Supplementary Material: Beyond deterministic translation for unsupervised domain adaptation

Eleni Chiou[1]
eleni.chiou.17@ucl.ac.uk
Eleftheria Panagiotaki[1]
e.panagiotaki@ucl.ac.uk
Iasonas Kokkinos[1,2]
i.kokkinos@cs.ucl.ac.uk

[1] Department of Computer Science
University College London
London, UK

[2] Snap Inc.

1 Overview

We provide additional details about the training procedure and additional results including quantitative results and qualitative results.

2 Training procedure

In Sec. 2.1 we provide some additional details regarding the training procedure of the stochastic translation network while in Sec. 2.2 we provide additional details about the pseudolabeling. Finally, in Sec. 2.3 we summarize in more detail the entire training process.

2.1 Stochastic translation for unsupervised domain adaptation

The stochastic translation network is based on MUNIT [4]. We have described it in the main paper but we also describe it here in more detail. For a more extensive presentation we refer the reader to [4]. As it is illustrated in Fig. 1, the stochastic translation network consists of content encoders $\{C_s, C_t\}$, style encoders $\{S_s, S_t\}$, generators $\{G_s, G_t\}$ and domain discriminators $\{D_s, D_t\}$ for the source domain s, and the target domain t respectively.

Given a source domain image $x \in \mathcal{S}$, we start by extracting a domain-invariant content code $c = C_s(x)$ and a domain-specific style code $s_s = S_s(x)$. Then, we perform with-in domain reconstruction (Fig. 1, top) and cross-domain translation (Fig. 1, bottom). We reconstruct the original image x, using the source generator G_s that takes as input at the first layer the content code c and its subsequent layers are modulated by Adaptive Instance Normalization (AdaIN) [3] driven by the style code s_s. This amounts in minimizing the following objective function:

$$L^s_{rec} = \sum_{x \in \mathcal{S}} \|x - G_s(C_s(x), S_s(x))\|.$$

We perform translation from the source domain to the target domain by passing the content code c as input to the target generator G_t, whose subsequent layers are modulated by AdaIn driven by a random style code $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This results in the following stochastic translation function from the source domain to the target domain:

$$\mathbf{T}[x, \mathbf{v}] \doteq G_t(C_s(x), \mathbf{v}).$$

We ensure that the resulting translation matches the distribution of the target domain data by employing the following GAN objective:

$$L^t_{GAN} = \sum_{x \in \mathcal{T}} \log D_t(x) + \sum_{x \in \mathcal{S}} \log(1 - D_t(\mathbf{T}[x, \mathbf{v}])),$$

where $\mathcal{T}$, $\mathcal{S}$ the target and source datasets respectively and D_t the adversarial discriminator for the the target domain t.

We ensure that the content codes c of the source image and the translated image are aligned by minimize the following objective:

$$L^{c_s}_{rec} = \sum_{x \in \mathcal{S}} \|C_t(G_t(C_s(x), \mathbf{v})) - C_s(x)\|_2.$$

Similarly, to align the target style code with the Gaussian prior distribution we use an objective of the following form:

$$L^{s_t}_{rec} = \sum_{x \in \mathcal{S}} \|S_t(G_t(C_s(x), \mathbf{v})) - \mathbf{v}\|_2.$$

As we described in detail in the main paper we ensure that the semantics are preserved during translation using the following objective function:

$$L^s_{sem} = \mathcal{L}_{ce}(F(\mathbf{T}[x, \mathbf{v}]), p),$$

where $\mathcal{L}_{ce}$, the cross-entropy loss, F a segmentation network trained on both source and target data, $F(\mathbf{T}[x, \mathbf{v}])$ the softmax output given the translated image $\mathbf{T}[x, \mathbf{v}]$ and $p = \text{argmax}(F(x))$ the predicted labels for the source image $x \in \mathcal{S}$.

The exact same procedure is followed for translating from the target to the source domain and the corresponding loss terms L^t_{rec}, L^s_{GAN}, $L^{c_t}_{rec}$, $L^{s_s}_{rec}$, L^t_{sem} are defined similarly.

The full objective is given by

$$\begin{aligned} \min_{\substack{C_s, S_s, G_s \\ C_t, S_t, G_t}} \max_{D_s, D_t} \quad & \lambda_x (L^s_{rec} + L^t_{rec}) + \lambda_{GAN} (L^s_{GAN} + L^t_{GAN}) \\ & + \lambda_c (L^{c_s}_{rec} + L^{c_t}_{rec}) + \lambda_s (L^{s_s}_{rec} + L^{s_t}_{rec}) \\ & + \lambda_{sem} (L^s_{sem} + L^t_{sem}), \end{aligned} \tag{11}$$

where λ_{GAN}, λ_x, λ_c, λ_s, λ_{sem} are weights that control the importance of each term. In our experiments, we set the weights as follows: $\lambda_{GAN} = 1$, $\lambda_x = 10$, $\lambda_c = 1$, $\lambda_s = 1$, $\lambda_{sem} = 1$.

2.2 Robust pseudolabel generation

As we mentioned in Sec. 3.4 of the main paper, we rely on three complementary networks to generate robust pseudolabels. We train two target networks $F_{t,\sigma^2=1}$, $F_{t,\sigma^2=10}$; one with

