reinforced Reweighting united with Self-supervised Adaptation (R2SA) method is proposed, as shown in Figure 1, to overcome the challenges in PDA. Reinforced reweighting aims to learn a source reweighting policy automatically through a DRL model to align the distributions of source and target domain features. In the meantime, the PDA model aims to derive a predictive model for the target domain based on the reweighted source domain data while returning state and reward signals to the DRL model. Last but not least, self-supervised adaptation aims to boost the adaptation performance through a SSL model which shares the feature extractor with the PDA model and improves its transferability via an additional task performed only on the target domain data.

B. Reinforced Reweighting

In R2SA, the source reweighting process is modeled as a Markov Decision Process defined using a tuple $M = (S, A, R, P, \gamma)$, where S denotes the state space, A represents

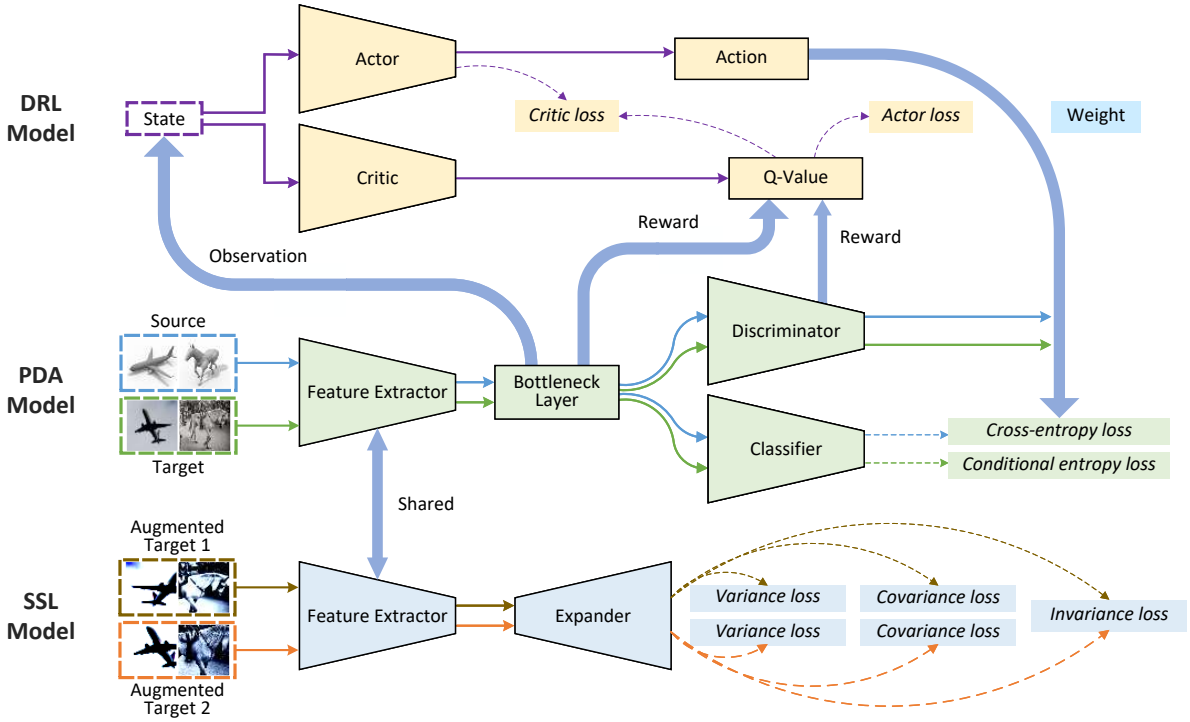


Fig. 1: Architecture overview of R2SA. A DRL model, a PDA model and a SSL model are joinly trained. The DRL model aims to learn a source reweighting policy automatically based on the information provided by the PDA model while the SSL model aims to boost the adaptation ability of the PDA model through performing additional representation learning only on the target domain.

the action space, R is the immediate reward, P describes the transition function, and $\gamma \in [0, 1]$ is the discount factor. This MDP formulation enables the use of RL algorithms to optimize the source reweighting policy. Specifically, at each time step, a batch of states $\mathbf{S}_b = \{s_i\}_{i=1}^n$ can be obtained through the PDA model given a batch of source samples $\mathbf{X}_b^s = \{x_i^s\}_{i=1}^n$, where n denotes the batch size. Our DRL model then determines the corresponding actions $\mathbf{A}_b = \{a_i\}_{i=1}^n$ to optimize the PDA model. The updated PDA model, in turn, provides rewards $\mathbf{R}_b = \{r_i\}_{i=1}^n$ and next states $\mathbf{S}_b' = \{s_i'\}_{i=1}^n$ to update the DRL model. To optimize this DRL model, we adopt the Deep Deterministic Policy Gradient (DDPG) algorithm [33] since it is commonly used for proof-of-concept. DDPG is a deterministic model-free actor-critic method which adapts the merits of DQN to domains with continuous action spaces. Within this framework, the actor network maps states to specific actions with the objective of maximizing rewards, while the critic network evaluates the actions by estimating their Q-values. That is, it learns the Q-value function via the Bellman equation and concurrently utilizes the Q-value function to learn the policy. This integrated strategy enables policy optimization based on the critic's feedback to effectively handle continuous action spaces. In the following, a detailed description of the state, action, and reward of our DRL model is provided.

