Bidirectional Domain Mixup for Domain Adaptive Semantic Segmentation*

Minseok Seo[2], Yuhyun Kim[1] and Dong-Geol Choi[1]

[1] Hanbat National University, Daejeon, South Korea
yuhyun.dev@gmail.com, dgchoi@hanbat.ac.kr
[2] SI Analytics, Daejeon, South Korea
minseok.seo@si-analytics.ai

Abstract. Mixup provides interpolated training samples and allows the model to obtain smoother decision boundaries for better generalization. The idea can be naturally applied to the domain adaptation task, where we can mix the source and target samples to obtain domain-mixed samples for better adaptation. However, the extension of the idea from classification to segmentation (i.e., structured output) is nontrivial. In this paper, we propose a new data mixing method, bidirectional domain mixup (BDM). In specific, we achieve domain mixup in two-step: cut and paste. Given the warm-up model trained from any adaptation techniques, we forward the source and target samples and perform a simple threshold-based cutout of the unconfident regions (**cut**). After then, we fill-in the dropped regions with the other domain region patches (**paste**). We coupled our proposal with various state-of-the-art adaptation models and observe significant improvement consistently.

Keywords: Semantic Segmentation · Unsupervised Learning · Domain Adaptation

1 Introduction

To reduce the annotation budget in semantic segmentation that require pixel level annotation, there have been many domain adaptation (DA) approaches using relatively inexpensive source (*e.g.* simulator-based) data [13, 14] and unlabeled target (*e.g.* real) data. However, deep neural networks show poor generalization performance in real data because they are sensitive to domain misalignments such as layout [9], texture [20], structure [16], and class distribution [27]. To deal with it, many approaches have been proposed, including adversarial training [16], entropy minimization [17], and self-training [27, 12, 23].

Among them, cross-domain data mixing based approaches [2, 25, 15, 6] recently show state-of-the-art performances. Early works [2] are largely inspired by a popular data augmentation method, CutMix [22], and borrow some rectangular patches from one domain to fill-in the random hole of other domain of image.

* This paper is the short version of AAAI'23 and is NEVER considered an official publication.

Many variants improve data mixing strategy with mixing the region of randomly sampled classes [15], heuristics on relationship between classes [25], and image-level soft mixup [6].

Motivated by the progress, we further delve into domain mixing approaches and propose **B**idirectional **D**omain **M**ixup (**BDM**) framework. Beyond the previous unidirectional sample mixing [2, 25, 15, 6], the framework mix the samples in both direction, mix the source patches on the target sample (*i.e.*source-to-target) and vice versa (*i.e.*target-to-source). Specifically, we mainly adopt two core steps of data mixing approach: 1) **Cut**: how can we identify uninformative patches and 2) **Paste**: which patches from other domains bring better supervision signals.

First, we promote to learn domain transferable and generalized features by cutting out the source-specific and nosily predicted region for source and target data, respectively. To supplement scarce supervisory signal due to the cut process, we design the paste step to fulfill the three key functionalities: 1) As semantic segmentation network heavily rely on the context [3, 1, 24, 21], it is important to **maintain intrinsic spatial structure** of images. Thus we leverage spatial continuity to pick a patch that will be pasted on given the hole region. 2) Previous data mixing approaches [25, 15] usually paste the randomly selected classes with its correlated classes. This design choice exacerbates the class imbalanced problem [7], resulting in low performance in sample-scarce class. Instead, we induce **class-balanced learning** by giving the high probability to paste the patches with rare classes. 3) Lastly, proposed mixing method **prevent the noisy learning** by avoiding to paste low-confident patches.

We combine these findings to achieve a new state-of-the-art in the standard benchmarks of domain adaptive semantic segmentation, GTA5 $\rightarrow$ Cityscapes and SYNTHIA $\rightarrow$ Cityscapes setting. In addition, BMD consistently provides significant performance improvements when build upon the various representative UDA approaches, including adversarial training [16], entropy minimization [17], and self-training [12, 23].

2 Method

To bridge source and target domains that come from different distributions, we simulate intermediate domains by generating domain mixed samples. In the next section, we first describe the overall pipeline to train a network with domain-mixed samples. Next, we introduce how these samples are generated in detail.

