DSP: Dual Soft-Paste for Unsupervised Domain Adaptive Semantic Segmentation

Li Gao∗
Wuhan University
China

Jing Zhang
The University of Sydney
Australia

Lefei Zhang†
zhanglefei@whu.edu.cn
Wuhan University
China

Dacheng Tao
taodacheng@jd.com
JD Explore Academy
China

ABSTRACT
Unsupervised domain adaptation (UDA) for semantic segmentation aims to adapt a segmentation model trained on the labeled source domain to the unlabeled target domain. Existing methods try to learn domain invariant features while suffering from large domain gaps that make it difficult to correctly align discrepant features, especially in the initial training phase. To address this issue, we propose a novel Dual Soft-Paste (DSP) method in this paper. Specifically, DSP selects some classes from a source domain image using a long-tail class first sampling strategy and softly pastes the corresponding image patch on both the source and target training images with a fusion weight. Technically, we adopt the mean teacher framework for domain adaptation, where the pasted source and target images go through the student network while the original target image goes through the teacher network. Output-level alignment is carried out by aligning the probability maps of the target fused image from both networks using a weighted cross-entropy loss. In addition, feature-level alignment is carried out by aligning the feature maps of the source and target images from student network using a weighted maximum mean discrepancy loss. DSP facilitates the model learning domain-invariant features from the intermediate domains, leading to faster convergence and better performance. Experiments on two challenging benchmarks demonstrate the superiority of DSP over state-of-the-art methods. Code is available at https://github.com/GaoLii/DSP.

KEYWORDS
unsupervised learning, semantic segmentation, convolutional neural networks, domain adaptation

ACM Reference Format:
Li Gao, Jing Zhang, Lefei Zhang, and Dacheng Tao. 2021. DSP: Dual Soft-Paste for Unsupervised Domain Adaptive Semantic Segmentation. In Proceedings of the 29th ACM International Conference on Multimedia (MM ’21), October 20–24, 2021, Virtual Event, China. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475186

1 INTRODUCTION
As one of the fundamental tasks in computer vision, semantic segmentation can be used as a preliminary step for many multimedia applications [3, 35, 36, 40], including image/video captioning, image-to-image translation, video content analysis. Training a well-performed deep semantic segmentation model usually requires a large amount of pixel-level labeled data, which is indeed very laborious and expensive to manually annotate. Alternatively, since it is much easier to generate synthetic images with dense pixel labels, e.g., via a 3D game engine, there are a lot of works focusing on training the segmentation model using synthetic labeled images. However, due to the appearance discrepancy between synthetic images and real images, which is also known as the domain shift, the model trained on synthetic images usually generalizes poorly on real images.

To address this issue, unsupervised domain adaptation (UDA) methods have been proposed to mitigate the domain shift between the source and target domains as shown in Figure 1(a). In the context of semantic segmentation, there are three categories of methods which perform domain adaptation at different levels, such as input level [8, 33], feature level [21, 41, 42], and output level [2, 14]. Input-level UDA methods aim to perform statistical matching at the input level to achieve uniformity in the visual appearance of the input images from different domains, e.g., style transfer [13]. Feature-level UDA methods aim to align the distribution of latent features (usually embedded by CNNs) in both domains to extract domain-invariant features, e.g., Maximum Mean Discrepancy (MMD) [19], adversarial learning [9]. Besides, since the predicted probability maps are in low-dimension and highly structured, it is effective to perform alignment of the probability maps from different domains, i.e., output-level UDA [2].
2 RELATED WORK

2.1 Mean Teacher-based Methods

Recently, the mean teacher framework [31] has been used by some methods for unsupervised domain adaptive semantic segmentation [2, 4]. These kinds of UDA methods perform output-level alignment by employing the consistency constraint on the target predictions from the student model and the teacher model, respectively. Though effective, they suffer from training instability and slow convergence due to inaccurate predictions on the unlabeled target domain, especially in the initial training phase. To address this issue, DACS [33] proposes to paste part of a source domain image onto the unlabeled target domain image, which leads to certain parts of the pseudo-labeled map always being injected with the ground truth semantic map, ensuring the accuracy of the target prediction. Although DACS benefits from the intermediate mixed target domain as shown in Figure 1(b), which helps to pull the target domain closer to the source domain, its performance is still limited due to the large gap between the two domains, incomplete structure, and inconsistent spatial layout issue by hard-paste, as well as the class imbalance issue.

