EasyChair Preprint

№ 13383

# BiFine: Bilateral Fine-Grained Alignment with Dual Channels for Partial Domain Adaptation

Zhongze Wu, Yitian Long, Shan You, Xiu Su, Jun Long, Yueyi Luo and Chang Xu



May 21, 2024

# BiFine: Bilateral Fine-Grained Alignment with Dual Channels for Partial Domain Adaptation

**Abstract.** Partial Domain Adaptation (PDA) often grapples with negative transfer when the target label space is a subset of the source domain's. Addressing this, we present BiFine, a dual-channel adversarial weighting framework for PDA that orchestrates a bilateral fine-grained alignment between domains. The Dual-Channel consists of two key components: the Shared-Private Weighting Diverger (SPW) and the Centroid-Based Similarity Discriminator (CSD). The SPW selectively modulates weights for shared classes, amplifying them to enhance positive transfer while suppressing those potentially leading to negative transfer from private source domain classes. Concurrently, CSD employs a bilateral strategy by adjusting target sample weights based on their cosine similarity to the centroids of shared source classes and attenuates intra-class variances to sharpen class boundaries. This holistic approach promotes a refined domain adaptation, securing closer alignment for shared classes and segregating outliers. Extensive evaluations on ImageCLEF, Office-31 and Caltech-office datasets affirm BiFine's efficacy, outperforming exsiting methods with classification accuracies of 91.99%, 97.78% and 96.49%, respectively.

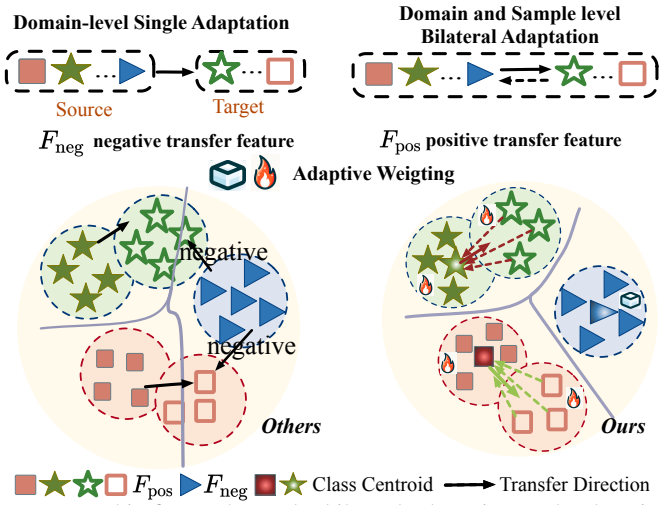


**Figure 1**: This figure shows the bilateral adaptation at the domain and sample levels using the BiFine method. Icons of fire and ice dynamically adjust the weights, amplifying the influence of $F_{pos}$ and diminishing that of $F_{neg}$, respectively.

## 1 Introduction

In machine learning, Domain Adaptation (DA) [21] plays a crucial role in adapting models from a labeled source domain to varied target domain distributions, particularly in out-of-distribution scenarios. Traditional DA approaches, however, often fall short in real-world applications where label spaces between domains shift. This challenge is addressed by Partial Domain Adaptation (PDA) [1], where the target domain's label space is a subset of the source domain's. PDA focuses on mitigating negative transfer caused by private class samples in the source domain, a common issue in conventional reweighted strategies [2, 3, 9, 23, 14].

Recently, several methods [9**?** , 16, 22, 23, 2, 3, 4, 24, 12] have emerged to tackle the challenges of Partial Domain Adaptation (PDA). The Selective Adversarial Network (SAN)[1] addresses the problem by designing a discriminator for each class in the source domain. While this enhances data distibution matching between classes, it also escalates computational demands and parameter counts as dataset scales. Building on the foundation laid by SAN, the Partial Adversarial Domain Adaptation (PADA)[2] method was introduced. It employs a single domain discriminator for data distribution alignment and introduces class-level weights to the source classifier. However, its performance can be constrained by shifts.

Recognizing the need for more refined judgment of source domain samples. The Importance Weighted Adversarial Nets (IWAN)[23] and the Example Transfer Network (ETN)[3] introduced an auxiliary domain discriminator to individually assess source domain samples, aiming to enhance the alignment between source and target domains. The Class Subset Selection Network (SSPDA)[24] proposed regularization terms to automatically reject abnormal samples, aiming to streamline the adaptation process. On the other hand, the Dual Alignment Approach (DAPDA)[12] emphasized minimizing both inter-domain and inter-class discrepancies through dual alignment techniques. CSDN [13] enhances PDA by dynamically weighting relevant source classes and focusing on ambiguous target samples. Yet, all above methods, they predominantly relied on a singular discriminative method to pinpoint source domain outlier classes. This singular focus becomes problematic, particularly when there's a pronounced similarity between cross-domain classes, leading to potential misalignments and reduced adaptation efficacy.