State: To consider both the information contained in each source sample and its transferability, which indicates the extent to which the knowledge from these samples is applicable to the target domain, a state s_i is defined as a vector that concatenates two features. These include (1) the output of the bottleneck layer given x_i^s, i.e., $T(F(x_i^s))$, where F and T denote the feature extractor and bottleneck layer respectively; and (2) the largest cosine similarity of source feature $T(F(x_i^s))$ to the target domain features, i.e.,

$$\xi(x_i^s) = \max_{x_j^t \in \mathcal{D}_t} \frac{T(F(x_i^s)) \cdot T(F(x_j^t))}{\|T(F(x_i^s))\|_2 \cdot \|T(F(x_j^t))\|_2}. \tag{1}$$

The first feature represents the encoded high-level information contained in each source sample. The second feature serves as an indicator of transferability by measuring the similarity of each source sample to the target domain in the feature space. In this way, a state can be expressed as:

$$s_i = [T(F(x_i^s)), \xi(x_i^s)], \tag{2}$$

and the next state considers the same source samples as the state but is calculated using the updated PDA model instead. Hence, each batch of source samples can add n transitions to the replay buffer of the DRL model.

Action: Different from previous DRL-based PDA methods, our DRL model learns a reweighting policy instead of a selection policy. As a result, the action space is continuous with each action a_i being a real number in $[-1, 1]$. Let the actor network be denoted as $\mu(s_i)$ with parameters θ^μ, the action can be derived as:

$$a_i = \text{clip}\left(\mu(s_i|\theta^\mu) + \epsilon, -1, 1\right), \tag{3}$$

where $\epsilon \sim \mathcal{N}$ represents the OU noise [40]. To update the PDA model, an action a_i is converted to a weight w_i^r in range of $[0, 1]$, that is, $w_i^r = \frac{a_i+1}{2}$.

Reward: Each time an action is determined, a reward signal is provided by the PDA model. To evaluate the policy, a reward r_i is composed of two terms. The first term measures the difference between the DRL weight w_i^r and the weight w_i^p calculated by the discriminator of the PDA model. By minimizing the Wasserstein distance [41] between the target domain and the reweighted source domain distributions, we implement adversarial reweighting to calculate weight w_i^p. That is, let D denote the discriminator with parameters θ^D, w_i^p is obtained by solving the following objective function:

$$\begin{aligned}\min_{\mathbf{w}^p}\max_{\theta^D}\frac{1}{n_s}\sum_{i=1}^{n_s} w_i^p D(T(F(x_i^s)))\\ -\frac{1}{n_t}\sum_{j=1}^{n_t} D(T(F(x_j^t))),\end{aligned} \tag{4}$$

where $\mathbf{w}^p = (w_1^p, w_2^p, \cdots, w_{n_s}^p)^T$, $w_i^p \geq 0$, $\sum_{i=1}^{n_s} w_i^p = n_s$, and $\sum_{i=1}^{n_s}(w_i^p - 1)^2 < \rho n_s$, with n_s denoting the total number of source samples. In addition, since not all w_i^p are reliable, the similarity of the source sample to the target domain is also considered when calculating the reward to achieve better performance. Hence, the designed reward function can be expressed as:

$$r_i = 1 - 2|\min(w_i^p, 1) - w_i^r| + w_i^r(\xi(x_i^s) - \bar{\xi}), \tag{5}$$

where $\bar{\xi} = \frac{1}{n}\sum_{i=1}^{n}\xi(x_i^s)$ indicates the mean similarity to the target domain of the source batch $\mathbf{X}_b^s$. Therefore, the reward for the $i-$th source sample, r_i, is affected by two factors. The first evaluates whether the weight yielded by the DRL model is close to w_i^p. A higher reward is given for closer alignment as it indicates the model's prediction is in accordance with the expected benchmark value. Meanwhile, the second criterion examines the relationship between the source and target domain features. Specifically, when the source sample x_i^s is closer to the target domain in the feature space, a higher reward is received if the weight output by the DRL model is larger since $(\xi(x_i^s) - \bar{\xi})$ is positive. Conversely, when the corresponding source feature is dissimilar to the target domain features, a higher reward is received if the weight output by the DRL model is smaller since $(\xi(x_i^s) - \bar{\xi})$ is negative. In this way, the reward signals are more distinguishable with proper upper and lower bounds to facilitate policy learning.

