PARTIAL LABEL UNSUPERVISED DOMAIN ADAPTATION WITH CLASS-PROTOTYPE ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Partial label learning (PLL) tackles the problem where each instance is associated with a set of candidate labels, only one of which is the ground-truth. Most existing PLL approaches assume that both the training and test sets share the identical data distribution. However, this assumption does not hold in many real-world scenarios where the training and test data come from different distributions. In this paper, we formalize this learning scenario as a new problem called partial label unsupervised domain adaptation (PLUDA). To address this challenging PLUDA problem, we propose a novel Prototype Alignment based PLUDA method named PAPLUDA, which dynamically refines the pseudo-labels of instances from both the source and target domains by consulting the outputs of a teacher-student model in a moving-average manner, and bridges the domain discrepancy through inter-domain class prototype alignment. In addition, a teacher-student model based contrastive regularization is also deployed to enhance the prediction stability and hence improve the class prototypes in both domains for PLUDA. Comprehensive experimental results demonstrate that PAPLUDA achieves the state-of-the-art performance on the widely used benchmark datasets.

1 INTRODUCTION

Partial label learning (PLL) is a typical weakly supervised learning problem, where each training instance is assigned a candidate label set, only one of which is valid. PLL has gained increasing attention from the research community due to its effectiveness of reducing annotation costs in the various real-world scenarios, such as face naming (Hüllermeier & Beringer, 2006), web mining (Luo & Orabona, 2010), and ecoinformatics (Liu & Dietterich, 2014). Nevertheless, standard PLL assumes the training and test data are sampled from the same distribution. With this assumption, a model learned from the training data can be expected to generalize well on the test data. However, this assumption does not always hold in many real-world scenarios where the training and test data come from different distributions due to many practical factors, e.g., the training and test data are collected from different sources, or we have an out-dated training set due to the fact that the data always change over time. In such cases, there would be a discrepancy between the training and test data distributions, and hence naively adopting the off-the-shelf PLL models can lead to the significant performance degradation. Meanwhile, the unavailability of the ground-truth labels prevents the deployment of existing unsupervised domain adaptation (UDA) methods (Tzeng et al., 2017; Dong et al., 2021; Na et al., 2021; Shen et al., 2022).

We formalize this new learning scenario of PLL with training-test distribution gaps as a partial label unsupervised domain adaptation (PLUDA) problem. By integrating the challenges of both PLL and UDA problems, the PLUDA problem has the following characteristics: 1) the source and target domains have different distributions but share the same set of classes; 2) data in the source domain is only partially labeled—each instance is associated with a candidate label set, and the target domain only has unlabeled data; 3) the candidate label set for each source instance can contain both the ground-truth and irrelevant noise labels, while labels outside of the candidate set are true negative labels. The goal of the PLUDA task is to learn a domain-invariant prediction model from the partially labeled source domain that can generalize well in the unlabeled target domain.

Although both PLL and UDA have been studied intensively in the literature, to the best of our knowledge, there is no research yet to address their integrated challenges of PLUDA in a unified

framework. PLUDA is related to but still distinct from the weakly supervised domain adaptation (WSDA) problem studied in the recent literature (Shu et al., 2019; Xie et al., 2022). WSDA assumes the ground-truth labels of the source domain instances are corrupted (e.g., replaced) with noisy labels, and has been studied as an effort of reducing annotation costs. Some researchers addressed the WSDA problem based on the off-the-shelf technologies and obtained good performance. For example, GearNet (Xie et al., 2022) exploits the bilateral relationships between the source and target domains to construct a universal framework based on the existential powerful domain adaptation methods of TCL (Shu et al., 2019) and DANN (Tzeng et al., 2017).

In this paper, we propose a novel prototype alignment based partial label unsupervised domain adaptation approach, PAPLUDA, to address the combined PLL and UDA problems simultaneously in a newly formalized PLUDA learning scenario. The proposed PAPLUDA approach contains three pseudo-label based components that cooperate with each other to tackle PLUDA learning. First, we conduct pseudo-label refinement to dynamically correct the pseudo-labels of instances from both domains towards the ground-truth labels, which aims to disambiguate the partial labels and set up the foundation for cross-domain class alignment. Second, we propose an inter-domain class prototype-based alignment to minimize the discrepancy between prototypes of the same class from both the source and target domains while maximizing the gaps between the prototypes of different classes. Finally, we deploy a teacher-student model based contrastive regularization to enhance the reliable pseudo-labels, and hence improve the class prototypes and the inter-domain prototype alignment. Overall, our contributions can be summarized as follows:

- A new challenging learning problem, PLUDA, is proposed to address both PLL and UDA challenges simultaneously, which is more practical by dropping the common supervised learning assumptions of accurate data labels and identical training and testing distributions.
- A novel PAPLUDA approach is proposed to tackle the PLUDA problem.
- Comprehensive experiments are conducted on benchmarks and the results validate the effectiveness of the proposed PAPLUDA.

2 Related Work

2.1 Partial Label Learning

Partial label learning is a prevalent weakly supervised learning problem (Zhou, 2018), where each training instance is associated with a set of candidate labels, only one of which is valid. Most existing methods fits the existing learning techniques to partial label training data and disambiguate the candidate noise labels by aggregating the prediction of data. For maximum likelihood techniques, the likelihood of each partial label training instance is achieved by consulting to the probability of each candidate label associated with the training instance (Jin & Ghahramani, 2002; Liu & Dietterich, 2012). For k-nearest neighbor techniques, the candidate labels from neighbor instances are merged for final perdition with a weighted voting manner (Hüllermeier & Beringer, 2006; Gong et al., 2017; Zhang & Yu, 2015). For maximum margin techniques, the classification margin on each partial label instance is defined by discriminating modeling outputs from candidate labels and non-candidate labels(Nguyen & Caruana, 2008; Yu & Zhang, 2016). Apart from fitting the off-the-shelf techniques to partial label data, some researchers propose to address PLL by fitting partial label data to the existing learning techniques. For example, (Zhang et al., 2017) proposes to transform partial label training set to multiple binary training sets which then be used to built multiple binary classifiers corresponding to ECOC coding matrix. (Wu & Zhang, 2018) adopts a one-vs-one decomposition strategy to enable binary decomposition for learning from partial label data. Although these works produce competitive performance, they can suffer from the drawback that they are restricted to the liner model and have difficulty to be performed on large-scale datasets.