In this paper, we go a step further and propose a novel paste method named dual soft-paste (DSP) for unsupervised domain adaptive semantic segmentation, which can create two new intermediate domains to facilitate aligning both source and target domains effectively, as shown in Figure 1. Specifically, DSP adopts a long-tail class first sampling strategy to select the candidate classes from a source domain template image and paste the corresponding image patch on both the source and target training images in a soft weighted-sum manner. It creates two new intermediate domains of fused images, which are used to perform domain alignment under the mean teacher framework. Technically, the pasted source and target images go through the student network while the original target image goes through the teacher network. Output-level alignment is carried out by aligning the probability maps of the fused target image from both networks using a weighted cross-entropy loss.

Feature-level alignment is also performed by aligning the feature maps of the fused source and target image from student networks using a weighted maximum mean discrepancy loss. The dual paste strategy guarantees that the same patch is shared by both domain images serving as an intermediate to bridge both domains and the soft-paste strategy preserves the original domain information by maintaining its structure layout, complete objects, as well as appearance styles. Consequently, DSP facilitates the model learning domain-invariant features from the intermediate domains, leading to faster convergence and better performance.

The contributions of this work can be summarized as follows:

- We propose a novel Dual Soft-Paste (DSP) method to create intermediate domains and facilitate domain alignment for semantic segmentation. DSP adopts a long-tail class first sampling strategy, which alleviates the class imbalance issue and shows its effectiveness in improving the performance.
- Based on the mean teacher framework, we propose a new UDA model by performing both feature-level and output-level alignment on the intermediate domains, which benefits from the softly pasted patches with ground truth labels.
- Extensive experiments on two challenging UDA semantic segmentation tasks, i.e., GTA5 to Cityscapes and SYNTHIA to Cityscapes, clearly demonstrate the superiority of the proposed model over state-of-the-art methods.

Figure 1: Illustration of different domain alignment paradigms. (a) Alignment on the original domains. (b) Alignment on the source domain and a mixed target domain, e.g., DACS [33]. (c) The proposed alignment method on two intermediate mixed domains by DSP. “SMD” and “TMD” refer to the source-mixed domain and target-mixed domain obtained by pasting source domain patches on the source and target domain images, respectively.
Figure 2: Overview of the proposed DSP model based on the mean teacher framework. The source image, source mixed image, and target mixed image are fed into the student network, while the target image is fed into the teacher network. DSP-induced output-level alignment and feature-level alignment are used to perform domain adaptation.

Different from DACS, we propose a novel dual soft-paste method to solve the aforementioned problems. First, our dual paste strategy can create two intermediate domains by pasting same source image patches on both the source and target images, which indeed serves as a bridge to reduce the gap between the two domains. Second, our soft-paste strategy can preserve the original domain information by keeping its structure layout, complete objects, as well as appearance styles. Third, we perform both feature-level and output-level alignment to learn domain-invariant features from the intermediate domains, leading to faster convergence and better performance.

2.2 Copy-and-Paste Strategies

There is a wide spectrum of work to improve the performance of deep models by using copy-and-paste methods for data augmentation in the supervised training setting. For example, CutMix [38] cut and pasted patches among training images where the labels are also mixed to the area of the patches. Remez et al. learned object masks by cutting-and-pasting with adversarial learning [26]. However, all these methods adopted the hard-paste strategy, which loses the original layout and semantic information of the original image when creating new images. MixUp [39] trained the network on convex combinations of image pairs and their labels with a mixing weight to address the aforementioned issue. Dwibedi et al. proposed to automatically cut object instances and paste them on random backgrounds to make detectors ignore these artifacts during training and generate data that gives competitive performance on real data [7]. FMix [11] proposed to use random binary masks obtained by applying a threshold to low frequency images sampled from fourier space. These random masks can take on a wide range of shapes and can be generated for use with one, two, and three dimensional data. In this paper, we also explore the soft-paste idea but specifically tailor it to the unsupervised domain adaptive semantic segmentation setting by handling the class imbalance issue and reducing the domain gap.