C. Self-supervised Adaptation

To boost the adaptability of the feature extractor, we perform a self-supervised auxiliary task only on the target domain. At each step, we sample a batch of target data $\mathbf{X}_b^t = \{x_j^t\}_{j=1}^n$, and for each target sample, we produce two different views x_j' and x_j'' by performing a set of augmentation operations, including random cropping, horizontal flip, color jittering, grayscale, Gaussian blur, solarization, and color normalization, in sequence. The two views are then encoded by the feature extractor F, which is shared with the PDA model, to generate their representations. The two feature representations are subsequently mapped to d-dimensional embeddings z_j' and z_j'' respectively by the expander H. Since the target samples are processed in batches, the self-supervised loss is calculated based on the two batches of embeddings, i.e., $\mathbf{Z}_b' = [z_1', z_2', \cdots, z_n']$ and $\mathbf{Z}_b'' = [z_1'', z_2'', \cdots, z_n'']$. To avoid the involvement of contrastive samples, VICReg loss [42] is adopted for optimization due to its theoretical soundness. Hence, our SSL loss is composed of three terms and can be written as:

$$\begin{aligned}L_s(\mathbf{Z}_b', \mathbf{Z}_b'') = \frac{\lambda_1}{n}\sum_{i=1}^{n}\|z_i' - z_i''\|_2^2\\ +\frac{\lambda_2}{2}\left[\frac{1}{d}\sum_{m=1}^{d}\max(0, 1 - S(\mathbf{Z}_b')_m)\right.\\ \left.+\frac{1}{d}\sum_{m=1}^{d}\max(0, 1 - S(\mathbf{Z}_b'')_m)\right]\\ +\lambda_3\left[\frac{1}{d}\sum_{m\neq k} C(\mathbf{Z}_b')_{mk}^2 + \frac{1}{d}\sum_{m\neq k} C(\mathbf{Z}_b'')_{mk}^2\right],\end{aligned} \tag{6}$$

where $S(\cdot)$ and $C(\cdot)$ denote the regularized standard deviation and the covariance matrix, respectively, and λ_1, λ_2 and λ_3 are hyperparameters controlling the importance of the three terms. In specific, the first term is the invariance term to minimize the mean square distance between the two embedding vectors from the same sample. The second term is the variance term to force the embedding vectors of samples within a batch to be different so that collapse with all the inputs mapped on the same vector is prevented. The last term is the covariance term to avoid informational collapse via decorrelating the variables of each embedding and thereby encourage the feature representation to encode more information.

D. Overall Objective of R2SA

In this work, we regard the self-supervised task as an auxiliary task instead of a pretraining task and propose to optimize the SSL model together with the PDA model. Therefore, the objective function of R2SA simultaneously minimizes the reweighted cross-entropy loss on the source domain, as well as the conditional entropy loss and self-supervised loss on the target domain:

$$\begin{aligned}E = \frac{1}{n_s}\sum_{i=1}^{n_s} w_i^r w_i^p L_y(y_i^s, C(T(F(x_i^s))))\\ +\frac{\nu_1}{n_t}\sum_{j=1}^{n_t} L_h(C(T(F(x_j^t))))\\ +\nu_2 L_s(H(F(\mathcal{D}_t')), H(F(\mathcal{D}_t''))),\end{aligned} \tag{7}$$

where $L_y(p, q) = -\sum_i p_i \log q_i$ denotes the cross-entropy loss, $L_h(p) = -\sum_i p_i \log p_i$ represents the conditional entropy loss, and ν_1 and ν_2 are two trade-off hyperparameters. The first term is designed to minimize classification loss using source data, which trains the entire prediction network effectively. The second term focuses on minimizing conditional entropy loss on target data, promoting low-density

separation between classes in the target domain. Finally, the last term seeks to minimize the SSL loss on target data, thereby enhancing representation learning.

In the PDA model, the discriminator is optimized using Eq. (4), and it is worth noting that the parameters of the discriminator and those of the rest components are optimized alternately. This is due to two reasons. First, reweighting the source domain features is a more robust approach to bridging the domain gap compared to adapting the feature extractor based on reweighted distribution alignment losses. Second, since the weights derived from the discriminator are taken into consideration when calculating the reward of the DRL model, fixing the discriminator can also help to stabilize the training of the DRL model.

While updating the PDA and SSL models according to Eq. (7), we follow the DDPG training strategy to optimize our DRL model. Consequently, our model comprises two sets of networks: the actor and critic networks with parameters θ^{μ} and θ^{Q} respectively, and the target actor and critic networks with parameters $\theta^{\mu'}$ and $\theta^{Q'}$ respectively. The separate target networks are softly updated to stabilize training.

At each time step, a random minibatch of N transitions (s_i, a_i, r_i, s'_i) is sampled from the replay buffer and the target Q-values are computed as:

$$y_i = r_i + \gamma Q'(s'_i, \mu'(s'_i|\theta^{\mu'})|\theta^{Q'}), \tag{8}$$

where γ represents the discount factor. The critic network is updated by minimizing:

$$L_q = \frac{1}{N}\sum_i \left(Q(s_i, a_i|\theta^Q) - y_i\right)^2, \tag{9}$$

and the actor network is updated by minimizing:

$$L_\mu = -\frac{1}{N}\sum_i Q(s_i, \mu(s_i|\theta^{\mu})|\theta^Q). \tag{10}$$

To slowly track the learned networks, the target critic and actor networks are softly updated by polyak averaging after the critic and actor networks are updated. The update rule is $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$, where $\tau \ll 1$.

IV. Experiments

The performance of the proposed method is evaluated and compared to state-of-the-art methods through experiments on five benchmark datasets.

