Spatial Class Distribution Shift in Unsupervised Domain Adaptation: Local Alignment Comes to Rescue

Safa Cicek[1] Ning Xu[2] Zhaowen Wang[2] Hailin Jin[2] Stefano Soatto[1]

[1] University of California, Los Angeles, {safacicek,soatto}@ucla.edu
[2] Adobe Research, {nxu,zhawang,hljin}@adobe.com

Abstract. We propose a method for semantic segmentation in the unsupervised domain adaptation (UDA) setting. We particularly examine the domain gap between spatial-class distributions and propose to align the local distributions of the segmentation predictions. Despite its simplicity, the proposed method achieves state-of-the-art results in UDA segmentation benchmarks.

1 Introduction

Unsupervised domain adaptation (UDA) consists of modifying a model trained on a labeled dataset, called the "source" so it can function on data from a different "target" domain, for which no annotations are available [1,2,3,4,5,6]. More in general, we want to train a model to operate on input data from both the source and target domains, despite the absence of annotated data for the latter. For instance, one may have a synthetic dataset, where annotation comes for free, but wish for the resulting model to work well on real data, where the manual annotation is scarce or absent. UDA setting is even more vital for semantic segmentation tasks where annotation and quality control per a single image requires more than 1.5 hours on average [7].

In this work, we focus on the UDA segmentation task where the goal is to estimate the segmentation map $y \in \{0,1\}^{K\times H\times W}$ for a given RGB image where K is the number of classes. For the source samples, we have access to ground-truth labels $y^s \in Y$ which are used to minimize cross-entropy loss[3],

$$L_{ce}(P^s; f) := \mathbb{E}_{(x^s,y^s)\sim P^s} \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \ell_{CE}(f(x^s)_{ij}; y^s_{ij}) \tag{1}$$

where f is the segmentation network and P^s is the distribution of source domain samples and corresponding labels, $\ell_{CE}(f(x)_{ij}; y_{ij}) := -\langle y_{ij}, \log f(x)_{ij} \rangle$ is calculated using the one-hot ground-truth vector $y_{ij} \in \{0,1\}^K$ and class label estimates $f(x)_{ij} \in \mathbb{R}^K$ from the output of a deep neural network with input x. Next, we discuss the proposed methods for leveraging unlabeled target data and

[3] We denote label and prediction corresponding to a pixel coordinates of (i,j) with $y_{ij} \in \{0,1\}^K$ and $f(x)_{ij} \in \mathbb{R}^K$ respectively.

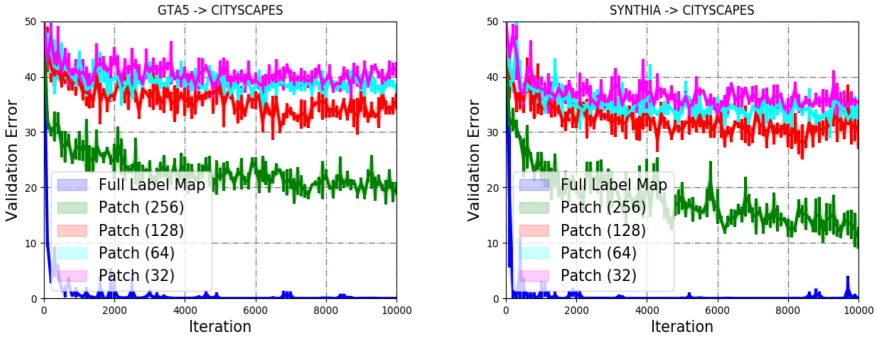


Fig. 1: **Spatial-class distribution shift correlates with the receptive field on the segmentation maps.** Validation errors for a binary classifier trained to distinguish binary domain labels from segmentation maps are given for GTA5 $\rightarrow$ Cityscapes (left) and SYNTHIA $\rightarrow$ Cityscapes (right). If the domain gap between segmentation maps are large, then error decays faster. We repeat this experiment for different receptive fields. When the binary classifier is trained on the entire segmentation maps (blue curve), errors decay quickly, whereas, for smaller patch sizes, learning slows down. For patch sizes smaller than 128, predictions of the classifiers are close to luck (50%). Errors for SYNTHIA are slightly lower due to the larger spatial-class shift between SYNTHIA and Cityscapes. This experiment verifies that even when the spatial class distribution shift is large between the global segmentation maps, local segmentation maps are still almost indistinguishable.

Fig. 2: **Samples from the datasets.** Spatial-class distributions vary from source to target domains for the UDA segmentation benchmarks; namely, SYNTHIA $\rightarrow$ Cityscapes and GTA5 $\rightarrow$ Cityscapes. SYNTHIA images are generated with random camera views, unlike Cityscapes which only have dashcam views. On the other hand, in GTA5, there are unrealistic scenarios e.g. ego-vehicle driving on the sidewalk.

our contribution in relation to this vast literature (Section 2). In the following section, we describe the proposed method (Section 3). Finally, we put it to the test on UDA segmentation benchmarks (Section 4).

2 Related Work

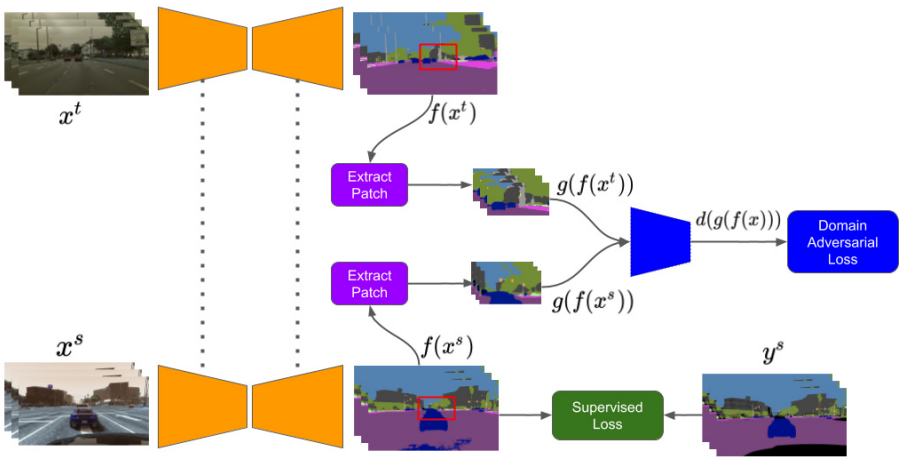


Fig. 3: The network structure of the proposed approach. Our binary domain discriminator (blue) acts on the *random* patches of predictions ($g(f(x))$) and not on the global segmentation maps ($f(x)$).