2.3 Self-Training

Self-training is a widely used strategy for semi/unsupervised learning by generating pseudo labels of the unlabeled data. For the semantic segmentation task, CBST [46] proposed an iterative self-training method that alternatively generated pseudo labels on target data via latent variable loss minimization and retrained the model using these labels. DAST [37] presented a discriminator attention-based self-training method to adaptively improve the decision boundary of the model for the target domain. IAST [23] developed a pseudo-label generation strategy, which uses an instance adaptive selector and a region-guided regularization to smooth the pseudo-label region and sharpen the non-pseudo-label region. Zheng et al. explicitly estimated the prediction uncertainty during training to rectify the pseudo label learning [43]. MetaCorrection [10] proposed to model the noise distribution of pseudo labels in the target domain to advance domain-aware meta learning. We also leverage the self-training idea but implement it together with the proposed DSP method under the mean teacher framework, where the target pseudo labels are generated by the teacher model, mixed with the pasted source patch labels, and updated during training.
3 METHOD

3.1 Preliminaries

Denoting the source domain by S, it contains images X_S and pixel-level labels Y_S, while the target domain T only contains unlabeled images X_T. The goal of UDA-based semantic segmentation is training a model on {X_S, Y_S, X_T} that can predict accurate semantic labels for X_T. To this end, we propose a novel DSP model under the mean teacher framework as illustrated in Figure 2. It has two segmentation networks, i.e., a student network f_θ with learnable parameters θ and a teacher network f_θ' with parameters θ' calculated by the exponential moving average (EMA) of θ'.

3.2 Dual Soft-Paste

3.2.1 Long-Tail Class First Sampling. Given the images from source domain S, we first calculate the frequency distribution of their classes as \{p_1, p_2, ..., p_c\}, i.e.,

\[
p_t = \frac{\sum_{i=1}^{N} c_{ij}}{N},
\]

where c represents the number of categories, c_{ij} indicates whether j-th source image contains class c_i, and N denotes the total number of images in S. Then, we choose the least frequent K categories as the long-tail classes and record those images containing these classes as a dataset D to facilitate the subsequent sampling process.

In this paper, we set K = 5 for the GTA5 dataset, including rider, bus, train, motorbike, bike. And k = 4 for the SYNTHIA dataset, including wall, light, bus, bike. During training, we randomly choose k long-tail classes and select k images from D, each of which contains at least one of the chosen long-tail classes. k is set to 2 in this paper. A hyper-parameter study of k is conducted in Section 4.4.

3.2.2 The Algorithm of Dual Soft-Paste. During training, we first randomly choose a source image from S and select half of its classes and corresponding image patch as the candidate patch used for subsequent pasting. Then, we choose long-tail classes and candidate images from S to form the final candidate patch.

Algorithm 1 The proposed dual soft-paste algorithm

Input: Source template image x_p and its label y_p, source image x_s, target image x_t, pre-defined long-tail dataset D, opacity β.

Output: The DSP mask M, mixed source image x_ps, mixed target image x_pt.

1: S_class ← set of classes appearing in y_p;
2: c ← randomly select \(\frac{|S_{\text{class}}|}{2}\) classes from S_class;
3: for each i, j do
4:     \(M(i, j) = \begin{cases} 1, & \text{if } y_p(i, j) \in c \\ 0, & \text{else} \end{cases}\)
5: end for
6: c_{tail} ← randomly select source images from D;
7: for each i, j do
8:     if \(y_{tail}(i, j) \in c_{tail}\) then \(M(i, j) = 1\)
9:     end if
10: end for
11: x_ps, x_pt ← calculate mixed images by Eq. 2 and Eq. 3
12: return M, x_ps, x_pt;

Figure 3: Visual demonstration of the DSP algorithm. Note that we show the binary DSP mask instead of the soft one by integrating the opacity β for better illustration.

For simplicity, we reuse M to represent βM by assigning the opacity value to those positive pixels in M. A hyper-parameter study of β is conducted in Section 4.4, which is set to 0.8 by default. The algorithm of DSP is presented in Algorithm 1. In addition, we show a visual example of DSP in Figure 3.

It is noteworthy that DSP has the following merits: First, it creates two mixed images that share an identical source template image patch at the same location, which can serve as a bridge to effectively reduce the domain gap between both domains. Second, it preserves the original domain information by keeping its structure layout, complete objects, as well as appearance style. Third, the ground truth labels of pasted source image patches can be leveraged for output-level alignment.