2.1 The Bidirectional Domain Mixup Framework

Given a labeled source dataset $\mathcal{D}^S = \{(x_i^S, y_i^S)\}_{i=1}^{N_S}$, and a unlabeled target dataset $\mathcal{D}^T = \{x_i^T\}_{i=1}^{N_T}$, our goal is to transfer the knowledge learned from source domain to unlabeled target domain. To do so, we utilize domain-mixed samples and propose a new data mixing method, bidirectional domain mixup (BDM), to generate them. As illustrated in Fig. 1, this framework comprises the following four major components.

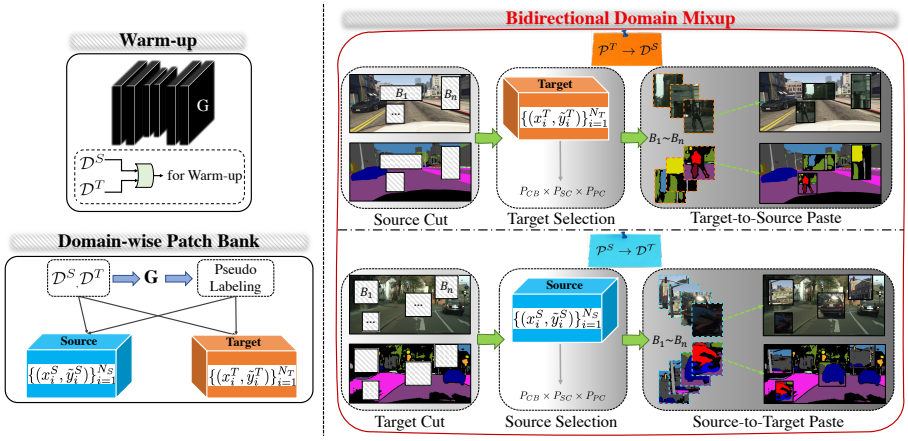

Fig. 1. Overview of BDM framework. Before training a network with the proposed BDM framework, we first warm up the model with any previous UDA method. Given the model $G(.)$, we generate pseudo labels for the source and target domain and store the images and corresponding pseudo labels in the domain-wise patch bank. The mixup process is conducted in a bidirectional way, target-to-source $P_T \rightarrow P_S$ and source-to-target $P_S \rightarrow P_T$. These samples are guided models to learn domain generalized features.

– We first apply a previous domain adaptation method as a *warm-up*. Our framework allows any previous domain adaptation method. To show the generality, we choose the representative methods including adversarial training [16], entropy minimization [17], and self-training [12, 23].

– *Domain-wise patch banks*, B^S and B^T, are constructed to store the samples of each domains. We divide images and corresponding pseudo labels into non-overlapping patches and the resulting pairs are stored in domain-wise patch banks. For the both domain, we adopt simple strategy [20] to generate pseudo labels, $\{\tilde{y}_i^S\}_{i=1}^{N_S}$ and $\{\tilde{y}_i^T\}_{i=1}^{N_T}$.

– During the training, a minibatch of source and target images, x^S and x^T, are sampled. *Bidirectional domain mixup(BDM)* generate domain-mixed samples, x_{mix}^S and x_{mix}^T, via mixture of images of one domain with patches from other domain. Thus, this cross-domain mixing is in bidirectional way, source-to-target and target-to-source. Specifically, some rectangular regions(*i.e.* patches) in target samples are cut and source patches retrieved from the source patch bank are pasted (*i.e.* source-to-target direction), and vice versa. We also apply same operation on labels, resulting in domain-mixed labels, y_{mix}^S and y_{mix}^T.

– Given the mixed images and labels, a segmentation network is trained with standard cross-entropy losses L_{cross}. The final loss is formulated as follows: $L_{final} = L_{cross}(x_{mix}^S, y_{mix}^S) + L_{cross}(x_{mix}^T, y_{mix}^T)$.

2.2 Cut

Cut is a process that masks out contiguous sections (*i.e.* multiple rectangular regions) of input and corresponding labels. We introduce widely used random patch cutout [5, 22, 15] and proposed confidence based cutout.

Confidence based cutout. Random region cutout is a simple but strong baseline that is adopted in various tasks such as UDA [15], Semi-DA [2] and SSL [19]. However, the random region cut the regions regardless of whether it is informative. Instead, we target to cut where provide noisy supervision in terms of learning the generalized features. To this end, we see that it is important to discard the regions with low confident predictions for the following reasons: 1) for the source domain, it remove non-transferable and source specific regions, 2) it prevent to learn from noisy pseudo labels of target data.