In this paper, the BiFine framework is introduced to bridge the distributional disparity between domains, as depicted in fig. 1, ensuring a more seamless alignment for shared classes. The SPW is manifested by the modulation of weights for shared and private source domain classes. Moreover, the essence of the CSD is captured through the strategic movement between the classes centroid. In this dual-channel approach, target samples are weighted based on their resemblance to shared source classes and distinctiveness within the source

domain classes is achieved by minimizing intra-class similarities.

The main contributions of this paper are as follows:

1) We propose BiFine, a Bilateral Fine-grained Alignment Network based on dual-channel adversarial weighting to selectively improve target transferability.

2) We design Shared-Private Weighting Diverger (SPW) that finely selects shared classes and dynamically curbs negative transfer from private classes.

3) We introduce the Centroid-Based Similarity Discriminator (CSD) for bilateral alignment, improving class distinction by aligning target with source class centroids.

4) Extensive experiments on ImageCLEF, Office31 and Caltech-office datasets show BiFine's effectiveness, achieving classification average accuracies of 91.99%, 97.78% and 96.49% respectively, surpassing exsisting methods.

## 2 Related work

**Domain Adaptation (DA).** The rise of deep learning has revitalized DA, notably through techniques that minimize distribution distances [19] or use adversarial methods for domain-invariant feature extraction [20]. Dual classifier training approaches [11] have also emerged, aligning classes by adversarially predicting unlabeled target samples. However, these methods often assume identical label spaces across domains, an assumption frequently violated in real-world applications.

**Partial Domain Adaptation (PDA)** addresses this challenge by focusing on scenarios where the target domain's labels are a subset of the source's. Techniques such as SAN [1], PADA [2], IWAN [23],[5] and ETN [3] introduce various weighting strategies for better adaptation of source instances. Innovations like DARL [4], DAPDA [12] and SSPDA [24] provide unique solutions, but some methods may induce negative transfer by including unknown class samples in alignment [12, 24, 16]. CSDN [13] aims for detailed matching and adaptive weighting, but its effectiveness is limited by a lack of class-specific discriminative knowledge. Existing PDA methods often focus on leveraging target predictions to weight source instances, neglecting the transferability of target samples and the risk of negative transfer from private classes.

## 3 Proposed Method

### 3.1 Definition of terminologies

In the field of machine learning, PDA represents a significant challenge, especially when dealing with diverse data distributions. In PDA, source domain data $X_s \in \mathbb{R}^{D\times n_s}$ with labels $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ are typically drawn from distribution $P(X_s)$. Conversely, target domain data $X_t \in \mathbb{R}^{D\times n_t}$ without labels $D_t = \{x_i^t\}_{i=1}^{n_t}$ come from a different distribution $P(X_t)$. Here, $D$ denotes the feature dimension, and $n_s$ and $n_t$ represent the number of source and target samples, respectively. Although the feature spaces of both domains are identical ($X_s = X_t$), the label space of the target domain $Y_t$ is a subset of the source domain's label space $Y_s$ ($Y_t \subseteq Y_s$), indicating a domain shift since $P_s \neq P_t$. The shared class samples in the source domain follow a distribution $P_{s,c}$, which differs from the target distribution $P_t$ ($P_{s,c} \neq P_t$).

### 3.2 Overall Framework

The BiFine framework utilizes a bilateral fine-grained adaptive approach at both sample and domain levels, as demonstrated in fig. 2.The framework employs SPW, leveraging an auxiliary classifier $\mathcal{C}_1$ and discriminator $\mathcal{D}_1$, to assign adaptive weights that emphasize shared classes through a weight inversion method detailed in eq. (1). Aditionally, CSD method enhances domain alignment by calculating cosine similarities $\mathbf{w}_2$ between target samples and source centroids, further refining this alignment to bolster shared class discrimination, as expounded in $\mathcal{L}_{ssd}$ (see eqs. (5) and (6)). Furthermore, the Domain Adversarial Weighting Network combines weights from both SPW and CSD to derive the final weight $\mathbf{w}$ (refer to eq. (7)), achieving a balance between classification loss $\mathcal{L}_{cls}$ and adversarial loss $\mathcal{L}_{adv}$ through a minimax game (outlined in eq. (8)), as defined in eqs. (9) and (10).

