

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

Li Gao\*

gaoli1218@whu.edu.cn  
Wuhan University  
China

Lefei Zhang†

zhanglefei@whu.edu.cn  
Wuhan University  
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>.

## CCS CONCEPTS

- Computing methodologies → Computer vision problems.

\*This work was done during Li Gao's internship at JD Explore Academy.

†Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

MM '21, October 20–24, 2021, Virtual Event, China

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8651-7/21/10...\$15.00

<https://doi.org/10.1145/3474085.3475186>

Jing Zhang\*

jing.zhang1@sydney.edu.au  
The University of Sydney  
Australia

Dacheng Tao

taodacheng@jd.com  
JD Explore Academy  
China

## KEYWORDS

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

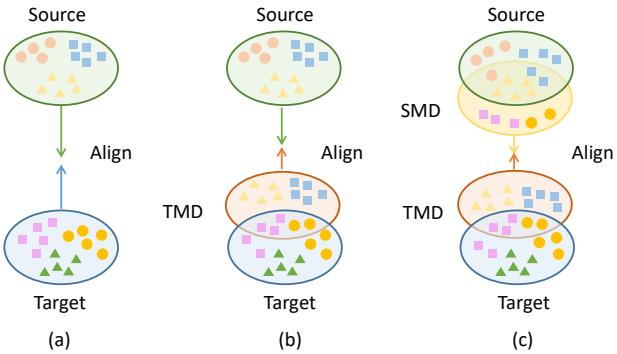
### ACM Reference Format:

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].



**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.**

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:

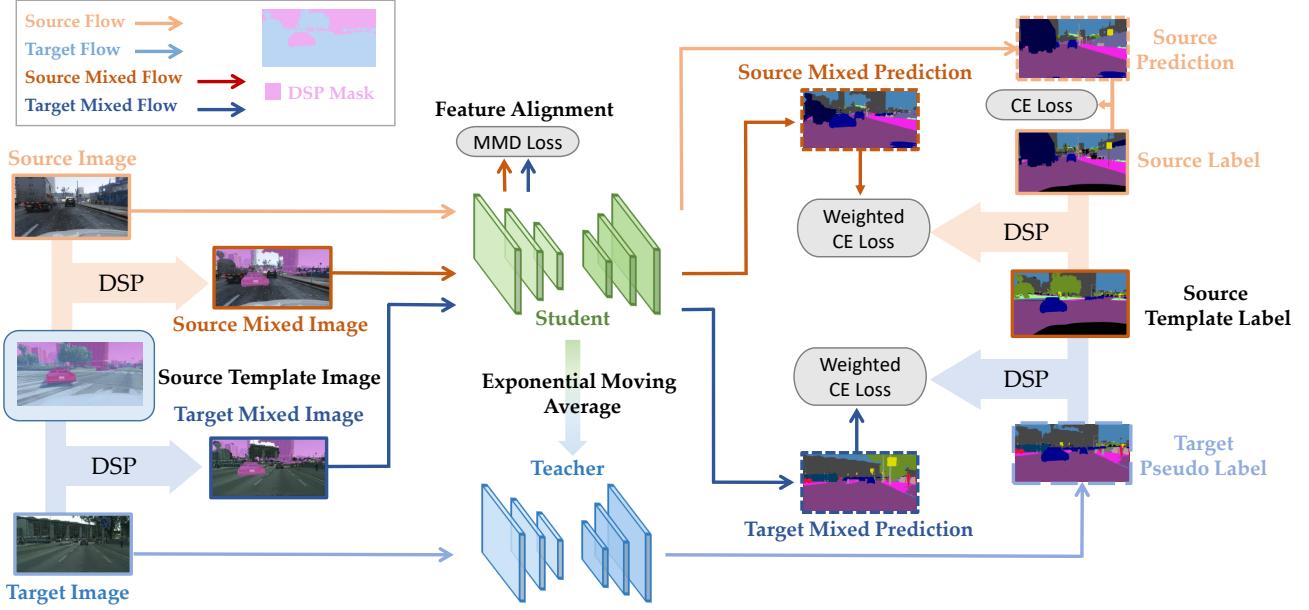
- We propose a novel Dual Soft-Paste (DSP) method to create intermediate domains and facilitate domain alignment for semantic segmentation. DSP adopts a long-tail class first sampling strategy, which alleviates the class imbalance issue and shows its effectiveness in improving the performance.
- Based 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.

