Semantically Adaptive Image-to-image Translation for Domain Adaptation of Semantic Segmentation

Luigi Musto
luigi.musto@studenti.unipr.it
Andrea Zinelli
andrea.zinelli1@studenti.unipr.it

University of Parma
Parma, IT

Abstract

Domain shift is a very challenging problem for semantic segmentation. Any model can be easily trained on synthetic data, where images and labels are artificially generated, but it will perform poorly when deployed on real environments. In this paper, we address the problem of domain adaptation for semantic segmentation of street scenes. Many state-of-the-art approaches focus on translating the source image while imposing that the result should be semantically consistent with the input. However, we advocate that the image semantics can also be exploited to guide the translation algorithm. To this end, we rethink the generative model to enforce this assumption and strengthen the connection between pixel-level and feature-level domain alignment. We conduct extensive experiments by training common semantic segmentation models with our method and show that the results we obtain on the synthetic-to-real benchmarks surpass the state-of-the-art.

1 Introduction

Deep neural networks for the semantic segmentation of street scenes require to be trained on large and heterogeneous datasets to achieve good accuracy and generalize well. Nevertheless, they still might fail in unseen scenarios and environments (e.g. because of adverse weather). Collecting and manually annotating datasets which can cover all these scenarios requires a huge effort, since the cost of per-pixel labeling is too high.

Simulators, instead, allow to generate unlimited labeled data with low effort. Driving simulators, for example, only require to setup the needed scene and to drive in it to collect the required data. Despite the advances and the photorealism of modern computer graphics, simulators still fail at generating images visually similar to the real ones, which is why models trained naively on such kind of data perform poorly when deployed in the real world.

This setting falls in the more general problem of Domain Adaptation: we have access to two domains, source and target, and we want to exploit the source domain to maximize the accuracy in the target domain, for a given task. When we do not have access to the target labels, but only source ones, we call this Unsupervised Domain Adaptation (UDA). In our case we can formalize the source and target domains to be a synthetic and real dataset.

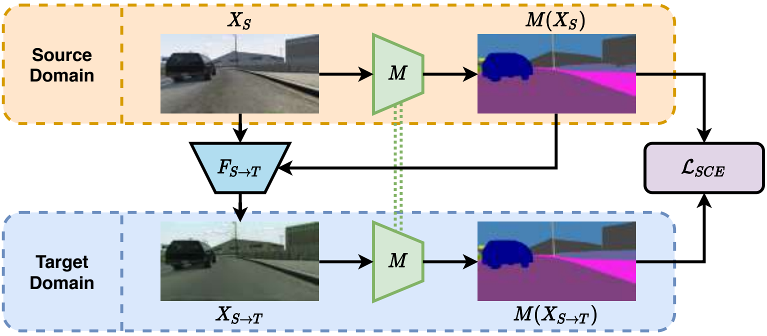


Figure 1: Core idea of our image-to-image translation system. We use the segmentation network M to get the semantic map $M(X_S)$ from the source image X_S. The semantic map acts as guidance for the translation model $F_{S\to T}$, which translates X_S to the target domain. The translated image $X_{S\to T}$ is then fed to M again and we get $M(X_{S\to T})$. Finally we impose the cross-domain semantic consistency by using the Symmetric Cross-Entropy Loss $\mathcal{L}_{SCE}$.

The most recent solutions to this problem adopt a two-steps approach. The first step is to perform image-to-image translation, where a generative model (*e.g*. CycleGAN [51]) or a stylization method [8] is employed to translate the source images to the target domain. The second step involves training the segmentation network on the translated images, where various methods can be employed to align the features extracted in the two domains.

We focus on improving the first step, making the translation model aware of the task that has to be performed on the resulting images. Different loss functions have been introduced to impose that the task network gives the same result on the two domains [5, 16, 24]. Here, instead, we rethink the generator architecture itself and design it to condition the image translation according to the predicted classes. This not only enhances the capabilities of the generator, but also strengthens the connection between translation and segmentation, since the generated features are connected to the corresponding class by the network itself.

Similar to the related work [5, 16, 24, 31, 42] we test our method by adapting both the GTA5 [36] and SYNTHIA [37] synthetic datasets to Cityscapes [6] and show that our results surpass the current state-of-the-art for the commonly used segmentation networks.

2 Related Work

Our work can be split into two cooperating parts: UDA for semantic segmentation and image-to-image translation. Here we separately review the most relevant approaches to these tasks, highlighting our contributions.

Unsupervised domain adaptation We aim at using synthetic data to perform semantic segmentation on real images, where no labels are available. This can be framed as a problem of UDA, where the main idea is to align the source and target distributions at either feature level, pixel level, or both. This has been applied to image classification [3, 9, 10, 29, 30, 43, 44] by minimizing the Maximum Mean Discrepancy [11, 29], measuring the correlation distance [41] or with adversarial learning [44].

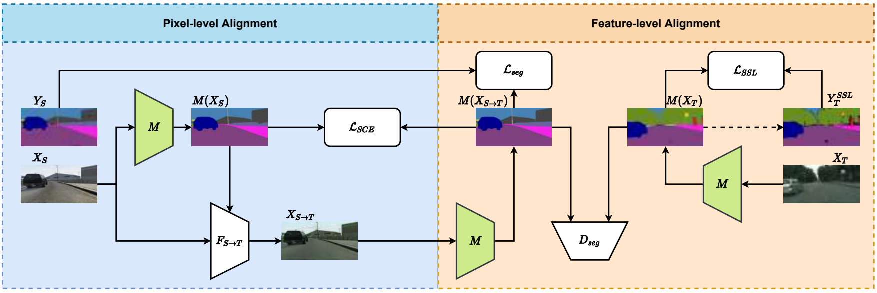


Figure 2: **Overview of our method**. M is the segmentation network, which is shared for all the steps. F is the translation network. D_{seg} is the segmentation discriminator. The losses are detailed in Section 3. We omitted the reconstruction and consistency pipelines, together with the target image discriminator. The pixel-level alignment for $T \rightarrow S$ is symmetric to $S \rightarrow T$. The dashed arrow between $M(X_T)$ and Y_T^{SSL} is used to indicate that the pseudo-label generation is performed offline, before feature-level alignment.