In the following, we present some of the previous works on the UDA segmentation tasks excluding the ones focusing on different tasks for the sake of space. AdvEnt [8] is the baseline method. It is observed that in the source-only training, entropy is mostly low for the source predictions and only high at the edges, while the entropy is mostly high on the predictions of the target images. Hence, they proposed to align the "weighted self-information" to minimize the entropy of target predictions while aligning them to source predictions. We improved this work in an orthogonal direction by performing a random-patch alignment. [9,10] proposed heuristics to have class-conditional alignments. [11] follows a curriculum learning approach: where they sequentially learn pseudo-labels for the entire image, superpixels, and finally the dense predictions.

[12] proposed to use a fully convolutional network (PatchGAN) for image translation which is later employed in several UDA works [13]. However, unlike PatchGAN that divides the image into a fixed collection of patches, we randomize the location of patches. Therefore, the distributions we align using the domain discriminator have supports defined by the values of these prediction patches. While the cardinality of the patch set exponentially increases with the size of the

image for the proposed method, PatchGAN always uses a small subset of this set. Hence, the aligned distributions for the proposed method and PatchGAN are significantly different.

Work of [14] looks similar to ours as they partition images into 3 by 3 regions before alignment. But actually, it has the exact opposite motivation and outcome as they only align the corresponding patches while we choose our partitions completely randomly. This saddle difference has great importance. Our motivation for aligning "random" patches is that the spatial-class distribution across domains can vary a lot wheres for them the motivation is to leverage the hypothesis that corresponding patches have similar spatial distributions. But, in reality, as camera views are random in SYNTHIA, corresponding patches should not be aligned to those of Cityscapes. [19,18] updates network parameters using

Table 1: **Comparison to SOA on SYNTHIA → Cityscapes.** All models are trained on the labeled source training data (SYNTHIA) and unlabeled target training data (Cityscapes) and performances on Cityscapes validation split are reported. A+E [8] refers to the ensemble of two networks: one trained with the adversarial loss and the other is with entropy minimization. In the literature, two different mIoU scores are reported for this task: one is for 16 common classes between two domains (mIoU) and the other is for the 13 classes (mIoU-13) excluding *wall*, *fence*, *pole*. Our method outperforms all the previous methods in both metrics.

Method	Road	SW	Build	Wall*	Fence*	Pole*	TL	TS	Veg.	Sky	PR	Rider	Car	Bus	Motor	Bike	mIoU	mIoU-13
Source	14.9	11.4	58.7	1.9	0.0	24.1	1.2	6.0	68.8	76.0	54.3	7.1	34.2	15.0	0.8	0.0	23.4	26.8
MCD [15]	84.8	43.6	79.0	3.9	0.2	29.1	7.2	5.5	83.8	83.1	51.0	11.7	79.9	27.2	6.2	0.0	37.3	43.5
Source	55.6	23.8	74.6	-	-	-	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	-	38.6
AdaptSegNet [13]	84.3	42.7	77.5	-	-	-	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	-	46.7
CLAN [16]	81.3	37.0	80.1	-	-	-	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	-	47.8
MinEnt [8]	73.5	29.2	77.1	7.7	0.2	27.0	7.1	11.4	76.7	82.1	57.2	21.3	69.4	29.2	12.9	27.9	38.1	44.2
AdvEnt [8]	87.0	44.1	79.7	9.6	0.6	24.3	4.8	7.2	80.1	83.6	56.4	23.7	72.7	32.6	12.8	33.7	40.8	47.6
A+E [8]	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	41.2	48.0
Source	64.3	21.3	73.1	2.4	1.1	31.4	7.0	27.7	63.1	67.6	42.2	19.9	73.1	15.3	10.5	38.9	34.9	40.3
CBST [17]	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	78.3	60.6	28.3	81.6	23.5	18.8	39.8	42.6	48.9
MRL2 [18]	63.4	27.1	76.4	14.2	1.4	35.2	23.6	29.4	78.5	77.8	61.4	29.5	82.2	22.8	18.9	42.3	42.8	48.7
MRENT [18]	69.6	32.6	75.8	12.2	1.8	35.3	23.3	29.5	77.7	78.9	60.0	28.5	81.5	25.9	19.6	41.8	43.4	49.6
MRKLD [18]	67.7	32.2	73.9	10.7	1.6	37.4	22.2	31.2	80.8	80.5	60.8	29.1	82.8	25.0	19.4	45.3	43.8	50.1
LRENT [18]	65.6	30.3	74.6	13.8	1.5	35.8	23.1	29.1	77.0	77.5	60.1	28.5	82.2	22.6	20.1	41.9	42.7	48.7
Ours	90.6	51.34	81.96	11.77	0.32	29.51	11.72	12.38	82.69	84.7	58.57	24.73	81.94	36.37	17.11	41.75	**44.84**	**51.99**

the pseudo-labels for which the network is confident. They incorporate spatial priors into the proposed CBST framework, leading to CBST with spatial priors (CBST-SP) by counting the class frequencies in the source domain, followed by smoothing with a Gaussian kernel to approximate the frequency of each class at a spatial location in the image space. Then, they simply modulate the network output with this spatial prior. Again, this idea contradicts the fact that the spatial distributions of segmentation maps differ across domains.