## 2 RELATED WORK

### 2.1 Mean Teacher-based Methods

Since our work is built upon the mean teacher framework, we briefly review related methods in this section. For other UDA methods for semantic segmentation, such as style transfer and adversarial learning, we recommend the excellent survey [32]. Mean teacher is a widely used framework in the field of semi-supervised learning, which is based on the simple idea that under the supervision of labeled data, unlabeled data should produce consistent predictions under different perturbations. It consists of two models, a student model and a teacher model, where the teacher model is an exponential moving average (EMA) of the student model. The teacher model transfers the learned knowledge to the student [31] by aligning the two domains at the output level with a consistency regularization.

SEANET [34] firstly introduced the mean teacher framework for unsupervised domain adaptive semantic segmentation, which adopted an attention mechanism to generate attention-aware features to guide the calculation of consistency loss in the target domain. Zhou et al. [44] proposed an uncertainty-aware consistency regularization method by exploiting the latent uncertainty information of the target samples. Recently, DACS [33] proposed to paste source image patches onto the target domain images to create a mixed domain, where the labels of the pasted patches can be used for supervised learning and the prediction consistency between the student model and teacher model in the mean teacher framework is also exploited. Although DACS has achieved promising results, it still suffers from several problems, including the large gap between two domains, the incomplete structure and inconsistent spatial layout problem by hard-paste, as well as the class imbalance problem, resulting in limited performance.



**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.

---

**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  $\beta$ ;  
**Output:** The DSP mask  $M$ , mixed source image  $x_{ps}$ , mixed target image  $x_{pt}$ ;

- 1:  $S_{class} \leftarrow$  set of classes appearing in  $y_p$  ;
- 2:  $c \leftarrow$  randomly select  $|S_{class}|/2$  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:  $c_{tail}, y_{tail} \leftarrow$  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:  $x_{ps}, x_{pt} \leftarrow$  calculate mixed images by Eq. 2 and Eq. 3
- 12: **return**  $M, x_{ps}, x_{pt}$ ;

---

### 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 proposed a novel DSP model under the mean teacher framework as illustrated in Figure 2. It has two segmentation networks, *i.e.*, a student network  $f_\theta$  with learnable parameters  $\theta$  and a teacher network  $f_{\theta'}$  with parameters  $\theta'$  calculated by the exponential moving average (EMA) of  $\theta'$ .

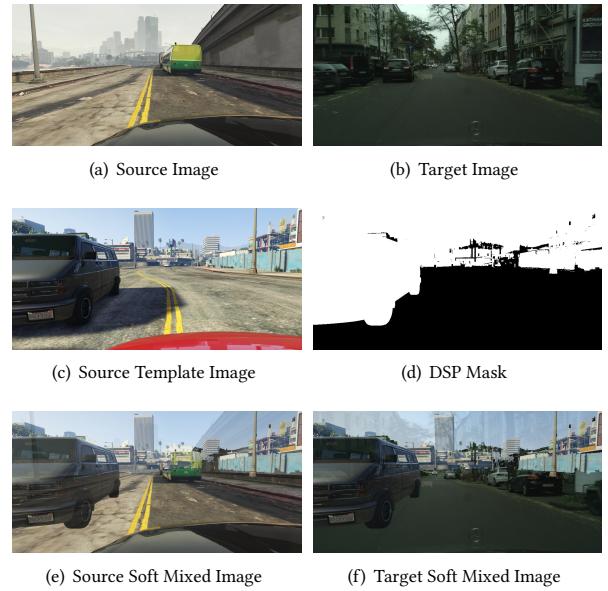
#### 3.2 Dual Soft-Paste

**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  $\{p_1, p_2, \dots, p_c\}$ , *i.e.*,

$$p_i = \frac{\sum_{j=1}^N c_{ij}}{N}, \quad (1)$$

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 categories and record those images containing these classes as a dataset  $D$  to facilitate the subsequent sampling process. In this paper, we set  $K$  to 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



**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  $\beta$  for better illustration.

images as described above. Next, we merge the candidate patch with the patches of long-tail classes to form the final candidate patch. In this way, we can guarantee that the candidate patches have both frequent classes and long-tail classes. Then, we paste this patch on a source image and target image via a soft weighted-sum manner using an opacity parameter  $\beta$ . Specifically, given a source image  $x_s$ , a target image  $x_t$ , and the source template image  $x_p$  with the corresponding binary mask  $M$ , the mixed source image  $x_{ps}$  can be obtained by:

$$x_{ps} = \beta M \odot x_p + (1 - \beta M) \odot x_s. \quad (2)$$

Similarly, the mixed target image  $x_{pt}$  can be obtained as follows:

$$x_{pt} = \beta M \odot x_p + (1 - \beta M) \odot x_t. \quad (3)$$

For simplicity, we reuse  $M$  to represent  $\beta M$  by assigning the opacity value to those positive pixels in  $M$ . A hyper-parameter study of  $\beta$  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.