Semantic segmentation is a much more complex task and the first solution for domain adaptation has been proposed in [15] with global distribution alignment at feature level. Curriculum learning was used in [50], where the authors proposed first to learn the global distribution of the image and the local distribution of superpixels, and then train the segmentation network according to these properties. Global feature alignment was also adopted in [42], where different discriminators are employed for features at different levels. Other works introduced class-wise adversarial learning [4] and pseudo-labels [4, 24, 52]. CLAN [31] improves feature level alignment by reweighting the adversarial loss with a discrepancy map based on categories. Feature level alignment, however, is not enough to adapt to different domains, which is why the most recent approaches [5, 8, 16, 24, 49] introduced also pixel-level alignment. CyCADA [16] trains the segmentation network on images translated with CycleGAN [51] and a semantic consistency loss. DCAN [49] adopts a custom image-to-image translation network and performs feature alignment both in the translation and in the segmentation step. CrDoCo [5] uses a cross-domain consistency loss to improve the translation. Similarly BDL [24] links translation and segmentation with a perceptual loss, where the training is iterated to gradually improve both tasks.

Our work builds on top of these ideas, but we rethink the generator architecture to condition the image-to-image translation with the semantic guidance of the segmentation network.

Image-to-image translation In order to translate synthetic images into real looking ones without using paired couples, the most common approach is to use Generative Adversarial Networks [12]. By learning how to trick the discriminator, the generator network becomes able to generate images aligned with the target distribution.

Nevertheless, only with the introduction of the cycle consistency [51] the generated images look realistic. UNIT [26] develops a more complex assumption: by combining VAEs [7, 35] with CoGANs [25], they enforce that two domains share a common latent space which can be used to move from one domain to the other and back. This approach evolves in MUNIT [18], where the shared latent space is formalized as the content and combined with the target style to generate multimodal realistic outputs.

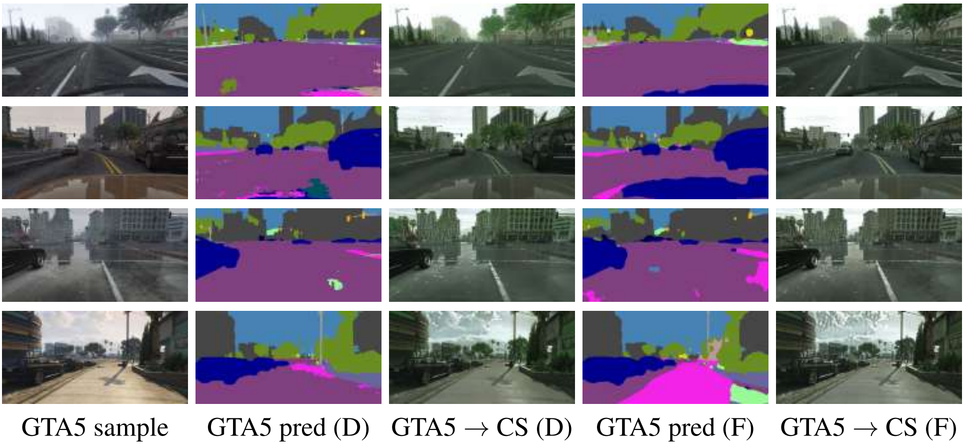


Figure 3: **Translation from GTA5 [36] to Cityscapes [6].** We take a sample from GTA5, get the predicted segmentation using M, and generate $X_{S \to T}$. We present the results obtained with both DeepLabV2 [2] and FCN8s [28] used as semantic guidance.

Normalization layers The key insight for image-to-image translation is in the ability to disentangle style and content. In fact, in order to move from one domain to the other, one has to be able to change the style while preserving the image content.

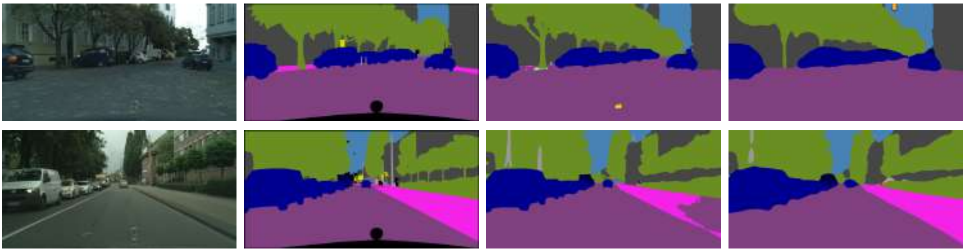
It has been noted [17] that the most effective way to swap styles is by using normalization layers. Batch Normalization [19] has been used in [45], but [47] found that replacing it with Instance Normalization (IN) [46] leads to significant improvements. IN works in the feature space in the same way Contrast Normalization works in the pixel space, which makes it much more effective. A more general approach has been introduced by Adaptive IN (AdaIN), which computes the affine transformation from a style input. UNIT [26] uses IN to swap the source and target style. Instead of performing a global translation with IN, we exploit the task network to translate each region of the image according to its semantic meaning. To this end, we choose to denormalize the generator activations with the SPADE layer [34], giving a result that naturally cooperates with the learning of the semantic segmentation task.

3 Method

Our objective is to train a deep neural network M to perform semantic segmentation on a target (real) dataset T. We assume we only have the target images X_T without the target labels Y_T. In order to do this, we use synthetic data from a source (synthetic) dataset S, where we have both the images X_S and the labels Y_S.

This problem setting is UDA for semantic segmentation, which means that we want to reduce the domain shift caused by the difference in visual appearance of the two domains.

As depicted in Figure 2, we follow the recent work [5, 16, 24] and take advantage of both pixel-level and feature-level alignment to reduce the domain shift. We can see them as two separate subtasks, but we will also show how they actually need to cooperate to improve each other in the following sections.



Test sample | Ground truth | GTA5 → CS w/ D | GTA5 → CS w/ F

Figure 4: **Semantic Segmentation adapted from GTA5 [36] to Cityscapes [6].** We take a sample X_T from the Cityscapes validation set and show the segmentation predictions $M(X_T)$ of both the adapted DeepLabV2 [2] and FCN8s [28] networks.

3.1 Pixel-level alignment

For pixel-level alignment, we make use of an image-to-image translation network $F_{S \to T}$, which learns through adversarial training to visually align X_S to X_T by generating $X_{S \to T} = F_{S \to T}(X_S)$. Some visual examples of the results of this pipeline are depicted in Figure 6. Inspired by [26] [18], we assume that S and T share a common latent space Z and design two coupled GANs to train the desired system.