[20] combined previous works of curriculum [11] and self training [19]. Instead of super-pixels, they use patches of sizes 4 and 8. The key idea is to

alternatively update labels and network weights. They apply average pooling on the predictions and pseudo labels and minimize a classification loss between them. Our approach fundamentally differs from this as we align the predictions on the source and the target images whereas they align the predictions of the pre-trained network (providing pseudo-labels) and the main network both on the target images. Hence, unlike us, their algorithm may not align the local-prediction distributions across domains, which is the main motivation of this work.

The proposed method is orthogonal to most of the methods from this rich and multi-faceted context and can be improved by incorporating some of these ideas.

3 Proposed Method

As can be seen in Fig 2, spatial-class distribution can greatly differ from source to target domains, due to scene structure, camera view changes, etc. This contradicts with the idea of aligning the segmentation network outputs globally via minimax losses.

Before describing the proposed method, we conducted a motivational experiment to verify and quantify this hypothesis. For this purpose, we train a binary domain classifier on the ground-truth segmentation maps to measure the identifiability of the domain label from segmentation maps. See Fig. 1 where the left panel is for GTA5-Cityscapes and the right one is for SYNTHIA-Cityscapes. When the task is to distinguish the global segmentation maps, classifiers can easily detect the domain (blue curves). As we decrease the receptive field on the segmentation maps, by cropping smaller patch sizes, it is getting harder for the classifier to find the correct domain label. For very small patch sizes, the performance of the classifier is close to chance (50%). Details on the training is given in the Section 4.1.

This experiment quantifies and verifies two almost obvious claims: First, the global segmentation maps can have different distributions across domains (i.e. spatial-class distribution shift) hence one should not align the global segmentation predictions at the training time. Second, even if the spatial-class distribution is very large (e.g. SYNTHIA → Cityscapes), the local segmentation maps can have similar distributions across domains. Assuming this as a fact for any cross-domain task -which we only verify for UDA segmentation benchmarks-, one should align random patches of predictions at the training time, for network predictions on different domains to abide by this phenomenon.

3.1 Loss Functions

A natural choice for aligning the cropped prediction distributions for the source and the target domains is to optimize a domain adversarial loss on the extracted patches from the segmentation predictions:

$$L_{adv}(P_x^s, P_x^t; f, d) := \mathbb{E}_{x^s \sim P_x^s, x^t \sim P_x^t} \ell_{CE}\Big(\psi(x^s), [0,1]\Big) + \ell_{CE}\Big(\psi(x^t), [1,0]\Big) \quad (2)$$

where $\psi(x) := d(g(f(x))$, g randomly extracts a patch of size $i < H$ and $j < W$ from the segmentation prediction $f(x)$, and ℓ_{CE} is cross entropy loss. P_x^s, P_x^t are marginal distributions of the source and the target domains. $d : x \mapsto \mathbb{R}^2$ is binary domain discriminator (see Fig. 3).

As in the previous work of [8], instead of applying domain adversarial loss on the segmentation maps $y \in \{0,1\}^{K \times H \times W}$ directly, we found aligning the "self-information maps", $\overline{y} = h(y) \in \mathbb{R}^{K \times H \times W}$ where $\overline{y}_{kij} = h(y_{kij}) := -y_{kij} \log y_{kij}$ more effective.[4] Hence, the final objective function becomes,

$$L_{advent}(P_x^s, P_x^t; f, d) := \mathbb{E}_{x^s \sim P_x^s, x^t \sim P_x^t} \ell_{CE}\Big(\overline{\psi}(x^s), [0,1]\Big) + \ell_{CE}\Big(\overline{\psi}(x^t), [1,0]\Big) \tag{3}$$

where $\overline{\psi}(x) := d(g(h(f(x)))$. Then, the overall optimization problem solved by the segmentation network f is,

$$\min_f \max_d L_{ce}(P^s; f) - \lambda L_{advent}(P_x^s, P_x^t; f, d). \tag{4}$$

Since there is no closed form solution, the objective function is optimized by the segmentation network f and the domain discriminator d in an alternating fashion using SGD.

4 Empirical Evaluation

4.1 Implementation Details

Datasets. To evaluate the performance of the proposed method, we put it to test on the standard UDA segmentation benchmarks: GTA5 → Cityscapes and SYNTHIA → Cityscapes and compare against SOA and baseline methods.

GTA5 dataset consists of 24966 images with a resolution of 1914 × 1052 and collected from the video game based on the city of Los Angeles. The resolution of the images in the target set Cityscapes is 2048 × 1024. The number of target training images is 2975. Methods are tested on the 500 samples of the validation split of Cityscapes. For GTA5 → Cityscapes, 19 common classes are used. These are the same classes as the ones used in Cityscapes benchmark [7] where rare classes are excluded from the evaluation.

SYNTHIA [21] is generated by rendering a virtual city created with the Unity development platform. RANDCITYSCAPES subset of SYNTHIA is used as the source training set. This subset consists of 9400 frames of the city taken

[4] Note that this is not exactly entropy of the predictions as the y_{kij} terms are not summed over the class dimension k. But, the source sample predictions have low entropy as cross-entropy is minimized on them. Adversarial alignment results in aligned $\overline{y}_{kij}$ distributions and thus, results in low entropy for the target predictions as well. As in [8], we found this weighted-scheme more effective because this adversarial loss promotes both the low entropy target predictions and aligned prediction maps across domains.

from a virtual array of cameras moving randomly. We choose the 16 overlapping classes between SYNTHIA and Cityscapes following the earlier works [17,19,9,11]. Classes that do not exist in this setting, compared to GTA5 setting are *terrain*, *truck*, *train*. There is another setting only considering 13 classes excluding the classes *wall*, *fence* and *pole* [13,10]. Mean scores corresponding to these 13 classes are reported as mIoU-13.

