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

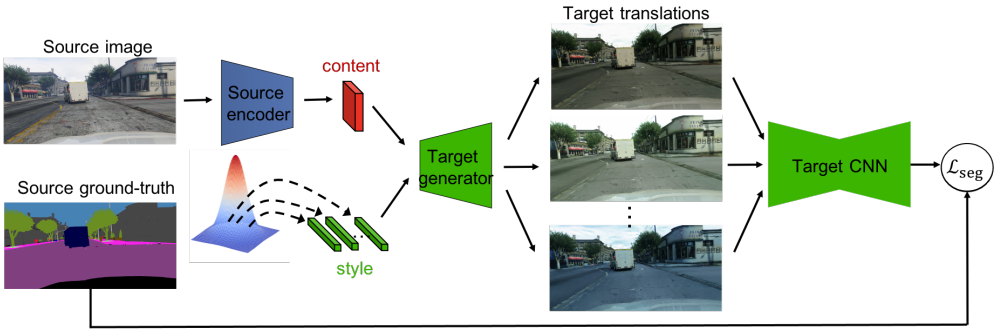

Figure 1. Unsupervised Domain Adaptation with stochastic translation: we rely on a content-style separation network to associate a synthetic image from the GTA5 dataset (source) with a distribution of image translations to the target domain. These translations preserve the content signal and adopt the appearance properties of the Cityscapes dataset (target) through randomly sampled style codes, S_t. We use the resulting images to train a target-domain network tasked with predicting the labels of the respective source-domain image, irrespective of the style variation. Stochasticity in UDA allows the translation networks to generate multiple, sharp outputs that better capture the diversity of the scenes in the target domain, and train the target-domain network with a more representative set of images.

Abstract

In this work we challenge the common approach of using a one-to-one mapping ('translation') between the source and target domains in unsupervised domain adaptation (UDA). Instead, we rely on stochastic translation to capture inherent translation ambiguities. This allows us to (i) train more accurate target networks by generating multiple outputs conditioned on the same source image, leveraging both accurate translation and data augmentation for appearance variability (ii) impute robust pseudo-labels for the target data by averaging the predictions of a source network on multiple translated versions of a single target image and (iii) train and ensemble diverse networks in the target domain by modulating the degree of stochasticity in the translations. We report improvements over strong recent baselines, leading to state-of-the-art UDA results on two challenging semantic segmentation benchmarks.

1. Introduction

Unsupervised Domain Adaptation (UDA) aims at accommodating the differing statistics between a 'source' and a 'target' domain, where the source domain comes with input-label pairs for a task, while the target domain only contains input samples. Successfully solving this problem can allow us for instance to exploit synthetically generated datasets that come with rich ground-truth to train models that can perform well in real images with different appearance properties. Translation-based approaches [6, 12, 17, 37, 38] rely on establishing a transformation between the two domains (often referred to as 'pixel space alignment') that bridges the difference in their statistics while preserving the semantics of the translated samples. This translation can then be used as a mechanism for generating supervision in the 'target' domain based on ground-truth originally available in a 'source' domain.

In this work we address a major shortcoming of this approach - namely the assumption that this translation is a deterministic function, mapping a single source to a single target image. Recent works on the closely related problem of unsupervised image translation [1, 14, 16, 42] have highlighted that this is a strong assumption and is frequently violated in practice. For instance a nighttime scene can have multiple daytime counterparts where originally invisible structures are revealed by the sun and also illuminated from different directions during the day. To mitigate this problem these techniques introduce methods for multi-

modal, or stochastic translation, that allows an image from one domain to be associated with a whole distribution of images in another. An earlier work [7] has shown the potential of generating multiple translations in the narrow setting of supervised domain adaptation across different medical imaging modalities. In this work we exploit stochasticity in the problem of UDA in three complementary ways and show that stochastic translation improves upon the current state-of-the-art in UDA on challenging semantic segmentation benchmarks.

Firstly, we use stochastic translation across the source and target domains by relying on multimodal translation [1, 14, 16, 42]. We show that allowing for stochastic translations yields clear improvements over the deterministic CycleGAN-based baseline, as well as all published pixel space alignment-based techniques. We attribute this to the ability of the multimodal translation to generate more diverse and sharper samples, that provide better training signals to the domain network.