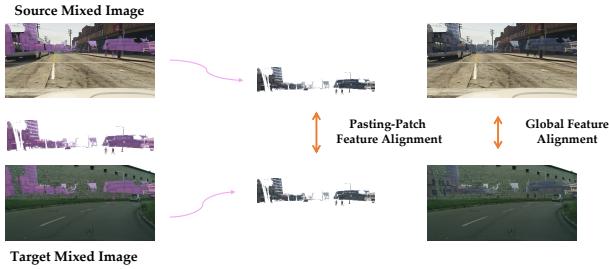
It is noteworthy that DSP has the following merits: First, it creates two mixed images that share an identical source template image patch at the same location, which can serve as a bridge to effectively reduce the domain gap between both domains. Second, it preserves the original domain information 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 ( $x_{ps}$

GTA5 → Cityscapes																								
Method	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	trunk	bus	train	motorbike	bike	mIoU				
Source Only	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6				
WeakSeg(ECCV20)[25]	91.6	47.4	84.0	30.4	28.3	31.4	37.4	35.4	83.9	38.3	83.9	61.2	28.2	83.7	28.8	41.3	8.8	24.7	46.4	48.2				
LSE(ECCV20)[30]	90.2	40.0	83.5	31.9	26.4	32.6	38.7	37.5	81.0	34.2	84.6	61.6	33.4	82.5	32.8	45.9	6.7	29.1	30.6	47.5				
IAST(ECCV20)[23]	94.1	58.8	85.4	<b>39.7</b>	29.2	25.1	43.1	34.2	84.8	34.6	88.7	62.7	30.3	87.6	42.3	50.3	24.7	35.2	40.2	52.2				
CrCDA(ECCV20)[15]	92.4	55.3	82.3	31.2	29.1	32.5	33.2	35.6	83.5	34.8	84.2	58.9	32.2	84.7	40.6	46.1	2.1	31.1	32.7	48.6				
LTIR(CVPR20)[16]	92.9	55.0	85.3	34.2	31.1	34.9	40.7	34.0	85.2	40.1	87.1	61.0	31.1	82.5	32.3	42.9	0.3	36.4	46.1	50.2				
UIDA(CVPR20)[24]	90.6	37.1	82.6	30.1	19.1	29.5	32.4	20.6	85.7	40.5	79.7	58.7	31.1	86.3	31.5	48.3	0.0	30.2	35.8	46.3				
PIT(CVPR20)[22]	87.5	43.4	78.8	31.2	30.2	36.3	39.9	42.0	79.2	37.1	79.3	65.4	<b>37.5</b>	83.2	46.0	45.6	25.7	23.5	49.9	50.6				
STAR(CVPR20)[20]	88.4	27.9	80.8	27.3	25.6	26.9	31.6	20.8	83.5	34.1	76.6	60.5	27.2	84.2	32.9	38.2	1.0	30.2	31.2	43.6				
ASA(TIP21)[45]	89.2	27.8	81.3	25.3	22.7	28.7	36.5	19.6	83.8	31.4	77.1	59.2	29.8	84.3	33.2	45.6	16.9	34.5	30.8	45.1				
CLAN(TPAMI21)[21]	88.7	35.5	80.3	27.5	25.0	29.3	36.4	28.1	84.5	37.0	76.6	58.4	29.7	81.2	38.8	40.9	5.6	32.9	28.8	45.5				
DACS(WACV21)[33]	89.9	39.7	<b>87.9</b>	<b>39.7</b>	<b>39.5</b>	<b>38.5</b>	46.4	<b>52.8</b>	<b>88.0</b>	<b>44.0</b>	88.8	<b>67.2</b>	35.8	84.5	45.7	50.2	0.0	27.3	34.0	52.1				
RPLL(IJCV21)[43]	90.4	31.2	85.1	36.9	25.6	37.5	<b>48.8</b>	48.5	85.3	34.8	81.1	64.4	36.8	86.3	34.9	52.2	1.7	29.0	44.6	50.3				
DAST(AAAI21)[37]	92.2	49.0	84.3	36.5	28.9	33.9	38.8	28.4	84.9	41.6	83.2	60.0	28.7	87.2	45.0	45.3	7.4	33.8	32.8	49.6				
ConTrans(AAAI21)[17]	<b>95.3</b>	<b>65.1</b>	84.6	33.2	23.7	32.8	32.7	36.9	86.0	41.0	85.6	56.1	25.9	86.3	34.5	39.1	11.5	28.3	43.0	49.6				
CIRN(AAAI21)[8]	91.5	48.7	85.2	33.1	26.0	32.3	33.8	34.6	85.1	43.6	86.9	62.2	28.5	84.6	37.9	47.6	0.0	35.0	36.0	49.1				
MetaCorrect(CVPR21)[10]	92.8	58.1	<b>86.2</b>	<b>39.7</b>	33.1	36.3	42.0	38.6	85.5	37.8	87.6	62.8	31.7	84.8	35.7	50.3	2.0	36.8	48.0	52.1				
ESL(CVPR21)[29]	90.2	43.9	84.7	35.9	28.5	31.2	37.9	34.0	84.5	42.2	83.9	59.0	32.2	81.8	36.7	49.4	1.8	30.6	34.1	48.6				
<b>Our DSP</b>	92.4	48.0	87.4	33.4	35.1	36.4	41.6	46.0	87.7	43.2	<b>89.8</b>	66.6	32.1	<b>89.9</b>	<b>57.0</b>	<b>56.1</b>	0.0	<b>44.1</b>	<b>57.8</b>	<b>55.0</b>				