Table 2: **Comparison to SOA on GTA5 → Cityscapes.** Same as Table 1 except for GTA5 → Cityscapes. Here, the results are reported on 19 common classes (mIoU). The proposed method outperforms 3 of 4 scores that previous SOA [18] reported and it is only 0.12% less than the best method (MRKLD) of [18] which selectively samples for hard classes. The proposed method can be combined with MRKLD, but we choose to report naked results to show the effectiveness of the proposed *random*-patch alignment.

Method	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Terrain	Sky	PR	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
Source	42.7	26.3	51.7	5.5	6.8	13.8	23.6	6.9	75.5	11.5	36.8	49.3	0.9	46.7	3.4	5.0	0.0	5.0	1.4	21.7
CyCADA [22]	79.1	33.1	77.9	23.4	17.3	32.1	33.3	31.8	81.5	26.7	69.0	62.8	14.7	74.5	20.9	25.6	6.9	18.8	20.4	39.5
Source	36.4	14.2	67.4	16.4	12.0	20.1	8.7	0.7	69.8	13.3	56.9	37.0	0.4	53.6	10.6	3.2	0.2	0.9	0.0	22.2
MCD [15]	90.3	31.0	78.5	19.7	17.3	28.6	30.9	16.1	83.7	30.0	69.1	58.5	19.6	81.5	23.8	30.0	5.7	25.7	14.3	39.7
Source	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
AdaptSegNet [13]	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
CLAN [16]	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
MinEnt [8]	84.4	18.7	80.6	23.8	23.2	28.4	36.9	23.4	83.2	25.2	79.4	59.0	29.9	78.5	33.7	29.6	1.7	29.9	33.6	42.3
MinEnt + ER [8]	84.2	25.2	77.0	17.0	23.3	24.2	33.3	26.4	80.7	32.1	78.7	57.5	30.0	77.0	37.9	44.3	1.8	31.4	36.9	43.1
AdvEnt [8]	89.9	36.5	81.6	29.2	25.2	28.5	32.3	22.4	83.9	34.0	77.1	57.4	27.9	83.7	29.4	39.1	1.5	28.4	23.3	43.8
A+E [8]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5
Source	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	29.2
FCAN [23]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	46.6
Source	71.3	19.2	69.1	18.4	10.0	35.7	27.3	6.8	79.6	24.8	72.1	57.6	19.5	55.5	15.5	15.1	11.7	21.1	12.0	33.8
CBST [17]	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
MRL2 [18]	91.9	55.2	80.9	32.1	21.5	36.7	30.0	19.0	84.8	34.9	80.1	56.1	23.8	83.9	28.0	29.4	20.5	24.0	40.3	46.0
MRENT [18]	91.8	53.4	80.6	32.6	20.8	34.3	29.7	21.0	84.0	34.1	80.6	53.9	24.6	82.8	30.8	34.9	16.6	26.4	42.6	46.1
MRKLD [18]	91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	26.9	26.0	42.3	**47.1**
LRENT [18]	91.8	53.5	80.5	32.7	21.0	34.0	29.0	20.3	83.9	34.2	80.9	53.1	23.9	82.7	30.2	35.6	16.3	25.9	42.8	45.9
Ours	89.72	32.54	82.19	31.27	25.12	30.11	38.0	24.64	84.68	41.36	76.59	60.25	29.2	86.27	39.05	51.43	1.37	29.07	39.73	46.98

Training details. We used ResNet101-based DeepLabv2 [24] without CRF post-processing [25] to have a fair comparison with SOA methods [17,18]. We compare our method with the ones reporting in the same setting where they do not exploit more advanced segmentation networks like DeepLabv3+ [26,27], thus we exclude [28,20] from the comparison. This is necessary to make fair comparison possible as DeepLabv3+ includes a decoder replacing bilinear interpolation with atrous convolutions. This results in better recovering of the object boundaries. Intersection Over Union (IoU) has been the standard evaluation metric for semantic segmentation task: $(IoU) = \frac{TP}{TP+FN+FP}$ where TP, FN and FP correspond to true positive, false negative and false positive respectively. Then, mean IoU (mIoU) is calculated by averaging IoU of all the classes. Pixel accuracy, $\frac{TP+TN}{TP+TN+FN+FP}$ also considers TN (pixels correctly identified as not belonging to the class). This is not a good metric if some classes are seen in a few pixels

Table 3: **Ablations on SYNTHIA → Cityscapes.** We compare the performance of the proposed method to the following baselines. (1) The source-only model is only trained on the labeled source examples minimizing the cross-entropy loss. (2) In AP-CI (Align Predictions of Cropped Images), *RGB images are cropped instead of prediction maps*. (3) AGP-GI (Align Global Predictions of Global Images) refers to minimizing the same adversarial loss (Eqn. 3) *on the global segmentation maps*. The proposed method, ALP-GI (Align Local Predictions of Global Images) outperforms all baselines with 15.13%, 31.59% and 4.69% (in mIoU) compared to Source-only, AP-CI (Align Predictions of Cropped Images) and AGP-GI (Align Global Predictions of Global Images) respectively. The last row shows the relative increase compared to the source-only baseline.