Given the source images x_i^S and their pseudo labels $\tilde{y}_i^S$, we calculates the ratio of the uncertain region over the randomly generated region $\mathbf{B} = (r_x, r_y, r_w, r_h)$. If the ratio of the uncertain region is above the cutout threshold γ, the region is cut. The proposed cutout is summarized as follows:

$$\hat{x}_i^S, \hat{y}_i^S = \begin{cases} \text{Cutout}(x_i^S, \tilde{y}_i^S), & \text{if} \quad \mathcal{H}(\tilde{y}_i^S, \mathbf{B}) > \gamma, \\ x_i^S, \tilde{y}_i^S & \text{otherwise}, \end{cases} \tag{1}$$

where $\mathcal{H}$ denote a function that computes the ratio of the uncertain region over region $\mathbf{B}$. And, the threshold γ is set to 0.2.

2.3 Paste

To generate the domain-mixed samples, we sample the patch from the other domain and paste it back to the region that are cutout. This enables the resulting samples to include both source and target patches.

We additionally consider three important factors during the patch sampling. First, we introduce **class-balanced** sampling. As random sampling tend to bias the mixup samples mainly toward the frequent classes, we offset this undesirable effect using class-balanced sampling. Since the long-tailed category distribution tends to be similar for different domains, we compute the class distribution of the source domain [23, 10] to provide more chance for patches that include rare classes. Second, we consider **spatial continuity** of the natural scene. The random sampling can produce mixed samples that are geometrically unnatural, and thus causes severe train and test time inconsistency. Instead, based on the fact that the vehicle egocentric view has strong fixed spatial priors [3, 26], we compute spatial priors for each semantic class [26] in the source domain and use this statistics during sampling. Finally, we take **pseudo label confidence** into account.

Patch Generation. In the source and target datasets, classes such as *train* and *bike* have a small number of samples, so to consider the class online, it is

necessary to find a sample including a 6 rare class from all samples. Since this method is memory inefficient, we cut the patch containing each class and save it.

We generate a patch by cutting the pseudo labeled datasets $\{(x_i^S, \tilde{y}_i^S)\}_{i=1}^{N_S}$ and $\{(x_i^T, \tilde{y}_i^T)\}_{i=1}^{N_T}$ at equal intervals by the number of horizontal $\mathbf{W}$ and vertical $\mathbf{H}$. Therefore, one image and its pseudo label are divided into $\mathbf{W} \times \mathbf{H}$ patches. After that, the patches extracted from each location are grouped into one patch sequence. Next, the patch containing the class $\{C_i\}_{i=1}^{K}$, $\mathbf{K}$ is the number of classes, from the pseudo label of the patch sequence representing each spatial location is stored in the child patch sequence, and it is possible to duplicate it. Therefore, patches considering spatial location and class existence are grouped into a total of $\mathbf{W} \times \mathbf{H} \times \mathbf{K}$ patch sequence. Finally, at the patch sequence of $\mathbf{W} \times \mathbf{H} \times \mathbf{K}$, the normalized confidence calculated by the method in Eq. 4 is sorted in ascending order, and $\mathbf{R}$ patch sequences are generated at regular intervals. The number of finally generated patch sequence is $\mathbf{W} \times \mathbf{H} \times \mathbf{K} \times \mathbf{R}$

Patch Selection

Class-balanced Patch Sampling. In addition to Zipfian distribution of object categories [11], the difference in the intrinsic size of each object makes pixel-level class imbalances more severe in semantic segmentation. However, the random patch sampling for paste make bias toward the frequent classes, as it naturally follow the probability of existence.

To alleviate the class imbalance problem in paste process, we propose patch-level oversampling. Given the total number of pixels for each classes in the source labels $\{\bar{N}^i\}_{i=1}^{K}$, the probability P_{CB} that each class patch is selected can be formulated as follows:

$$\begin{aligned} \hat{P}_{CB} &= \{(-\log(\frac{\bar{N}^i}{\sum_{i=0}^{\mathbf{K}} \bar{N}^i}))^{\alpha}\}_{i=1}^{K}, \\ P_{CB} &= \{(\frac{\hat{P}_{CB}^i}{\sum_{i=0}^{\mathbf{K}} \hat{P}_{CB}^i})\}_{i=1}^{K}, \end{aligned} \tag{2}$$

where $\mathbf{K}$ is the number of classes, α is the sharpening coefficient. We set α to 2 in all experiments.