Table 1: Results of different domain adaptation methods for the GTA5 → Cityscapes task.



and  $x_{pt}$ ) are fed into the student network  $f_\theta$ , while the original target image ( $x_t$ ) is fed into the teacher network  $f_{\theta'}$ . 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  $p_s$  of  $x_s$ , a cross-entropy based *semantic segmentation loss* is used for training the network:

$$\mathcal{L}_{seg} = - \sum_{i=1}^{H \times W} \sum_{c=1}^C y_s^{H \times W \times C} \log p_s^{H \times W \times C}, \quad (5)$$

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  $f_\theta$ , we use a weighted cross-entropy based *soft semantic segmentation loss* to train the network, i.e.,

$$\begin{aligned} \mathcal{L}_{seg\_soft} = & - \sum_{i=1}^{H \times W} \sum_{c=1}^C y_p^{H \times W \times C} \log p_{ps}^{H \times W \times C} \odot M \\ & - \sum_{i=1}^{H \times W} \sum_{c=1}^C y_s^{H \times W \times C} \log p_{ps}^{H \times W \times C} \odot (1 - M), \end{aligned} \quad (6)$$

where  $y_p$  is the ground truth semantic label of the pasted source image patch.

From the teacher model  $f_{\theta'}$ , the pseudo label of the original target image  $x_t$  can be obtained as  $\hat{y}_t = f_{\theta'}(x_t)$ . Meanwhile, we can obtain the predict semantic map  $p_{pt}$  of  $x_{pt}$  from  $f_\theta$ . 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.,

$$\begin{aligned} \mathcal{L}_{cons} = & - \sum_{i=1}^{H \times W} \sum_{c=1}^C y_p^{H \times W \times C} \log p_{pt}^{H \times W \times C} \odot M \\ & - \sum_{i=1}^{H \times W} \sum_{c=1}^C \hat{y}_t^{H \times W \times C} \log p_{pt}^{H \times W \times C} \odot (1 - M). \end{aligned} \quad (7)$$

3.3.2 *DSP-induced Feature-level Alignment*. Since the source mixed image  $x_{ps}$  and target mixed image  $x_{pt}$  have the same pasted source image patch, the features extracted in this region from  $x_{ps}$  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:

$$\mathcal{L}_{paste} = \|\mu(f_e(x_{ps}) \odot M) - \mu(f_e(x_{pt}) \odot M)\|_{\mathcal{H}}^2, \quad (8)$$

where  $\mu(\cdot)$  denotes the kernel mean embedding,  $f_e$  represents the feature extractor of the student model  $f_\theta$  (i.e., the network before