Method	Road	SW	Build	Wall*	Fence*	Pole*	TL	TS	Veg.	Sky	PR	Rider	Car	Bus	Motor	Bike	mIoU	mIoU-13
Source-only	57.18	26.41	72.05	6.29	0.15	25.56	8.87	11.12	74.06	80.6	53.69	12.39	49.46	5.25	7.66	20.39	31.95	36.86
AP-CI	33.12	21.75	58.02	0.21	0.0	10.82	0.0	0.37	53.43	42.52	24.12	0.58	24.16	1.22	0.01	1.24	16.97	20.04
AGP-GI	83.42	38.16	77.33	4.34	0.17	25.5	6.2	6.82	77.81	83.97	55.29	18.76	79.13	40.55	15.86	31.62	40.31	47.3
ALP-GI	90.6	51.34	81.96	11.77	0.32	29.51	11.72	12.38	82.69	84.7	58.57	24.73	81.94	36.37	17.11	41.75	44.84	51.99
Relative	33.42	24.93	9.91	5.48	0.17	3.95	2.85	1.26	8.63	4.1	4.88	12.34	32.48	31.12	9.45	21.36	12.89	15.13

only. A trivial solution would be to never estimate such classes. So, we also do not report on this metric. Batch size is set to be one due to memory constraints.

Table 4: **Ablations on GTA5 → Cityscapes.** Same as Table 3 except for GTA5 → Cityscapes. The proposed method, ALP-GI (Align Local Predictions of Global Images) outperforms all baselines with 9.22%, 26.36% and 6.42% (in mIoU) compared to Source-only, AP-CI (Align Predictions of Cropped Images) and AGP-GI (Align Global Predictions of Global Images) respectively. The last row shows the relative increase compared to the source-only baseline.

Method	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Terrain	Sky	PR	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
Source-only	75.46	24.6	65.09	11.91	10.58	28.19	27.45	14.64	79.71	32.04	70.56	52.26	20.1	71.91	28.7	48.5	1.42	16.89	37.36	37.76
AP-CI	44.41	14.72	58.47	12.14	1.0	16.64	1.28	1.67	70.77	12.19	65.4	41.38	0.16	27.9	9.43	13.93	0.0	0.25	0.0	20.62
AGP-GI	86.62	10.48	81.79	27.41	17.29	25.19	29.36	14.85	84.27	34.7	78.16	57.3	28.3	83.65	31.93	35.52	0.15	22.59	21.05	40.56
ALP-GI	89.72	32.54	82.19	31.27	25.12	30.11	38.0	24.64	84.68	41.36	76.59	60.25	29.2	86.27	39.05	51.43	1.37	29.07	39.73	46.98
Relative	14.26	7.94	17.1	19.36	14.54	1.92	10.55	10	4.97	9.32	6.03	7.99	9.1	14.36	10.35	2.93	-0.05	12.18	2.37	9.22

Hence, batch norm parameters are updated with momentum but current batch statistics are not used during training. Image width and height are resized to (760, 1280) for SYNTHIA, (720, 1280) for GTA5 and (512, 1024) for Cityscapes during training. The weight for the adversarial loss (λ) given in Eqn. 3 is set to be 0.001. The number of training iterations is 150000. The learning rate for segmentation network and domain discriminator are 2.5×10^{-4} and 10^{-4} respectively. SGD and Adam are used for optimizing segmentation and discriminator networks respectively.

Training details for the experiment in Fig 1. The implementation details for the motivational experiment are as follows. We report validation errors after each 100 training iterations. We randomly choose 500 samples from the source domains for validation and did not use them during training. Cityscapes have already the validation split of size 500 samples. In total, 1000 samples is used to calculate the validation errors. Errors are averaged over three runs. Deviations over different runs are shaded but in some regions, they are too small to be visible. Standard classifier, ResNet18 [29] is used as a binary classifier. Label maps are resized to (512, 1024) before and after cropping, to have the same size segmentation maps for both domains. SGD with momentum 0.9, weight decay 10^{-4}, and fixed learning rate of 10^{-3} is used.

4.2 Quantitative Evaluation

In Tables 1,2, we compare the proposed method against SOA methods. The proposed method especially shines on SYNTHIA $\rightarrow$ Cityscapes as for this task, spatial-class distribution shift is larger (See Fig. 1, 2).

Our method surpasses all the previous SOA methods in both metrics of SYNTHIA $\rightarrow$ Cityscapes. Previous SOA [18] applies mining on rarely predicted classes and manages to get relatively high scores even in the very challenging classes. For instance, our method could only achieve 1.37% in *train* class of GTA5 setting, while previous SOA [18] could perform 26.9%. Similarly, for *fence* class of SYNTHIA setting, we perform poorly. Domain shift for segmentation maps and RGB images of these classes is too large for achieving robust performance in these classes.

Other methods we compare are as follows. FCAN [23] which proposed to combine the image alignment and translation losses. FCAN does not report class-wise performance and only report on GTA5. [13] applied domain adversarial loss both at the hidden layers and the network outputs. [15] encourages the consistency of different classifiers by having one encoder and two classifiers. Both classifiers are trained on the labeled source samples. The distance between predictions of two classifiers on the same target sample is minimized by the encoder and maximized by classifiers. [16] leverages the consistency between two classifiers in a different way. If two classifiers agree on the prediction, they keep the adversarial loss weight for that prediction small. In both tasks, we outperform all these methods which are orthogonal to ours and it could be combined with them but here we report the naked results to highlight the role of *random*-patch alignment.

The closest apple-to-apple comparison to our method is with [8] AdvEnt which applies the same loss without random cropping on the predictions. The proposed method improves baseline AdvEnt method [8] from 47.6% to 51.99% in SYNTHIA $\rightarrow$ Cityscapes and 43.8% to 46.98% in GTA5 $\rightarrow$ Cityscapes. Moreover, note that A+E reported in [8] is an ensemble of two networks trained with different losses, so it is not directly comparable to our method. Nonetheless, the relative accuracy improvements to their reported numbers are 3.15% for GTA5 and 8.12% for SYNTHIA. More controlled ablations are discussed next.