A. Datasets

The *Office-31* dataset [46] consists of 4,652 images distributed among 31 categories collected from three domains: Amazon (A), DSLR (D), and Webcam (W). We follow the same practice as [15] and select 10 categories to create the target domains. The *Office-Home* [47] dataset is larger, comprising approximately 15,500 images spread across 65 categories collected from four domains: Artistic (Ar), Clipart (Cl), Product (Pr), and Real-World (Rw). We select images from the first 25 categories in alphabetical order to create target domains. The *VisDA2017* [48] dataset is a challenging large-scale dataset containing over 280,000 images in 12 categories obtained from two domains: Synthetic (S) and Real (R). We select the first 6 categories in alphabetical order to create the target domain. The *ImageNet-Caltech* dataset is built from Caltech-256 (C) [49] and ImageNet-1K (I) [14], with 256 and 1000 classes, respectively. We select 84 shared classes as the target classes, and only the validation set of ImageNet is used when I is the target domain. The *DomainNet* [50] dataset is another challenging large-scale dataset. We follow the practice of [51] and select four domains with 126 categories, namely Clipart (C), Painting (P), Real (R), and Sketch (S). We adopt the first 40 categories in alphabetical order to create target domains.

B. Implementation Details

Our implementation is carried out on an Nvidia RTX A6000 GPU using Pytorch. The ResNet-50 [43] pre-trained on ImageNet [14] is adopted as the feature extractor, with the last fully-connected layer being excluded. The architecture of the discriminator consists of three fully-connected layers with 1024, 1024, and 1 nodes, respectively. The expander of the SSL model consists of three fully-connected layers, each with 8192 nodes. For the DRL model, the actor network has three fully-connected layers with 1024, 1024 and 1 nodes, respectively, while the critic network has two fully-connected layers with 1024 and 1 nodes, respectively. The hyperparameters, λ_1, λ_2, λ_3, ν_1, ν_2, τ, are set to 25, 25, 1, 0.1, 0.1, 1e-3, respectively, and the discount factor of the DRL model is 0.99. All the network layers are trained from scratch except the feature extractor. The feature extractor, bottleneck layer, classifier and expander are trained using the SGD algorithm with batch size 36 and momentum 0.9. The learning rate of the feature extractor is one-tenth that of the others. The discriminator, actor and critic are trained using the Adam algorithm. The learning rate and batch size of the discriminator are set to 1e-3 and 36, respectively, while we set the learning rate to 1e-4 and the batch size to 32 for the actor and critic. To compare with existing methods, we calculate the average accuracy based on three runs.

C. Experimental Results

Table I, II, III, IV and V present the experimental results on *Office-31*, *Office-Home*, *VisDA2017*, *ImageNet-Caltech* and *DomainNet* datasets, respectively. Remarkably, DA methods such as RTN, DANN, and ADDA exhibit inferior performance on many tasks compared to ResNet, indicating that directly aligning the target domain with the entire source domain can result in negative transfer. However, most PDA methods achieve significantly better results than ResNet by attempting to mitigate the influence of irrelevant source samples.

Among the PDA methods, our R2SA method gains state-of-the-art accuracy on four out of six *Office-31* tasks, eight out of twelve *Office-Home* tasks, one out of two *ImageNet-Caltech* tasks and eight out of twelve *DomainNet* tasks. Generally, R2SA achieves the highest average accuracies on four out of the five datasets. R2SA obtains larger performance gains on more challenging tasks and achieves significant

TABLE I: Average Accuracy (%) on *Office-31* Dataset using ResNet50 as backbone.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet [43]	75.59	96.27	98.09	83.44	83.92	84.97	87.05
DANN [24]	73.56	96.27	98.73	81.53	82.78	86.12	86.50
RTN [44]	78.98	93.22	85.35	77.07	89.25	89.46	85.56
ADDA [9]	75.67	95.38	99.85	83.41	83.62	84.25	87.03
SAN [15]	93.90	99.32	99.36	94.27	94.15	88.73	94.96
IWAN [16]	89.15	99.32	99.36	90.45	95.62	94.26	94.69
PADA [17]	86.54	99.32	**100.00**	82.17	92.69	95.41	92.69
ETN [25]	94.52	**100.00**	**100.00**	95.03	96.21	94.64	96.73
SSPDA [22]	91.52	92.88	98.94	90.87	90.61	94.36	93.20
DARL [18]	90.17	99.32	**100.00**	90.45	93.42	93.11	94.41
DRCN [26]	90.80	**100.00**	**100.00**	94.30	95.20	94.80	95.85
RTNet [19]	96.20	**100.00**	**100.00**	97.60	92.30	95.40	96.92
BA^{3}US [27]	98.98	**100.00**	98.73	99.36	94.82	94.99	97.81
DAPDA [28]	95.06	**100.00**	**100.00**	92.15	95.13	97.4	96.62
DRL-DS [20]	99.55	**100.00**	**100.00**	98.52	96.24	96.17	98.41
AR [29]	93.54	**100.00**	99.67	96.82	95.51	96.04	96.93
AGAN [45]	97.28	**100.00**	**100.00**	94.26	95.72	95.72	97.16
CSDN [30]	98.93	**100.00**	**100.00**	98.73	94.26	94.63	97.76
RAN [21]	98.98	**100.00**	**100.00**	97.67	96.28	96.24	98.19
SLM [31]	**99.80**	**100.00**	99.80	98.70	96.10	95.90	98.40
R2SA	98.87±1.09	**100.00**±0.0	99.36±0.0	**99.79**±0.37	**96.45**±0.11	96.49±0.24	**98.49**
R2SA (w/o DRL)	96.50±2.72	**100.00**±0.0	99.36±0.0	98.94±1.84	**96.45**±0.21	96.38±0.26	97.94
R2SA (w/o SSL)	97.74±2.15	**100.00**±0.0	99.36±0.0	98.73±0.64	96.28±0.30	**96.56**±0.21	98.11

TABLE II: Average Accuracy (%) on *Office-Home* Dataset using ResNet50 as backbone.