### 3.3 Shared-Private Weighting Diverger

The Shared-Private Weighting Diverger (SPW) is designed to amplify the influence of shared class samples from the source domain while minimizing the impact of private classes. In this approach, an auxiliary classifier and discriminator system is utilized to initially assign weights $\hat{w}(\mathbf{X}_s)$ to the source domain data, where a weight of 1 indicates a source sample and 0 indicates a target sample.

The key innovation lies in the adjustment of sample weights to prioritize shared classes. This is achieved by inverting the initial weights:

$$w_1(\mathbf{x}_i^s) = 1 - \hat{w}(\mathbf{x}_i^s), \tag{1}$$

where $w_1(\mathbf{x}_i^s)$ is the adjusted weight for the source sample $\mathbf{x}_i^s$, thereby enhancing the prominence of shared classes in the learning process.

The auxiliary classifier is trained exclusively on source domain data, leading to the auxiliary classification loss $\mathcal{L}_{\text{aux}}^{\text{cls}}$:

$$\begin{aligned}\mathcal{L}_{\text{aux}}^{\text{cls}} = -\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{c=1}^{C}\big[&y_{i,c}^s \log(\hat{y}_{i,c}^s) \\ &+ (1-y_{i,c}^s)\log(1-\hat{y}_{i,c}^s)\big],\end{aligned} \tag{2}$$

where $C$ the number of source domain classes and $y_{i,c}^s$, $\hat{y}_{i,c}^s$ denoting the true label and the predicted probability for the $i$-th source sample in class $c$, respectively.

Conversely, the auxiliary discriminator is trained using both source and target domain data, yielding the auxiliary adversarial loss $\mathcal{L}_{\text{aux}}^{\text{adv}}$:

$$\begin{aligned}\mathcal{L}_{\text{aux}}^{\text{adv}} = &-\frac{1}{n_s}\sum_{i=1}^{n_s}\log\left(\mathcal{D}_1(\mathcal{F}(\mathbf{x}_i^s))\right) \\ &-\frac{1}{n_t}\sum_{j=1}^{n_t}\log\left(1-\mathcal{D}_1(\mathcal{F}(\mathbf{x}_j^t))\right).\end{aligned} \tag{3}$$

### 3.4 Centroid-Based Similarity Discriminator

The proposed CSD, leveraging two key strategies for effective bilateral fine-grained alignment based on the similarity between target and source domains.

***Cosine Similarity with Source Centroid.*** We utilize the cosine similarity between the target batch samples and the source domain class centroids to infer the likelihood of a class being shared. High cosine similarity scores indicate that a target sample closely aligns with a source class, suggesting shared class characteristics. For a given source centroid vector $\mathbf{S} = [s_1, s_2, \ldots, s_{mb}]$ and a target sample batch $\mathbf{X}_i^t = [x_1, x_2, \ldots, x_{mt}]$, weights are assigned as:

$$w_{mb} = \mathbf{X}_i^t \cdot (\mathbf{S} \cdot \mathbf{1}_{mt\times mb}), \tag{4}$$