To alleviate the limitations of the standard PLL methods, PLL in the deep learning paradigm has recently started gaining attention from the research community. Yao et al. (2020) attempt to address PLL with deep convolutional neural network by exploiting a temporal-ensembling technique. Meanwhile, Yan & Guo (2020) handle PLL with multilayer perceptrons by means of batch label correction. Wen et al. (2021) present a group of loss functions called leveraged weighed loss, which takes the work (Lv et al., 2020) as its special case. Feng et al. (2020) present two provably consistent methods from the perspective of partial label generation procedural: a risk-consistent method and

a classifier-consistent method. Xu et al. (2021) adopt variational inference along the network training to progressively refine the latent label distribution under the assumption of instance-dependent partial label generation process. In addition, Wang et al. (2022) propose a class prototype-based label disambiguation method by contrastive learning, which achieves impressive performance on several image classification benchmarks. Though theses deep learning methods have produced notable progress on PLL, the study however is still limited to the supervised learning scenario where both training and test datasets share the identical data distribution.

2.2 Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA), which aims to transfer knowledge learned from a label-rich source domain to unlabeled target domain, has gained tremendous attention from the research community (Hoffman et al., 2018; Ghafoorian et al., 2017; Kamnitsas et al., 2017; Wang & Zheng, 2015; Fang et al., 2020; Dong et al., 2021). The main challenge of UDA is to bridge the cross-domain distribution gaps and induce transferable prediction models. Some previous research works have attempted to minimize the inter-domain gap based on some standard distribution discrepancy criteria, such as the maximum mean discrepancy (Pan et al., 2010), Kullback-Leibler divergence (Zhuang et al., 2015), central moment discrepancy (Zellinger et al., 2017), and wasserstein distance (Lee & Raginsky, 2018). Some studies also attempt to minimize the domain discrepancy by exploiting a domain discriminator in an adversarial learning manner, including domain adversarial network (Ganin et al., 2016) and adversarial discriminative domain adaptation (Tzeng et al., 2017). Recently, some researchers consider enhancing the domain adaptation procedural by exploiting the unlabeled data from the target domain, and propose new UDA approaches based on self-training models (Zou et al., 2019; Chen et al., 2020b).

These standard UDA methods however all assume the training data in the source domain are precisely annotated with accurate true labels, which is difficult to satisfy in real-world scenarios. More recently, weakly supervised domain adaptation (WSDA), which addresses both the UDA problem and the noise label learning issue, has gained increasing attention due to its effectiveness of reducing annotation costs for high quality labels (Shu et al., 2019; Tzeng et al., 2017; Liu et al., 2019). For example, TCL (Shu et al., 2019) proposes to train a domain adaptation network by exploiting the selected clean and transferable source instances. GearNet (Xie et al., 2022) proposes a universal framework based on the off-the-shelf powerful domain adaptation methods, TCL (Shu et al., 2019) and DANN (Tzeng et al., 2017), and further exploits bilateral relationships between both the source and target domains to enhance the WSDA training procedure. In this work, we relax the label requirement in the source domain by considering partial labels, which can be easily obtained with crowdsourcing. This new PLUDA problem however cannot be directly addressed by these existing PLL or UDA methods.

3 The Proposed Approach

We assume a source domain $D_S = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ with n_s partial label instances and target domain $D_T = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$ with n_t unlabeled instances, where $\mathbf{x}_i^s$ and $\mathbf{x}_i^t$ denote the i-th instance from source domain D_S and target domain D_T respectively, and $\mathbf{y}_i^s \in \{0,1\}^L$ denotes the label indicator vector associated with $\mathbf{x}_i^s$. The multiple 1 values in $\mathbf{y}_i^s$ denote either the true label or irrelevant label noise indistinguishably, which forms the candidate label set S_i for $\mathbf{x}_i^s$. We use $B_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{m_s}$ and $B_t = \{\mathbf{x}_i^s\}_{i=1}^{m_t}$ to denote a mini-batch of instances sampled from D_S and D_T respectively. The task of PLUDA is to induce a multi-class prediction model from D_S and D_T that can generalize well in the target domain.

We propose to address this novel task of PLUDA by simultaneously performing domain adaptation and partial label disambiguation with inter-domain class-prototype alignment and teacher-student model based pseudo-label refinement. The proposed PAPLUDA model is illustrated in Figure 1. The model architecture includes a shared feature extractor G, a teacher-student prediction network (student network F and teacher network F'), and a feature projection network H. Towards the objective of PAPLUDA, three loss components are designed on the model architecture: (1) *classification loss with soft label disambiguation*, which trains a teacher-student model from the partial labeled data in the source domain by performing moving average based soft label disambiguation; (2) *inter-domain class-prototype alignment loss*, which aims to bridge the cross-domain distribution

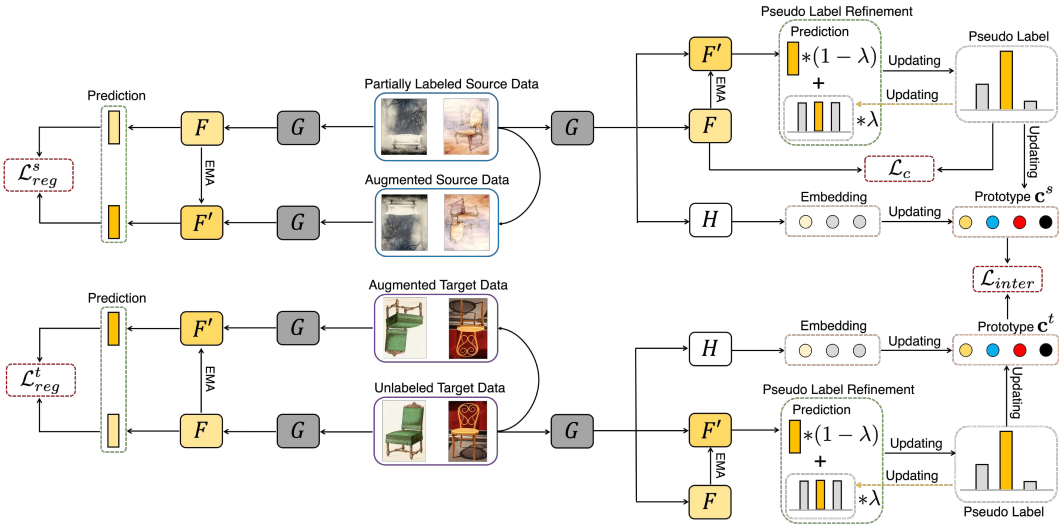


Figure 1: The proposed PAPLUDA model, which mainly has three loss components: classification loss with soft label disambiguation $\mathcal{L}_c$; inter-domain class-prototype alignment loss $\mathcal{L}_{inter}$; teacher-student model based contrastive regularization loss $\mathcal{L}_{reg}$ ($\mathcal{L}_{reg}^s$ and $\mathcal{L}_{reg}^t$). All the components mutually benefit from each other to improve the PLUDA learning cooperatively.

gap for information sharing by performing contrastive alignment between the prototypes of the same class in the two domains; (3) *teacher-student model based contrastive regularization loss*, which enhances the prediction stability and consequently improves the class-prototypes and inter-domain prototype alignment for PLUDA learning. We elaborate these loss components and the proposed approach in the following subsections.