Method	SYNTHIA → Cityscapes																	mIoU	mIoU*
	road	sidewalk	building	wall*	fence*	pole*	light	sign	vegetation	sky	person	rider	car	bus	motorbike	bike			
Source Only	55.6	23.8	74.6	9.2	0.2	24.4	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	33.5	38.6	
WeakSeg(ECCV20)[25]	92.0	53.5	80.9	11.4	0.4	21.8	3.8	6.0	81.6	84.4	60.8	24.4	80.5	39.0	26.0	41.7	44.3	51.9	
LSE(ECCV20)[30]	82.9	43.1	78.1	9.3	0.6	28.2	9.1	14.4	77.0	83.5	58.1	25.9	71.9	38.0	29.4	31.2	42.6	49.4	
IAST(ECCV20)[23]	81.9	41.5	<b>83.3</b>	17.7	<b>4.6</b>	32.3	30.9	28.8	83.4	85.0	65.5	30.8	<b>86.5</b>	38.2	<b>33.1</b>	52.7	49.8	57.0	
CrCDA(ECCV20)[15]	86.2	44.9	79.5	8.3	0.7	27.8	9.4	11.8	78.6	86.5	57.2	26.1	76.8	39.9	21.5	32.1	42.9	50.0	
LTIR(CVPR20)[16]	92.6	53.2	79.2	-	-	-	1.6	7.5	78.6	84.4	52.6	20.0	82.1	34.8	14.6	39.4	-	49.3	
UIDA(CVPR20)[24]	84.3	37.7	79.5	5.3	0.4	24.9	9.2	8.4	80.0	84.1	57.2	23.0	78.0	38.1	20.3	36.5	41.7	48.9	
PIT(CVPR20)[22]	83.1	27.6	81.5	8.9	0.3	21.8	26.4	<b>33.8</b>	76.4	78.8	64.2	27.6	79.6	31.2	31.0	31.3	44.0	51.8	
STAR(CVPR20)[20]	82.6	36.2	81.1	-	-	-	12.2	8.7	78.4	82.2	59.0	22.5	76.3	33.6	11.9	40.8	-	48.1	
ASA(TIP21)[45]	91.2	48.5	80.4	3.7	0.3	21.7	5.5	5.2	79.5	83.6	56.4	21.9	80.3	36.2	20.0	32.9	41.7	49.3	
CLAN(TPAMI21)[21]	82.7	37.2	81.5	-	-	-	17.1	13.1	81.2	83.3	55.5	22.1	76.6	30.1	23.5	30.7	-	48.8	
DACS(WACV21)[33]	80.6	25.1	81.9	<b>21.5</b>	2.9	<b>37.2</b>	22.7	24.0	83.7	<b>90.8</b>	<b>67.6</b>	<b>38.3</b>	82.9	38.9	28.5	47.6	48.3	54.8	
RPLL(IJCV21)[43]	87.6	41.9	83.1	14.7	1.7	36.2	31.3	19.9	81.6	80.6	63.0	21.8	86.2	40.7	23.6	53.1	47.9	54.9	
DAST(AAAI21)[37]	87.1	44.5	82.3	10.7	0.8	29.9	13.9	13.1	81.6	86.0	60.3	25.1	83.1	40.1	24.4	40.5	45.2	52.5	
ConTrans(AAAI21)[17]	<b>93.3</b>	<b>54.0</b>	81.3	14.3	0.7	28.8	21.3	22.8	82.6	83.3	57.7	22.8	83.4	30.7	20.2	47.2	46.5	53.9	
CIRN(AAAI21)[8]	85.8	40.4	80.4	4.7	1.8	30.8	16.4	18.6	80.7	80.4	55.2	26.3	83.9	43.8	18.6	34.3	43.9	51.1	
MetaCorrect[10](CVPR21)	92.6	52.7	81.3	8.9	2.4	28.1	13.0	7.3	83.5	85.0	60.1	19.7	84.8	37.2	21.5	43.9	45.1	52.5	
ESL(CVPR21)[29]	84.3	39.7	79.0	9.4	0.7	27.7	16.0	14.3	78.3	83.8	59.1	26.6	72.7	35.8	23.6	45.8	43.5	50.7	
<b>Our DSP</b>	86.4	42.0	82.0	2.1	1.8	34.0	<b>31.6</b>	33.2	<b>87.2</b>	88.5	64.1	31.9	83.8	<b>65.4</b>	28.8	<b>54.0</b>	<b>51.0</b>	<b>59.9</b>	

**Table 2: Results of different domain adaptation methods for the SYNTHIA → Cityscapes task. mIoU\* denotes the mean IoU of 13 classes, excluding the classes marked by the asterisk.**

the ASPP module), and  $\mathcal{H}$  denotes the **reproducing kernel Hilbert space (RKHS)**. Note that since the pasted patches in  $x_{ps}$  and  $x_{pt}$  have different context of source and target domain information, which may be embedded in the extracted features, the paste-patch alignment loss can reduce the domain gap implicitly.

In addition, we try to minimize the MMD of the image features of  $x_{ps}$  and  $x_{pt}$  to align the feature distributions of both domains. This **global feature alignment loss** is:

$$\mathcal{L}_{global} = \|\mu(f_e(x_{ps})) - \mu(f_e(x_{pt}))\|_{\mathcal{H}}^2. \quad (9)$$

An illustration of these two losses is shown in Figure 4. The overall training objective can be defined as:

$$\mathcal{L} = \mathcal{L}_{seg} + \mathcal{L}_{seg\_soft} + \mathcal{L}_{cons} + \lambda_{feature}(\mathcal{L}_{paste} + \mathcal{L}_{global}), \quad (10)$$

where  $\lambda_{feature}$  is a hyper-parameter to balance different losses.