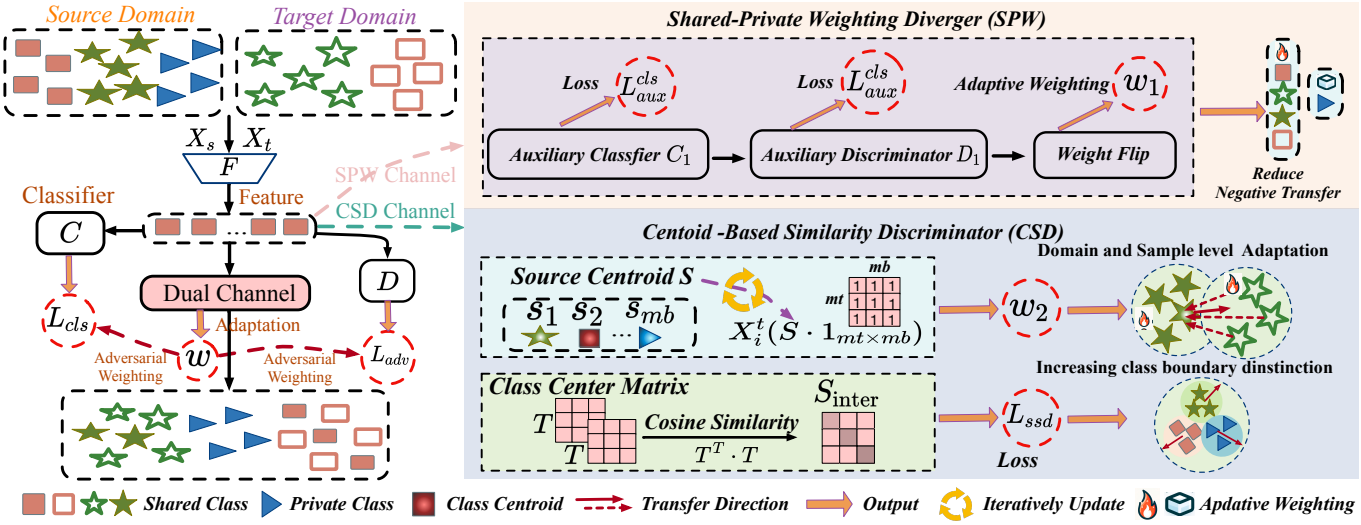


**Figure 2**: The Framework of proposed BiFine, where $F$ represents feature extractors, $C$ represents classifier, $D$ represents discriminator, $C_1$ represents auxiliary classifier, $D_1$ represents auxiliary discriminator, $S_{inter}$ represents the target and source centroid similarity discrimination, fire and ice icon represents increasing and decreasing weight respectively.

where $\mathbf{1}_{mt\times mb}$ facilitates the dimensional alignment of $\mathbf{S}$ and $\mathbf{X}_i^t$. After obtaining $w_{mb}$, the class weights corresponding to the $i$-th batch of source domain samples are extracted to form a new weight vector $\mathbf{w}_2$.

***Discriminative Source Space Construction.*** To address the challenge of data disorder and distribution disparities, we develop a discriminative source space based on cosine similarity. This approach focuses on widening the inter-class gaps within the source domain to foster more accurate classification. Assuming $\mathbf{T} = [t_1, t_2, \ldots, t_{cs}]$ represents the normalized source-class center matrix, we define the inter-class cosine similarity matrix as:

$$\mathbf{S}_{\text{inter}} = \mathbf{T}^T \cdot \mathbf{T}. \tag{5}$$

Subsequently, to quantify the inter-class differences and enhance classification boundaries, we compute a scalar loss:

$$\mathcal{L}_{ssd} = \gamma \sum_{i<j} \mathbf{S}_{\text{inter}}(i,j), \tag{6}$$

which aims to reduce inter-class similarity and enforce class separability in the source domain.

### 3.5 Dual-Channel Based Adversarial Weighting

In this work, BiFine computes the final weight $w(x)$ from the cosine similarity of vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, which are outputs of the SPW and CSD channel. The final weight $w(x)$ for a source domain sample is computed as:

$$w(x) = \frac{\mathbf{w}_1 \cdot \mathbf{w}_2}{\text{mean}(\mathbf{w}_1 \cdot \mathbf{w}_2)}, \tag{7}$$

where mean denotes the mean dot product of $\mathbf{w}_1$ and $\mathbf{w}_2$ across the source domain.

Then, this weight is applied to modulate the classification and adversarial losses within a minimax game:

$$\min_{\theta_f,\theta_y} \max_{\theta_d} \left[ \frac{1}{n_s} \sum_{i=1}^{n_s} w(x_i^s) \left( \mathcal{L}_{\text{cls}}^i(\theta_f,\theta_c) - \lambda \mathcal{L}_{\text{adv}}^i(\theta_f,\theta_d) \right) \right], \tag{8}$$

where $\theta_f$, $\theta_y$ and $\theta_d$ are the parameters to be learned, $n_s$ is the number of source domain samples, $\mathcal{L}_{\text{cls}}$ is the classification loss, $\mathcal{L}_{\text{adv}}$ is the adversarial loss and $\lambda$ is the trade-off parameter between the two losses.

The classification loss $\mathcal{L}_{\text{cls}}$ is detailed as:

$$\begin{aligned} \mathcal{L}_{\text{cls}} = & \frac{1}{n_s} \sum_{i=1}^{n_s} w(x_i^s) \left[ \mathcal{L}_{\text{ent}}\left(\mathcal{C}(\mathcal{F}(x_i^s)), y_i^s\right) \right. \\ & \left. + \mathcal{L}_{\text{ent}}\left(\mathcal{C}_1(\mathcal{F}(x_i^s)), y_i^s\right) \right], \end{aligned} \tag{9}$$

where $\mathcal{L}_{\text{ent}}$ represents the entropy loss and defined in eq. (11).

The adversarial loss $\mathcal{L}_{\text{adv}}$, which focuses on domain classification, is given by:

$$\begin{aligned} \mathcal{L}_{\text{adv}} = & -\frac{1}{n_s} \sum_{i=1}^{n_s} w(x_i^s) \log\left(\mathcal{D}(\mathcal{F}(x_i^s))\right) \\ & -\frac{1}{n_t} \sum_{j=1}^{n_t} \log\left(1 - \mathcal{D}(\mathcal{F}(x_j^t))\right). \end{aligned} \tag{10}$$

The entropy loss $\mathcal{L}_{\text{ent}}$, applied to both the primary and auxiliary classifiers, aims to minimize uncertainty in target label predictions:

$$\mathcal{L}_{\text{ent}} = -\sum_{j=1}^{n_t} \hat{y}_j \log(\hat{y}_j), \tag{11}$$

where $\hat{y}_j$ denotes the pseudo-labels for the target samples.