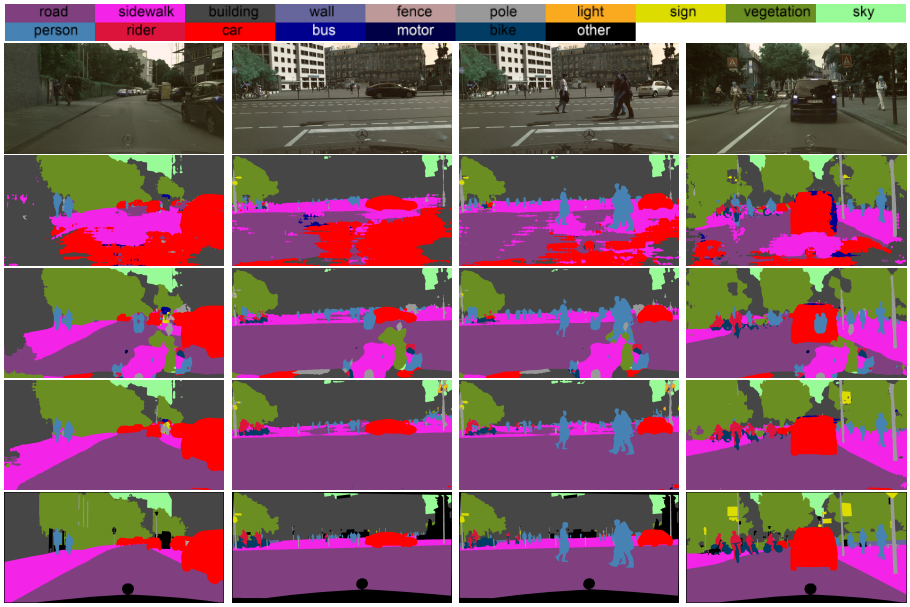


Fig. 4: **Qualitative results for SYNTHIA → Cityscapes**. Visuals of Cityscapes test set predictions are presented along with the corresponding RGB images. From top to bottom: (1) RGB image, (2) source-only prediction, (3) global alignment prediction, (4) our prediction and (5) ground truth segmentation. The proposed method especially performs well on the more common classes like *road* or *sidewalks* whereas it misses some of the small and rare objects like *traffic signs*.

In Tables 3,4, we compare the proposed method Align Local Predictions of Global Images (ALP-GI) against the following baselines: (1) Source-only, (2) Align Predictions of Cropped Images (AP-CI) and (3) Align Global Predictions of Global Images (AGP-GI).

Source-only baselines are only trained on the labeled source samples minimizing the cross-entropy loss. Our method improves source-only baselines with 15.13% and 9.22% for SYNTHIA → Cityscapes and GTA5 → Cityscapes respectively. Since the network and other training details are the same for all the methods, the improvement verifies the effectiveness of the proposed loss in leveraging the unlabeled target samples.

Similarly, the proposed method improves AGP-GI (Align Global Predictions of Global Images) baselines with 4.69% and 6.42% for SYNTHIA → Cityscapes and GTA5 → Cityscapes respectively. The improvement compared to AGP-GI verifies the significance of the *random*-patch alignment. Note that the results reported here for AGP-GI are slightly lower than those reported in [8]. The difference from AdvEnt-only (ours is 47.3, theirs is 47.6) is due to implementation differences. Moreover, they report the best results by ensembling two differ-

ent networks (A+E) whereas we report a single network's predictions for each method for having a controlled-experimental setting.

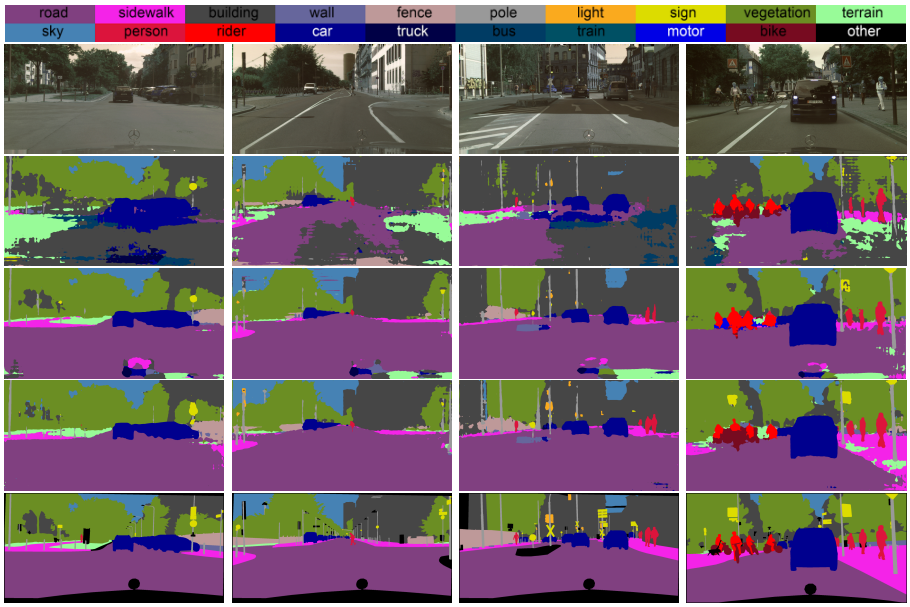


Fig. 5: **Qualitative results for GTA5 → Cityscapes**. Same as Fig. 4 except for GTA5 → Cityscapes. Again, network can correctly capture objects belonging to classes *road*, *car* while missing tiny objects (e.g. *traffic light*, *traffic sign*) or classes with large domain gap (e.g. *fence*).

Another way to align the prediction patches is simply to feed the cropped images to the network. But, the problem with this approach is that the network cannot leverage the scene information (i.e. larger context) when inferring the semantic segmentation map for small crop sizes. That is why in the proposed method, we minimize the cross-entropy loss on the entire images, and for the adversarial loss, *we randomly extract the patches of the predictions and not of the RGB images*. For completeness, we also evaluate this choice and report AP-CI (Align Predictions of Cropped Images) in Tables 3,4 where we compare the proposed method to simply cropping the RGB images with the same crop sizes. This baseline gives terrible results for the patch size of 256 as such a small receptive field makes it hard for the network to correctly capture the scene. As a result, the performance of this baseline is even lower than the source-only baseline which does not leverage any target sample. The proposed method surpasses this baseline with 31.59% and 26.36% for SYNTHIA → Cityscapes and GTA5 → Cityscapes respectively.