3.1 Classification with Soft Label Disambiguation

As the true labels are hidden within each candidate label set in the source training data, it is critical to separate the true labels from the noisy labels—that is, label disambiguation—for inducing a good classification model. We deploy a teacher-student model as the classification model due to its robustness against noisy labels for stable prediction. We assume the teacher and the student share the same feature extractor G but maintain two different prediction networks: student network F and teacher network F'. F and F' have identical structures but different parameters, and they produce multi-class soft-max probability prediction vectors via soft-max function. The parameters of the student network, Θ_F, will be learned from the training data, while the parameters of the teacher network, $\Theta_{F'}$, will be updated as an exponential moving average of the student network parameters, such as:

$$\Theta_{F'} = \eta\Theta_{F'} + (1-\eta)\Theta_F \tag{1}$$

where $\eta \in (0,1)$ is a weight hyperparameter. Importantly, we propose to use the teacher network that is typically more robust than the student network to perform label disambiguation:

$$\widehat{\mathbf{y}}_{ij}^s = \mathbb{I}\big[j = \arg\max_{j \in S_i} F'(G(\mathbf{x}_i^s))\big], \tag{2}$$

where $\mathbb{I}$ is an indicator function, and $\widehat{\mathbf{y}}_i^s$ is the disambiguated one-hot label indicator vector for instance $\mathbf{x}_i^s$ using the teacher network. Such disambiguated one-hot label vectors for the source training instances however would certainly contain mistakes. Hence directly using them for model training could cause oscillations by largely deviating from the original candidate labels. Hence we propose to perform *soft label disambiguation* by updating the candidate label distribution vectors with the one-hot vectors $\{\widehat{\mathbf{y}}_i^s\}$ in a moving average fashion:

$$\mathbf{p}_i^s = \lambda\mathbf{p}_i^s + (1-\lambda)\widehat{\mathbf{y}}_i^s, \tag{3}$$

where $\lambda \in (0,1)$ is a hyperparameter, and $\mathbf{p}_i^s$ denotes the soft disambiguated label vector for the source instance $\mathbf{x}_i^s$. We initialize $\mathbf{p}_i^s$ as the original candidate label distribution vector for $\mathbf{x}_i^s$, such as $\mathbf{p}_{ij}^s = \frac{1}{|S_i|}\mathbb{I}[j \in S_i]$.

Benefiting from the long memory of the moving average update, we expect this soft label disambiguation procedure can steadily and progressively approach the underlying true label vectors through the iterative training steps. With the soft disambiguated label vectors, we can update the student network F and the feature extractor G by minimizing following classification loss for each instance:

$$\mathcal{L}_c^i(\mathbf{x}_i^s, S_i) = \sum_{j=1}^{L} -\mathbf{p}_{ij} \log F_j(G(\mathbf{x}_i^s)) \tag{4}$$

where j denotes the indices of the class labels, and $F_j(\cdot)$ denotes the j-th output entry of the student prediction network F, which indicates the predicted probability of the given instance belonging to the j-th class. By iteratively performing soft label disambiguation and prediction model update, we expect the true labels can be gradually and statistical identified from the source training data, and hence induce a good classification model.

3.2 Inter-Domain Class-Prototype Alignment

The goal of PLUDA is to induce a classification model that generalize well in the target domain. Hence it is important to bridge the cross-domain feature distribution discrepancy and take the unlabeled target data into consideration for the training process. We propose to discriminatively bridge the cross-domain divergence and match the two domains specifically for the classification task by performing prediction-aware inter-domain class-prototype alignment.

A class prototype is defined as the mean embedding vector for the instances belonging to the class. Following self-supervised learning (Chen et al., 2020a), we compute the class prototypes in a low-dimensional embedding space produced by a projection network H on the extracted features. As the true labels are unknown in the source and target domains, we propose to compute the class prototypes using the predicted pseudo-labels from the teacher-student model. Specifically, we compute the class-prototype for each training batch B_d as follows:

$$\mathbf{c}_k^d = \frac{\sum_{\mathbf{x}_i^d \in B_d} H(G(\mathbf{x}_i^d)) \cdot \mathbb{I}[k = \arg\max_{k \in \mathcal{Y}^d} \mathbf{p}_{ik}^d]}{\sum_{i=1}^{|B_d|} \mathbb{I}[k = \arg\max_{k \in \mathcal{Y}^d} \mathbf{p}_{ik}^d]} \tag{5}$$

where $d \in \{s, t\}$ is a domain indicator; $\mathcal{Y}^s = S_i$ in the source domain and $\mathcal{Y}^t = \{1, 2, \cdots, L\}$ in the target domain. The discrete pseudo-labels in the source domain are produced from the soft disambiguated label vectors $\{\mathbf{p}_i^s\}$ computed in Eq.(3). For the unlabeled target domain, we propose to maintain soft label probability vectors $\{\mathbf{p}_i^t\}$ during training in the following way:

$$\mathbf{p}_i^t = \lambda \mathbf{p}_i^t + (1-\lambda) F'(G(\mathbf{x}_i^t)) \tag{6}$$

To alleviate the negative impact from the partially labeled data in a local batch, we maintain a dynamic dictionary $C'^s = [\mathbf{c}_1'^s, \cdots, \mathbf{c}_k'^s, \cdots, \mathbf{c}_L'^s]$ in the source domain to store the class prototypes from the last batch for prototype calibration. After computing the class prototypes using Eq.(5) for the current source batch, we further update them by consulting the dynamic dictionary in a moving average manner:

$$\mathbf{c}_k^s = \gamma \mathbf{c}_k^s + (1-\gamma) \mathbf{c}_k'^s, \tag{7}$$

where $\gamma \in [0, 1]$ is a trade-off parameter. This allows us to take the prototype information from previous batches into account.