**Overall Loss Function.** The aggregated loss for the BiFine network, trainable end-to-end, is given by:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{cls}} + \beta \mathcal{L}_{\text{adv}} + \gamma \mathcal{L}_{\text{aux}}^{\text{adv}} + \delta \mathcal{L}_{\text{aux}}^{\text{cls}} + \zeta \mathcal{L}_{\text{ssd}}, \tag{12}$$

where the hyperparameters $\alpha, \beta, \gamma, \delta$ and $\zeta$ balance each loss component's contribution to the training objective.

## 4 Experiments

### 4.1 Experimental setup

We evaluate our model's performance via experiments on established benchmarks: Office-31[18], ImageCLEF [1] and Caltech-office

[1] https://www.ImageCLEF.org/2014/adaptation

[8]. Following the same settings as [3], we train our model lasts for 1000 iterations, using SGD optimizer (learning rate: 0.001, weight decay: 0.05) and batch size of 36. To fine-tune the hyperparameters, we use an importance-weighted cross-validation scheme, which ensures effective optimization tailored to the specific needs of our model.

**Table 1**: CLASSIFICATION ACCURACY (%) ON ImageCLEF DATASET

| UDA | | | | | | | |
|---|---|---|---|---|---|---|---|
| Method | $I \to P$ | $P \to I$ | $I \to C$ | $C \to I$ | $C \to P$ | $P \to C$ | Avg |
| Resnet-50[10] | 74.80 | 83.90 | 91.50 | 78.00 | 65.50 | 91.20 | 80.70 |
| DAN[15] | 74.50 | 82.20 | 92.80 | 86.30 | 69.20 | 89.80 | 82.50 |
| DANN[7] | 75.00 | 86.00 | 96.20 | 87.00 | 74.30 | 91.50 | 85.00 |
| DRMEA[17] | 80.70 | 92.50 | 97.20 | 90.50 | 77.70 | 96.20 | 89.10 |
| **BiFine** | 80.21 | **92.74** | 96.28 | **92.57** | **78.15** | 95.95 | **89.32** |
| PDA | | | | | | | |
| Method | $I \to P$ | $P \to I$ | $I \to C$ | $C \to I$ | $C \to P$ | $P \to C$ | Avg |
| Resnet-50[10] | 79.16 | 95.56 | 99.31 | 83.33 | 52.05 | 52.84 | 77.04 |
| PADA[2] | 82.49 | 91.68 | 96.67 | 91.33 | 78.79 | 93.33 | 89.05 |
| ETN[3] | 81.82 | 92.00 | 97.00 | 94.00 | 79.46 | 96.00 | 90.05 |
| SAFN[22] | 79.5 | 90.7 | 93.0 | 90.3 | 77.8 | 94.0 | 87.5 |
| DMP [16] | 82.4 | 94.5 | 96.7 | 94.3 | 78.7 | 96.4 | 90.5 |
| **BiFine** | **84.38** | 94.34 | 97.67 | **95.34** | **82.91** | **97.34** | **91.99** |

### *4.2 Experimental Results*

**Results Analysis on ImageCLEF.** The analytical results on the ImageCLEF dataset, as shown in table 1. In the rigorous tasks, BiFine notably excels, especially in the $C \to I$ task with a $92.57\%$ accuracy, eclipsing DRMEA [17] by $2.07\%$.This improvement is credited to BiFine's bilateral fine-grained domain alignment that maintains target data adaption direction, unlike DRMEA's risk-prone pseudo labels. In tasks like $C \to P$, BiFine's selective source class weighting circumvents the pitfalls of negative transfer that SAFN [22] encounters, leading to a substantial $5.11\%$ accuracy improvement by focusing on the shared classes and utilizing a more precise sample and domain adaptation strategy.

**Results Analysis on Office-31.** BiFine's performance on Office-31 dataset's are reported in table 3. Achieving an accuracy of $97.22\%$ in the $A \to W$ task, BiFine outperforms the ETN [3] by $2.70\%$, showcasing the advantage of its class-specific weighting mechanism which precisely discriminates features relevant to the target domain. Additionaly, in fig. 4, BiFine converges faster than ETN even though the maximum accuracy is the same. Furthermore, BiFine's robust feature space alignment exceeds DARL [4] and PADA [2], especially in the $D \to A$ task with an accuracy of $95.46\%$, which is a significant increase over DARL's $94.57\%$. Overall, BiFine presents an impressive average accuracy of $97.78\%$ across tasks,which shows effectively mitigates domain shift and avoids the negative transfer.