Sampling with Spatial Continuity. The spatial layout between the source (synthetic) and target dataset share large similarities. Therefore, we propose spatial continuity based paste that considers spatial relationship instead of pasting patches at random positions. The probability of selecting each location $\{\text{Patches}_i\}_{i=1}^{\mathbf{W}\times\mathbf{H}}$ of the patches generated through Patch Generation section to be mixed with the patch locations cutout through Cut section is calculated as follows:

$$\begin{aligned} \hat{SC} &= \text{argmax}\{\text{SC}_i(o_w, o_h)_{i=1}^{K}\}, \\ P_{SC} &= \{\hat{SC}(\text{Patches}_i(o_{\hat{w}^i}, o_{\hat{h}^i}))\}_{i=1}^{\mathbf{W}\times\mathbf{H}}, \end{aligned} \tag{3}$$

where $\{\mathrm{SC}_i\}_{i=1}^{\mathbf{K}}$ is the source domain class-wise spatial prior kernel map generated by CBST-SP [26]. where o_h, o_w is the center coordinates of the cutout patch, o_h^i, o_w^i is the center coordinates of the patch at the i-th location. Note that, we normalized the sum of the set P_{SC} to 1.

Sampling with Normalized Confidence. Opposite to confidence based cutout, in the paste, we give high probability to the patches that include confident pixels. To faithfully measure the confidence level of a patch, we take the difficulty of each class into account and design the normalized confidence of a patch. We first calculate the average confidence of classes using the set of pseudo labels and use it to represent the difficulty of each class. Then, the confidence of patches in the patch bank is normalized at pixel-level by subtracting the difficulty score according to predicted classes (Norm). Intuitively, it measure the *relative* confidence level. The resulting normalized confidence maps are spatially averaged (Average), and patches are sorted in ascending order according to it (Sort).

$$\begin{aligned} \hat{B}^T &= \{Average(Norm(B_i^T))\}_{i=1}^{N_T}, \\ \hat{B}^S &= \{Average(Norm(B_i^S))\}_{i=1}^{N_S}, \\ \bar{B}^T &= Sort(\hat{B}^T), \\ \bar{B}^S &= Sort(\hat{B}^S) \end{aligned} \tag{4}$$

Finally, the patch is divided into three batches: the low, the middle, and the high confidence group. The probability P_{PC} that the patch in each group is selected is $\{0.1, 0.3, 0.6\}$.

Probability of selection of each patch sequence. We select the patch jointly considering class balance, spatial continuity, and pseudo label confidence for BDM. Since each probability is independent, the probability that each patch sequence is selected is $P_{CB} \times P_{SC} \times P_{PC}$.

3 Experiments

In this section, we present experimental results to validate the proposed BDM for domain adaptive semantic segmentation.

We first describe experimental configurations in detail. After that, we validate our BDM on two public benchmark datasets. Note that the Intersection-over-Union (IoU) metric is used for all the experiments.

3.1 Experimental Settings

Dataset. We evaluate our proposed Bidirectional Domain Mixup on two popular domain adaptive semantic segmentation benchmarks(SYNTHIA $\rightarrow$ Cityscapes, and GTA5 $\rightarrow$ Cityscapes). Cityscapes [4] is a real-world urban scene dataset consisting of a training set with 2,975 images, a validation set with 500 images

Table 1. Comparison with state-of-the-art models on GTA5 → Cityscapes. We highlight the mIoU of tail classes (*i.e.* mIoU-tail) along with per-class IoU and overall mIoU. Our results are averaged over five runs.