With the class-prototypes computed in both domains, we propose to perform inter-domain prototype alignment using the following contrastive loss for each class k:

$$\mathcal{L}_{inter}^k(\mathbf{c}_k^s, \mathbf{c}_k^t) = -\log \frac{\Omega(\mathbf{c}_k^s, \mathbf{c}_k^t)}{\Omega(\mathbf{c}_k^s, \mathbf{c}_k^t) + \sum_{j=1}^{L} \mathbb{I}[j \neq k] \cdot \big(\Omega(\mathbf{c}_j^s, \mathbf{c}_k^t) + \Omega(\mathbf{c}_j^t, \mathbf{c}_k^s)\big)}, \tag{8}$$

where $\Omega(a, b) = \exp(\text{cosine}(a, b)/\tau)$ measures the similarity of the given pair, and τ is the temperature hyperparameter. This contrastive loss uses the inter-domain prototypes from the same class as

the positive pair and the ones from different classes as the negative pairs. We expect this contrastive loss can push the inter-domain prototype pair of the same class closer to each other and push the inter-domain pairs from different classes apart in the embedded feature space, aiming to induce a good feature representation space (via G), in which the two domains can be discriminatively aligned for classification model training.

3.3 Teacher-Student Model based Contrastive Regularization

To exploit the consistent statistical knowledge learned along the model training across both domains and avoid prediction perturbations caused by the noisy label information, we propose a self-supervised contrastive loss in the output space of the prediction networks and calculate the loss in both domains as regularizers to enhance the stability of the prediction model.

The self-supervised contrastive loss aims to push the prediction outputs for the variants of the same instance to be similar and for the different instances to be dissimilar. It can conveniently exploit the unlabeled target instances. Specifically, for each training instance $\mathbf{x}_i$, we first generate an augmented version for it: $\widehat{\mathbf{x}}_i = \mathrm{Aug}(\mathbf{x}_i)$, where $\mathrm{Aug}(\cdot)$ denotes the data augmentation function (Cubuk et al., 2020). Then the contrastive loss for each instance $\mathbf{x}_i^d$ in each domain $d \in \{s, t\}$ can be computed as:

$$\mathcal{L}_{reg}^{i,d}(\mathbf{x}_i^d, \widehat{\mathbf{x}}_i^d) = -\log \frac{h(\mathbf{x}_i^d, \widehat{\mathbf{x}}_i^d)}{\sum_{j=1}^{|B_d|} h(\mathbf{x}_j^d, \widehat{\mathbf{x}}_i^d) + \sum_{j=1}^{|B_d|} \mathbb{I}[j \neq i] \cdot h(\widehat{\mathbf{x}}_j^d, \widehat{\mathbf{x}}_i^d)}, \tag{9}$$

where $h(a, b) = \exp(\mathrm{cosine}(F(G(a)), F'(G(b)))/\tau)$ denotes the similarity of the predicted label vectors on the given input instance pair using the student network and teacher network respectively, and τ is the temperature hyperparameter. By deploying this contrastive loss as regularizers on both domains, we have the following batch-wise total regularization loss:

$$\mathcal{L}_{reg} = \mathbb{E}_{\mathbf{x}_i^s \in B_s} \mathcal{L}_{reg}^{i,s}(\mathbf{x}_i^s, \widehat{\mathbf{x}}_i^s) + \mathbb{E}_{\mathbf{x}_i^t \in B_t} \mathcal{L}_{reg}^{i,t}(\mathbf{x}_i^t, \widehat{\mathbf{x}}_i^t) \tag{10}$$

We expect this prediction consistency regularization loss in both domains can help induce similar feature representations across domains for the classification task and improve the inter-domain class-prototype alignment to facilitate domain adaptation.

3.4 Overall Training Loss

Finally, by integrating the classification loss in Eq.(4), the inter-domain class-prototype alignment loss in Eq.(8) and the teacher-student model based contrastive regularization loss in Eq.(10) together, we have the following overall batch-wise training loss:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_i^s \in B_s} \mathcal{L}_c^i + \alpha \sum\nolimits_k \mathcal{L}_{inter}^k + \beta \mathcal{L}_{reg} \tag{11}$$

where α and β are trade-off hyperparameter that balances the weights of the inter-domain class-prototype alignment loss and the teacher-student model based contrastive regularization loss respectively. The training algorithm is presented in Algorithm 1 in Appendix.

4 Experiments

We conduct experiments on two benchmark domain adaptation datasets by generating partial labels on the source domains. In this section, we present the experimental setting and results.

4.1 Datasets

We validate the effectiveness of the proposed approach on two different domain adaptation datasets: Office-31 (Saenko et al., 2010) and Office-Home (Venkateswara et al., 2017). Office-31 is a classical dataset containing 4652 images with 31 classes across the three domains: Amazon (A), Webcam (W), and DSLR (D). The A, W, and D represent that images are collected from the amazon.com, web camera, and digital SLR camera respectively. Office-Home is a widely used domain adaptation benchmark dataset contraining 15500 images with 65 classes across the four domains: Artistic (Ar), Clip Art (Cl), Product (Pr) and Real-World (Rw). These domains represent four different

Table 1: Accuracy (mean±std) comparisons on Office-31 datasets with partial labels on ambiguity level q = 0.2. The best result among each column is highlighted in bold.

Tasks	A → W	A → D	W → A	W → D	D → A	D → W	Average
DANN	33.46±1.87	35.02±1.05	25.42±0.79	45.62±1.18	24.21±1.47	48.41±0.47	35.36±1.14
TCL	50.62±0.85	54.33±1.18	30.42±0.36	56.72±0.96	28.22±0.68	52.70±0.58	45.50±0.77
$\text{GearNet}_{\text{Co-teaching}}$	30.33±2.05	27.71±1.25	21.23±0.68	49.79±2.13	25.04±1.48	43.20±1.93	32.88±1.58
$\text{GearNet}_{\text{DANN}}$	53.25±0.41	55.83±1.14	32.17±1.16	63.75±1.59	32.06±0.94	58.69±0.77	49.29±1.01
$\text{GearNet}_{\text{TCL}}$	59.77±0.71	62.15±1.03	43.91±1.27	76.50±0.88	37.67±0.73	64.94±0.68	57.49±0.88
Ours	**77.86±0.52**	**74.37±1.21**	**55.82±0.85**	**91.67±0.77**	**46.75±0.59**	**82.68±0.77**	**71.52±0.78**

Table 2: Accuracy (mean±std) comparisons on Office-31 datasets with partial labels on ambiguity level q = 0.4. The best result among each column is highlighted in bold.