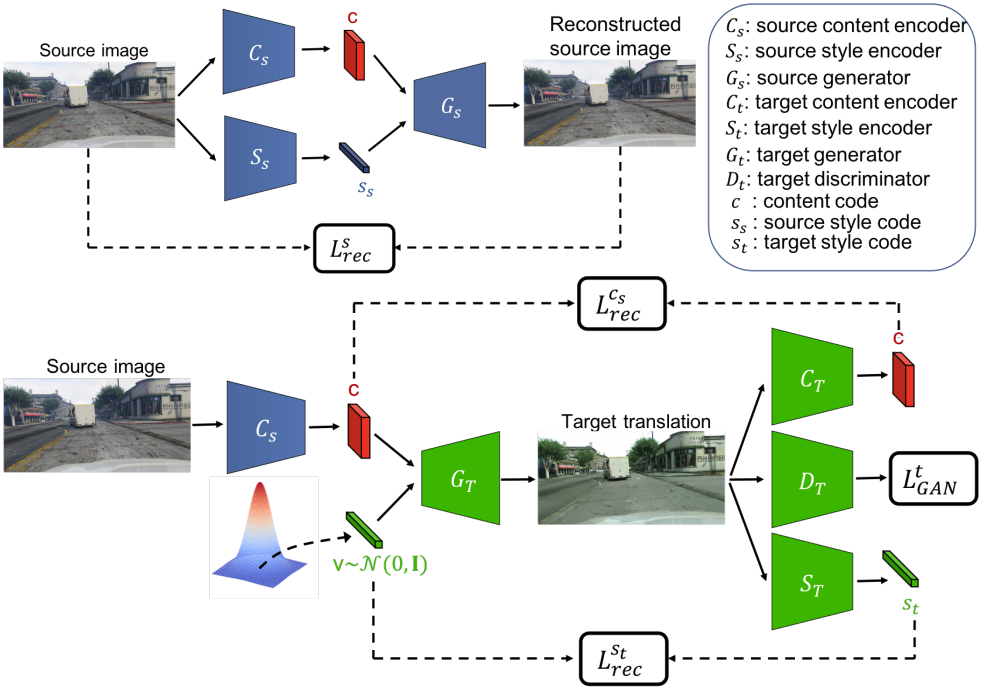

Figure 1: Stochastic translation [2] from the source to the target domain: with-in domain reconstruction (top) and cross-domain translation (bottom) allows us to reconstruct the input and pass the content from the source image to its counterparts respectively. The target cycle is omitted for clarity.

the variance left intact and the other with the variance scaled by 10. We also train a source network, F_s, and exploit multiple samples to obtain a better estimate of the pseudo-labels as we described in Sec. 3.3 of the main paper.

The enhanced probability map used to generate pseudolabels is obtained by the weighted average of the predictions of the three networks:

$$\hat{y} = \frac{1}{3}\hat{y}_s + \frac{1}{3}\hat{y}_{t,\sigma^2=1} + \frac{1}{3}\hat{y}_{t,\sigma^2=10}, \tag{12}$$

where $\hat{y}_s$, $\hat{y}_{t,\sigma^2=1}$, $\hat{y}_{t,\sigma^2=10}$ indicates the pixel-level posterior distribution on labels obtained by F_s, $F_{t,\sigma^2=1}$, $F_{t,\sigma^2=10}$ respectively.

We assign pseudolabels to samples for which the dominant class has a score above a certain threshold θ. Similarly to [15] we use class-wise confidence thresholds to assign pseudolabels. In particular for each class c, the threshold θ_c equals the probability ranked at $r * N_c$, where N_c is the number of pixels predicted to belong in class c and r is the proportion of pseudolabels we want to retain. In cases where $\theta_c > 0.9$, we set $\theta_c = 0.9$. In Sec. 3.1 we provide results for the selection of r based on the mean intersection-over-union (mIoU) of the validation set.

2.3 Training process

Algorithm 1 summarizes the training process. Initially we train a segmentation network, F, operating on both domains and use this to impose a semantic consistency constrain to the stochastic translation network trained using Eq. 11. Using translated images obtained by the stochastic translation network, we train two target-domain networks and one source-domain network. We generate robust pseudolabels for the target-domain data by combining the predictions of the three models. The pseudolabels are used in the next round of training as supervision for the networks when they are driven by target-domain data. We perform two rounds (R) of pseudo-labeling and training.

Algorithm 1 Training process

Input: $\mathcal{S}, \mathcal{T}$
Output: $F_{t,\sigma^2=1}^{(R=2)}, F_{t,\sigma^2=10}^{(R=2)}, F_s^{(R=2)}$
train F with Eq. 2 (main paper)
train the stochastic translation network with Eq. 11
train $F_{t,\sigma^2=1}^{(R=0)}$ with Eq. 4 (main paper), where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
train $F_{t,\sigma^2=10}^{(R=0)}$ with Eq. 4 (main paper), where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, 10\mathbf{I})$
train $F_s^{(R=0)}$ with Eq. 9 (main paper), where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i \leftarrow 1$ to 2 **do**
  generate $\hat{y}$ with Eq. 12
  train $F_{t,\sigma^2=1}^{(R=i)}$ with Eq. 8 (main paper), where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  train $F_{t,\sigma^2=10}^{(R=i)}$ with Eq. 8 (main paper), where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, 10\mathbf{I})$
  train $F_{\mathcal{S}}^{(R=i)}$ with source-domain counterpart of Eq. 8 (main paper), where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
end for

3 Additional results

We provide results (mIoU and per-class IoU) obtained using both DeepLab-V2 with ResNet-101 and FCN-8s with VGG-16 and compare with state-of-the-art methods on both GTA-to-Cityscapes and SYNTHIA-to-Cityscapes benchmarks. mIoU results for DeepLab-V2 with ResNet-101 for both GTA-to-Cityscapes and SYNTHIA-to-Cityscapes benchmarks have already been presented in the main paper.

We also report results from additional ablation studies and class-wise IoU for ablation studies already presented in the main paper. We report results on GTA-to-Cityscapes using DeepLab-V2 with ResNet-101.

3.1 Quantitative results

Comparison with state-of-the-art methods on GTA-to-Cityscapes and SYNTHIA-to-Cityscapes benchmarks.
In Table 1 we provide results (per-class IoU and mIoU) and comparison with recent state-of-the-art methods for the GTA-to-Cityscapes benchmark. (mIoU results are also reported in Table 4 of the main paper). Our results show that our method achieves state-of-the-art performance and outperforms previous methods.