			Head Classes													Tail Classes					
Method	mIoU	mIoU-tail	road	sidewalk	building	wall	fence	pole	vegetation	terrain	sky	person	car	truck	bus	light	sign	rider	train	motorcycle	bike
Source Only	36.6	24.0	75.8	16.8	77.2	12.5	21.0	25.5	81.3	24.6	70.3	53.8	49.9	17.2	25.9	30.1	20.1	26.4	6.5	25.3	36.0
Adaptseg [16]	41.4	25.0	86.5	25.9	79.8	22.1	20.0	23.6	81.8	25.9	75.9	57.3	76.3	29.8	32.1	33.1	21.8	26.2	7.2	29.5	32.5
ADVENT [17]	45.5	25.7	89.4	33.1	81.0	26.6	26.8	27.2	83.9	36.7	78.8	58.7	84.8	38.5	44.5	33.5	24.7	30.5	1.7	31.6	32.4
CCM [9]	49.9	26.9	93.5	57.6	84.6	39.3	24.1	25.2	85.0	40.6	86.5	58.7	85.8	49.0	56.4	35.0	17.3	28.7	5.4	31.9	43.2
IAST [12]	51.5	34.0	93.8	57.8	85.1	39.5	26.7	26.2	84.9	32.9	88.0	62.6	87.3	39.2	49.6	43.1	34.7	29.0	23.2	34.7	39.6
DACS [15]	52.1	32.6	89.9	39.6	87.8	30.7	39.5	38.5	87.9	43.9	88.7	67.2	84.4	45.7	50.1	46.4	52.7	35.7	0.0	27.2	33.9
DSP [6]	55.0	36.9	92.4	48.0	87.4	33.4	35.1	36.4	87.7	43.2	89.8	66.6	89.9	57.1	56.1	41.6	46.0	32.1	0.0	44.1	57.8
CAMix [25]	55.2	37.9	93.3	58.2	86.5	36.8	31.5	36.4	87.2	44.6	88.1	65.0	89.7	46.9	56.8	35.0	43.5	24.7	27.5	41.1	56.0
CorDA [18]	56.6	38.5	**94.7**	63.1	87.6	30.7	40.6	40.2	87.6	47.0	**89.7**	66.7	90.2	48.9	57.5	47.8	51.6	35.9	0.0	39.7	56.0
ProDA [23]	57.5	42.0	87.8	56.0	79.7	46.3	44.8	45.6	88.6	45.2	82.1	70.7	88.8	45.5	59.4	53.5	53.5	39.2	1.0	48.9	56.4
DAP [8]	59.8	44.6	94.5	**63.1**	89.1	29.8	47.5	50.4	89.5	50.2	87.0	**73.6**	**91.3**	50.2	52.9	56.7	58.7	38.6	0.0	50.2	**63.5**
Adaptseg [16] + Ours	57.4(+16.0)	44.4	89.3	50.0	88.4	45.6	45.4	41.1	78.0	35.6	82.3	69.2	87.5	**55.7**	57.8	49.9	**60.2**	45.6	8.1	45.5	57.2
ADVENT [17] + Ours	57.6(+12.1)	38.9	91.3	51.8	86.7	49.9	49.2	53.3	85.8	47.9	85.7	62.3	87.8	55.5	54.4	43.1	43.3	45.9	4.4	46.3	50.4
IAST [12] + Ours	61.0(+9.5)	46.5	92.1	59.6	**89.9**	52.9	55.7	49.2	89.3	46.7	86.3	59.1	88.3	54.8	55.9	44.6	45.8	42.0	**39.2**	50.3	57.3
ProDA [23] + Ours	**63.9**(+6.4)	**47.8**	89.2	60.1	83.8	**61.5**	**63.6**	**66.7**	**90.4**	**51.1**	83.5	72.6	88.0	51.2	**65.3**	**58.2**	59.3	**47.8**	1.0	**60.1**	60.9
Target Only	64.5	53.0	96.2	75.5	87.7	38.0	39.6	43.4	88.2	52.4	89.5	69.7	91.4	66.2	69.7	46.6	62.8	49.5	45.0	49.0	65.1

and a testing set with 1,525 images. We use the unlabeled training dataset as $\{D_i^T\}_{i=1}^{2,975}$ and evaluate our Bidirectional Domain Mixup with 500 images from the validation set. SYNTHIA [14] is a synthetic urban scene dataset. We pick SYNTHIA-RAND-CITYSCAPES subset as the source domain, which shares 16 semantic classes with Cityscapes. In total, 9,400 images from SYNTHIA dataset are used as source domain training data $\{D_i^S\}_{i=1}^{9,400}$ for the task. GTA5 [13] dataset is another synthetic dataset sharing 19 semantic classes with Cityscapes. 24,966 urban scene images are collected from a physically-based rendered video game Grand Theft Auto V (GTAV) and are used as source training data $\{D_i^S\}_{i=1}^{24,966}$. We view 6 and 5 with relatively few training samples as tail-classes for each source domain(GTA5, SYNTHIA), respectively.