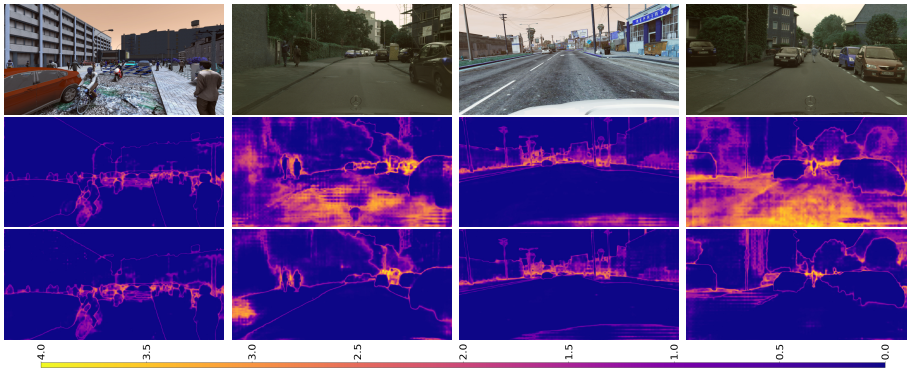


Fig. 6: **Entropy of predictions**. The entropy of the predictions on the source samples (columns 1,3) and the target test samples (columns 2,4) are given. From top to bottom: RGB image, the entropy of the source-only trained model and the entropy of the proposed method. Left two columns are for models trained on SYNTHIA → Cityscapes and right two columns are for GTA5 → Cityscapes. Color transitions from purple to yellow as the value of entropy increases. Entropy values are low on the source images for both the source-only and the proposed models (column 1,3) except edges due to the cross-entropy loss. For the target test samples (column 2,4), the entropy of predictions are small for the proposed method thanks to the adversarial loss while source-only models have high uncertainty on the target images.

For SYNTHIA → Cityscapes, our method outperforms the source-only model for all the classes as can be seen in the last row Table 3. But, the improvements are especially significant for the classes *road*, *sidewalk*, *car*, *bus* where accuracies (IoU) increased from 57.18, 26.41, 49.46, 5.25 to 90.6, 51.34, 81.94, 36.37 respectively. These are more common classes and the shapes of these objects do not significantly differ from one domain to other as in *fence* class. Hence, the proposed *random*-patch alignment method can leverage the object shapes learned from the source data. For GTA5 → Cityscapes, the proposed method improves the source-only baseline in all classes except *train*, which is a challenging class for this task as objects belonging to *train* class in GTA5 are far away from the ego-vehicle and they are hardly perceivable. The advantage of the proposed method is most apparent in the classes *building* and *wall* where IoU scores increased from 65.09 and 11.91 to 82.19 and 31.27 respectively.

4.3 Qualitative Evaluation

In Fig. 4 and 5, we present several qualitative results for SYNTHIA → Cityscapes and GTA5 → Cityscapes. In each figure, predictions of the source-only, global alignment, and the proposed methods along with corresponding RGB images and ground-truth segmentation maps are given. Black regions in the ground-truth maps belong to *other* class which are not evaluated at the test time.

Thanks to the proposed random-patch alignment regularization, the network learns the shape of the objects like *car*, *traffic signs*, and corrects the mistakes of the global alignment method. Even though the proposed method is quite successful in capturing the common classes *road*, *sidewalks* accurately, sometimes it can miss tiny and rare objects (e.g. belonging to class *fence*). In Fig. 6, we

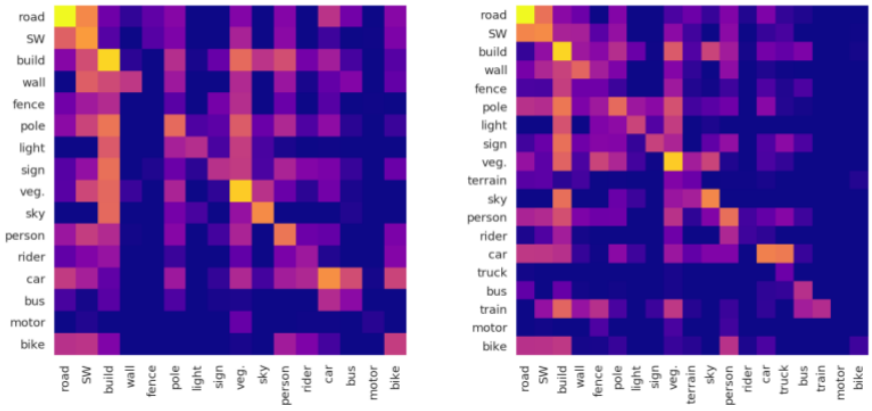
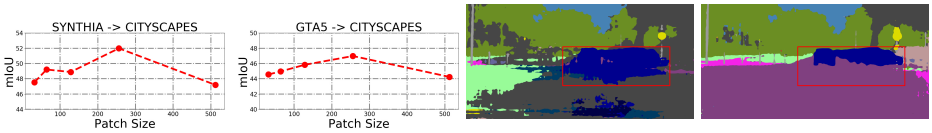


Fig. 7: **Confusion matrices**. Log-scaled confusion matrices for SYNTHIA → Cityscapes (left) and GTA5 → Cityscapes (right) are given. Confusion matrices are calculated by averaging over all the target test set. The value of the matrix at row i and column j is equal to the number of observations that should be classified as i and predicted to be j. As the value increases, color changes from purple to yellow. Networks are confused between *sidewalk* and *road* classes in both tasks. The classes *building* and *vegetation* are attracting classes that networks tend to misclassify objects belonging to other classes as one of two.