The training objective for the image translation model is comprised of several loss functions computed as the sum of two components, one per domain. The final objective is then:

$$\mathcal{L} = \lambda_{recon}\mathcal{L}_{recon} + \lambda_{GAN}\mathcal{L}_{GAN} + \lambda_{CC_I}\mathcal{L}_{CC_I} + \lambda_{CC_H}\mathcal{L}_{CC_H} + \lambda_{SCE}\mathcal{L}_{SCE} \tag{1}$$

Image reconstruction We have an encoder for each domain, E_S and E_T, coupled with a corresponding generator for each domain, G_S and G_T, to form two Autoencoders. The encoders extract the latent code $z \sim Z$, which is fed to the generators along with the semantic features predicted by M. Therefore, a translated image is indicated as $x_{A \to B} = G_B(E_A(x_A), M(x_A))$ and the image reconstruction loss is:

$$\mathcal{L}_{recon}^{S} = \mathbb{E}_{x_S \sim X_S}\left[||x_{S \to S} - x_S||\right] \qquad \mathcal{L}_{recon}^{T} = \mathbb{E}_{x_T \sim X_T}\left[||x_{T \to T} - x_T||\right] \tag{2}$$

Adversarial loss By combining E_S with G_T and vice versa, we get the actual translation models $F_{S \to T}$ and $F_{T \to S}$, which are trained in an adversarial fashion to trick the corresponding discriminators D_T and D_S:

$$\begin{aligned}
\mathcal{L}_{GAN}^{S} &= \frac{1}{2}\mathbb{E}_{x_S \sim X_S}\left[(D_S(x_S))^2\right] + \frac{1}{2}\mathbb{E}_{x_T \sim X_T}\left[(D_S(x_{T \to S}) - 1)^2\right] \\
\mathcal{L}_{GAN}^{T} &= \frac{1}{2}\mathbb{E}_{x_T \sim X_T}\left[(D_T(x_T))^2\right] + \frac{1}{2}\mathbb{E}_{x_S \sim X_S}\left[(D_T(x_{S \to T}) - 1)^2\right]
\end{aligned} \tag{3}$$

Cycle consistency By combining $F_{S \to T}$ with $F_{T \to S}$ and viceversa we can now apply the cycle consistency loss to images and latent spaces:

$$\begin{aligned} \mathcal{L}^S_{CC_I} &= \mathbb{E}_{x_S \sim X_S}[||x_{S \rightleftarrows T} - x_S||] & \mathcal{L}^T_{CC_I} &= \mathbb{E}_{x_T \sim X_T}[||x_{T \rightleftarrows S} - x_T||] \\ \mathcal{L}^S_{CC_H} &= \mathbb{E}_{z_S \sim Z}[||z_{S \to T} - z_S||] & \mathcal{L}^T_{CC_H} &= \mathbb{E}_{z_T \sim Z}[||z_{T \to S} - z_T||] \end{aligned} \tag{4}$$

where $z_{S \to T}$ refers to the latent space extracted by E_T from $x_{S \to T}$ and viceversa.

Symmetric cross-entropy Finally, we impose that the segmentation predicted for the translated image has to be consistent with the one predicted for the original one through a symmetric cross-entropy loss, which is made of two contributions. For the $S \to T$ case, the first contribution assumes that $M(x_{S \to T})$ is the ground truth label and tries to align $M(x_S)$ with it. The second contribution assumes that $M(x_S)$ is the ground truth label and tries to align $M(x_{S \to T})$ with it. The $T \to S$ case is symmetrical to the first one.

$$\begin{aligned} \mathcal{L}^S_{SCE} &= -\,\mathbb{E}_{x_S \sim X_S}\left[M(x_{S \to T}) \log M(x_S)\right] - \mathbb{E}_{x_S \sim X_S}\left[M(x_S) \log M(x_{S \to T})\right] \\ \mathcal{L}^T_{SCE} &= -\,\mathbb{E}_{x_T \sim X_T}\left[M(x_{T \to S}) \log M(x_T)\right] - \mathbb{E}_{x_T \sim X_T}\left[M(x_T) \log M(x_{T \to S})\right] \end{aligned} \tag{5}$$

3.2 Semantically adaptive generator

Recent generator architectures [18, 21, 27] make use of AdaIN to remove the source style and inject the target one. However, we observe that the global denormalization performed by AdaIN might be suboptimal for the image translation task. This is why we redesigned our generator to adaptively denormalize each pixel based on its semantics.

We use M to extract a segmentation map $m \in \mathbb{R}^{B \times C \times H \times W}$ from the input image, where C is the number of classes. When feeding it to the generator, we choose to represent this semantic guidance as the unnormalized output of M. In the supplementary material we detail the reasons behind this choice and the other possibilities.

Given an input activation $x \in \mathbb{R}^{B \times C' \times H' \times W'}$, m is resized to $H' \times W'$ and fed to the SPADE layer, which outputs $\gamma, \beta \in \mathbb{R}^{B \times C' \times H' \times W'}$. We then normalize x by using Instance Normalization and use γ and β to denormalize it:

$$y_{b,c,h,w} = \gamma_{b,c,h,w} \frac{x_{b,c,h,w} - \mu_{b,c}}{\sigma_{b,c}} + \beta_{b,c,h,w} \tag{6}$$

3.3 Analysis

Pixel-level alignment has given a great boost to the research in UDA problems, but the gap with the performance achievable with full supervision is still huge. We believe that the image translation methods still need a lot of improvements and this is why we focused on redesigning the generator to include a semantic conditioning. Our claim is that adaptively denormalizing each pixel based on its class allows the translation model to produce results which are better for domain adaptation, since each region gets injected with features that are more consistent with its semantic. This connection strengthens the bridge with feature-level alignment (see Figure 1), which before our work was induced only by consistency losses.