Secondly, we exploit the ability to sample multiple translations for a given image in order to obtain better pseudo-labels for the unlabelled target images: we generate multiple translations of every target image into the source domain, label each according to a source-domain CNN, and average the resulting predictions to form a reliable estimate of the class probability. This is used as supervision for target-domain networks, and is shown to be increasingly useful as the number of averaged samples per image grows.

Thirdly, we modify the variance of the latent style code in order to train and ensemble complementary target-domain networks, each of which is adapted to handle a different degree of appearance variability. The results of ensembling these networks on the target data are then used to train a single target-domain network that outperforms all methods that also rely on ensembling-based supervision in the target domain.

We show that each of our proposed contributions yields additional improvements over strong recent baselines, leading to state-of-the-art UDA results on two challenging semantic segmentation benchmarks.

2. Related Work

UDA approaches [3,6,12,17,20,21,23,30,33,41,44] aim at learning domain invariant representations by aligning the distributions of the two domains at feature/output level or at image level. Based on the observation that the source and the target domain share a similar semantic layout, [30, 32] rely on adversarial training to align the raw output and entropy distributions respectively. However, such a global alignment does not guarantee that individual target samples are correctly classified. Category-based feature alignment methods [19, 27, 31, 33, 35, 41] attempt to address this by mapping target-domain features closer to the corresponding source-domain features.

Image-level UDA methods aim at aligning the two domain at the raw pixel space. [6, 12, 17, 38] rely on CycleGAN [43] to translate source domain images to the style of the target domain. Two recent works [20, 39] bypass the need for training an image translation network by relying on simple Fourier transform and global photometric alignment respectively.

Complementary to the idea of translation is the use of self-training [28, 40, 44, 45] which has been originally used in semi-supervised learning. Self-training iteratively generates pseudo-labels for the target domain based on confident predictions and uses those to supervise the model, implicitly encouraging category-based feature alignment between the source and the target domain. Another direction pursued in [8, 22] is to leverage the unlabeled target data by using consistency regularization to make the model predictions invariant to perturbations imposed in the input images.

Two recent works [6, 17] that rely on both image-level alignment and self-training are more closely related to our work. [17] relies on CycleGan to translate source images to the style of the target domain. They train the image translation network and the segmentation network alternatively and introduce a perceptual supervision based on the segmentation network to enforce semantic consistency during translation. They also generate pseudo-labels for the target data based on high confident predictions of the target network and use those to supervise the target network. [6] improves upon [17] by replacing the single-domain perceptual supervision with a cross-domain perceptual supervision using two segmentation networks operating in the source and the target domain respectively. In addition, they rely on both the source and the target networks to generate pseudo labels for the target data. Similar to these works we rely on image-to-image translation to translate source images to the style of the target domain, but we go beyond their one-to-one mapping approach which allows to leverage both accurate translation and data augmentation for appearance variability. In addition, as in [6] we use source and target networks to generate pseudo-labels, but we exploit stochasticity in the translation to generate more robust pseudo-labels.

3. Methods

We start in Sec. 3.1 by introducing the background of using translation in UDA, and then introduce our technical contributions from Sec. 3.2 onwards. Our presentation gradually introduces different components, loss terms, and processes used in UDA, and we summarize how everything is pieced together in Sec. 3.5.

3.1. Domain Translation and UDA

In UDA we consider a source dataset with paired image-label data: $\mathcal{S} = \{(x_s^i, y_s^i)\}, i \in [1, S]$ and a target dataset

comprising only image data $\mathcal{T} = \{x_t^i\}, i \in [1, T]$. Our task is to learn a segmentation system that provides accurate predictions in the target domain; we assume a substantial domain gap, precluding the naive approach of training a network on $\mathcal{S}$ and then deploying it in the target domain.

Output-space alignment UDA approaches [30] train a single segmentation network, F on both the source and the target images, using a cross-entropy loss in the source domain and an adversarial loss in the target domain to statistically align the predictions on target images to the distribution of source predictions. This results in a training objective of the following form:

$$\mathcal{L}(F) = \sum_{(x,y)\in\mathcal{S}} \mathcal{L}_{ce}(F(x), y) + \sum_{x\in\mathcal{T}} \mathcal{L}_{adv}(F(x)), \quad (1)$$

where $F(x)$ the softmax output.