## 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.

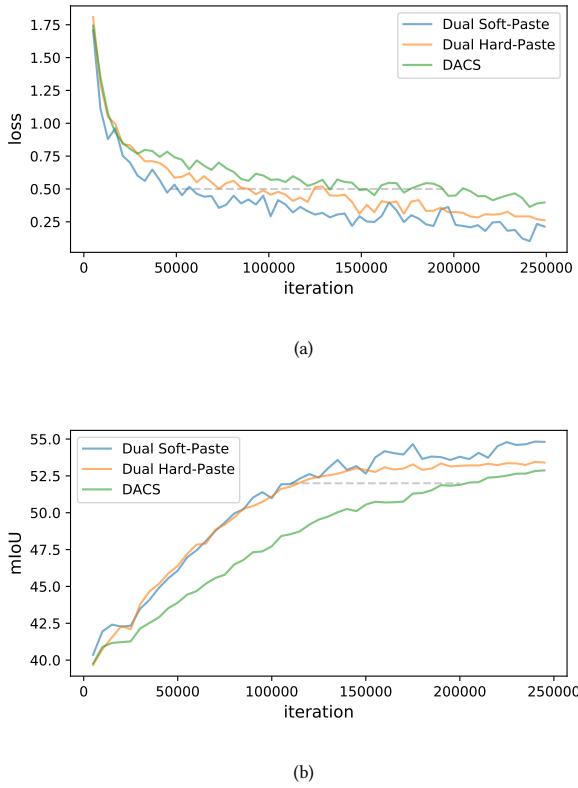
### 4.2 Implementation Details

The proposed model is implemented using PyTorch on a single NVIDIA Tesla V100 GPU with 16 GB memory. Following previous work, 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  $\lambda_{feature}$  to 0.005, the EMA decay coefficient  $\alpha$  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 19 common classes. Our DSP model achieves



**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.**

the best mIoU score of 55.0%, significantly outperforming state-of-the-art methods by large margins from 2.8% to 11.4%. Besides, our method shows its effectiveness in predicting long-tail classes. Moreover, DSP outperforms the source-only segmentation model by 18.4% mIoU, showing a good cross-domain generalization ability. 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 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 GTA5 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.

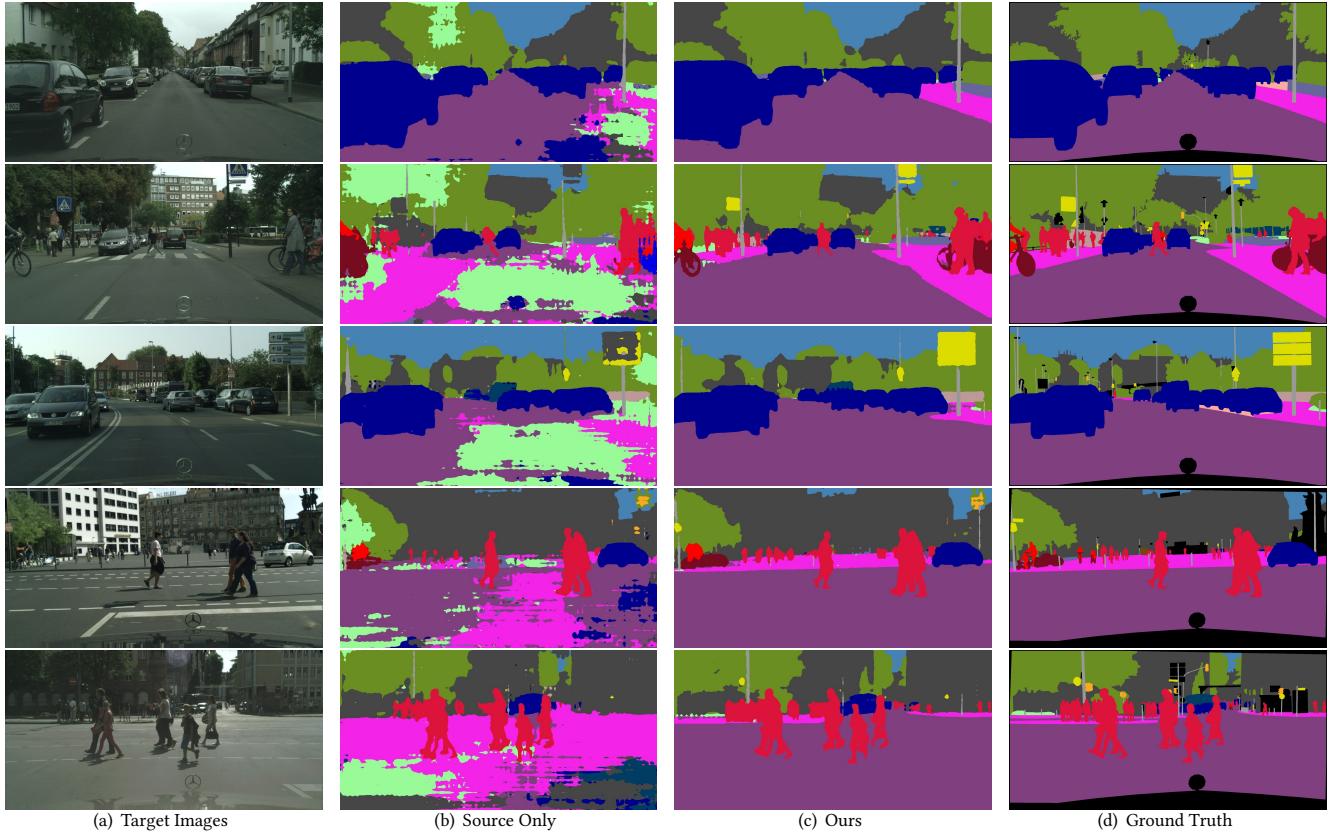
GTAV → Cityscapes		
Methods	mIoU (%)	Gain(%)
Source Only	36.6	-
+Mean Teacher (MT)	42.3	5.7
+MT + Single Paste	52.1	15.5
+MT + Dual Paste	53.6	17.0
+MT + Dual Soft-Paste	54.5	17.9
+MT + DSP + Feature Alignment	55.0	18.4