GTA5 → Cityscapes																					
Method	Arch.	road	sidewalk	building	wall	fence	pole	light	sign	veget	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
Cycada [16]	D	86.7	35.6	80.1	19.8	17.5	**38.0**	39.9	**41.5**	82.7	27.9	73.6	**64.9**	19	65.0	12.0	28.6	4.5	31.1	**42.0**	42.7
AdaptSegNet [42]	D	86.5	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	**7.2**	29.5	32.5	41.4
DCAN [49]	D	85.0	30.8	81.3	25.8	21.2	22.2	25.4	26.6	83.4	36.7	76.2	58.9	24.9	80.7	29.5	42.9	2.5	26.9	11.6	41.7
CLAN [31]	D	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
BDL [24]	D	91.0	**44.7**	84.2	34.6	**27.6**	30.2	36.0	36.0	85.0	43.6	83.0	58.6	31.6	83.3	**35.3**	**49.7**	3.3	28.8	35.6	48.5
Ours	D	**91.2**	43.3	**85.2**	**38.6**	25.9	34.7	**41.3**	41.0	**85.5**	**46.0**	**86.5**	61.7	**33.8**	**85.5**	34.4	48.7	0.0	**36.1**	37.8	**50.4**
Curriculum [51]	F	74.9	22.0	71.7	6.0	11.9	8.4	16.3	11.1	75.7	13.3	66.5	38.0	9.3	55.2	18.8	18.9	0.0	16.8	16.6	28.9
CBST [54]	F	66.7	26.8	73.7	14.8	9.5	**28.3**	25.9	10.1	75.5	15.7	51.6	47.2	6.2	71.9	3.7	2.2	5.4	18.9	32.4	30.9
Cycada [16]	F	85.2	37.2	76.5	21.8	15.0	23.8	22.9	21.5	80.5	31.3	60.7	50.5	9.0	76.9	17.1	28.2	4.5	9.8	0.0	35.4
DCAN [49]	F	82.3	26.7	77.4	23.7	20.5	20.4	**30.3**	15.9	80.9	25.4	69.5	52.6	11.1	79.6	24.9	21.2	1.3	17.0	6.7	36.2
LSD [39]	F	88.0	30.5	78.6	25.2	23.5	16.7	23.5	11.6	78.7	27.2	71.9	51.3	19.5	80.4	19.8	18.3	0.9	20.8	18.4	37.1
CLAN [31]	F	88.0	30.6	79.2	23.4	20.5	26.1	23.0	14.8	81.6	34.5	72.0	45.8	7.9	80.5	**26.6**	29.9	0.0	10.7	0.0	36.6
CrDoCo [5]	F	89.1	33.2	80.1	26.9	25.0	18.3	23.4	12.8	77.0	29.1	72.4	**55.1**	20.2	79.9	22.3	19.5	1.0	20.1	18.7	38.1
BDL [24]	F	89.2	40.9	81.2	29.1	19.2	14.2	29.0	19.6	83.7	35.9	80.7	54.7	23.3	**82.7**	25.8	28.0	2.3	25.7	19.9	41.3
Ours	F	**91.1**	**46.4**	**82.9**	**33.2**	**27.9**	20.6	29.0	**28.2**	**84.5**	**40.9**	**82.3**	52.4	**24.4**	81.2	21.8	**44.8**	**31.5**	**26.5**	**33.7**	**46.5**

Table 1: Results of adapting GTA5 [36] to Cityscapes [6]. D stands for DeepLabV2 [2] with ResNet101 [13], while F stands for FCN8s [28] with VGG16 [40] as backbone network.

3.4 Feature-level alignment

For feature-level alignment, we train M on X_T and $X_{S \to T}$ by combining supervision on $X_{S \to T}$, self-supervision on X_T and adversarial learning. The loss, in this case, is given by

$$\mathcal{L} = \lambda_{seg}\mathcal{L}_{seg} + \lambda_{SSL}\mathcal{L}_{SSL} + \lambda_{adv}\mathcal{L}_{adv} \tag{7}$$

We set $\lambda_{seg} = 1$, $\lambda_{SSL} = 1$, $\lambda_{adv} = 10^{-3}$ for DeepLabV2, $\lambda_{adv} = 10^{-4}$ for FCN8s and use the same optimization hyperparameters of [24] to train both networks.

Segmentation loss The main supervision for the segmentation task is given by training the network on $(X_{S \to T}, Y_S)$, where $X_{S \to T}$ are images translated from the synthetic to the real domain. This is formulated as the common cross-entropy loss:

$$\mathcal{L}_{seg} = -\mathbb{E}_{x \sim X_{S \to T}, y \sim Y_S} \sum_{k=1}^{K} \mathbf{1}_{[k=y]} \log(M(x)_k) \tag{8}$$

Self-supervised segmentation Following [24], we also adopt self-supervision to improve the adaptation model. To this end, we compute $M(X_T)$ and use as labels the high confidence predictions, creating Y_T^{SSL}:

$$Y_T^{SSL} = \begin{cases} \arg\max\limits_{1 \leq k \leq K} M(X_T)_k, & \text{if } M(X_T)_k \geq th_{SSL} \\ -1, & \text{otherwise} \end{cases} \tag{9}$$

where K is the number of classes, -1 is the index ignored and th_{SSL} is the confidence threshold, which we use to filter the uncertain predictions. In our experiments we set $th_{SSL} = 0.9$.

This makes us able to compute a cross-entropy loss also on the target dataset:

$$\mathcal{L}_{SSL} = -\mathbb{E}_{x \sim X_T, y \sim Y_T^{SSL}} \sum_{k=1}^{K} \mathbf{1}_{[k=y]} \log(M(x)_k) \tag{10}$$

SYNTHIA → Cityscapes																	
Method	Arch.	road	sidewalk	building	wall	fence	pole	light	sign	veget	sky	person	rider	car	bus	mbike	bike	mIoU
AdaptSegNet [42]	D	79.2	37.2	78.8	-	-	-	9.9	10.5	78.2	80.5	53.5	19.6	67.0	29.5	21.6	31.3	45.9
CLAN [31]	D	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
BDL [24]	D	86.0	46.7	80.3	-	-	-	14.1	11.6	79.2	81.3	54.1	27.9	**73.7**	42.2	25.7	45.3	51.4
Ours	D	**87.7**	**49.7**	**81.6**	-	-	-	**19.3**	**18.5**	**81.1**	**83.7**	**58.7**	**31.8**	73.3	**47.9**	**37.1**	**45.7**	**55.1**
FCNsITW [15]	F	11.5	19.6	30.8	4.4	0.0	20.3	0.1	11.7	42.3	68.7	51.2	3.8	54.0	3.2	0.2	0.6	20.2
Curriculum [50]	F	65.2	26.1	74.9	0.1	0.5	10.7	3.5	3.0	76.1	70.6	47.1	8.2	43.2	20.7	0.7	13.1	29.0
CBST [52]	F	69.6	28.7	69.5	**12.1**	0.1	25.4	11.9	13.6	**82.0**	**81.9**	49.1	14.5	66.0	6.6	3.7	32.4	35.4
DCAN [49]	F	79.9	30.4	70.8	1.6	**0.6**	22.3	6.7	23.0	76.9	73.9	41.9	16.7	61.7	11.5	10.3	38.6	35.4
CLAN [31]	F	80.4	30.7	74.7	-	-	-	1.4	8.0	77.1	79.0	46.5	8.9	73.8	18.2	2.2	9.9	39.3
CrDoCo [5]	F	**84.9**	32.8	**80.1**	4.3	0.4	**29.4**	14.2	21.0	79.2	78.3	50.2	15.9	69.8	23.4	11.0	15.6	38.2
BDL [24]	F	72.0	30.3	74.5	0.1	0.3	24.6	10.2	25.2	80.5	80.0	54.7	**23.2**	72.7	**24.0**	7.5	44.9	39.0
Ours	F	79.1	**34.0**	78.3	0.3	**0.6**	26.7	**15.9**	**29.5**	81.0	81.1	**55.5**	21.9	**77.2**	23.5	**11.8**	**47.5**	**41.5**

Table 2: Results of adapting SYNTHIA [37] to Cityscapes [37]. D stands for DeepLabV2 [2] with ResNet101 [13], while F stands for FCN8s [28] with VGG16 [40] as backbone network.