In [32] entropy-based adversarial training is used to align the target entropy distribution to the source entropy distribution instead of aligning the raw predictions, resulting in the following objective:

$$\mathcal{L}(F) = \sum_{(x,y)\in\mathcal{S}} \mathcal{L}_{ce}(F(x), y) + \sum_{x\in\mathcal{T}} \mathcal{L}_{adv}(E(F(x))), \quad (2)$$

where $E(F(x)) = -F(x)\log(F(x))$ is the weighed self-information.

Given that the network provides low-entropy predictions on source images, adversarial entropy minimization promotes low-entropy predictions in the target domain. The entropy-based objective forces the target points to be classified confidently, and aims at reducing misclassifications by aligning the decision boundaries of F with low-density areas of the target domain - reflecting a desired property under the cluster assumption [4]. Still, having a single network F that successfully operates in both domains can be challenging due to the broader intra-class variability caused by the domain gap.

Pixel-space alignment approaches try to mitigate this problem by establishing a relation between the distributions of the source and target domain images and using that to supervise a network that only operates with target-domain images. In its simplest form, adopted also in [2, 12, 17, 36, 38] this relation is a deterministic translation function $\mathbf{T}$ that maps source images to the target domain, resulting in the following objective:

$$\mathcal{L}(F_t) = \sum_{(x,y)\in\mathcal{S}} \mathcal{L}_{ce}(F_t(\mathbf{T}[x]), y) + \sum_{x\in\mathcal{T}} \mathcal{L}_{adv}(E(F_t(x))), \quad (3)$$

where the difference with respect to Eq. 2 is that the translated version of x, $\mathbf{T}[x]$ is passed to the target-domain segmentation network, F_t. A straightforward way of obtaining such a translation function is through unsupervised translation between the two domains [43]; more sophisticated approaches [12, 17, 38] train the translation network in tandem with the UDA task, using for instance semantic losses to ensure the semantics of the source domains are preserved during cyclic translation. Other methods that implicitly use translation include [39], where a Fourier domain-based approach is used to align the two domains, effectively bypassing the need for a pixel-level translation network.

Source Target translations

Figure 2. Diverse translations of images from the GTA 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.

This approach creates a target-adapted variant of the source-domain dataset, allowing us to train a single network that is tuned exclusively to the statistics of the target domain. This reduces the intra-class variance and puts less strain on the segmentation network, but relies on the strong assumption that such a deterministic translation function exists. In this work we relax this assumption and work with a *distribution on translated images*. This better reflects most UDA scenarios and provides us with novel and simple tools to improve UDA performance, as described below.

3.2. Stochastic Translation and UDA

We propose to replace the determinstic translation function $\mathbf{T}[x]$, with a distribution over images given by $\mathbf{T}[x, \mathbf{v}], \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathbf{v}$ is a random vector sampled from a normal distribution with zero mean and unit covariance [14]. For instance when translating a nighttime scene into its daytime scene, the random argument can reflect the (unpredictable) position of the sun, clouds or obscured objects. For the synthetic-to-real case that we handle in our experiments we can see from Fig. 1 that the translation network can indeed generate scenes illuminated differently as well as different cloud patterns, allowing us to capture more faithfully the range of scenes encountered in the target domain. We note that $\mathbf{T}$ remains deterministic and can be expressed by a neural network, but has a random argument which results in a distribution on translated images.

This change is reflected in the UDA training objective by replacing the loss of the translated image with the *expected loss* of the translated image:

$$\mathcal{L}(F_t) = \sum_{(x,y)\in\mathcal{S}} \mathbf{E}_{\mathbf{v}}\left[\mathcal{L}_{ce}(F_t(\mathbf{T}[x,\mathbf{v}]),y)\right] + \sum_{x\in\mathcal{T}} \mathcal{L}_{adv}(E(F_t(x))), \tag{4}$$

where the expectation is taken with respect to the random vector $\mathbf{v}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, driving the stochastic translation. We note that during training we create minibatches by first sampling images from $\mathcal{S}$ and then sampling $\mathbf{v}$ once per image, effectively replacing the integration in the expectation with a Monte Carlo approximation.