Tasks	A → W	A → D	W → A	W → D	D → A	D → W	Average
DANN	23.17±1.55	28.33±1.25	14.31±1.63	33.54±1.66	18.71±1.89	37.34±1.78	25.90±1.63
TCL	33.97±1.67	40.12±1.97	20.31±0.88	45.33±1.85	22.12±1.83	42.57±2.73	34.07±1.82
$\text{GearNet}_{\text{Co-teaching}}$	14.32±1.13	19.16±0.96	10.76±1.05	31.04±1.44	15.58±1.66	26.14±1.33	19.50±1.26
$\text{GearNet}_{\text{DANN}}$	36.84±1.58	46.25±1.88	18.04±1.44	50.62±1.83	22.51±0.97	43.33±1.25	36.26±1.49
$\text{GearNet}_{\text{TCL}}$	49.62±0.75	55.19±1.03	33.15±1.88	61.04±2.07	28.25±1.44	57.15±1.56	47.40±1.45
Ours	**76.17±1.03**	**71.25±0.96**	**45.81±0.93**	**90.40±1.66**	**37.39±1.05**	**80.86±1.06**	**66.98±1.12**

image styles including artistic depictions, clipart images, images without background and pictures captured by cameras respectively. Following the partial label generation process (Lv et al., 2020), we manually corrupt two datasets into partially labeled version for PLUDA learning. Specially, any irrelevant label for source image is uniformly picked as a candidate label with probability q. When encountering a special case that no one irrelevant label is chosen, we randomly pick a irrelevant label as the candidate label to ensure all training set are fully corrupted. For all comparison experiments, we report the average test accuracy and standard deviation based on five independent runs.

4.2 Comparison Methods

We compare the proposed approach with three state-of-the-art weakly supervised domain adaptation methods and two baseline methods, each configured with the suggested parameters according to the respective literature:

- DANN (Tzeng et al., 2017), which utilizes a domain discriminator with adversarial optimization objective to reduce the domain gap.
- TCL (Shu et al., 2019), which devises a transferable curriculum to select transferable instance for model training and improve domain adaptation via self-paced learning.
- $\text{GearNet}_{\text{Co-teaching}}$ (Xie et al., 2022), which exploits bilateral relationship between source and target domains for model training and integrates off-the-shelf framework co-teaching (Han et al., 2018) to improve the model robustness against label noise.
- $\text{GearNet}_{\text{DANN}}$ (Xie et al., 2022) and $\text{GearNet}_{\text{TCL}}$ (Xie et al., 2022), which is a universal framework by exploiting the knowledge from both domains, and adopt DANN (Tzeng et al., 2017) and TCL (Shu et al., 2019) as the backbone respectively to address WSDA problem.

4.3 Implementation Details

We use Resnet50 as the backbone network for feature extraction. The projection network is a two-layer multilayer percept (MLP) that outputs 128-dimensional embeddings. The prediction network is a two-layer MLP, while the Leaky ReLu activation function is used in the middle layer and a softmax activation is adopted in the output layer. The smoothing coefficient η and monument coefficient for prototype updating are set to 0.999 and 0.99 respectively. For pseudo label updating, we linearly ramp down λ from 0.95 to 0.8. The temperature parameter is set as τ = 5. We employ a standard stochastic gradient descent optimizer with a momentum of 0.9, an initial learning rate of 0.1 with cosine learning rate decay for all our experiments. Weight decay is set to 0.0005 for all comparision experiments. The mini-batch size is set as 32. The trade-off parameter α and β are chosen from $\{0.1, 0.3, 0.5, 0.7, 0.8, 1\}$ according to the validation performance of A → W on Office-31 dataset and Ar → CI on Office-Home dataset respectively.

Table 3: Accuracy (mean±std) comparisons on Office-Home datasets with partial labels on ambiguity level $q = 0.2$. The best result among each column is highlighted in bold.

Tasks	DANN	TCL	$\text{GearNet}_{\text{Co-teaching}}$	$\text{GearNet}_{\text{DANN}}$	$\text{GearNet}_{\text{TCL}}$	Ours
Ar → CI	9.64±1.13	10.23±1.47	13.55±0.53	11.41±1.06	15.36±0.75	**20.77±0.93**
Ar → Pr	15.13±1.15	16.02±1.41	21.42±0.72	19.90±1.19	21.33±1.01	**25.18±1.15**
Ar → Rw	20.49±2.11	21.26±0.72	26.82±0.66	25.27±1.33	26.78±1.57	**29.57±1.24**
CI → Ar	11.75±0.77	14.88±1.61	11.37±0.78	15.79±0.75	18.45±1.55	**22.63±0.88**
CI → Pr	14.58±1.22	17.47±1.35	18.95±0.83	24.16±1.13	25.35±1.02	**28.26±1.23**
CI → Rw	13.35±1.48	14.61±1.40	16.89±0.89	16.89±0.61	18.96±0.98	**31.69±0.75**
Pr → Ar	13.46±0.57	14.88±0.62	15.71±1.35	17.17±0.88	21.37±1.31	**27.51±0.82**
Pr → CI	12.38±1.88	16.79±1.77	12.54±1.21	14.97±1.17	19.61±1.48	**35.68±1.24**
Pr → Rw	20.71±1.04	27.38±0.83	25.06±0.75	25.64±1.04	31.65±1.19	**53.60±1.12**
Rw → Ar	16.25±1.15	19.87±1.06	17.50±1.01	20.50±0.68	25.69±1.52	**46.19±0.74**
Rw → CI	15.52±0.88	17.02±2.02	15.71±1.16	18.37±1.04	23.33±1.43	**44.39±1.20**
Rw → Pr	27.88±1.57	32.57±1.15	35.82±1.28	37.93±1.31	41.96±1.18	**56.15±1.09**
Average	15.93±1.24	18.58±1.28	19.28±0.93	20.67±1.01	24.15±1.25	**35.13±1.03**

4.4 Comparison Results

Results on Office-31. The comparison results of target domain accuracy and standard deviation on six transfer tasks that are combined in pairs by the three domains (A, W, and D) from Office-31 dataset are reported in Table 1 and Table 2.

Table 1 reports the comparison results under the noise level $q = 0.2$. From table 1, we can observe that: PAPLUDA produces the best results on all six transfer tasks, and outperforms other comparison methods with significant performance gains. For example, the proposed approach outperforms the best alternative comparison methods by 18.09%, 17.74%, 15.17%, and 12.22% on the domain adaptation tasks A → W, D → W, W → D, and A → D respectively, which demonstrates the effectiveness of the proposed approach in addressing PLUDA problem.