3.2 Comparison with State-of-the art

In this section, we compare our proposed method with the top-performing UDA approach.

Table 1 shows the comparisons on GTA5 → Cityscapes setting. DACS, which classmix at random locations without considering the class distribution of the source dataset, significantly improved performance in classes with a large number of samples, but significantly decreased in tail classes such as *bike* and *train*. CAMix, a classmix method considering the contextual relationship, solved the problem of performance degradation of tail classes through consistency loss and dynamic threshold. However, considering only the contextual relationship, the performance decreased in *wall*, *light*, and *rider* classes, which have significantly different contextual relationship between GTA5 and Cityscapes.

On the other hand, our BDM jointly consider class balance, spatial continuity, and pseudo label confidence and achieve state-of-the-art with an mIoU score of

Table 2. Comparison with state-of-the-art models on SYNTHIA → Cityscapes. Our results are averaged over five runs.

			Head Classes								Tail Classes				
Method	mIoU	mIoU-tail	road	sidewalk	building	vegetation	sky	person	car	bus	light	sign	rider	motorcycle	bike
Source Only	40.3	20.8	64.3	21.3	73.1	63.1	67.6	42.2	73.1	15.3	7.0	27.7	19.9	10.5	38.9
Adaptseg [16]	45.8	18.5	79.5	37.1	78.2	78.0	80.3	53.7	67.1	29.4	9.3	10.6	19.2	21.8	31.6
ADVENT [9]	48.0	16.9	85.6	42.2	79.7	80.4	84.1	57.9	73.3	36.4	5.4	8.1	23.8	14.2	33.0
CCM [9]	52.9	29.0	79.6	36.4	80.6	81.8	77.4	56.8	80.7	45.2	22.4	14.9	25.9	29.9	52.0
IAST [12]	57.0	35.2	81.9	41.5	83.3	83.4	85.0	65.5	86.5	38.2	30.9	28.8	30.8	33.1	52.7
DACS [15]	54.8	32.1	80.5	25.1	81.9	83.6	**90.7**	67.6	82.9	38.9	22.6	23.9	38.3	28.4	47.5
DSP [6]	59.9	35.9	86.4	42.0	82.0	87.2	88.5	64.1	83.8	65.4	31.6	33.2	31.9	28.8	54.0
CAMix [25]	59.7	33.8	91.8	54.9	83.6	83.8	87.1	65.0	85.5	55.1	23.0	29.0	26.4	36.8	54.1
CorDA [18]	62.8	41.5	**93.3**	**61.6**	85.3	84.9	90.4	69.7	85.6	38.4	36.6	42.8	**41.8**	32.6	53.9
ProDA [23]	62.0	40.3	87.8	45.7	84.6	88.1	84.4	74.2	88.2	51.1	54.6	37.0	24.3	40.5	45.6
DAP [8]	64.3	**48.7**	84.2	46.5	82.5	**89.3**	87.5	75.7	**91.7**	**73.5**	53.6	**45.7**	34.6	**49.4**	**60.5**
Adaptseg*[16] + Ours	50.6(+4.8)	24.1	84.5	44.3	79.5	84.2	83.3	60.1	69.3	33.2	19.6	18.7	23.6	24.0	34.7
ADVENT [16] + Ours	53.8(+5.8)	27.1	85.4	47.7	82.9	86.5	85.7	64.5	72.0	39.3	25.2	22.2	25.6	26.1	36.4
IAST [12] + Ours	62.9(+5.9)	42.6	88.2	50.2	**88.5**	85.6	89.7	70.2	88.3	44.8	42.3	33.5	39.0	39.4	59.0
ProDA [23] + Ours	**66.8**(+4.8)	47.7	91.0	**55.8**	86.9	85.8	85.7	**84.1**	86.0	55.2	**58.3**	44.7	40.3	45.0	50.6
Target Only	72.3	54.6	96.2	75.5	87.7	88.2	89.5	69.7	91.4	69.7	46.6	62.8	49.5	49.0	65.1

* Note that pre-trained weights are not provided, so use them after reproduction.