Method	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	Avg
ResNet [43]	46.33	67.51	75.87	59.14	59.94	62.73	58.22	41.79	74.88	67.40	48.18	74.17	61.35
DANN [24]	43.76	67.90	77.47	63.73	58.99	67.59	56.84	37.07	76.37	69.15	44.30	77.48	61.72
RTN [44]	49.31	57.70	80.07	63.54	63.47	73.38	65.11	41.73	75.32	63.18	43.57	80.50	63.07
ADDA [9]	45.23	68.79	79.21	64.56	60.01	68.29	57.56	38.89	77.45	70.28	45.23	78.32	62.82
SAN [15]	44.42	68.68	74.60	67.49	64.99	77.80	59.78	44.72	80.07	72.18	50.21	78.66	65.30
IWAN [16]	53.94	54.45	78.12	61.31	47.95	63.32	54.17	52.02	81.28	76.46	56.75	82.90	63.56
PADA [17]	51.95	67.00	78.74	52.16	53.78	59.03	52.61	43.22	78.79	73.73	56.60	77.09	62.06
ETN [25]	59.24	77.03	79.54	62.92	65.73	75.01	68.29	55.37	84.37	75.72	57.66	84.54	70.45
SSPDA [22]	52.02	63.64	77.95	65.66	59.31	73.48	70.49	51.54	84.89	76.25	60.74	80.86	68.07
DARL [18]	55.31	80.73	86.36	67.93	66.16	78.52	68.74	50.93	87.74	79.45	57.19	85.60	72.06
DRCN [26]	54.00	76.40	83.00	62.10	64.50	71.00	70.80	49.80	80.50	77.50	59.10	79.90	69.05
RTNet [19]	63.20	80.10	80.70	66.70	69.30	77.20	71.60	53.90	84.60	77.40	57.90	85.50	72.34
BA^{3}US [27]	60.62	83.16	88.39	71.75	72.79	83.40	75.45	61.59	86.53	79.25	62.80	86.05	75.98
DAPDA [28]	56.49	77.56	80.29	65.73	71.52	77.28	66.53	55.96	85.65	77.02	60.82	84.82	71.64
DRL-DS [20]	60.96	80.75	84.47	75.51	75.82	80.05	75.97	60.06	83.42	79.00	64.34	83.21	75.30
AR [29]	**67.4**	85.32	90.00	**77.32**	70.59	85.15	78.97	64.78	89.51	80.44	**66.21**	86.44	78.51
AGAN [45]	56.36	77.25	85.09	74.20	73.84	81.12	70.80	51.52	84.54	78.97	56.78	83.42	72.82
CSDN [30]	57.31	78.10	87.02	70.98	70.08	79.02	75.76	54.93	86.03	79.61	61.25	84.65	73.73
RAN [21]	63.26	83.08	89.03	74.99	74.47	82.90	77.99	61.19	86.68	79.86	63.52	85.04	76.84
SLM [31]	61.10	84.00	91.40	76.50	75.00	81.80	74.60	55.60	87.80	**82.30**	57.80	83.50	76.00
R2SA	66.82 ±1.66	**85.34** ±1.49	91.86 ±0.35	77.08 ±0.70	**75.95** ±0.96	**86.71** ±1.42	79.40 ±1.25	**65.35** ±0.78	**89.91** ±0.38	81.88 ±0.77	65.55 ±1.38	**87.64** ±0.58	**79.46**
R2SA (w/o DRL)	66.27 ±1.68	85.26 ±1.54	**91.87** ±0.50	76.55 ±0.61	75.46 ±2.04	85.88 ±1.75	78.66 ±2.16	64.60 ±0.84	89.71 ±0.54	81.73 ±0.96	64.34 ±0.61	87.17 ±0.65	78.96
R2SA (w/o SSL)	64.46 ±2.96	85.25 ±1.88	91.68 ±0.18	76.09 ±0.19	73.14 ±2.03	85.76 ±0.77	**79.46** ±1.34	64.18 ±0.48	89.69 ±0.56	81.60 ±1.02	63.86 ±1.44	86.91 ±0.90	78.51

improvements in difficult scenarios where knowledge is transferred from smaller-scale to larger-scale datasets. For instance, R2SA achieves a 5.04%, a 4.27% and a 5.63% improvement on the C → R, P → R and S → R tasks on *DomainNet* dataset, respectively. These encouraging results have verified the effectiveness of R2SA and demonstrate its capability of filtering out irrelevant information from the source domain. Although our method does not achieve the best performance on the *VisDA2017* dataset, it provides results comparable to SLM and significantly outperforms the other methods. In fact, our method surpasses SLM and attains state-of-the-art performance on all other datasets. Overall, among all 33 tasks spanning the five datasets, our method achieves the best performance in 21 tasks and secures the second-best in 10 tasks, with minimal performance gaps from the top-performing methods. In the remaining two simpler tasks where several methods achieve closely competitive results, our method is also among the top performers. Furthermore, the observation that no single method dominates across tasks where our method doesn't lead, underscores the adaptability of our approach to a wide range of PDA scenarios. Meanwhile, the analysis also demonstrates the complex nature of PDA challenges and underscores the necessity to enhance our approach's effectiveness across varying tasks.