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

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.

GTAV → Cityscapes							
$\beta$	0	0.5	0.6	0.7	0.8	0.9	1
mIoU	42.3	51.0	52.2	54.5	<b>55.0</b>	54.9	53.6

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



**Figure 6: Some visual segmentation results for the GTA5 to Cityscapes task.**

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 → Cityscapes				
long-tail classes to choose	0	1	2	3
mIoU	53.2	54.3	<b>55.0</b>	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.

## REFERENCES

- [1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2018. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40, 4 (2018), 834–848.
- [2] Yun Chun Chen, Yen Yu Lin, Ming Hsuan Yang, and Jia Bin Huang. 2019. Crdco: Pixel-Level Domain Transfer with Cross-Domain Consistency. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1791–1800.
- [3] Zhe Chen, Jing Zhang, and Dacheng Tao. 2019. Progressive lidar adaptation for road detection. *IEEE/CAA Journal of Automatica Sinica* 6, 3 (2019), 693–702.
- [4] Jaehoon Choi, Taekyung Kim, and Changick Kim. 2019. Self-Ensembling with GAN-Based Data Augmentation for Domain Adaptation in Semantic Segmentation. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 6829–6839.
- [5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3213–3223.
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 248–255.
- [7] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. 2017. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 1301–1310.
- [8] Li Gao, Lefei Zhang, and Qian Zhang. 2021. Addressing Domain Gap via Content Invariant Representation for Semantic Segmentation. *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 7528–7536.
- [9] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. arXiv:1406.2661
- [10] Xiaoqing Guo, Chen Yang, Baopu Li, and Yixuan Yuan. 2021. MetaCorrection : Domain-Aware Meta Loss Correction for Unsupervised Domain Adaptation in Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–10.
- [11] Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, Adam Prügel-Bennett, and Jonathon S. Hare. 2020. Understanding and Enhancing Mixed Sample Data Augmentation. (2020). arXiv:2002.12047
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.
- [13] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1989–1998.
- [14] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. 2016. FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation. arXiv:1612.02649
- [15] Jiaxing Huang, Shijian Lu, Dayan Guan, and Xiaobing Zhang. 2020. Contextual-Relation Consistent Domain Adaptation for Semantic Segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 705–722.
- [16] Myeongjin Kim and Hyeran Byun. 2020. Learning Texture Invariant Representation for Domain Adaptation of Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 12972–12981.
- [17] Suhyeon Lee, Junhyuk Hyun, Hongje Seong, and Euntai Kim. 2021. Unsupervised Domain Adaptation for Semantic Segmentation by Content Transfer. *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 1–10.
- [18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 740–755.
- [19] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning Transferable Features with Deep Adaptation Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, Vol. 37. 97–105.
- [20] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi Zhe Song, and Tao Xiang. 2020. Stochastic Classifiers for Unsupervised Domain Adaptation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9108–9117.
- [21] Yawei Luo, Ping Liu, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. 2021. Category-Level Adversarial Adaptation for Semantic Segmentation using Purified Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2021), 1–16.
- [22] Fengmao Lv, Tao Liang, Xiang Chen, and Guosheng Lin. 2020. Cross-Domain Semantic Segmentation via Domain-Invariant Interactive Relation Transfer. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4333–4342.
- [23] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. 2020. Instance Adaptive Self-Training for Unsupervised Domain Adaptation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 415–430.