Method	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	bicycle	mIoU
VGG backbone																				
AdaptSegNet[ ]	87.3	29.8	78.6	21.1	18.2	22.5	21.5	11.0	79.7	29.6	71.3	46.8	6.5	80.1	23.0	26.9	0.0	10.6	0.3	35.0
AdvEnt[ ]	86.9	28.7	78.7	28.5	25.2	17.1	20.3	10.9	80.0	26.4	70.2	47.1	8.4	81.5	26.0	17.2	18.9	11.7	1.6	36.1
BDL [ ]	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
LTIR [ ]	**92.5**	**54.5**	83.9	34.5	25.5	31.0	30.4	18.0	84.1	39.6	**83.9**	53.6	19.3	81.7	21.1	13.6	17.7	12.3	6.5	42.3
FDA-MBT [ ]	86.1	35.1	80.6	30.8	20.4	27.5	30.0	26.0	82.1	30.3	73.6	52.5	21.7	81.7	24.0	30.5	**29.9**	14.6	24.0	42.2
PCEDA [ ]	90.7	49.8	81.9	23.4	18.5	**37.3**	35.5	34.3	82.9	36.5	75.8	**61.8**	12.4	83.2	19.2	26.1	4.0	14.3	21.8	42.6
DPL-Dual (Ensemble) [ ]	89.2	44.0	83.5	**35.0**	24.7	27.8	**38.3**	25.3	84.2	39.5	81.6	54.7	**25.8**	83.3	29.3	49.0	5.2	30.2	32.6	46.5
Ours	91.1	43.2	84.1	34.6	25.5	25.8	33.7	31.3	84.7	**44.9**	83.1	55.3	23.5	81.6	23.1	34.3	6.3	**32.7**	34.8	46.0
Ours (Ensemble)	91.0	40.7	**84.7**	33.8	**27.1**	30.9	33.1	**35.1**	**85.3**	44.7	82.9	56.8	23.4	**86.2**	**36.5**	**50.3**	2.8	27.8	**36.6**	**47.9**
ResNet101 backbone																				
AdvEnt[ ]	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
BDL [ ]	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
LTIR [ ]	92.9	55.0	85.3	34.2	31.1	34.9	40.7	34.0	85.2	40.1	87.1	61.0	31.1	82.5	32.3	42.9	0.3	36.4	46.1	50.2
FDA-MBT [ ]	92.5	53.3	82.4	26.5	27.6	36.4	40.6	38.9	82.3	39.8	78.0	62.6	34.4	84.9	34.1	53.1	16.9	27.7	46.4	50.5
PCEDA [ ]	91.0	49.2	85.6	37.2	29.7	33.7	38.1	39.2	85.4	35.4	85.1	61.1	32.8	84.1	**45.6**	46.9	0.0	34.2	44.5	50.5
TPLD [ ]	**94.2**	**60.5**	82.8	36.6	16.6	39.3	29.0	25.5	85.6	44.9	84.4	60.6	27.4	84.1	37.0	47.0	**31.2**	36.1	**50.3**	51.2
Wang et al. [ ]	90.5	38.7	**86.5**	41.1	32.9	**40.5**	48.2	42.1	86.5	36.8	84.2	**64.5**	**38.1**	87.2	34.8	50.4	0.2	41.8	54.6	52.6
PixMatch [ ]	91.6	51.2	84.7	37.3	29.1	24.6	31.3	37.2	86.5	44.3	85.3	62.8	22.6	87.6	38.9	52.3	0.65	37.2	50.0	50.3
DPL-Dual (Ensemble) [ ]	92.8	54.4	86.2	41.6	32.7	36.4	**49.0**	34.0	85.8	41.3	**86.0**	63.2	34.2	87.2	39.3	44.5	18.7	**42.6**	43.1	53.3
SUDA [ ]	91.1	52.3	82.9	30.1	25.7	38.0	44.9	38.2	83.9	39.1	79.2	58.4	26.4	84.5	37.7	45.6	10.1	23.1	36.0	48.8
CaCo [ ]	91.9	54.3	82.7	31.7	25.0	38.1	46.7	39.2	82.6	39.7	76.2	63.5	23.6	85.1	38.6	47.8	10.3	23.4	35.1	49.2
Ours	93.3	56.5	85.9	41.0	33.1	34.8	43.8	**43.8**	86.6	46.5	82.5	61.1	30.4	87.0	39.7	50.7	8.8	34.9	46.8	53.0
Ours (Ensemble)	93.4	55.8	86.4	**44.4**	**36.1**	34.6	45.0	39.8	**86.9**	**48.0**	84.4	61.7	30.9	**87.7**	44.9	**55.9**	11.1	38.4	45.4	**54.3**

Table 1: Quantitative comparison on GTA5→Cityscapes. We present per-class IoU and mean IoU (mIoU) obtained using VGG and ResNet101 backbones.

In Table 2 we provide results (per-class IoU and mIoU) and comparison with recent state-of-the-art methods for the SYNTHIA-to-Cityscapes benchmark. (mIoU results are also reported in Table 5 of the main paper). Our results show that our method achieves state-of-the-art performance and outperforms previous methods.