Our stochastic translation network is based on MUNIT [14]: we start from reconstructing images in each domain through content and style encodings, where content is fed to the first layer of a generator whose subsequent layers are modulated by style-driven Adaptive Instance Normalization [13] - this amounts to minimizing the following domain-specific autoencoding objectives:

$$L_s = \sum_{x\in\mathcal{S}} \|x - G_s(C_s(x),S_s(x))\|,$$
$$L_t = \sum_{x\in\mathcal{T}} \|x - G_t(C_t(x),S_t(x))\|$$

where C_s, S_s, G_s are the content-encoder and style-encoder and generator networks for the source domain s respectively, while C_t, S_t, G_t are those of the target domain t.

The basic assumption is that the commonalities between two domains are captured by the shared content space, allowing us to pass content from the source image to its target counterparts, as also shown in Fig. 1. The uncertainty in the translation is captured by a domain-specific style encoding that is inherently uncertain given the source image.

This results in the following stochastic translation function from source to target:

$$\mathbf{T}[x,\mathbf{v}] \doteq G_t(C_s(x),\mathbf{v}),\ \mathbf{v}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\ x\in\mathcal{S}$$

where we encode the content of the source image through $C_s(x)$ and then pass it to the target-domain generator G_t that is driven by a random style code $\mathbf{v}$. A similar translation is established between the target and source domains, and adversarial losses on both domains ensure that the resulting translations appear as realistic samples of the respective domains.

The alignment of the shared latent space for content is enforced by a cycle translation objective:

$$L_{cycle}(x) = \|C_t(G_t(C_s(x),\mathbf{v})) - C_s(x)\|_2,\ x\in\mathcal{S}$$

ensuring that regardless of the random style code, we can recover the original content $C_s(x)$ by encoding the translated image through the respective content encoder. A similar loss is used for the style code, while the losses are applied to translations to both domains.

We preserve semantic information during translation by imposing a semantic consistency constraint to our stochastic translation network using a fixed segmentation network F pretrained on source and target data using Eq. 2. Given an image x we obtain the predicted labels before translation as $p = \text{argmax}(F(x))$ and enforce semantic consistency during translation using an objective of the following form:

$$L_{sem}(x) = \mathcal{L}_{ce}(F(\mathbf{T}[x,\mathbf{v}]),p). \tag{5}$$

We argue that stochastic translation provides us with a natural mechanism to handle UDA problems with large domain gaps where things may unavoidably get 'lost in translation'; the content cycle constraint can help preserve semantics during translation, while the random style allows the translated image appearance to vary freely, avoiding a deterministic and blunt translation.

This is demonstrated in Fig. 2, where we show some of the samples obtained by our method: we observe that our method generates sharp samples of high variability and noticeable diversity. As we show in the experimental results section, this results in substantially improved UDA accuracy. We also note that our approach includes deterministic translation as a special case, since the network can always learn to ignore the source of stochasticity if that is not useful - hence deterministic translation-based results provide effectively a lower bound on what our method can deliver.

3.3. Stochastic translation and pseudo-labelling

Having shown how stochastic translation from the source to the target domain can be integrated in the basic formulation of UDA, we now turn to exploiting stochastic translation from the target to the source domain, which is freely provided by the cycle-consistent formulation of [14].

In particular we consider a complementary segmentation network, F_s, that operates in the source domain and can be directly supervised from the labeled source dataset based on a cross-entropy objective:

$$\mathcal{L}(F_s) = \sum_{(x,y)\in\mathcal{S}} \mathcal{L}_{ce}(F_s(x),y) \tag{6}$$

This network can provide labels for the target-domain images, once these are translated from the target to the source domain; these pseudo-labels of the target data can in turn be used to supervise the target-domain network through a cross-entropy loss.

In the case of deterministic translation pseudo-labels would be obtained by the following expression:

$$\hat{y}(x) = F_s(\mathbf{I}[x]), \quad x\in\mathcal{T}, \tag{7}$$

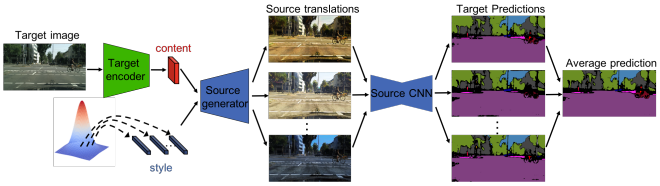

Figure 3. Stochastic translation for pseudo-labeling: the target image (left) results in multiple target-domain translations (middle) which are processed by the source-domain network, F_s and averaged to produce pseudo-labels for the target image; the latter are used to supervise the target-domain network F_t through a cross-entropy loss.