D. Analysis

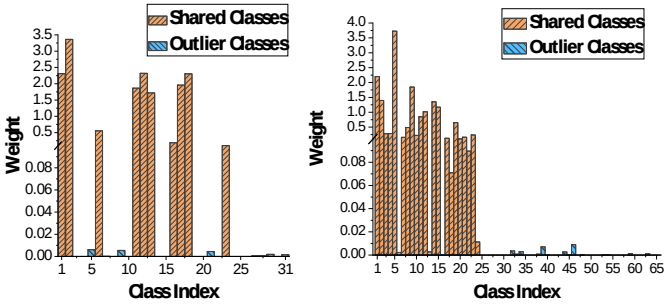
We perform an ablation study to examine the effectiveness of each component in R2SA and compare it with two variants. Specifically, the variant without a DRL model is denoted

TABLE III: Average Accuracy (%) on *VisDA2017* Dataset using ResNet50 as backbone.

Method	S → R
ResNet [43]	45.26
DANN [24]	51.01
RTN [44]	50.04
SAN [15]	49.90
IWAN [16]	48.60
PADA [17]	53.53
ETN [25]	63.35
SSPDA [22]	68.89
DARL [18]	67.77
DRCN [26]	58.20
BA^{3}US [27]	69.86
DRL-DS [20]	81.08
AR [29]	88.75
AGAN [45]	67.71
CSDN [30]	67.62
RAN [21]	75.10
SLM [31]	**91.70**
R2SA	90.08±0.39

TABLE IV: Average Accuracy (%) on *ImageNet-Caltech* Dataset using ResNet50 as backbone.

Method	C → I	I → C	Avg
ResNet [43]	71.29	69.69	70.49
DANN [24]	67.71	70.80	69.23
SAN [15]	75.26	77.75	76.51
IWAN [16]	73.33	78.06	75.70
PADA [17]	70.48	75.03	72.76
ETN [25]	74.93	83.23	79.08
DRCN [26]	78.90	75.30	77.10
BA^{3}US [27]	**83.35**	84.00	83.68
AR [29]	82.24	87.12	84.69
AGAN [45]	80.43	79.63	80.03
CSDN [30]	79.85	82.19	81.02
SLM [31]	81.40	82.30	81.90
R2SA	82.90±0.23	**88.20±1.35**	**85.55**



(a) A → W (b) Ar → Rw

Fig. 2: Learned class weights on two tasks. Orange and blue bins represent the average weights of shared classes and source-only classes, respectively.

as R2SA (w/o DRL), and the variant without a SSL model is denoted as R2SA (w/o SSL). The comparison results on *Office-31* and *Office-Home* are included in Tables I and II, respectively. On both datasets, R2SA outperforms its two variants and achieves the best results. Meanwhile, it is observed that the two variants also lead to better performance than most of the previous methods. These verify the effectiveness of our DRL model as well as SSL model, while demonstrating the superiority of solving the PDA problem through merging the

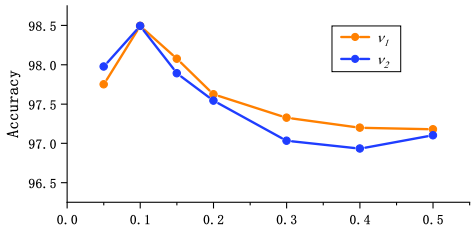
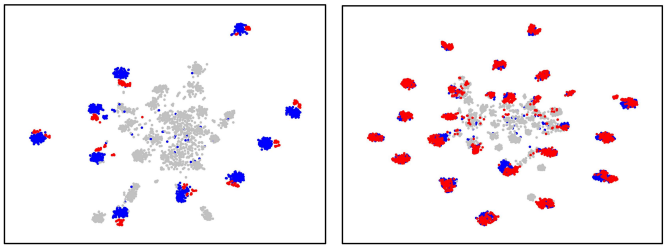


Fig. 3: Analysis of the trade-off parameters based on the Office-31 dataset.



(a) A → W (b) Ar → Rw

Fig. 4: Visualization of features learned by R2SA on two tasks. The blue and gray points represent the source samples from shared and outlier classes, respectively, while the red points represent the target samples.

merits of DRL and SSL simultaneously.

Alongside the ablation study, we also compare the proposed DRL-based reweighting method (R2SA w/o SSL) with the latest DRL-based selection method [21] utilizing the same baseline method [29]. The comparison results, in terms of average accuracy, are detailed in Tables VI and VII for the respective datasets. In both tables, results better than the baseline method are marked with up arrows. From these results, it is evident that the DRL-based reweighting method generally outperforms the DRL-based selection method across a broad spectrum of tasks and achieves a higher average accuracy across the datasets. Moreover, the reweighting method also leads to a much greater number of tasks surpassing the baseline method. This further highlights the enhanced performance of the DRL-based reweighting method compared to the DRL-based selection method.