Table 2 represents a harder case with noise level $q = 0.4$. Compared with the experimental results in Table 1, the performance of each method in general degrades. Nevertheless, the proposed approach consistently outperforms all the other methods even in such a harder noise level case. Moreover, the performance gains yield by the proposed approach upon all the other comparison methods are quite notable in several cases. For example, it significantly outperforms the best alternative comparison methods by 29.36%, 26.55%, 23.71%, and 16.06% on the domain adaptation tasks W → D, A → W, D → W, and A → D respectively, which again validates the efficacy of the proposed approach.

From the 1 and Table 2, we can see that: it is reasonable that the performance of each comparison method degrades when they encounter such a harder case from the noise level $q = 0.2$ to $q = 0.4$; while the performance gains between the proposed method and the best alternative comparison method becomes larger. For example, the performance gains on the domain adaptation tasks W → D, A → W, D → W, and A → D from 15.17%, 18.09%, 17.74%, and 12.22% to 29.36%, 26.55%, 23.71%, and 16.06% respectively when they perform experiments with a noise level from $q = 0.2$ to $q = 0.4$. Note that performance of the proposed approach on the domain adaptation tasks A → W, W → D, and D → W actually is quite stable with increase of the noise level from 0.2 to 0.4, which demonstrates that the proposed PAPLUDA model is robust against the noisy labels.

In summary, these results on Office-31 dataset under different noise levels clearly demonstrate the effectiveness of the proposed method in dealing with the PLUDA problem.

Results on Office-Home. Table 3 reports the comparison results of the average target domain accuracy and standard deviation on twelve tasks that are combined in pairs by the four domains (Ar, CI, Pr, and Rw) from Office-Home. Table 3 shows the detail comparison results under the noise level $q = 0.2$, from which we can observe that: 1) Compared with all five domain adaptation methods, the proposed approach achieves the promising performance on the Office-Home dataset with twelve tasks, while some of them obtain the remarkable performance gains. For example, the proposed approach outperforms the best alternative comparison methods by 23.54%, 21.06%, 20.50% and 16.07% on the tasks Pr → Rw, Rw → CI, Rw → Ar and Pr → CI respectively. 2) Out of the total 60 comparison cases over 5 comparison methods and 12 tasks, the proposed approach outperforms all the comparison methods in all cases, which is not easy given the comparison methods with different

Table 4: Results of ablation study on various tasks

Ablation variant	D → A (q = 0.2)	D → A (q = 0.4)	Ar → Pr (q = 0.2)	Rw → Pr (q = 0.2)
CLS-w/o-pa	25.35	18.85	18.67	40.05
CLS-w/o-cpc	38.95	30.12	22.36	48.82
CLS-w/o-plr	29.24	23.55	19.68	42.88
CLS-w/o-tch	44.53	33.08	22.61	54.37
CLS	18.77	12.73	11.35	19.86
Full Model	46.75	37.39	25.28	56.15

advantages on various classification tasks. These reslults on Office-Home again validate the efficacy of the proposed method in addressing PLUDA problem.

4.5 Ablation Study

The proposed approach mainly contains three components that contribute the PLUDA learning: the pseudo-label refinement, the inter-domain class-prototype alignment loss, and the teacher-student model based contrastive regularization loss. Moreover, the effectiveness of pseudo-label refinement relies on the reliable teacher model to conduct the soft label disambiguation. To validate the importance of each component, we perform an ablation study to compare the proposed method with the following ablation variants: 1) CLS-w/o-pa, which drops the class-prototype alignment loss; 2) CLS-w/o-cpc, which drops the teacher-student model based contrastive loss; 3) CLS-w/o-plr, which drops the pseudo-label refinement; 4) CLS-w/o-tch, which drops the teacher model F' and uses the prediction network F instead; 5) CLS, which only use the classification loss by dropping all the other components including the class-prototype alignment loss, the teacher-student model based contrastive loss, and the pseudo-label refinement. The comparison results are reported in Table 4. Compared to the full model, we can observe that all the ablation variants have some performance degradation to different degree. Moreover, all four ablation variants CLS-w/o-pa, CLS-w/o-cpc, CLS-w/o-plr, and CLS-w/o-tch perform well than the ablation baseline CLS, which demonstrates that all the components are important and contribute to the full model with their different strength respectively. Among the four ablation variants, we can see that the CLS-w/o-pa, which drops the class-prototype alignment loss, leads to a large performance degradation no matter what is the noise level. It suggests that the class-prototype alignment loss is critical for the proposed method in addressing the domain discrepancy across both domains. It is interesting that the variant CLS-w/o-plr gets a larger performance degradation under the noise level q = 0.4 than q = 0.2. This makes sense since the pseudo-label refinement method can be more important in such a harder noise level case, that is to say the pseudo-label refinement method can be useless when noise level q infinitely approximates 0, in which case the PLUDA would be degraded into the UDA problem. We can observe the similar results from the variant CLS-w/o-tch that drops teacher model, which demonstrates that teacher model is important in refining the pseudo label and learning the consistent knowledge. The variant CLS-w/o-cpc also produces some performance degradation since the teacher-student model based contrastive regularization loss can enhance the stable pseudo-label refinement and improve the class prototype, thus consequently helps the inter-domain class-prototype alignment across both domains and improves the PLUDA learning. Overall, the ablation study results demonstrates that it is effective to integrate these parts in addressing the PLUDA problem.

5 Conclusion

In this paper, we propose a new challenging learning problem called partial label unsupervised domain adaptation (PLUDA), which addresses both PLL and UDA challenges simultaneously. Since there are naturally no methodologies designed for this learning scenario yet, we propose a novel prototype alignment PLUDA approach named PAPLUDA. It performs pseudo-label refinement in both source and target domains, and minimizes the domain discrepancy by an inter-domain prototype alignment method. In addition, a teacher-student model based contrastive regularization is deployed to enhance the stable prediction and hence improve the class prototype in both domains for PLUDA, thus consequently help the inter-domain prototype alignment to facilitate domain adaption. Extensive experimental results demonstrate that PAPLUDA achieves the state-of-the-art performance on the widely used benchmark datasets.

REFERENCES

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020a.

Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. Self-training avoids using spurious features under domain shift. *NeurIPS*, 2020b.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR*, 2020.

Jiahua Dong, Zhen Fang, Anjin Liu, Gan Sun, and Tongliang Liu. Confident anchor-induced multi-source free domain adaptation. In *NeurIPS*, 2021.

Zhen Fang, Jie Lu, Feng Liu, Junyu Xuan, and Guangquan Zhang. Open set domain adaptation: Theoretical bound and algorithm. *IEEE transactions on neural networks and learning systems*, 32(10):4309–4322, 2020.

Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In *NeurIPS*, 2020.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Mohsen Ghafoorian, Alireza Mehrtash, Tina Kapur, Nico Karssemeijer, Elena Marchiori, Mehran Pesteie, Charles RG Guttmann, Frank-Erik de Leeuw, Clare M Tempany, Bram van Ginneken, et al. Transfer learning for domain adaptation in mri: Application in brain lesion segmentation. In *MICCAI*, 2017.

Chen Gong, Tongliang Liu, Yuanyan Tang, Jian Yang, Jie Yang, and Dacheng Tao. A regularization approach for instance-based superset label learning. *IEEE transactions on cybernetics*, pp. 967–978, 2017.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *NeurIPS*, 2002.

Konstantinos Kamnitsas, Christian Baumgartner, Christian Ledig, Virginia Newcombe, Joanna Simpson, Andrew Kane, David Menon, Aditya Nori, Antonio Criminisi, Daniel Rueckert, et al. Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In *IPMI*, 2017.

Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. In *NeurIPS*, 2018.

Feng Liu, Jie Lu, Bo Han, Gang Niu, Guangquan Zhang, and Masashi Sugiyama. Butterfly: A panacea for all difficulties in wildly unsupervised domain adaptation. In *NeurIPS LTS Workshop*, 2019.

Liping Liu and Thomas Dietterich. A conditional multinomial mixture model for superset label learning. *NeurIPS*, 2012.

Liping Liu and Thomas Dietterich. Learnability of the superset label learning problem. In *ICML*, 2014.

Jie Luo and Francesco Orabona. Learning from candidate labeling sets. *NeurIPS*, 2010.

Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *ICML*, 2020.

Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *CVPR*, 2021.

Nam Nguyen and Rich Caruana. Classification with partial labels. In *SIGKDD*, pp. 551–559, 2008.

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2):199–210, 2010.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

Kendrick Shen, Robbie M Jones, Ananya Kumar, Sang Michael Xie, Jeff Z HaoChen, Tengyu Ma, and Percy Liang. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In *ICML*, 2022.

Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Transferable curriculum for weakly-supervised domain adaptation. In *AAAI*, 2019.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

Dong Wang and Thomas Fang Zheng. Transfer learning for speech and language processing. In *APSIPA*, 2015.

Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. Pico: Contrastive label disambiguation for partial label learning. In *ICLR*, 2022.

Hongwei Wen, Jingyi Cui, Hanyuan Hang, Jiabin Liu, Yisen Wang, and Zhouchen Lin. Leveraged weighted loss for partial label learning. In *ICML*, 2021.

Xuan Wu and Min-Ling Zhang. Towards enabling binary decomposition for partial label learning. In *IJCAI*, pp. 2868–2874, 2018.

Renchunzi Xie, Hongxin Wei, Lei Feng, and Bo An. Gearnet: Stepwise dual learning for weakly supervised domain adaptation. *arXiv preprint arXiv:2201.06001*, 2022.

Ning Xu, Congyu Qiao, Xin Geng, and Min-Ling Zhang. Instance-dependent partial label learning. In *NeurIPS*, 2021.

Yan Yan and Yuhong Guo. Partial label learning with batch label correction. In *AAAI*, 2020.

Yao Yao, Jiehui Deng, Xiuhua Chen, Chen Gong, Jianxin Wu, and Jian Yang. Deep discriminative cnn with temporal ensembling for ambiguously-labeled image classification. In *AAAI*, 2020.

Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. In *ACML*, 2016.

Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017.

Min-Ling Zhang and Fei Yu. Solving the partial label learning problem: An instance-based approach. In *IJCAI*, 2015.

Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, pp. 2155–2167, 2017.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1): 44–53, 2018.

Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. Supervised representation learning: Transfer learning with deep autoencoders. In *IJCAI*, 2015.

Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019.

A APPENDIX

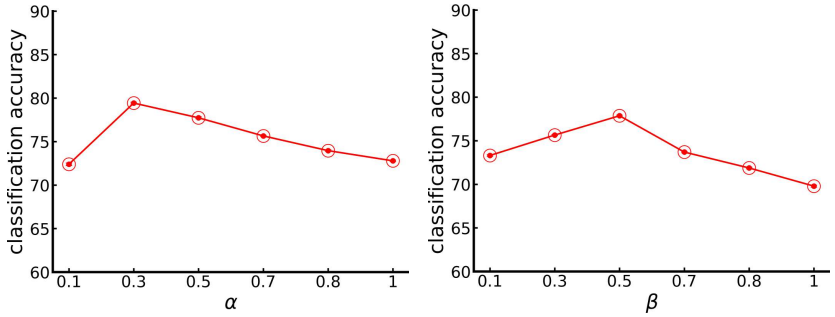
In this appendix, we provide the complete implementation of our proposed approach in Algorithm 1. Moreover, the additional experimental results is provided to further validate the effectiveness of the proposed approach, and the results of parameter sensitivity analysis is reported to study how would the trade-off parameters impact performance of the proposed model. In addition, we conduct extended experiments on the W $\rightarrow$ D task from Office-31 dataset to show how would the noise level affect performance of the proposed PAPLUDA method.

Algorithm 1 Training Algorithm of PAPLUDA.

Input:
D_S: partially labeled source dataset. D_T: unlabeled target dataset. F: prediction network. F': teacher model. G: feature extractor. H: projection network. α, β: trade-off hyperparameters.

for number of training iterations **do**
  Sample a mini-batch of source samples $B_S = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{|B|}$ from partially labeled source dataset D_S and target samples $B_T = \{\mathbf{x}_i^t\}_{i=1}^{|B|}$ from unlabeled target dataset D_T respectively.
  Perform pseudo-label refinement on B_S according to Eq.(3).
  Compute and update the class-prototype according to Eq.(5) and Eq.(7).
  Compute classification loss $\mathcal{L}_c$ via Eq.(4) by taking the corrected pseudo-label as the target.
  Perform pseudo-label refinement on B_T according to Eq.(6).
  Compute the inter-domain class-prototype alignment loss $\mathcal{L}_{inter}$ via Eq.(8).
  Compute the teacher-student model based contrastive regularization loss $\mathcal{L}_{reg}$ via Eq.(10).
  Update $\{F, G, H\}$ using the overall batch-wise training loss Eq.(11).
  Update the teacher model according to Eq.(1).
end for

A.1 PARAMETER SENSITIVITY ANALYSIS



(a) Vary the parameter α (b) Vary the parameter β

Figure 2: Test performance with different α and β value for the proposed approach on task A $\rightarrow$ W from Office-31.