3.3 Mean Teacher-based Domain Adaptation

3.3.1 DSP-induced Output-level Alignment. During training, the original source image (x_s), the mixed source and target image (x_ps)
and $x_{pt}$ are fed into the student network $f_{S}$, while the original target image ($x_{t}$) is fed into the teacher network $f_{T}$. While the parameters $\theta$ of the student network are optimized via gradient back-propagation, the parameters $\theta_{t}'$ of the teacher network at training step $t$ are updated using EMA as follows:

$$\theta_{t}' = \alpha \cdot \theta_{t-1}' + (1 - \alpha) \cdot \theta_{t}, \quad (4)$$

where $\alpha$ denotes the EMA decay coefficient.

After obtaining the predict semantic map $p_{t}$ of $x_{t}$, a cross-entropy based semantic segmentation loss is used for training the network:

$$L_{seg} = -\sum_{i=1}^{H} \sum_{c=1}^{W} H^{x \times W \times C} \log p_{t}^{x \times W \times C}, \quad (5)$$

where $H, W, C$ represent the height and width of the image, and number of classes, respectively. $y_{s}$ denotes the ground truth semantic labels.

Similarly, after obtaining the predict semantic map $p_{ps}$ of $x_{ps}$ from $f_{S}$, we use a weighted cross-entropy based soft semantic segmentation loss to train the network, i.e.,

$$L_{seg_{soft}} = -\sum_{i=1}^{H} \sum_{c=1}^{W} y_{s} H^{x \times W \times C} \log p_{ps}^{x \times W \times C} \circ M \quad (6)$$

$$= -\sum_{i=1}^{H} \sum_{c=1}^{W} y_{s} H^{x \times W \times C} \log p_{ps}^{x \times W \times C} \circ (1 - M),$$

where $y_{s}$ is the ground truth semantic label of the pasted source image patch.

From the teacher model $f_{T}$, the pseudo label of the original target image $x_{t}$ can be obtained as $y_{t} = f_{T}(x_{t})$. Meanwhile, we can obtain the predict semantic map $p_{pt}$ of $x_{pt}$ from $f_{S}$. Since these two models are assumed to produce a same prediction for a same image under different perturbations, so we adopt a prediction consistency loss to train the network, i.e.,

$$L_{cons} = -\sum_{i=1}^{H} \sum_{c=1}^{W} y_{s} H^{x \times W \times C} \log p_{pt}^{x \times W \times C} \circ M \quad (7)$$

$$= -\sum_{i=1}^{H} \sum_{c=1}^{W} y_{s} H^{x \times W \times C} \log p_{pt}^{x \times W \times C} \circ (1 - M).$$

3.3.2 DSP-induced Feature-level Alignment. Since the source mixed image $x_{s}$ and target mixed image $x_{pt}$ have the same pasted source image patch, the features extracted in this region from $x_{s}$ and $x_{pt}$ should be as similar as possible. To this end, we adopt Maximum Mean Discrepancy (MMD) [19] to learn transferable features by minimizing the MMD of their kernel embeddings. This paste-patch feature alignment loss can be formulated as:

$$L_{paste} = \|\mu(f_{S}(x_{s}) \circ M) - \mu(f_{S}(x_{pt}) \circ M)\|_{H}^{2}, \quad (8)$$

where $\mu(\cdot)$ denotes the kernel mean embedding, $f_{S}$ represents the feature extractor of the student model $f_{S}$ (i.e., the network before
we adopt ResNet-101 [12] pre-trained on ImageNet [6] and on MSCOCO [18] as the backbone network to extract features, and ASPP [1] is adopted to be the classifier to predict semantic maps. We use Stochastic Gradient Descent (SGD) with Nesterov acceleration as the optimizer, an initial learning rate of $2.5 \times 10^{-3}$ for the feature extractor, and an initial learning rate of $2.5 \times 10^{-4}$ for the classifier, which are then decreased based on a polynomial decay policy with an exponent of 0.9. Weight decay is set to $5 \times 10^{-4}$ and momentum is set to 0.9. During training, we resize images in Cityscapes, GTA5, and SYNTHIA to $1,024 \times 512, 1,280 \times 720$, and $1,280 \times 760$, respectively, after which the input images are randomly cropped into patches with a size of $512 \times 512$. We also apply color jittering and Gaussian blurring for data augmentation. The model is trained for a total of 250,000 iterations with a batch size of 2. We set $\lambda_{\text{feature}}$ to 0.005, the EMA decay coefficient $\gamma$ to 0.99, opacity $\beta$ to 0.8, and the number of long-tail classes selected at each training iteration $k$ to 2.