Since reweighting plays an essential role to filter out irrelevant source samples, we also calculate the average learned weights for each source class on two tasks, as shown in Figure 2. On both tasks, the weights output by R2SA for the shared classes are significantly larger than those for source-only classes. On the easier A → W task, all the ten shared categories are recognized and even the smallest weight of the shared classes is about 17 times of the largest weight of the source-only classes. On the more challenging Ar → Rw task, most of the shared categories are recognized while weights of source-only classes are all close to zero. These observations reveal that R2SA is able to improve its performance through reweighting source samples based on their relevance.

Furthermore, to ensure the generalizability of our approach, we use standard default values for the DRL and SSL model pa-

TABLE V: Average Accuracy (%) on *DomainNet* Dataset using ResNet50 as backbone.

Method	C → P	C → R	C → S	P → C	P → R	P → S	R → C	R → P	R → S	S → C	S → P	S → R	Avg
ResNet [43]	41.21	60.01	42.13	54.52	70.80	48.32	63.10	58.63	50.26	45.43	39.30	49.75	51.96
DANN [24]	27.83	36.64	29.91	31.79	41.98	36.58	47.64	46.81	40.85	25.82	29.54	32.72	35.68
SAN [15]	34.35	51.62	46.23	57.13	70.21	58.25	69.61	67.49	67.88	41.69	41.15	48.44	54.50
PADA [17]	22.49	32.85	29.95	25.71	56.47	30.45	65.28	63.35	54.17	17.45	23.89	26.91	37.41
BA^{3}US [27]	42.87	54.72	53.79	64.03	76.39	64.69	**79.99**	**74.31**	**74.02**	50.36	42.69	49.65	60.63
AR [29]	52.66	68.24	**58.29**	66.78	77.53	74.38	76.70	71.77	70.48	53.66	53.60	61.57	65.47
R2SA	**54.68** ±2.04	**73.28** ±0.12	56.82 ±1.77	**70.99** ±0.84	**81.80** ±0.38	**76.65** ±0.48	79.82 ±0.37	73.05 ±0.15	73.41 ±0.97	**60.48** ±1.12	**57.50** ±0.47	**67.20** ±0.62	**68.81**

TABLE VI: Average Accuracy (%) Comparison on *Office-31* Dataset.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
DRL Selection	96.38↑	100↑	99.36↑	97.45↑	96.28↑	96.45↑	97.65↑
DRL Reweighting	97.74↑	100↑	99.36↑	98.73↑	96.28↑	96.56↑	98.11↑

TABLE VII: Average Accuracy (%) Comparison on *Office-Home* Dataset.

Method	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	Avg
DRL Selection	65.91↑	83.81	91.78↑	76.77↑	73.61↑	84.63	78.06	62.54	88.63	81.87↑	64.87↑	86.47	78.25↑
DRL Reweighting	64.46↑	85.25↑	91.68	76.09↑	73.14↑	85.76↑	79.46↑	64.18↑	89.69↑	81.60	63.86↑	86.91↑	78.51↑

rameters instead of relying on distinct dataset-specific optimal parameters. As a result, our method only requires the tuning of two trade-off parameters, ν_1 and ν_2. Both of these parameters achieve optimal or near-optimal performance on the majority of tasks when set to 0.1. Therefore, we assign a value of 0.1 to both parameters for all experiments. We present an analysis of these two parameters based on the Office-31 dataset, which averages the results of six tasks over three runs, as shown in Figure 3.

In addition, the t-SNE embeddings of feature representations learned by R2SA on task A → W and Ar → Rw are illustrated in Figure 4. The source samples in the shared classes, the source samples in outlier classes and the target samples are represented by blue, gray and red dots, respectively. It can be observed that R2SA aligns target features with source features in the shared classes. Moreover, it also separates different target classes clearly with compact cluster structures, which further demonstrates the superiority of R2SA.

E. Future Work

Based on the analysis of experimental results, we recognize the need to refine our method to better address the varied challenges of PDA. One focal point of this refinement is enhancing our reward function, making it more adaptive and responsive to domain differences. This effort includes a closer examination of factors like class distribution, domain similarity measures, and the PDA model's performance on the target domain. By adjusting the rewards based on both the training progress and the PDA model's performance, we aim to enhance the guidance provided to optimize the DRL model.

Additionally, we are keen on exploring more advanced DRL algorithms with better stability and robustness, such as T3D and SAC. These algorithms could potentially refine the reweighting policy, leading to better adaptation outcomes.

Moreover, we plan to enrich the information our DRL model uses to make decisions by designing a more detailed state representation. By incorporating more relevant information from the PDA model, including comprehensive feature similarity and feature importance metrics, a deeper understanding of the PDA model's context can be fostered. This, in turn, would enable more informed decision-making by the DRL model.

By focusing on these enhancements, we believe our method can achieve better effectiveness across a broader range of PDA challenges.

V. Conclusion

In this paper, a Reinforced Reweighting united with Self-supervised Adaptation method (R2SA) is proposed to address the PDA problem. Reinforced reweighting aims to learn a reweighting policy automatically. Meanwhile, self-supervised adaptation conducts additional representation learning only on the target domain data to boost adaptability of the PDA model. Extensive experimental results demonstrate the remarkable performance of R2SA.