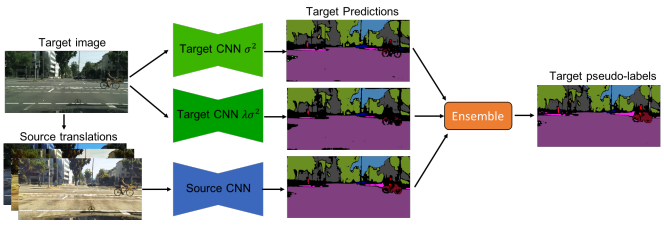

Figure 4. Ensembling of a triplet of networks — two target networks trained with different degrees of stochasticity in the translation (σ^2) and a source network — for robust pseudo-labeling.

where $\mathbf{I}$ is the inverse transform from the target to the source domain, and $\hat{y}$ indicates the pixel-level posterior distribution on labels.

In our case however we have a whole distribution on translations for every image in $\mathcal{T}$. We realise that we can exploit multiple samples to obtain a better estimate of the pseudo-labels. In particular we form the following Monte Carlo estimate of pseudo-labels:

$$\begin{aligned}\hat{y}(x) &= E_{\mathbf{v}}\left[F_s(\mathbf{I}[x,\mathbf{v}])\right], \quad x\in\mathcal{T}, \mathbf{v}\sim\mathcal{N}(\mathbf{0},\mathbf{I}) \\ &\simeq \frac{1}{K}\sum_{k=1}^{K} F_s(\mathbf{I}[x,\mathbf{v}_k]),\end{aligned}$$

where $\mathbf{v}_k$ are independently sampled from the normal distribution. As shown in Fig. 3 the label maps obtained through this process tend to have fewer errors and be more confident, since averaging the results obtained by different translations can be expected to cancel out the fluctuation of the predictions around their ground-truth value.

Our experimental results indicate that using $K = 10$ yields substantially better results that using a single sample. We also note that pseudo-label generation is a one-off process done prior to training the target-domain network, and consequently the number of samples, K, does not affect training time.

3.4. Stochasticity-driven training of diverse network ensembles

An experimental approach that has been recently adopted by several recent works [6, 39] consists in ensembling different networks trained for UDA, and using their predictions as an enhanced pseudo-labeling mechanism. For instance in [39] this was accomplished by modifying one of the main design parameters of their phase-driven translation algorithm. A main recipe for successful network ensembling is to generate complementary networks, so that they make uncorrelated errors, which hopefully cancel out.

Based on the understanding that the stochasticity driving our translation mechanism can be seen as implementing appearance-level dataset augmentation in the target domain, we introduce a simple twist to the translation mechanism that allows us to train networks that operate in different regimes. For this we scale by a constant the variance of the normal distribution used to sample the random style code - this amounts to generating more diverse translations than those suggested by the image statistics of the target domain. On one hand this trains a target network that can handle a broader range of inputs, but on the other hand it may waste capacity to handle unrepresentative samples.

We train two such networks, one with the variance left intact and the other with the variance scaled by 10, and average their predictions with those of the source-domain network described in the previous subsection as shown in Fig. 4. Our results show that this triplet of networks yields a clear boost over the baseline of operating with a single network.

Further following common practice in UDA we use the resulting ensembling results as pseudo-labels in the next round of training - this yields further improvements, as documented in detail in the experimental results section.

3.5. Training objectives

Having described the components of our method, we now summarize the losses used for training our networks.

Firstly, we train our stochastic translation network using the process of [14] and introduce a semantic consistency loss as in [12] to preserve semantics during translation. We provide a more detailed description in the supplementary material.

For the target-domain network the basic objective has already been provided in Eq. 4, where $\mathcal{L}_{ce}$ is the standard cross-entropy loss and $\mathcal{L}_{adv}$ is the adversarial entropy minimization objective [32]. A more sophisticated objective can train this network with pseudo-labels, obtained either from a source-domain network as described Sec. 3.3 or from the ensembling of multiple networks, as described in Sec. 3.4.