- [24] Fei Pan, Inkyu Shin, Francois Fleuret, Seokju Lee, and In So Kweon. 2020. Unsupervised Intra-Domain Adaptation for Semantic Segmentation through Self-Supervision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3763–3772.
- [25] Sujoy Paul, Yi Hsuan Tsai, Samuel Schulter, Amit K. Roy-Chowdhury, and Manmohan Chandraker. 2020. Domain Adaptive Semantic Segmentation Using Weak Labels. *Proceedings of the European Conference on Computer Vision (ECCV)*, 571–587.
- [26] Tal Remez, Jonathan Huang, and Matthew Brown. 2018. Learning to Segment via Cut-and-Paste. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 37–52.
- [27] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. Playing for Data: Ground Truth from Computer Games. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 102–118.
- [28] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. 2016. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3234–3243.
- [29] Antoine Saporta, Tuan-hung Vu, Matthieu Cord, and P. Patrice. 2021. ESL: Entropy-Guided Self-Supervised Learning for Domain Adaptation in Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.
- [30] M Naseer Subhani and Mohsen Ali. 2020. Learning from Scale-Invariant Examples for Domain Adaptation in Semantic Segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 290–306.
- [31] Antti Tarvainen and Harri Valpola. 2017. Mean Teachers are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-supervised Deep Learning Results. *Proceeding of the Advances in Neural Information Processing Systems (NIPS)*, 1196–1205.
- [32] Marco Toldo, Andrea Maracani, Umberto Michieli, and Pietro Zanuttigh. 2020. Unsupervised Domain Adaptation in Semantic Segmentation: A Review. arXiv:2005.10876
- [33] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. 2021. DACS: Domain Adaptation via Cross-Domain Mixed Sampling. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. 1379–1389.
- [34] Yonghao Xu, Bo Du, Lefei Zhang, Qian Zhang, Guoli Wang, and Liangpei Zhang. 2019. Self-Ensembling Attention Networks: Addressing Domain Shift for Semantic Segmentation. *Proceedings of the 33th AAAI Conference on Artificial Intelligence (AAAI)* 33, 5581–5588.
- [35] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. 2021. ViTAE: Vision Transformer Advanced by Exploring Intrinsic Inductive Bias. arXiv:2106.03348
- [36] Jian Ye, Zhe Chen, Juhua Liu, and Bo Du. 2020. TextFuseNet: Scene Text Detection with Richer Fused Features. In *International Joint Conference on Artificial Intelligence (IJCAI)*. 516–522.
- [37] Fei Yu, Mo Zhang, Hexin Dong, Sheng Hu, Bin Dong, and Li Zhang. 2021. DAST : Unsupervised Domain Adaptation in Semantic Segmentation Based on Discriminator Attention and Self-Training. *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 10754–10762.
- [38] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 6023–6032.
- [39] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-paz. 2018. Mixup: Beyond Empirical Risk Minimization. arXiv:1710.09412v2
- [40] Jing Zhang and Dacheng Tao. 2020. Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal* 8, 10 (2020), 7789–7817.
- [41] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. 2021. Prototypical Pseudo Label Denoising and Target Structure Learning for Domain Adaptive Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [42] Qimeng Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. 2019. Category Anchor-Guided Unsupervised Domain Adaptation for Semantic Segmentation. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*. 1–11.
- [43] Zhedong Zheng and Yi Yang. 2021. Rectifying Pseudo Label Learning via Uncertainty Estimation for Domain Adaptive Semantic Segmentation. *International Journal of Computer Vision (IJCV)* 129 (2021), 1106–1120.
- [44] Qianyu Zhou, Zhengyang Feng, Guangliang Cheng, Xin Tan, Jianping Shi, and Lizhuang Ma. 2020. Uncertainty-Aware Consistency Regularization for Cross-Domain Semantic Segmentation. arXiv:2004.08878v3
- [45] Wei Zhou, Yukang Wang, Jiajia Chu, Jiehua Yang, Xiang Bai, and Yongchao Xu. 2021. Affinity Space Adaptation for Semantic Segmentation Across Domains. *IEEE Transactions on Image Processing (TIP)* 30 (2021), 2549–2561.
- [46] Yang Zou, Zhiding Yu, B V K Vijaya Kumar, and Jinsong Wang. 2018. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training. *Proceedings of the European Conference on Computer Vision (ECCV)*, 289–305.