Method	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	sky	person	rider	car	bus	motocycle	bicycle	mIoU	mIoU*
VGG backbone																		
AdvEnt[ ]	67.9	29.4	71.9	6.3	0.3	19.9	0.6	2.6	74.9	74.9	35.4	9.6	67.8	21.4	4.1	15.5	31.4	36.6
BDL [ ]	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	46.1
FDA-MBT [ ]	84.2	35.1	78.0	**6.1**	0.4	27.0	8.5	22.1	77.2	79.6	55.5	19.9	74.8	24.9	14.3	40.7	40.5	47.3
PCEDA [ ]	79.7	35.2	78.7	1.4	0.6	23.1	10.0	28.9	79.6	81.2	51.2	**25.1**	72.2	24.1	16.7	**50.4**	41.1	48.7
DPL-Dual (Ensemble) [ ]	83.5	38.2	**80.4**	1.3	**1.1**	**29.1**	**20.2**	32.7	81.8	83.6	55.9	20.3	79.4	26.6	7.4	46.2	43.0	50.5
Ours	83.3	40.9	80.3	1.4	0.6	24.8	16.9	31.1	**82.4**	**84.1**	**57.4**	20.1	83.2	30.3	16.0	44.5	43.6	51.5
Ours (Ensemble)	**88.7**	**41.6**	80.3	1.0	0.7	23.6	14.3	**33.1**	81.9	81.1	57.2	21.1	**84.1**	**33.4**	**19.1**	44.3	**44.1**	**52.3**
ResNet101 backbone																		
AdvEnt[ ]	85.6	42.2	79.7	-	-	-	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	-	48.0
LTIR [ ]	92.6	53.2	79.2	-	-	-	1.6	7.5	78.6	84.4	52.6	20.0	82.1	34.8	14.6	39.4	-	49.3
BDL [ ]	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
FDA-MBT [ ]	79.3	35.0	73.2	-	-	-	19.9	24.0	61.7	82.6	61.4	**31.1**	83.9	40.8	**38.4**	51.1	-	52.5
PCEDA [ ]	85.9	44.6	80.8	-	-	-	24.8	23.1	79.5	83.1	57.2	29.3	73.5	34.8	32.4	48.2	-	53.6
TPLD [ ]	80.9	44.3	82.2	19.9	0.3	**40.6**	20.5	30.1	77.2	80.9	60.6	25.5	84.8	41.1	24.7	43.7	47.3	53.5
Wang et al. [ ]	79.4	34.6	**83.5**	**19.3**	**2.8**	35.3	**32.1**	**26.9**	78.8	79.6	**66.6**	30.3	86.1	36.6	19.5	**56.9**	48.0	54.6
PixMatch [ ]	**92.5**	**54.6**	79.8	4.7	0.08	24.1	22.8	17.8	79.4	76.5	60.8	24.7	85.7	33.5	26.4	54.4	46.1	54.5
DPL-Dual (Ensemble) [ ]	87.5	45.7	82.8	13.3	0.6	33.2	22.0	20.1	83.1	86.0	56.6	21.9	83.1	40.3	29.8	45.7	47.0	54.2
SUDA [ ]	83.4	36.0	71.3	8.7	0.1	26.0	18.2	26.7	72.4	80.2	58.4	30.8	80.6	38.7	36.1	46.1	44.6	52.2
CaCo [ ]	87.4	48.9	79.6	8.8	0.2	30.1	17.4	28.3	79.9	81.2	56.3	24.2	78.6	39.2	28.1	48.3	46.0	53.6
Ours	85.8	41.7	82.4	7.6	1.9	33.2	26.5	18.4	83.3	86.5	62.0	29.7	83.9	52.1	34.6	51.4	48.8	56.8
Ours (Ensemble)	87.2	44.1	82.1	6.5	1.4	33.1	24.7	17.9	**83.4**	**86.6**	62.4	30.4	**86.1**	**58.5**	36.8	52.8	**49.6**	**57.9**

Table 2: Quantitative comparison on SYNTHIA→Cityscapes. We present per-class IoU and mean IoU (mIoU) obtained using VGG and ResNet101 backbones. mIoU and mIoU* are the mean IoU computed on the 16 classes and the 13 subclasses respectively.

Training objective of the source domain network.

As we mentioned in the main paper, for the source domain network we observed experimentally that we obtained better results by adding an entropy-based adversarial loss $\mathcal{L}_{adv}$, to the output of source-domain network F_s when it is driven by translated target images. In Table 3 we report results obtained with and without the entropy-based adversarial loss (Eq. 9 in

the main paper). Adding the entropy-based regularization improves performance for most classes.

Loss	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	bicycle	mIoU
$\mathcal{L}_{CE}$	90.2	38.0	81.2	**29.1**	16.2	**24.4**	23.7	15.5	**84.0**	**38.8**	78.5	56.9	24.0	85.0	**36.4**	**47.0**	0.3	**31.8**	26.8	43.6
$\mathcal{L}_{CE}+\mathcal{L}_{adv}$	**90.5**	**39.4**	**82.0**	29.0	**21.4**	23.6	**28.6**	**17.8**	83.9	38.2	**79.8**	**56.9**	**26.0**	**85.1**	32.2	44.1	**3.8**	31.5	**30.1**	**44.4**

Table 3: Better performance is achieved by adding an entropy-based regularization $\mathcal{L}_{adv}$ to the output of source-domain network F_s when it is driven by translated target images.

Selection of r for pseudolabel generation.
As we mentioned in Sec. 2.2, we select the proportion of r based on the mIoU of the validation set. In Table 4 and Table 5 we provide the per-class IoU and mean IoU (mIoU) obtained on the validation set for different values of r in the first (R=0) and second (R=1) round of pseudo-labeling respectively. In both rounds the best performance is achieved for $r = 0.6$. In the second round of pseudolabeling the networks provide more confident predictions since the performance remains the same for almost all classes when $r \leq 0.5$.

r	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	bicycle	mIoU
1.0	92.1	47.8	84.3	36.5	27.9	31.5	36.6	24.5	85.4	41.2	81.6	61.4	30.1	86.3	37.6	47.3	1.3	28.7	32.7	48.2
0.8	97.4	69.8	93.4.	52.0	42.5	47.1	55.2	37.7	94.2	53.5	91.1	78.2	40.2	94.4	47.4	55.7	2.2	40.9	55.4	60.4
0.7	98.2	**72.1**	95.9.	63.4.	51.5	**55.2**	**64.0**	**42.5**	96.5	65.0	93.7	86.9	51.7	96.4	57.0	61.2	**2.7**	53.1	65.8	67.0
0.6	98.5	70.2	96.3	73.4	**57.5**	54.1	58.7	33.3	97.2	77.4	**93.8**	90.8	60.3	97.5	65.5	69.6	2.5	65.8	**71.9**	**70.2**
0.5	**98.6**	59.4	96.5	**79.0**	55.2	44.6	54.3	20.7	97.6	81.4	93.8	92.2	61.0	**97.9**	**70.6**	**74.1**	1.3	**73.3**	70.3	69.6
0.4	98.6	50.1	**96.6**	76.5	42.2	44.6	54.6	20.6	**97.7**	**82.3**	93.8	**92.3**	**61.1**	97.9	70.6	74.1	0.5	73.4	70.5	68.3
0.3	98.6	50.1	96.6	76.6	38.3	44.7	54.6	20.6	97.7	82.3	93.8	92.3	61.1	97.9	70.6	74.1	0.2	73.4	70.5	68.1

Table 4: Per-class IoU and mean (mIoU) obtained using different values of r for class-wise confidence threshold selection in the first round (R=0) of pseudolabeling. We observe that $r = 0.6$ gives the best results.