4 EXPERIMENTS

4.1 Datasets and Evaluation Metrics

We evaluate the performance of the proposed method for two challenging UDA tasks: GTA5 [27] to Cityscapes [5] and SYNTHIA [28] to Cityscapes. GTA5 is a synthetic dataset created using a photo-realistic open-world computer game engine. Dense pixel-level semantic annotations are provided for 24,966 urban landscape images with a resolution of $1,914 \times 1,052$. 19 common classes in the Cityscapes dataset are chosen in our experiments. SYNTHIA is another synthetic collection of 9,400 diverse urban images with a resolution of $1,280 \times 760$. We consider 16 common categories in the Cityscapes dataset for evaluation while the results on 13 common classes are also reported following a common practice. Cityscapes is a large-scale real-world urban scenes benchmark for semantic segmentation, which provides 5,000 densely annotated images with a resolution of $2,048 \times 1,024$. We use 2,975 unlabeled training images during training and 500 validation images for testing. In all experiments, we use the mIoU metric for evaluation.

4.2 Implementation Details

The proposed model is implemented using PyTorch on a single NVIDIA Tesla V100 GPU with 16 GB memory. Following previous works, we adopt ResNet-101 [12] pre-trained on ImageNet [6] and on MSCOCO [18] as the backbone network to extract features, and ASPP [1] is adopted to be the classifier to predict semantic maps. We use Stochastic Gradient Descent (SGD) with Nesterov acceleration as the optimizer, an initial learning rate of $2.5 \times 10^{-3}$ for the feature extractor, and an initial learning rate of $2.5 \times 10^{-4}$ for the classifier, which are then decreased based on a polynomial decay policy with an exponent of 0.9. Weight decay is set to $5 \times 10^{-4}$ and momentum is set to 0.9. During training, we resize images in Cityscapes, GTA5, and SYNTHIA to $1,024 \times 512, 1,280 \times 720$, and $1,280 \times 760$, respectively, after which the input images are randomly cropped into patches with a size of $512 \times 512$. We also apply color jittering and Gaussian blurring for data augmentation. The model is trained for a total of 250,000 iterations with a batch size of 2. We set $\lambda_{\text{feature}}$ to 0.005, the EMA decay coefficient $\gamma$ to 0.99, opacity $\beta$ to 0.8, and the number of long-tail classes selected at each training iteration $k$ to 2.

4.3 Comparison with State-of-the-art Methods

In this section, we evaluate our DSP model on the two challenging UDA semantic segmentation tasks and compare it with several state-of-the-art methods.

Table 1 shows the results of different methods for the GTA5 to Cityscapes task over 13 common classes. Our DSP model achieves...
As for another SYNTHIA to Cityscapes task, we report the mIoU results of both 13 and 16 classes in Table 2. As can be seen, DSP achieves the best performance in terms of mIoU of 16 and 13 classes, i.e., 51.0% mIoU and 59.9% mIoU*, respectively. It also outperforms all existing methods, and achieves a significant improvement over all existing methods, and achieves a significant improvement over the source-only model by 17.5% and 21.3% over 16 and 13 classes in terms of mIoU.

### 4.4 Parameter Analysis and Ablation Study

To investigate the impact of different components in our DSP model, we conduct an ablation study on the GTA5 to Cityscapes setting as shown in Table 3. “Source Only” denotes the model without domain adaptation. “Mean Teacher” (MT) denotes the vanilla mean teacher framework. “Single Paste” denotes the pasting strategy proposed in DACS [33] that only pasting source image patches to the target domain images. “Dual Paste” denotes the dual paste strategy based on hard-paste while “Dual Soft-Paste” denotes the proposed dual paste strategy based on soft-paste. “Feature Alignment” denotes the DSP-induced feature-level alignment. As can be seen, the source only model obtains 36.6 mIoU on the target domain. After using the mean teacher framework, a gain of 5.7% mIoU can be observed. The single paste strategy (DACS) brings another 9.8% mIoU improvement, 15.5% mIoU in total. By contrast, our dual soft-paste strategy achieves a gain of 17.9% mIoU over the source only baseline model. Compared with hard-paste, the proposed soft-paste strategy is more effective. After using the proposed DSP-induced feature-level alignment, our DSP model improves the baseline model by a significant margin, i.e., 18.4% mIoU.