Adversarial loss Supervision on pixel-level aligned images and self-supervision on target images are not enough to learn a full model. This is why we also make use of adversarial training by feeding the semantic maps to a discriminator D_{seg}, which has to distinguish the maps predicted by M for S and T, giving:

$$\mathcal{L}_{adv} = \mathbb{E}_{x_T \sim X_T}[\log(D_{seg}(M(x_T)))] + \mathbb{E}_{x_{S\to T} \sim X_{S\to T}}[\log(1 - D_{seg}(M(x_{S\to T})))] \tag{11}$$

This loss enforces an output space alignment [42], which means that M has to learn how to predict semantic maps with distributions that are aligned regardless of the input domain.

4 Experiments

We present our experimental results for the synthetic to real adaptation using two dataset settings: GTA5 [36] to Cityscapes [6] and SYNTHIA [37] to Cityscapes. We evaluate the mean intersection-over-union (IoU) on the Cityscapes validation set and show how our method outperforms the current state-of-the-art by adopting the same segmentation models. Finally, we conduct an ablation study to highlight the value of our contributions.

Segmentation network We choose to adapt two segmentation networks: DeepLabV2 [2] with ResNet101 [13] and FCN8s [28] with VGG16 [40]. Both networks are trained on images downsampled to 1024x512 with batch size 1.

We initialize the segmentation networks from [24] to speed up the training process. In order to show the independence from this initialization, we also conduct one experiment where we train DeepLabV2 from scratch for the GTA→Cityscapes task, and we find this to be in line with the results that we get by initializing it with [24].

Translation network For the translation part, we describe the architecture of the encoders, generators and discriminators.

The encoder is made by few downsampling blocks, followed by residual blocks for further processing of the latent code and they all use IN [46]. Symmetrically, the generators take in the latent code and process it with residual blocks, where IN and SPADE are combined to normalize the feature maps. These are followed by upsampling blocks with Layer Normalization [1]. We found LN to better preserve the style in the generated activations.

In each domain we have discriminators for multiple scales [48], each being a Patch Discriminator [20, 23]. The GAN [12] objective we choose is the one proposed in LSGAN [32]. We apply Spectral Normalization [33] to all the models described here.

When training the translation model we resize the input images to 1024x512 and take 512x512 random crops out of them. We use Adam [22] as optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We apply TTUR [14] and set the initial learning rate to be 10^{-4}. The learning rate is scheduled to decay to 0 after 1000000 iterations with a 'poly' scheduling where the power is 0.9. The batch size is 1 for all the experiments. The loss weights are set to $\lambda_{recon} = 10$, $\lambda_{GAN} = 1$, $\lambda_{CC_I} = 10$, $\lambda_{CC_H} = 1$, $\lambda_{SCE} = 10$.

Bidirectional learning Pixel-level and feature-level alignment are not performed in an end-to-end fashion. Besides being highly expensive in terms of memory requirements, we found this approach to be very unstable and it did not lead to good results.

We adopt a policy similar to [24] and iteratively recreate $X_{S\to T}$ when M stops improving on the target dataset. Before each training of the segmentation network, we also generate new pseudo-labels Y_T^{SSL}. We found this procedure to significantly improve the final mIoU compared to a single iteration of pixel-level and feature-level alignment.

4.1 Comparison with State of the Art

GTA5 to Cityscapes For the GTA5 [36] to Cityscapes [6] task, we evaluate on all the 19 classes used in the Cityscapes benchmark since the datasets are fully compatible. Some visual results for this setting are presented in Figure 4. In this case, the upper bounds in terms of mIoU are 65.1 for DeepLabV2 [2] and 60.3 for FCN8s [28], which are the results achievable by training with the target labels. In Table 1 we compare our results with the related work. In terms of mIoU, we get respectively +1.9% and +4.2% over the state-of-the-art with the two networks.

SYNTHIA to Cityscapes SYNTHIA [37] has been adopted in the past by the other works for its overlapping with 16 of the Cityscapes classes. For the SYNTHIA to Cityscapes task we compare our results with the state-of-the-art in Table 2 and present some visual results in the supplementary material. For a fair comparison, the results of the DeepLabV2 architecture are limited to the 13 classes adopted by the other works [24, 31, 42]. The upper bounds in terms of mIoU are 71.7 for DeepLabV2 and 59.5 for FCN8s. In the case of DeepLabV2 we surpass the current state-of-the-art in mIoU by +3.7%. For FCN8s, instead, we get +2.5% on the mIoU.

4.2 Ablation study

In order to weight our contribution, we perform an ablation study of the proposed method (see Table 3). For each experiment, we report 3 values: the mIoU; the gain *wrt* the lower bound, which is a naive training on the source dataset; the remaining gap *wrt* the upper bound, which is the result for training with target labels (called oracle prediction).

SPADE	$\mathcal{L}_{SCE}$	mIoU	Gain	Gap to UB
		49.2	15.6	15.9
✓		49.5	15.9	15.6
	✓	49.5	15.9	15.6
✓	✓	50.4	16.8	14.7

Table 3: **Ablation study**. We report the mIoU, the gain *wrt* the lower bound (*i.e.* training naively on source), the gap *wrt* the upper bound (*i.e.* training on target).

Setting	Network	IS	FID
GTA5 → Cityscapes	DeepLabV2	4.9	27.9
GTA5 → Cityscapes	FCN8s	4.8	40.3
SYNTHIA → Cityscapes	DeepLabV2	5.0	100.8
SYNTHIA → Cityscapes	FCN8s	4.9	113.7

Table 4: **Image quality evaluation**. We report the Inception Score (IS) [38] and the Fréchet Inception Distance (FID) [14] of the images generated in each setting of our experiments.

The experiments are conducted with DeepLabV2 [2] for the GTA5 [36] to Cityscapes [6] task, for which the lower bound is 33.6 and the upper bound is 65.1.

We first show the baseline results that we get by using the generator with no Symmetric Cross-Entropy $\mathcal{L}_{SCE}$ and no semantic guidance. In this setting, the residual blocks of the generator use IN [46] layers and the image-to-image translation is completely unrelated to the semantic segmentation. Secondly, we add the semantic guidance with the SPADE [34] layer. This setting can still benefit from the semantic guidance in the translation, but loses the ability to enforce the cross-domain consistency for the segmentation task. Then we swap back the SPADE layer with IN and enable $\mathcal{L}_{SCE}$. This setting resembles the one used in [5], where the architecture of CycleGAN [51] is replaced by ours. Finally, we show that the best results are achieved by the combination of the two elements, which completely bridges the translation and segmentation tasks and is the final setting of our work.