r	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	bicycle	mIoU
1.0	93.0	53.3	85.8	41.2	33.1	33.4	39.1	29.7	86.4	45.4	84.5	60.0	29.3	86.9	45.8	57.7	2.7	34.6	45.8	52.0
0.8	96.8	**67.4**	93.6	57.3	45.2	46.7	**54.2**	**41.9**	94.2	59.2	92.4	74.8	37.1	94.4	57.1	71.3	4.6	49.2	60.8	63.1
0.7	97.2	67.3	94.2	64.5	49.4	**49.0**	51.9	37.1	95.3	67.6	**92.5**	80.8	46.3	95.9	68.6	77.9	**5.4**	59.7	70.5	66.9
0.6	**97.3**	65.4	**94.5**	67.8	**51.3**	44.0	48.5	36.0	**95.7**	71.9	92.5	84.4	52.5	96.9	85.3	81.2	4.9	67.9	74.7	69.1
0.5	97.3	65.4	94.5	**68.1**	49.3	42.8	48.6	36.3	95.7	**72.2**	92.5	**84.7**	**52.7**	**97.0**	**87.5**	**81.6**	3.0	**68.6**	**75.1**	**69.1**
0.4	97.3	65.4	94.5	68.1	49.3	42.8	48.6	36.3	95.7	72.2	92.5	84.7	52.7	97.0	87.5	81.6	1.3	68.6	75.1	69.0
0.3	97.3	65.4	94.5	68.1	49.3	42.8	48.6	36.3	95.7	72.2	92.5	84.7	52.7	97.0	87.5	81.6	0.4	68.6	75.1	69.0

Table 5: Per-class IoU and mean (mIoU) obtained using different values of r for class-wise confidence threshold selection in the second round (R=1) of pseudolabeling. We observe that $r = 0.6$ gives the best results.

Class-wise IoU for ablation studies reported in the main paper.
In Table 6 we report the per-class IoU obtained from deterministic and stochastic translation (mIoU results are reported in Table 1 of the main paper). In particular the comparison builds on directly on the ADVENT baseline[10]; the first two rows compare the originally published and our reproduced numbers respectively. The third row shows the substantial improvement attained by training the system of ADVENT using translated images. The forth

row reports our stochastic translation-based result. We observe a substantial improvement, that can be attributed solely to the stochasticity of the translation. The last row shows that imposing a semantic consistency constraint as described in Eq. 5 (main paper) further improves the performance. In Table 7 we report the per-class IoU obtained from multiple rounds R of pseudo-labeling and training (mIoU results are provided in Table 3 of the main paper). Multiple rounds of pseudolabeling and training yield improved performance.

Model	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	bicycle	mIoU
ADVENT	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
ADVENT *	87.2	38.5	78.2	25.9	24.6	30.4	36.3	21.7	84.0	28.7	76.7	60.1	28.8	80.0	28.0	45.2	0.7	19.7	19.9	42.9
ADVENT *+ CycleGAN*	**91.9**	**51.5**	83.1	30.8	23.6	**32.0**	32.1	24.3	83.8	**38.5**	**82.3**	58.7	28.5	84.1	33.3	35.9	0.6	21.7	20.0	45.1
Ours	90.2	37.6	**84.1**	**33.0**	**25.1**	30.1	36.8	**28.4**	83.8	36.1	82.2	58.1	**29.6**	84.6	**34.4**	45.4	1.0	**26.2**	**30.8**	46.2
Ours w/ L_{sem}	92.1	49.9	83.5	29.1	24.7	30.3	**38.3**	27.2	**84.8**	34.4	81.1	**60.4**	28.1	**85.2**	33.0	**45.7**	**2.5**	23.8	30.4	**46.6**

Table 6: GTA to Cityscapes UDA using stochastic translation: We train ADVENT using synthetic images obtained from deterministic translation (CycleGAN) and stochastic translation (Ours). We observe a clear improvement thanks to pixel-space alignment based on stochastic translation. * denotes our retrained models.

Model	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	bicycle	mIoU
source (R=0)	90.5	39.4	82.0	29.0	21.4	23.6	28.6	17.8	83.9	38.2	79.8	56.9	26.0	85.1	32.2	44.1	3.8	31.5	30.1	44.4
target, $\sigma^2 = 1$ (R=0)	92.1	49.9	83.5	29.1	24.7	30.3	38.3	27.2	84.8	34.4	81.1	60.4	28.1	85.2	33.0	45.7	2.5	23.8	30.4	46.6
target, $\sigma^2 = 10$ (R=0)	90.9	43.0	83.4	30.6	29.3	30.6	34.1	27.1	84.4	36.2	79.9	60.6	29.5	84.5	32.5	40.3	3.1	29.2	26.4	46.1
Ens (R=0)	92.1	47.8	84.3	36.5	27.9	31.5	36.6	24.5	85.4	41.2	81.6	61.4	30.1	86.3	37.6	47.3	1.3	28.7	32.7	48.2
source (R=1)	92.1	48.4	84.3	36.4	29.5	30.5	35.9	26.5	85.4	42.9	82.1	59.8	29.6	85.5	38.2	52.9	3.4	32.7	37.3	49.1
target, $\sigma^2 = 1$ (R=1)	92.1	47.5	85.1	38.3	29.4	32.9	35.4	32.1	85.9	46.8	81.7	60.5	30.4	86.6	35.7	51.1	4.4	34.9	41.0	50.1
target, $\sigma^2 = 10$ (R=1)	92.9	55.2	85.1	38.1	30.6	32.8	39.8	34.8	85.9	42.2	84.0	59.0	26.1	85.4	47.9	46.3	10.1	28.4	42.8	50.9
Ens (R=1)	93.0	53.3	85.8	41.2	33.1	33.4	39.1	29.7	86.4	45.4	84.5	60.0	29.3	86.9	45.8	57.7	2.7	34.6	45.8	52.0
source (R=2)	92.3	48.2	85.1	40.7	34.3	29.8	38.5	28.2	86.5	46.7	83.3	60.9	30.2	86.9	41.3	53.1	10.4	38.4	40.5	51.3
target, $\sigma^2 = 1$ (R=2)	93.3	56.5	85.9	41.0	33.1	34.8	43.8	43.8	86.6	46.5	82.5	61.1	30.4	87.0	39.7	50.7	8.8	34.9	46.8	53.0
target, $\sigma^2 = 10$ (R=2)	93.4	56.3	85.6	40.6	33.5	35.9	43.5	41.1	85.7	43.8	84.1	60.6	29.2	87.2	44.2	53.7	13.7	33.8	39.2	52.8
Ens (R=2)	93.4	55.8	86.4	44.4	36.1	34.6	45.0	39.8	86.9	48.0	84.4	61.7	30.9	87.7	44.9	55.9	11.1	38.4	45.4	54.3

Table 7: Ablation study on GTA→Cityscapes. Averaging the predictions (Ens) of a source network F_s, and two target networks F_t trained with different degrees of stochasticity (σ^2) in the translation allows to obtain robust pseudo-labels, while using multiple rounds R of pseudo-labeling and training improves the overall performance.