give the entropy of predictions for the source-only baseline and the proposed method for both tasks along with the corresponding RGB images. As expected, entropy values of the predictions on the target test set (column 2,4) are less for the proposed method thanks to the adversarial loss. Entropies have high values mostly on the edges. For source images (column 1,3), both models have small uncertainty as both minimize cross-entropy loss on them. In Fig. 7, we give log-confusion matrices on the target test set predictions of the proposed method for both tasks. As can be observed, some classes are more likely to be confused (e.g. sidewalks and road). Furthermore, the false-positive ratio is high for some classes like building and vegetation (i.e. network is tempted to predict objects belonging to other classes as one of the two). In Fig. 8a, we plot the performance on the target test set as a function of patch size. We get the best results when aligning the prediction crops of size 256 for both tasks. Based on the experiment in Fig. 1, for this size of segmentation maps, a strong discriminator can have



(a) Performance as a function of the patch size.

(b) The proposed method learns the shape of the objects from the labeled source domain.

Fig. 8: **(a)** mIoU on the target test samples are given for models trained to align different patch size predictions. The left panel is for SYNTHIA → Cityscapes and the right one is for GTA5 → Cityscapes. Both models achieve the best results when aligning the predictions of size 256. For smaller and larger patch sizes, the performance of the model decays. **(b)** Blue regions denoted with the red rectangles are estimated as a *car*. Such a *car* shape does not exist in any of the source segmentation patches, hence unless we perform the proposed adversarial loss, a discriminator can easily tell apart domains only by looking at the segmentation maps. So, this prediction will be corrected with the proposed loss. On the other hand, we do not promote global segmentation map alignment unlike previous works as the global segmentation distributions are not necessarily the same across domains.

predictions that are better than luck. However, still, the discriminator cannot reduce validation error below 20% (green curve) unlike the global alignment case (blue curve) where the validation error quickly drops to 0%. Moreover, aligning the predictions with this crop size is sufficient to have aligned predictions for smaller path sizes. In Fig. 8b, we present an illustrative example of how the proposed adversarial loss helps to learn the shapes of the objects from the labeled source domain and results in improved predictions compared to the source-only predictions.

5 Conclusion

We proposed a simple yet, effective solution to the spatial-class distribution shift problem and proved its effectiveness by performing at the state-of-the-art in UDA segmentation benchmarks. We further verified its success in the more controlled settings with the ablation studies. The method performs the best for UDA tasks with large spatial-class shifts (e.g. SYNTHIA Cityscapes). The proposed method adds no computational cost to the baseline method and it takes approximately 25 hours to train with a single Nvidia Tesla V100. Results on the Berkeley Deep Driving dataset are given in Supp. Mat.

Acknowledgment Research supported by ONR N00014-19-1-2229.

References

1. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing (2018) 1
2. Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: AAAI. Volume 6. (2016) 8 1
3. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Computer Vision and Pattern Recognition (CVPR). Volume 1. (2017) 4 1
4. Cicek, S., Soatto, S.: Unsupervised domain adaptation via regularized conditional alignment. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 1416–1425 1
5. Cicek, S., Fawzi, A., Soatto, S.: Saas: Speed as a supervisor for semi-supervised learning. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 149–163 1
6. Cicek, S., Soatto, S.: Input and weight space smoothing for semi-supervised learning. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. (2019) 0–0 1
7. 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. (2016) 3213–3223 1, 6
8. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. arXiv preprint arXiv:1811.12833 (2018) 3, 4, 6, 7, 9, 10
9. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016) 3, 7
10. Chen, Y.H., Chen, W.Y., Chen, Y.T., Tsai, B.C., Frank Wang, Y.C., Sun, M.: No more discrimination: Cross city adaptation of road scene segmenters. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 1992–2001 3, 7
11. Zhang, Y., David, P., Gong, B.: Curriculum domain adaptation for semantic segmentation of urban scenes. In: The IEEE International Conference on Computer Vision (ICCV). Volume 2. (2017) 6 3, 4, 7
12. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 1125–1134 3
13. 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. (2018) 7472–7481 3, 4, 7, 9
14. Chen, Y., Li, W., Van Gool, L.: Road: Reality oriented adaptation for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 7892–7901 4
15. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 3723–3732 4, 7, 9
16. Luo, Y., Zheng, L., Guan, T., Yu, J., Yang, Y.: Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 2507–2516 4, 7, 9

17. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Domain adaptation for semantic segmentation via class-balanced self-training. arXiv preprint arXiv:1810.07911 (2018) 4, 7
18. Zou, Y., Yu, Z., Liu, X., Kumar, B., Wang, J.: Confidence regularized self-training. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 5982–5991 4, 7, 9
19. Zou, Y., Yu, Z., Vijaya Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European conference on computer vision (ECCV). (2018) 289–305 4, 7
20. Lian, Q., Lv, F., Duan, L., Gong, B.: Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 6758–6767 4, 7
21. 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. (2016) 3234–3243 6
22. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. arXiv preprint arXiv:1711.03213 (2017) 7
23. Zhang, Y., Qiu, Z., Yao, T., Liu, D., Mei, T.: Fully convolutional adaptation networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 6810–6818 7, 9
24. 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** (2017) 834–848 7
25. Krähenbühl, P., Koltun, V.: Parameter learning and convergent inference for dense random fields. In: International Conference on Machine Learning. (2013) 513–521 7
26. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017) 7
27. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). (2018) 801–818 7
28. Zhang, Q., Zhang, J., Liu, W., Tao, D.: Category anchor-guided unsupervised domain adaptation for semantic segmentation. In: Advances in Neural Information Processing Systems. (2019) 433–443 7
29. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778 9