We can see that when we remove SPADE or $\mathcal{L}_{SCE}$ the mIoU drops, suggesting that they both have an important contribution to get the best result.

4.3 Generated image quality

We also report the quality of the images generated by our image-to-image translation model. In Table 4 we report the Inception Score (IS) [38] of the images $X_{S\to T}$ and the Fréchet Inception Distance (FID) [14] with the Cityscapes training set. Although the IS of the produced images is low in every setting, the FID results indicate that the semantic guidance induced by DeepLabV2 is the one that best visually aligns the synthetic domain to Cityscapes. The images translated from SYNTHIA, however, have a much greater distance from Cityscapes than the ones translated from GTA5, regardless of the network used as semantic guidance. We note that this is possibly due to the bigger initial gap in visual appearance between the two domains, since the FID between the original SYNTHIA and Cityscapes is 156.92, while the FID between the original GTA5 and Cityscapes is only 62.42.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 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.

[3] Qingchao Chen, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty. Reweighted adversarial adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7976–7985, 2018.

[4] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1992–2001, 2017.

[5] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1791–1800, 2019.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[7] P Kingma Diederik, Max Welling, et al. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[8] Aysegul Dundar, Ming-Yu Liu, Ting-Chun Wang, John Zedlewski, and Jan Kautz. Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation. *arXiv preprint arXiv:1807.09384*, 2018.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 1180–1189. JMLR. org, 2015.

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[11] Bo Geng, Dacheng Tao, and Chao Xu. Daml: Domain adaptation metric learning. *IEEE Transactions on Image Processing*, 20(10):2980–2989, 2011.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[15] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[16] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[18] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[24] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019.

[25] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[26] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017.

[27] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[29] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 97–105. JMLR. org, 2015.

[30] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[31] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. 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*, pages 2507–2516, 2019.

[32] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[33] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

[34] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[35] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

[36] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016.

[37] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. 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*, pages 3234–3243, 2016.

[38] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[39] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3752–3761, 2018.

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[41] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer, 2016.

[42] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018.

[43] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

[44] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[45] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, volume 1, page 4, 2016.

[46] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[47] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6924–6932, 2017.

[48] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[49] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–534, 2018.

[50] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2020–2030, 2017.

[51] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[52] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 289–305, 2018.

A Representation of the semantic input

The image synthesis network of SPADE takes as input a *one-hot* encoding of the ground truth semantic segmentation. Here, instead, we use the unnormalized output of M for every translation that we perform. This is a consequence of the cycle consistency constraints.

As explained in the main article, we have to perform both the $S \rightleftarrows T$ and $T \rightleftarrows S$ cycles, which is why we have to train both G_S and G_T by feeding them semantic maps aligned with the input images. In UDA problems, we do not have access to Y_T, which is why we use $M(X_T)$ for the $T \rightleftarrows S$ cycle.

However, we note that the refined output classes predicted by M are far from the ground truth and cannot give an accurate conditioning, especially in the target domain when the segmentation is still in the initial training phases. Because of this, we choose to use as semantic guidance the unnormalized output of M. This representation has the advantage of carrying the confidence of the prediction, which could potentially be used by the SPADE layers to avoid denormalizing a region with the incorrect class (*e.g.* on the borders of objects, where the segmentation tends to fail more easily).

In the $S \rightleftarrows T$ cycle, we could use Y_S as semantic guidance, but this would lead to inconsistent input distributions for the SPADE layers, which is why we adopt $M(X_S)$ as semantic guidance in this case too.

B Detailed architecture

Encoder

Kernel size	Stride	Input channels	Output channels	Output upsampling	Residual	Activation function	Normalization	Spectral normalization
7	1	3	64	-	-	ReLU	IN	✓
4	2	64	128	-	-	ReLU	IN	✓
4	2	128	256	-	-	ReLU	IN	✓
3	1	256	256	-	✓	ReLU	IN	✓
3	1	256	256	-	✓	ReLU	IN	✓
3	1	256	256	-	✓	ReLU	IN	✓
3	1	256	256	-	✓	ReLU	IN	✓

Generator

Kernel size	Stride	Input channels	Output channels	Output upsampling	Residual	Activation function	Normalization	Spectral normalization
3	1	256	256	-	✓	ReLU	IN+SPADE	✓
3	1	256	256	-	✓	ReLU	IN+SPADE	✓
3	1	256	256	-	✓	ReLU	IN+SPADE	✓
3	1	256	256	✓	✓	ReLU	IN+SPADE	✓
5	1	256	128	✓	-	ReLU	LN	✓
5	1	128	64	-	-	ReLU	LN	✓
7	1	64	3	-	-	Tanh	-	✓

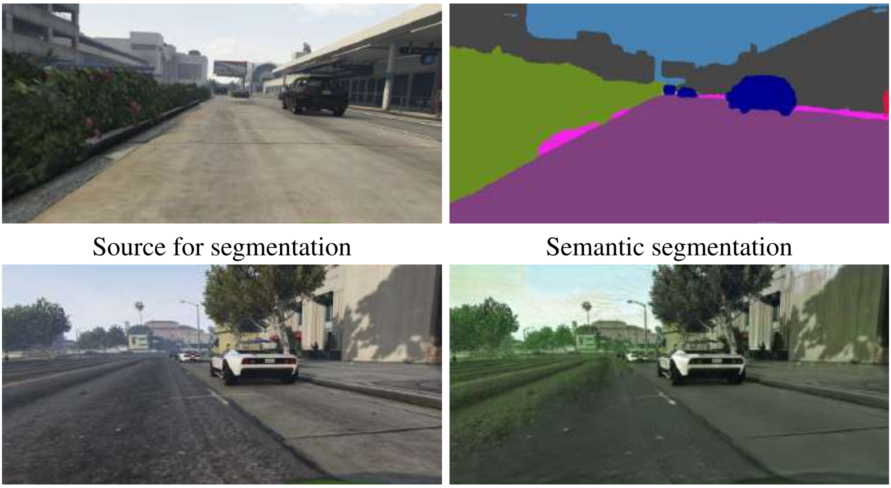
Discriminator (x3)

Kernel size	Stride	Input channels	Output channels	Output upsampling	Residual	Activation function	Normalization	Spectral normalization
4	2	3	64	-	-	$\text{LReLU}_{0.2}$	-	✓
4	2	64	128	-	-	$\text{LReLU}_{0.2}$	-	✓
4	2	128	256	-	-	$\text{LReLU}_{0.2}$	-	✓
4	2	256	512	-	-	$\text{LReLU}_{0.2}$	-	✓
1	1	512	1	-	-	-	-	✓

Table 5: **Detailed architecture of encoders, generators and discriminators in the image-to-image translation step.** The architectures follow the schemes adopted by CycleGAN and UNIT. *Output upsampling* indicates that we use a 2× nearest-neighbor upsampling of the output feature maps. *Residual* indicates that the layer is actually a residual block, not a simple convolutional one. $\text{LReLU}_{0.2}$ indicates the Leaky Rectified Linear Unit with slope $\alpha = 0.2$.