3.2 Qualitative results

Diverse translation obtained using stochastic translation.
Fig. 2 shows diverse translations of images from the SYNTHIA source dataset to the Cityscapes target dataset. Fig. 3 and Fig. 4 show diverse translations of images from the Cityscapes target dataset to the SYNTHIA and GTA source datasets respectively. We observe that stochastic translation generates diverse samples that capture more faithfully the data distribution of the source domain and preserve the content of the original image allowing us to obtain more robust pseudolabels for the target data.
Stochastic versus deterministic translation.
Fig. 5 shows stochastic and deterministic translation of images from the GTA source dataset to the Cityscapes target dataset while Fig. 6 shows stochastic and deterministic translation of images from the SYNTHIA source dataset to the Cityscapes target dataset. Fig. 7 shows stochastic and deterministic translation of images from the Cityscapes target dataset to the

Figure 2: Diverse translations of images from the SYNTHIA source dataset to the Cityscapes target dataset: we observe that even though the content and pixel semantics stay intact, we generate diverse variants of the same scene, effectively capturing more faithfully the data distribution in the target domain.

GTA source dataset while Fig. 8 shows stochastic and deterministic translation of images from the Cityscapes target dataset to the SYNTHIA source dataset. We observe that stochastic translation generates sharp samples of noticeable diversity compared to the deterministic translation that generates a single output.

Multiple rounds of pseudolabeling.

Fig. 9 shows the pseudo-labels obtained from the first (R=0) and second (R=1) round of pseudolabeling. We observe that the pseudolabels we obtained in the second round are more accurate allowing us to train more accurate models in the last round of training.

Robust pseudolabeling through ensembling.

Fig. 10 shows the pseudo-labels obtained by averaging the predictions of two target networks $F_{t,\sigma^2=1}$, $F_{t,\sigma^2=10}$ and a one source network F_s. Averaging the predictions allows us to generate more accurate pseudolabels.

Ensembling for improved segmentation performance.

Fig. 11 shows the predictions obtained by averaging the predictions of two target networks $F_{t,\sigma^2=1}$, $F_{t,\sigma^2=10}$ and a one source network F_s. Averaging the predictions allows us to further improve performance by better distinguishing similar structures (e.g., road, sidewalk) and identifying small objects.

Qualitative comparison of the segmentation results.

Fig. 12 shows results segmentation results obtained by our method and DPL[11]. Our method generates better predictions that are closer to the ground-truth.

Figure 3: Diverse translations of images from the Cityscapes target dataset to the SYNTHIA source dataset: we observe that even though the content and pixel semantics stay intact, we generate diverse variants of the same scene, effectively capturing more faithfully the data distribution of the source domain. This allows us to generate more robust pseudolabels.

Figure 4: Diverse translations of images from the Cityscapes target dataset to the GTA source dataset: we observe that even though the content and pixel semantics stay intact, we generate diverse variants of the same scene, effectively capturing more faithfully the data distribution of the source domain. This allows us to generate more robust pseudolabels.

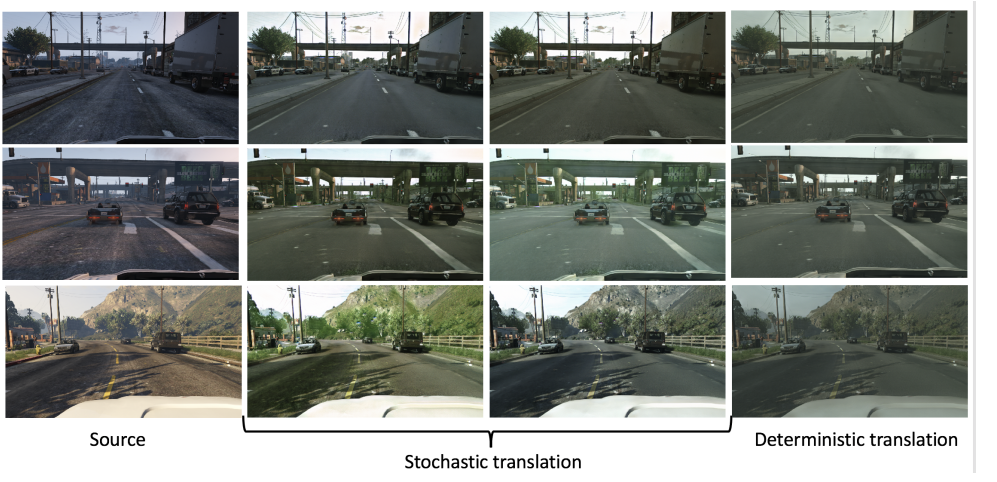

Figure 5: Stochastic and deterministic translation of images from the GTA source dataset to the Cityscapes target dataset.

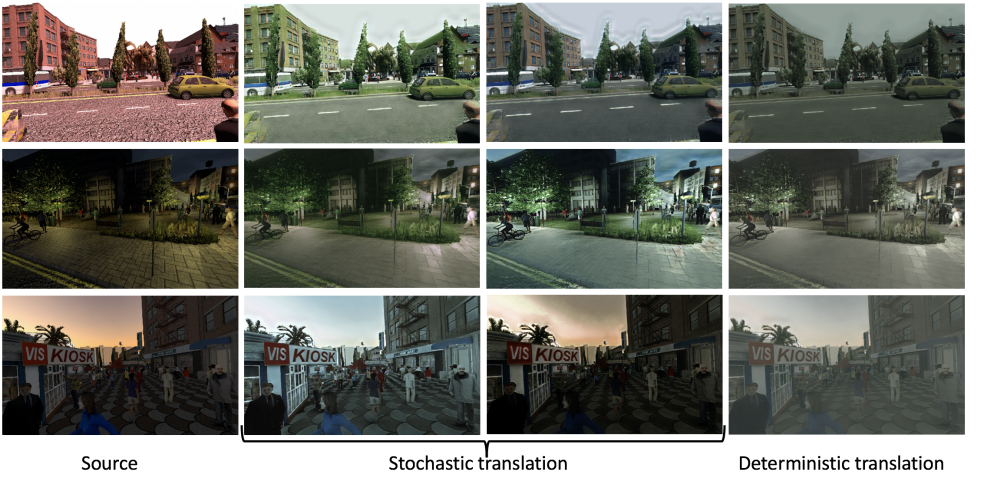

Figure 6: Stochastic and deterministic translation of images from the SYNTHIA source dataset to the Cityscapes target dataset.