**Results Analysis on Caltech-office.** The results on Caltech-office is shown in table 2. In the $C10 \to A5$ task, BiFine achieves an accuracy of 96.98%, which is 1.55% higher than DAPDA's [12] method. This accurate class alignment demonstrates BiFine's superior ability in situations where methods such as DAPDA may not be able to reach due to their rigid alignment strategies. Furthermore, in the $W10 \to C5$ task, BiFine's 93.92% accuracy, a significant 2.92% improvement over the AR method, showcases its flexible bidirectional fine-grained alignment and robustness against negative transfer.

### *4.3 Feature Visualization*

The effectiveness of the BiFine approach is visualized using t-SNE [6] for the $A \to W$ task of the Office-31 dataset, as shown in fig. 3. Initially, the use of ResNet-50 for feature embeddings results in a scattered distribution, suggesting a lack of clear domain alignment, especially in the A $\to$ W scenario. This scattered distribution could potentially lead to misclassification. However, after applying BiFine, there is a noticeable improvement. The embeddings show not only a more compact intra-class grouping but also a closer alignment between the centroids of shared classes, marked as T0-T9 and S0-S9. This improvement indicates that BiFine effectively promote class-wise separation and reduce negative transfer.

### *4.4 Ablation Study*

In our ablation studies, we evaluated the components of the BiFine framework, with findings summarized in table 4. The implementation of the SPW module alone slightly improved performance to $95.83\%$, highlighting its role in emphasizing shared class features. In contrast, using only the CSD module notably increased accuracy to $96.18\%$, demonstrating its effectiveness in aligning domain features and minimizing intra-class variations. Combining SPW and CSD in the BiFine framework led to a significant performance boost, reaching an average accuracy of $96.49\%$. This was particularly evident in complex tasks like $C10 \to A5$ and $W10 \to C5$, where this dual approach was highly effective. The BiFine framework, through SPW, adeptly filters outlier classes from the source domain to mitigate negative transfer. Simultaneously, CSD facilitates bilateral fine-grained alignment, accommodating label space shifts in Partial Domain Adaptation (PDA) scenarios.

## 5 CONCLUSION

In this paper, we present a Bilateral Fine-Grained Alignment Network (BiFine) for PDA. BiFine stands out by implementing a dual-channel architecture for bilateral adaptation at both the domain and sample levels. The BIFINE's Shared-Private Weighting (SPW) component dynamically adjusts class weights, reducing negative transfer, while the Centroid-Based Similarity Discriminator (CSD) ensures bilateral fine-grained alignment. These mechanisms work in adversarial training to significantly improve the performance of PDA tasks. Extensive experiments on datasets such as ImageCLEF, Office-31 show that BiFine achieves average classification accuracies of 91.99% and 97.78% respectively, surpassing the performance of original setting by 1.94% and 1.02%.

## References


[1] 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*, pages 2724–2732, 2018.

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

[3] 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*, pages 2985–2994, 2019.

[4] 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*, 33(2):539–553, 2020.

**Table 2**: Classification accuracy (%) on Caltech-Office dataset

| Method | C10→A5 | C10→W5 | C10→D5 | A10→C5 | W10→C5 | D10→C5 | Avg |
|---|---|---|---|---|---|---|---|
| Resnet-50[10] | 91.86 | 93.33 | 80.88 | 80.99 | 59.08 | 59.08 | 77.54 |
| IWAN[23] | 94.22 | 97.78 | 98.53 | 89.90 | 90.24 | 91.61 | 93.71 |
| PADA[2] | 93.79 | 89.63 | 94.11 | 86.30 | 86.64 | 82.02 | 88.75 |
| ETN[3] | 96.15 | 100.00 | 98.53 | 91.26 | 93.15 | 93.32 | 95.40 |
| DARL[4] | 96.36 | 98.52 | 100.00 | 92.47 | 93.15 | 92.64 | 95.52 |
| DAPDA[12] | 98.66 | 97.93 | 100.00 | 93.05 | **94.35** | **94.62** | 96.43 |
| BiFine | 96.98 | **100.00** | **100.00** | **93.40** | 93.92 | **94.62** | **96.49** |