C Fake segmentation

The effect of using the SPADE layers in the image-to-image translation model can be seen better when there is a mismatch between the source image and the semantic guidance. To show this effect, we feed the SPADE layers with a segmentation map extracted from an image that is different from the one being translated. In Figure 5, we can see how the denormalization wrongly creates some features in the region of the image they do not belong to (*i.e.* green on the road).



Source for segmentation

Semantic segmentation

Source for translation

Translated image

Figure 5: **Fake segmentation for image-to-image translation.** We take two different samples X_S^1 (a) and X_S^2 (c) from GTA5. We then use M to get the predicted segmentation $M(X_S^1)$ (b) and use it as semantic guidance for the translation of X_S^1 to get $X_{S\rightarrow T} = F_{S\rightarrow T}(X_S^1, M(X_S^2))$. The result (d) emphasizes the effect of the semantic guidance in our image-to-image translation method.

D Additional results

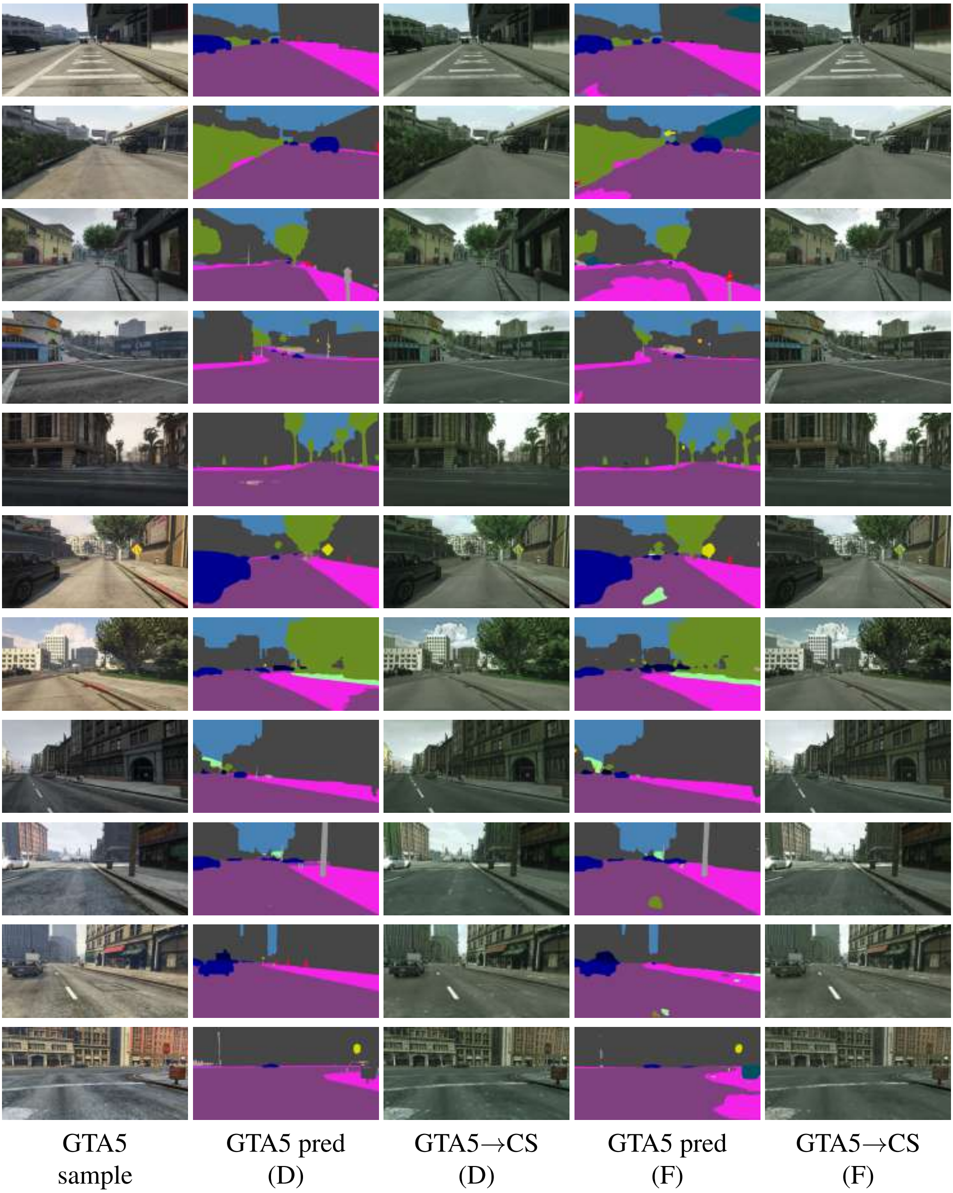


Figure 6: **Additional translations from GTA5 to Cityscapes.** We take a sample X_S from GTA5, get the predicted segmentation using M, and generate $X_{S \to T}$. We present the results obtained with both DeepLabV2 and FCN8s used as semantic guidance.