Acknowledgment

This work was supported by the National Research Foundation of Singapore, AME Young Individual Research Grant (A2084c0167).

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[2] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[3] L. Zhang, P. Wang, W. Wei, H. Lu, C. Shen, A. van den Hengel, and Y. Zhang, "Unsupervised domain adaptation using robust class-wise matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1339–1349, 2019.

[4] W. Deng, L. Zheng, Y. Sun, and J. Jiao, "Rethinking triplet loss for domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 1, pp. 29–37, 2021.

[5] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.

[6] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 443–450.

[7] Z. Wang, B. Du, and Y. Guo, "Domain adaptation with neural embedding matching," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2387–2397, 2019.

[8] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning (ICML)*. PMLR, 2015, pp. 1180–1189.

[9] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 7167–7176.

[10] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7404–7413.

[11] Q. Kang, S. Yao, M. Zhou, K. Zhang, and A. Abusorrah, "Effective visual domain adaptation via generative adversarial distribution matching," *IEEE Transactions on neural networks and learning systems*, vol. 32, no. 9, pp. 3919–3929, 2020.

[12] X. Xu, H. He, H. Zhang, Y. Xu, and S. He, "Unsupervised domain adaptation via importance sampling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4688–4699, 2020.

[13] Y. Tian and S. Zhu, "Partial domain adaptation on semantic segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 3798–3809, 2022.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[15] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018, pp. 2724–2732.

[16] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018, pp. 8156–8164.

[17] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 135–150.

[18] J. Chen, X. Wu, L. Duan, and S. Gao, "Domain adversarial reinforcement learning for partial domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[19] Z. Chen, C. Chen, Z. Cheng, B. Jiang, K. Fang, and X. Jin, "Selective transfer with reinforced transfer network for partial domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 706–12 714.

[20] K. Wu, M. Wu, J. Yang, Z. Chen, Z. Li, and X. Li, "Deep reinforcement learning boosted partial domain adaptation," in *Proceedings of the International Conference on International Joint Conferences on Artificial Intelligence (IJCAI)*, 2021, pp. 3192–3199.

[21] K. Wu, M. Wu, Z. Chen, R. Jin, W. Cui, Z. Cao, and X. Li, "Reinforced adaptation network for partial domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 5, pp. 2370–2380, 2023.

[22] S. Bucci, A. D'Innocente, and T. Tommasi, "Tackling partial domain adaptation with self-supervision," in *International Conference on Image Analysis and Processing*. Springer, 2019, pp. 70–81.

[23] Z. Cao, K. You, Z. Zhang, J. Wang, and M. Long, "From big to small: Adaptive learning to partial-set domains," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[24] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[25] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2019, pp. 2985–2994.

[26] S. Li, C. H. Liu, Q. Lin, Q. Wen, L. Su, G. Huang, and Z. Ding, "Deep residual correction network for partial domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 7, pp. 2329–2344, 2020.

[27] J. Liang, Y. Wang, D. Hu, R. He, and J. Feng, "A balanced and uncertainty-aware approach for partial domain adaptation," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 123–140.

[28] L. Li, Z. Wan, and H. He, "Dual alignment for partial domain adaptation," *IEEE transactions on cybernetics*, vol. 51, no. 7, pp. 3404–3416, 2020.

[29] X. Gu, X. Yu, J. Sun, Z. Xu *et al.*, "Adversarial reweighting for partial domain adaptation," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 14 860–14 872, 2021.

[30] S. Li, K. Gong, B. Xie, C. H. Liu, W. Cao, and S. Tian, "Critical classes and samples discovering for partial domain adaptation," *IEEE Transactions on Cybernetics*, 2022.

[31] A. Sahoo, R. Panda, R. Feris, K. Saenko, and A. Das, "Select, label, and mix: Learning discriminative invariant feature representations for partial domain adaptation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4210–4219.

[32] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[33] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[34] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in neural information processing systems*, vol. 12, 1999.

[35] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.

[36] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[37] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," *arXiv preprint arXiv:1909.11825*, 2019.

[38] J. Xu, L. Xiao, and A. M. López, "Self-supervised domain adaptation for computer vision tasks," *IEEE Access*, vol. 7, pp. 156 694–156 706, 2019.

[39] K. Saito, D. Kim, S. Sclaroff, and K. Saenko, "Universal domain adaptation through self supervision," *Advances in neural information processing systems*, vol. 33, pp. 16 282–16 292, 2020.

[40] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Physical review*, vol. 36, no. 5, p. 823, 1930.

[41] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[42] A. Bardes, J. Ponce, and Y. LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," in *International Conference on Learning Representations*, 2021.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 770–778.

[44] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," *Advances in neural information processing systems (NeurIPS)*, vol. 29, 2016.

[45] Y. Kim and S. Hong, "Adaptive graph adversarial networks for partial domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 1, pp. 172–182, 2022.

[46] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*. Springer, 2010, pp. 213–226.

[47] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.

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

[49] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[50] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1406–1415.

[51] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8050–8058.