Supplementary Material
Beyond deterministic translation for unsupervised domain adaptation

Eleni Chiou
University College London
eleni.chiou.17@ucl.ac.uk

Eleftheria Panagiotaki
University College London
panagio@cs.ucl.ac.uk

Iasonas Kokkinos
University College London
i.kokkinos@cs.ucl.ac.uk

1. Overview

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

2. Training procedure

In Sec. 2.1 we provide some additional details regarding the training procedure of our stochastic translation network while in Sec. 2.2 we provide additional details about the pseudolabeling.

2.1. Stochastic translation for UDA

We have already discussed stochastic translation for UDA in the main paper; here we just provide a more detailed description. Our stochastic translation network is based on MUNIT [2]. 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. 1a) and cross-domain translation (Fig. 1b). We reconstruct the original image x, using the source generator G_s, which takes as input at the first layer the content code c and its subsequent layers are modulated by Adaptive Instance Normalization (AdaIN) [1] 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:

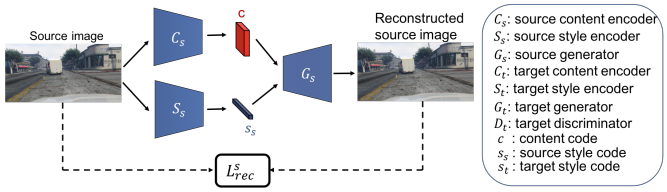
$$L^{s_t}_{rec} = \sum_{x \in \mathcal{S}} \|S_t(G_t(C_s(x), \mathbf{v})) - \mathbf{v}\|_2.$$

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

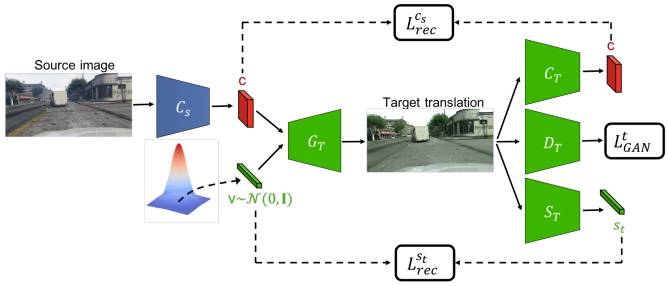
$$L^s_{sem} = \mathcal{L}_{ce}(F(\mathbf{T}[x, \mathbf{v}]), p),$$

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.



(a) With-in domain reconstruction.



(b) Cross-domain translation.

Figure 1. Training procedure of the stochastic translation network: we perform with-in domain reconstruction and cross-domain translation. The target cycle is omitted for clarity.

The full objective is given by

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

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. Pseudolabeling with class-wise confidence thresholds

We assign pseudolabels to samples for which the dominant class has a score above a certain threshold θ. Similarly to [3] 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 pseudollabels we want to retain. We set $r=0.6$ based on the mIoU of the validation set. In cases where $\theta_c>0.9$, we set $\theta_c=0.9$. In Sec. 3 we provide results for the selection of r.

3. Additional results

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 1 we report results obtained with and without the entropy-based adversarial loss (Eq. 9 in the main paper). We report results on GTA-to-Cityscapes using DeepLab-V2 with ResNet-101.

Loss	mIoU
$\mathcal{L}_{CE}$	43.8
$\mathcal{L}_{CE}+\mathcal{L}_{adv}$	**44.4**

Table 1. We obtain better results by adding a entropy-based regularization $\mathcal{L}_{adv}$, to the output of source-domain network F_s when it is driven by translated target images.

As we mentioned in Sec. 2.2, we set $r=0.6$ based on the mIoU of the validation set. In Table 2 we provide the mIoU obtained on the validation set for different values of r. We report results on GTA-to-Cityscapes using DeepLab-V2 with ResNet-101.

r	mIoU
1.0	48.2
0.8	60.9
0.7	67.2
0.6	**69.8**
0.5	68.7
0.4	67.3
0.3	67.2

Table 2. mIoU obtained using different values of r for class-wise confidence threshold selection. We observe that $r=0.6$ gives the best results.

Fig. 2 shows diverse translations of images from the SYNTHIA source dataset to the Cityscapes target dataset. We observe that our method generates sharp samples of high variability and noticeable diversity that results in substantially improved segmentation performance of the target-domain network.

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 our method 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 obtrain more robust pseudolabels for the target data.

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.

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.

References

[1] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 1

[2] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 1

[3] 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. 2