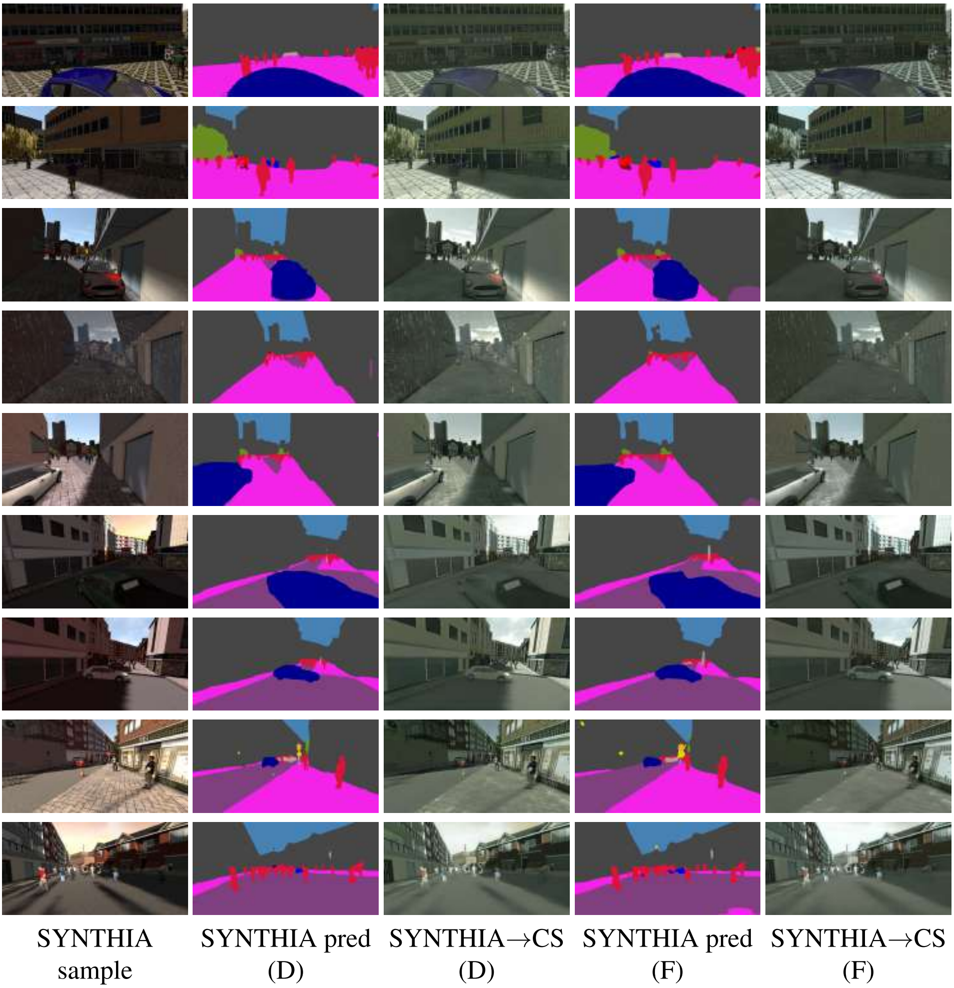


Figure 7: **Translations from SYNTHIA to Cityscapes.** We take a sample X_S from SYNTHIA, get the predicted segmentation using M, and generate $X_{S \to T}$. We present the results obtained with both DeepLabV2 and FCN8s used as semantic guidance.

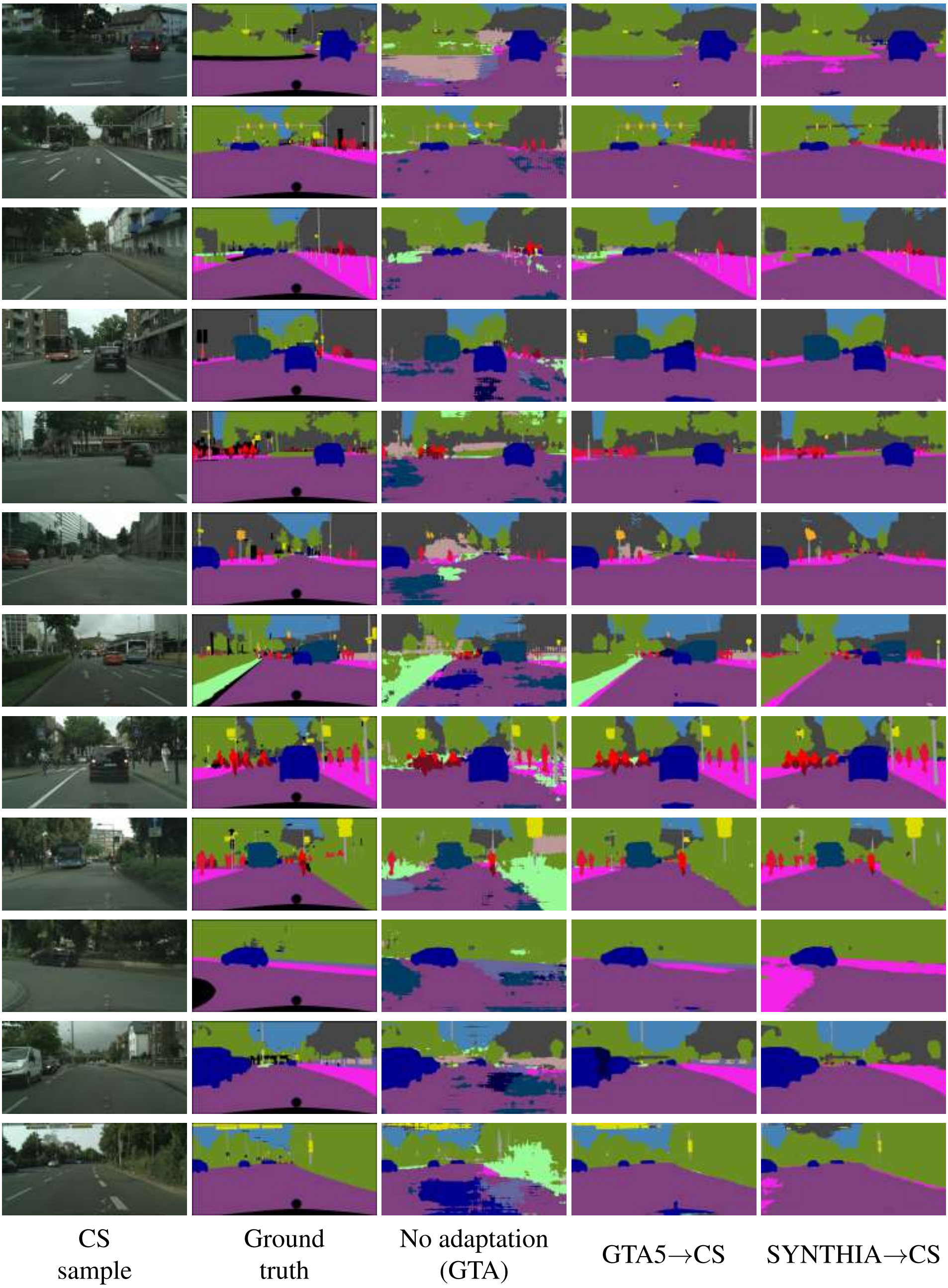


Figure 8: **Additional segmentation results.** We take a sample X_T from the Cityscapes validation set and get the predicted segmentation using M. Here we show the different results obtainable with M being DeepLabV2. First we show the results obtained with M trained with no adaptation on GTA5, then the results obtained by adapting GTA5 and SYNTHIA.