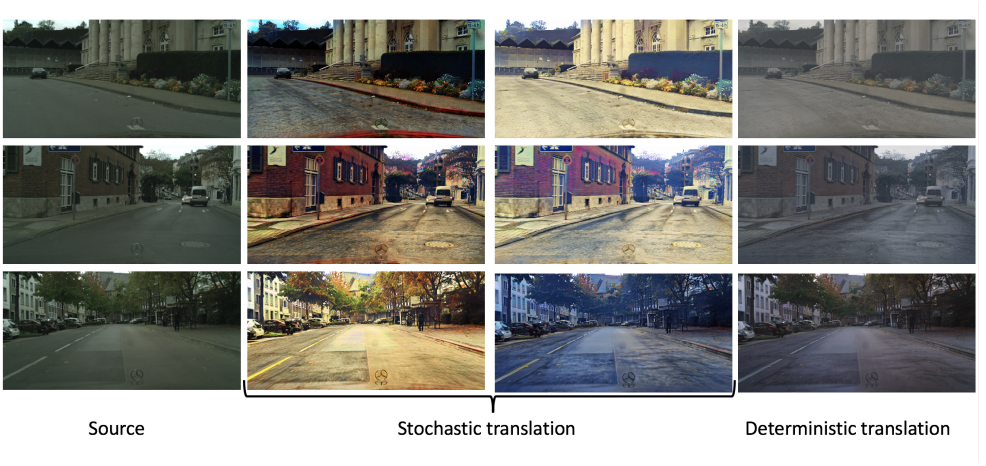

Figure 7: Stochastic and deterministic translation of images from the Cityscapes target dataset to the GTA source dataset.

Figure 8: Stochastic and deterministic translation of images from the Cityscapes target dataset to the SYNTHIA source dataset.

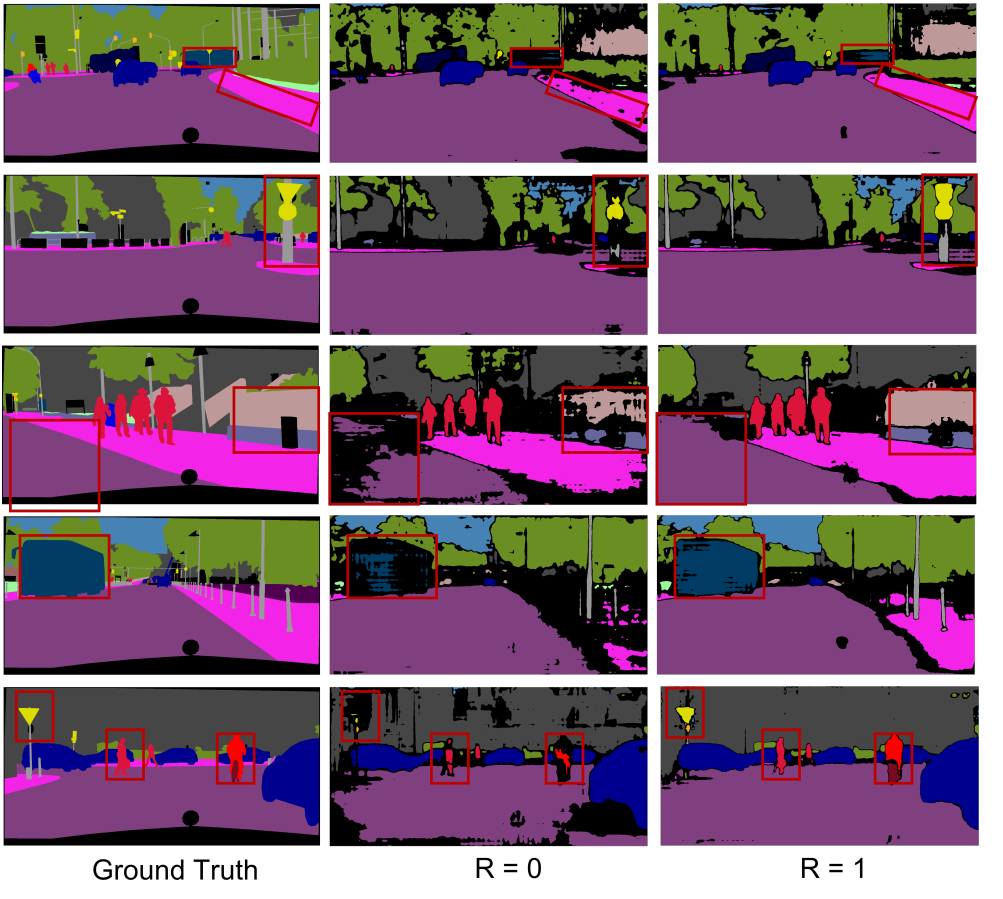

Figure 9: Visualization of pseudolabels obtained from the first (R=0) and second (R=1) round. Pseudolabels obtained in the second round are more accurate allowing us to train more robust models in the last round of training.