In this subsection, we study the trade-off parameters α and β with different values from $\{0.1, 0.3, 0.5, 0.7, 0.8, 1\}$ to see how would the trade-off parameters affect the performance of the model. Note that a larger α represents larger weight to the inter-domain class-prototype alignment loss. We adjust one parameter while keep the other one with its best performance.

Figure 2 reports the test classification accuracy results on unlabeled target data for trade-off parameter α and β with different values. From the Figure 2, we can see that when α is very small, the

performance is relatively poor. With the increase of α, the performance improve, which suggests that the inter-domain class-prototype alignment loss works and begins to contribute to the model. When α is too large, the performance degrades as the inter-domain class-prototype alignment loss dominants gradually. This is reasonable since we just expect the inter-domain class-prototype alignment loss help the model training rather than dominant the learning procedure. For the parameter β, we can see the similar performance changes as the parameter α.

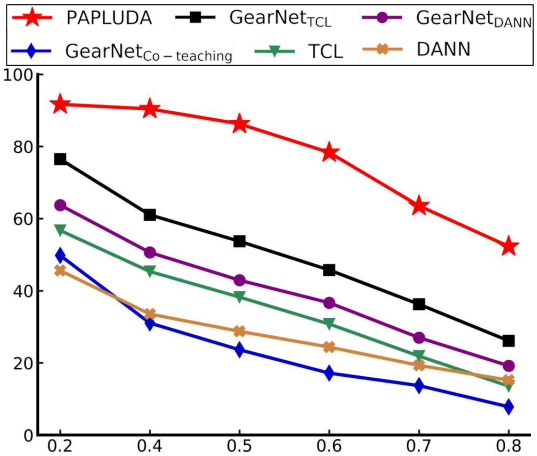


Figure 3: Test performance with different noise level q value for PAPLUDA on the W $\rightarrow$ D task from Office-31 dataset.

A.2 Impact of Noise Level

To further validate the effectiveness of the proposed PAPLUDA method, we perform extended experiments to see how the noise level q impacts the performance of PAPLUDA. As shown in Figure 3, we plot the experimental results on the domain adaptation task W $\rightarrow$ D with different noise level values. From the Figure 3, we can observe that: with the increase of the noise level q, thus consequently increase the number of condidate labels, the performance of all the comparison methods degrade significantly. Nevertheless, the proposed PAPLUDA consistently outperforms all the other methods. Moreover, when encountered an extremely harder case with q = 0.8 (the average number of candidate labels is 25.0138 within 31 classes), the comparison methods produce the worse performance or even destroyed by the high noise level case; While the proposed PAPLUDA also can produce a reasonable performance with a large performance gains. This further demonstrate the effectiveness of PAPLUDA in pseudo label refinement and consequently boost the PLUDA learning.

Table 5: Accuracy (mean$\pm$std) comparisons on Office-Home datasets with partial labels on ambiguity level q = 0.1. The best result among each column is highlighted in bold.

Tasks	DANN	TCL	GearNet$_{\text{Co-teaching}}$	GearNet$_{\text{DANN}}$	GearNet$_{\text{TCL}}$	Ours
Ar $\rightarrow$ CI	14.45$\pm$1.25	16.13$\pm$1.55	15.87$\pm$0.47	17.55$\pm$1.12	19.94$\pm$0.81	**23.62$\pm$0.99**
Ar $\rightarrow$ Pr	22.21$\pm$1.17	24.05$\pm$1.36	25.61$\pm$0.78	26.35$\pm$1.25	**28.76$\pm$1.06**	27.37$\pm$1.21
Ar $\rightarrow$ Rw	29.12$\pm$2.23	31.55$\pm$0.72	33.28$\pm$0.71	34.13$\pm$1.36	**36.62$\pm$1.63**	35.27$\pm$1.19
CI $\rightarrow$ Ar	14.13$\pm$0.69	16.51$\pm$1.67	16.83$\pm$0.83	23.54$\pm$0.78	25.79$\pm$1.61	**26.50$\pm$0.92**
CI $\rightarrow$ Pr	20.18$\pm$1.14	22.47$\pm$1.43	28.24$\pm$0.86	30.66$\pm$1.18	**32.13$\pm$1.07**	31.52$\pm$1.26
CI $\rightarrow$ Rw	22.86$\pm$1.53	24.35$\pm$1.35	31.27$\pm$1.03	34.65$\pm$0.67	36.24$\pm$1.12	**36.57$\pm$0.81**
Pr $\rightarrow$ Ar	15.33$\pm$0.52	17.33$\pm$0.74	17.29$\pm$1.42	20.54$\pm$0.93	25.46$\pm$1.39	**29.16$\pm$0.87**
Pr $\rightarrow$ CI	16.62$\pm$1.93	18.25$\pm$1.81	15.98$\pm$1.26	19.22$\pm$1.25	23.28$\pm$1.55	**37.08$\pm$1.27**
Pr $\rightarrow$ Rw	29.41$\pm$1.16	32.46$\pm$0.88	35.18$\pm$0.81	32.65$\pm$1.32	38.63$\pm$1.25	**55.32$\pm$1.21**
Rw $\rightarrow$ Ar	22.37$\pm$1.22	25.58$\pm$1.21	24.17$\pm$1.05	27.42$\pm$0.78	33.79$\pm$1.57	**48.70$\pm$0.77**
Rw $\rightarrow$ CI	17.61$\pm$0.95	19.62$\pm$2.13	17.79$\pm$1.21	21.68$\pm$1.12	27.92$\pm$1.47	**46.43$\pm$1.26**
Rw $\rightarrow$ Pr	35.03$\pm$1.66	39.45$\pm$1.22	46.63$\pm$1.32	45.17$\pm$1.36	49.68$\pm$1.25	**59.58$\pm$1.15**
Average	21.61$\pm$1.29	23.98$\pm$1.34	25.68$\pm$0.98	27.79$\pm$1.09	31.52$\pm$1.32	**38.09$\pm$1.08**

A.3 ADDITIONAL EXPERIMENTAL RESULTS

To further validate the effectiveness of the proposed approach, we provide the additional experimental results on Office-Home dataset with noise level q = 0.1. Table 5 shows the detail comparison results of the average target domain accuracy and standard deviation on twelve tasks. From Table 5, we can see that the proposed method produced the best results in most cases. From Table 3 and Table 5, we can see that: all the comparison methods' performance degrades in general when vary the noise level from $q = 0.1$ to $q = 0.2$; while the proposed method performs better than the comparison methods in a harder noise level case with larger performance gains, which again demonstrates the efficacy of the proposed approach in addressing competitive label noise.