63.9% when ProDA was selected as the warm-up model. Despite the great overall scores, it showed a low IoU in the *train* class. The rationale behind this is the severely poor performance of the chosen warm-up model in that class. Instead, when the warm-up model is switched to IAST, we achieved much improved scores (23.2% → 39.2%) IoU in the *train* class.

Last but not least, our method shows consistent performance improvement with four different warm-up models, showing the generality of our framework.

Table 2 shows the comparisons of SYNTHIA → Cityscapes adaptation. Again, our BDM achieved state-of-the-art with an mIoU score of 66.8% when ProDA was selected as the warm-up model. These experimental results indicate that BDM is valid not only in the GTA5 dataset, where the scene layout between the source and target dataset is highly similar but also in the SYNTHIA dataset which includes images with different viewpoints (e.g. top-down view).

4 Conclusions

In this paper, we proposed Bidirectional Domain Mixup (BDM), a cutmix method that cut the low confidence region and selects a patch to paste according to class-balance(CB), spatial continuity(SC), and pseudo label confidence(PC) in the corresponding region. Our proposed BDM achieves state-of-the-art in GTA5 to cityscapes benchmark and SYNTHIA to cityscapes benchmark with a large gap.

References

1. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
2. Chen, S., Jia, X., He, J., Shi, Y., Liu, J.: Semi-supervised domain adaptation based on dual-level domain mixing for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11018–11027 (2021)
3. Choi, S., Kim, J.T., Choo, J.: Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9373–9383 (2020)
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
5. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
6. Gao, L., Zhang, J., Zhang, L., Tao, D.: Dsp: Dual soft-paste for unsupervised domain adaptive semantic segmentation. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 2825–2833 (2021)
7. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5356–5364 (2019)
8. Huo, X., Xie, L., Hu, H., Zhou, W., Li, H., Tian, Q.: Domain-agnostic prior for transfer semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7075–7085 (2022)
9. Li, G., Kang, G., Liu, W., Wei, Y., Yang, Y.: Content-consistent matching for domain adaptive semantic segmentation. In: European conference on computer vision. pp. 440–456. Springer (2020)
10. Li, R., Li, S., He, C., Zhang, Y., Jia, X., Zhang, L.: Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11593–11603 (2022)
11. Manning, C., Schutze, H.: Foundations of statistical natural language processing. MIT press (1999)
12. Mei, K., Zhu, C., Zou, J., Zhang, S.: Instance adaptive self-training for unsupervised domain adaptation. In: European conference on computer vision. pp. 415–430. Springer (2020)
13. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: European conference on computer vision. pp. 102–118. Springer (2016)
14. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: 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. pp. 3234–3243 (2016)
15. Tranheden, W., Olsson, V., Pinto, J., Svensson, L.: Dacs: Domain adaptation via cross-domain mixed sampling. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1379–1389 (2021)

16. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7472–7481 (2018)
17. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2517–2526 (2019)
18. Wang, Q., Dai, D., Hoyer, L., Van Gool, L., Fink, O.: Domain adaptive semantic segmentation with self-supervised depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
19. Wang, Y., Wang, H., Shen, Y., Fei, J., Li, W., Jin, G., Wu, L., Zhao, R., Le, X.: Semi-supervised semantic segmentation using unreliable pseudo-labels. arXiv preprint arXiv:2203.03884 (2022)
20. Yang, Y., Soatto, S.: Fda: Fourier domain adaptation for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4085–4095 (2020)
21. Yi, J.S.K., Seo, M., Park, J., Choi, D.G.: Using self-supervised pretext tasks for active learning. In: Proc. ECCV (2022)
22. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6023–6032 (2019)
23. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12414–12424 (2021)
24. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6881–6890 (2021)
25. Zhou, Q., Feng, Z., Gu, Q., Pang, J., Cheng, G., Lu, X., Shi, J., Ma, L.: Context-aware mixup for domain adaptive semantic segmentation. IEEE Transactions on Circuits and Systems for Video Technology (2022)
26. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Domain adaptation for semantic segmentation via class-balanced self-training. arXiv preprint arXiv:1810.07911 (2018)
27. Zou, Y., Yu, Z., Kumar, B.V., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)