**Table 3**: Accuracy (%) on Office-31 Dataset

| Method | $A \rightarrow W$ | $D \rightarrow W$ | $W \rightarrow D$ | $A \rightarrow D$ | $D \rightarrow A$ | $W \rightarrow A$ | Avg |
|---|---|---|---|---|---|---|---|
| Resnet-50[10] | 79.16 | 95.56 | 99.31 | 83.33 | 52.05 | 52.84 | 77.04 |
| DRMEA[17] | 90.64 | **100.00** | 98.70 | 80.52 | 71.48 | 68.20 | 84.92 |
| SAN[1] | 93.90 | 99.32 | 99.36 | 94.27 | 94.15 | 88.73 | 94.96 |
| IWAN[23] | 89.15 | 99.32 | 99.36 | 90.45 | 95.62 | 94.26 | 94.69 |
| PADA[2] | 86.54 | 99.32 | **100.00** | 82.17 | 92.69 | 95.41 | 92.69 |
| ETN[3] | 94.52 | **100.00** | **100.00** | 95.03 | **96.21** | 94.64 | 96.73 |
| SSPDA[24] | 93.42 | 97.62 | **100.00** | 90.43 | 93.45 | 95.53 | 95.07 |
| DARL[4] | 94.58 | 99.66 | **100.00** | 98.73 | 94.57 | 94.26 | 96.97 |
| DAPDA[12] | 95.06 | **100.00** | **100.00** | 92.15 | 95.13 | 97.40 | 96.62 |
| RCDA[5] | 97.13 | 97.58 | 95.93 | **100.00** | 95.82 | **100.00** | 97.74 |
| CSDN[13] | **98.93** | **100.00** | **100.00** | 98.73 | 94.26 | 94.63 | 97.76 |
| DMP [16] | 96.6 | **100.00** | **100** | 96.4 | 95.1 | 95.4 | 97.20 |
| AR [9] | 93.5 | **100.00** | 99.7 | 96.8 | 95.5 | 96.0 | 96.90 |
| **BiFine** | 97.22 | **100.00** | **100.00** | 98.72 | 95.46 | 95.25 | **97.78** |

**Table 4**: Ablation studies On Caltech-office dataset

| +SPW | +CSD | C10→A5 | C10→W5 | C10→D5 | A10→C5 | W10→C5 | D10→C5 | Avg |
|---|---|---|---|---|---|---|---|---|
| ✗ | ✗ | 96.16 | 100.00 | 98.53 | 91.26 | 93.15 | 93.32 | 95.40 |
| ✓ | ✗ | 96.27 | 100.00 | 98.49 | 92.79 | 93.25 | 94.35 | 95.83 |
| ✗ | ✓ | 96.63 | 100.00 | 100.00 | 92.86 | 93.20 | 94.16 | 96.18 |
| ✓ | ✓ | **96.98** | **100.00** | **100.00** | **93.40** | **93.92** | **94.62** | **96.49** |

**Table 5**: Stability and GPU throughput comparison on Caltech-office Dataset.

| VQA-RAD | | | | |
|---|---|---|---|---|
| Method | Closed | Open | Overall | GPU Throughput (samples/sec) |
| LLaVA | 65.02 ± 0.12 | 49.95 ± 0.18 | - | 2.86 |
| LLaVA-Med | 84.15 ± 0.13 | 61.5 ± 0.16 | - | 3.23 |
| MedVInT-PMC-VQA | 86.75 ± 0.1 | 73.65 ± 0.15 | 81.55 ± 0.12 | 2.57 |
| Uni-Med | 87.15 ± 0.08 | 74.18 ± 0.12 | 81.95 ± 0.09 | 2.48 |
| SLAKE | | | | |
| Method | Closed | Open | Overall | GPU Throughput (samples/sec) |
| LLaVA | 63.2 ± 0.14 | 78.1 ± 0.17 | - | 2.78 |
| LLaVA-Med | 85.3 ± 0.12 | 83.05 ± 0.15 | - | 3.12 |
| MedVInT-PMC-VQA | 86.25 ± 0.1 | 84.45 ± 0.13 | 85.15 ± 0.11 | 2.51 |
| Uni-Med | 87.48 ± 0.07 | 85.25 ± 0.11 | 86.05 ± 0.10 | 2.42 |


[5] S. Choudhuri and A. Sen. Robust class-conditional distribution alignment for partial domain adaptation. *arXiv preprint arXiv:2310.12060*, 2023.

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.

[7] 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*, 17(1): 2096–2030, 2016.

[8] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

[9] X. Gu, X. Yu, y. yang, J. Sun, and Z. Xu. Adversarial reweighting for partial domain adaptation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14860–14872. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/7ce3284b743aefde80ffd9aec500e085-Paper.pdf.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[11] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10285–10295, 2019.