The training losses and mIoU results of DACS [33], i.e., single hard-paste, Dual Hard-Paste (a variant of the proposed DSP based on hard-paste), and the proposed DSP on the GTAV to Cityscapes setting are plotted in Figure 5. In the early training phase, DACS suffers from the large domain gap and may produce incorrect predictions, especially for those long-tail classes. Consequently, these incorrect predictions may mislead the adaptation process, leading to a slow convergence speed and limited performance. By contrast, our model adopts a dual soft-paste strategy and a long-tail class first sampling strategy to create intermediate domains having smaller domain gaps, thereby facilitating the domain adaptation. In addition, it can be seen that the dual paste strategy contributes to the faster convergence speed while the soft-paste strategy matters for better cross-domain generalization performance.

#### Table 3: Ablation study of the proposed DSP model.

| Methods                  | mIoU (%) | Gain(%) |
|--------------------------|----------|---------|
| Source Only              | 36.6     | -       |
| +Mean Teacher (MT)       | 42.3     | 5.7     |
| +MT + Single Paste       | 52.1     | 15.5    |
| +MT + Dual Paste         | 53.6     | 17.0    |
| +MT + Dual Soft-Paste    | 54.5     | 17.9    |
| +MT + DSP + Feature Alignment | 55.0 | 18.4    |

Table 4 shows the results of different hyper-parameter settings of the opacity $\beta$. When $\beta = 0.8$, the model achieves the best performance, i.e., 55.0% mIoU. When $\beta = 0$, the model is the vanilla mean teacher model, which only obtains 42.3% mIoU. And when $\beta = 1$, the model becomes the mean teacher model using the dual hard-paste strategy, obtaining a better mIoU of 53.6%. Besides, when $\beta$ is less than 0.7, the weight of the pasted source image patch is too small, which may result in inaccurate predictions of target images, especially in the early training phase, thereby affecting the final performance.

#### Table 4: Hyper-parameter study of opacity $\beta$.

| $\beta$ | 0 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1     |
|---------|---|-----|-----|-----|-----|-----|-------|
| mIoU    | 42.3 | 51.0 | 52.2 | 54.3 | 55.0 | 54.9 | 53.6  |

Figure 5: Convergence analysis of DACS [33], Dual Hard-Paste (a variant of the proposed DSP based on hard-paste), and the proposed DSP on the GTA5 to Cityscapes setting. (a) Training losses. (b) mIoU results. DSP converges faster, i.e., achieving better performance in the same training steps.
Table 5 shows the results of using different numbers of long-tailed classes $k$ during pasting. As can be seen, the performance peaks at $k = 2$. When $k = 0$, the performance drops by a margin of 1.8% mIoU, implying the proposed long-tail class first sampling strategy matters for mitigating the class imbalance issue. Besides, when $k$ becomes larger, the sampled patches from different images may overlap each other and also result in inconsistent spatial layouts in the pasted patch, which will affect the performance.

| GTAV $\rightarrow$ Cityscapes |
|-------------------------------|
| long-tail classes to choose | 0  | 1 | 2  | 3  |
| mIoU                        | 53.2 | 54.3 | 55.0 | 53.7 |

Table 5: Hyper-parameter study of the number of long-tailed classes $k$.

In Figure 6, we present some visual segmentation results of the source only model and our DSP model. As can be seen, the source only model has limited cross-domain generalization ability without any domain adaptation. There are large areas of incorrect predictions, such as road and trees. By contrast, our DSP model shows a fairly good cross-domain generalization performance.

5 CONCLUSION

In this paper, we investigate the unsupervised domain adaptive semantic segmentation problem from the perspective of image manipulation. Specifically, we propose a novel Dual Soft-Paste (DSP) method to create new intermediate domains with smaller domain gaps. Based on the mean teacher framework, DSP-induced output-level alignment and feature-level alignment are performed, which help to learn domain-invariant features. Besides, the long-tail class first sampling strategy used in DSP shows its effectiveness in addressing the class-imbalance issue. Experiments on two challenging benchmarks demonstrate the superiority of DSP over state-of-the-art methods. In the future, we plan to investigate the impact of DSP in other domain adaptation frameworks as well as develop an adaptive sampling strategy using reinforcement learning, which can actively sample both normal and long-tail classes during training.

6 ACKNOWLEDGMENTS

This work was supported by the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170 and the Fundamental Research Funds for the Central Universities under Grant 2042021kf0196. The numerical calculations in this paper had been supported by the supercomputing system in the Supercomputing Center of Wuhan University.