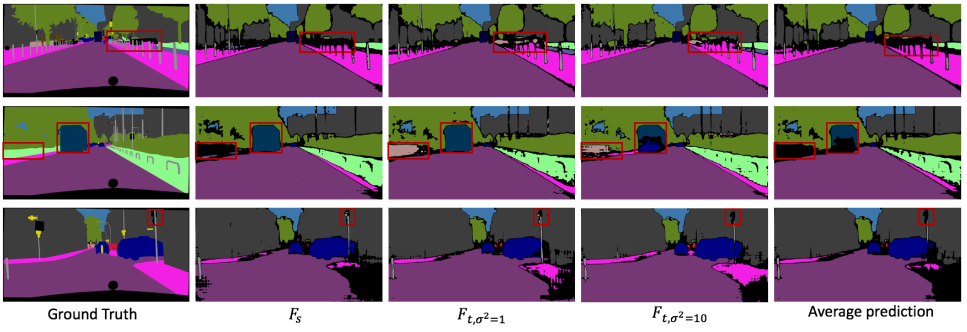

Figure 10: Visualization of pseudolabels obtained by averaging the predictions of two target networks $F_{t,\sigma^2=1}$, $F_{t,\sigma^2=10}$ and a one source network F_s. Averaging the predictions allows us to generate more accurate pseudolabels.

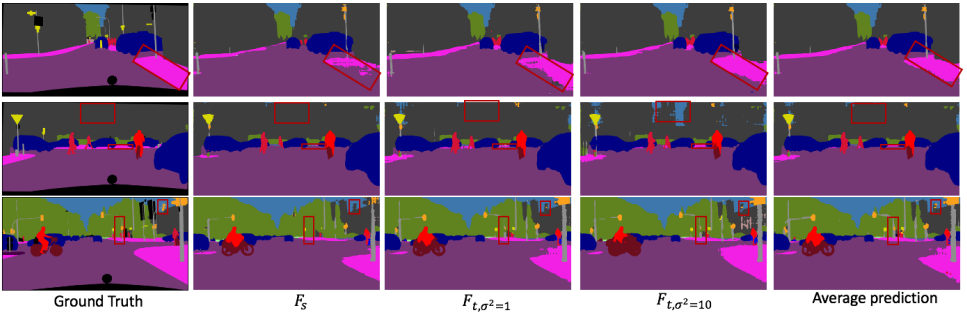

Figure 11: Visualization of results obtained by averaging the predictions of two target networks $F_{t,\sigma^2=1}$, $F_{t,\sigma^2=10}$ and a one source network F_s. Averaging the predictions allows us to further improve performance by better distinguishing similar structures (e.g., road, sidewalk) and identifying small objects.

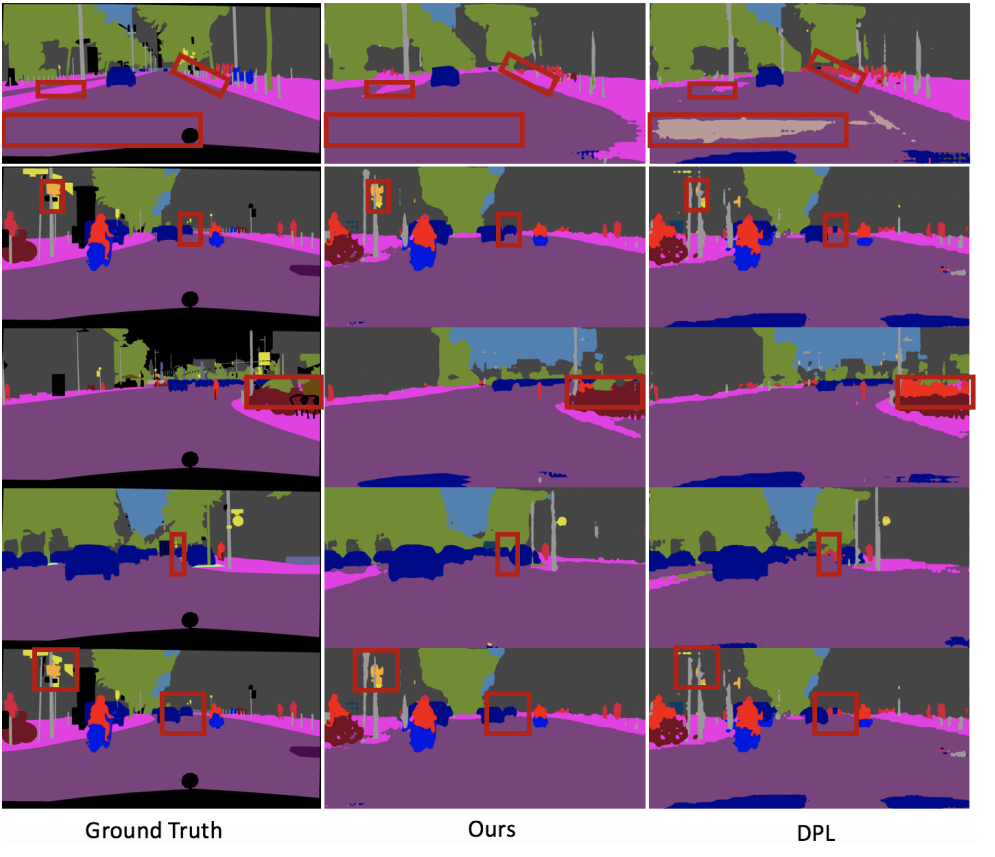

Figure 12: Qualitative comparison of our method with DPL[1]. Our method generates better predictions that are closer to the ground-truth.

References

[1] Yiting Cheng, Fangyun Wei, Jianmin Bao, Dong Chen, Fang Wen, and Wenqiang Zhang. Dual path learning for domain adaptation of semantic segmentation. In *ICCV*, 2021.

[2] Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *CVPR*, 2022.

[3] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[4] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[5] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *CVPR*, 2020.

[6] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, 2019.

[7] Luke Melas-Kyriazi and Arjun K Manrai. Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In *CVPR*, 2021.

[8] Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, 2020.

[9] 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 *CVPR*, 2018.

[10] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019.

[11] Yuxi Wang, Junran Peng, and ZhaoXiang Zhang. Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation. In *ICCV*, 2021.

[12] Yanchao Yang and Stefano Soatto. FDA: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020.

[13] Yanchao Yang, Dong Lao, Ganesh Sundaramoorthi, and Stefano Soatto. Phase consistent ecological domain adaptation. In *CVPR*, 2020.

[14] Jingyi Zhang, Jiaxing Huang, Zichen Tian, and Shijian Lu. Spectral unsupervised domain adaptation for visual recognition. In *CVPR*, 2022.

[15] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.