[12] L. Li, Z. Wan, and H. He. Dual alignment for partial domain adaptation. *IEEE transactions on cybernetics*, 51(7):3404–3416, 2020.

[13] 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*, 53(9):5641–5654, 2023. doi: 10.1109/TCYB.2022.3163432.

[14] J. Liang, R. He, Z. Sun, and T. Tan. Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2975–2984, 2019.

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

[16] Y.-W. Luo, C.-X. Ren, D.-Q. Dai, and H. Yan. Unsupervised domain adaptation via discriminative manifold propagation. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1653–1669, 2020.

[17] Y. W. Luo, C. X. Ren, P. Ge, K. K. Huang, and Y. F. Yu. Unsupervised domain adaptation via discriminative manifold embedding and alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, page 5943. AAAI, 2020. doi: 10.1609/aaai.v34i04.5943.

[18] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 213–226. Springer, 2010.

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

[20] 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*, pages 7167–7176, 2017.

[21] Y. Xiao, P. Wei, C. Liu, and L. Lin. Adversarially robust source-free domain adaptation with relaxed adversarial training. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2681–2686, 2023. doi: 10.1109/ICME55011.2023.00456.

[22] R. Xu, G. Li, J. Yang, and L. Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation.

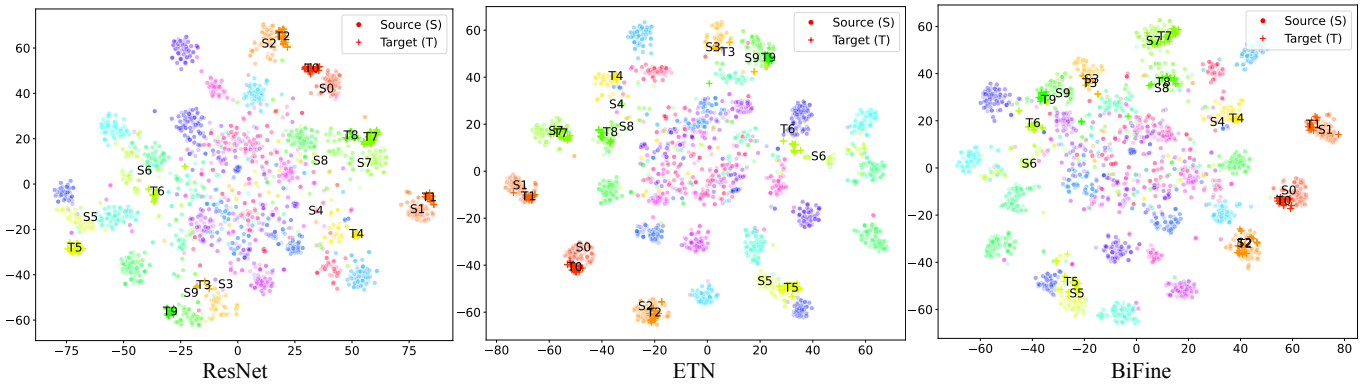


**Figure 3**: The t-SNE visualization of the Office-31 dataset $A \rightarrow W$ task. Shared classes are color-coded with "S" and "T" representing source ("o") and target domains ("+") respectively. "S1" represents the source class with class number 1.

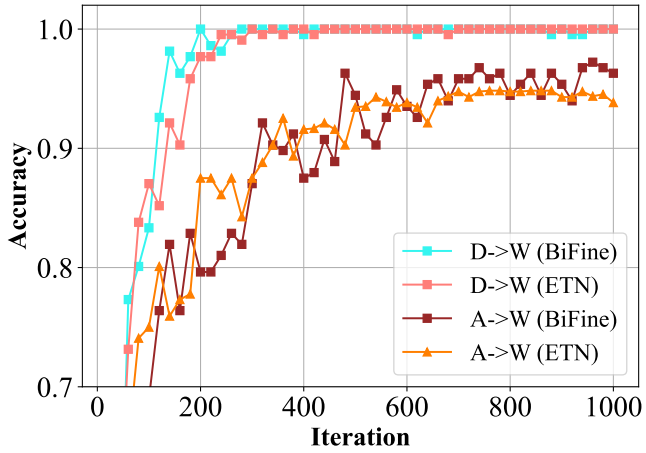


**Figure 4**: Testing accuracy on Office-31 dataset for the tasks $D \rightarrow W$ and $A \rightarrow W$.

In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1426–1435, 2019.

[23] J. Zhang, Z. Ding, W. Li, and P. Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8156–8164, 2018. doi: 10.1109/CVPR.2018.00851.

[24] F. Zohrizadeh, M. Kheirandishfard, and F. Kamangar. Class subset selection for partial domain adaptation. In *CVPR Workshops*, 2019.