In that case the objective becomes:

$$\mathcal{L}(F_t) = \sum_{(x,y)\in\mathcal{S}} \mathbf{E}_{\mathbf{v}} \left[\mathcal{L}_{ce}(F_t(\mathbf{T}[x,\mathbf{v}]), y)\right] + \sum_{x\in\mathcal{T}} \mathcal{L}_{adv}(E(F_t(x))) + \sum_{x\in\mathcal{T}} \mathcal{L}^{\theta}_{ce}(F_t(x), \text{argmax}(\hat{y})), \tag{8}$$

where the cross entropy loss $\mathcal{L}^{\theta}_{ce}(F_t(x))$ is only applied on pseudo-labels where the dominant class has a score above the threshold θ. Similar to [44] we use class-wise confidence thresholds to address the inter-class imbalance and avoid ignoring hard classes. Specifically, for each class c the threshold θ_c equals to 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 pseudo-labels we want retain. We provide more details in the supplementary material.

Finally, for the source-domain network, we observed experimentally that we obtain better results by adding an entropy-based regularization to the output of F_s when it is driven by translated target images - this ensures that the source network will correctly classify the source images, while placing its boundaries far from areas populated by synthetic source-domain images. The objective function for the source network becomes:

$$\mathcal{L}(F_s) = \sum_{(x,y)\in\mathcal{S}} \mathcal{L}_{ce}(F_s(x), y) + \sum_{x\in\mathcal{T}} \mathbf{E}_{\mathbf{v}} \left[\mathcal{L}_{adv}(F_s(\mathbf{I}[x,\mathbf{v}]))\right], \tag{9}$$

forming the source-domain counterpart to the objective encountered in Eq. 4.

4. Experiments

We evaluate the proposed approach on two common UDA benchmarks for semantic segmentation. In particular we use the synthetic dataset GTA5 [25] or SYNTHIA [26] with ground-truth annotations as the source domain and the Cityscapes [9] dataset as the target domain with no available annotations during training. We evaluate the performance using the mean intersection over union score (mIoU) across semantic classes on the Cityscapes validation set.

4.1. Datasets

Cityscapes [9] is a real-world dataset of diverse urban street scenes collected from different cities. We use 2975 training images and 500 validation images with resolution 2048×1024. We resize the images to 1024×512. We train the image translation network and the segmentation network using the training set and report the results on the validation set.

GTA5 [25] consists of 24966 synthesized images captured from a video game. The original images have resolution 1914×1052 and they are resized to 1024×512 for training. GTA5 provides pixel-level semantic annotations of 33 classes. Similar to other studies, we use the 19 common classes between GTA5 and Cityscapes.

SYNTHIA [26] consists of synthesized images rendered from a virtual city. We use SYNTHIA-RAND-CITYSCAPES subset which has 9400 annotated images with resolution 1280×760. We use the 16 common classes between SYNTHIA and Cityscapes for training and we evaluate the performance on 16 classes and a subset of 13 classes following previous studies [17, 32, 38, 39].

4.2. Implementation Details

Stochastic translation network We rely on MUNIT [14] to establish a stochastic translation across the source and target domain. Images from the source and the target domain are resized to 1024×512 and cropped to 400×400. We train the network for 600000 iterations with batch size 1 and a learning rate starting 0.0001 and decreasing by half every 100000 iterations. We provide more details in the supplementary material.

Semantic Segmentation network We train two different architectures, i.e., DeepLabV2 [5] with ResNet101 [11] backbone, and FCN-8s [18] with VGG-16 [29] backbone. We train DeepLabV2 with ResNet101 using Stochastic Gradient Descent optimizer with initial learning rate 2.5×10^{-4}, momentum 0.9 and weight decay 1×10^{-4}. The learning rate is adjusted according to the poly learning rate scheduler with a power of 0.9. We train FCN-8s with VGG-16 using ADAM with initial learning rate 1×10^{-5} and momentum 0.9 and 0.99. The learning is decreased by a factor $\gamma = 0.1$ every 50000 iterations. We use the same discriminator for both the DeepLabV2 and FCN-8s. The discriminator used to adapt the entropy maps is similar to [24]. It has 4 convolutional layers, each followed by a leaky-ReLU layer with negative slope of 0.2. The last layer is a binary classification layer classifying the inputs either as source or target.

4.3. Results

Stochastic translation: we start by examining in how stochastic translation improves performance compared to deterministic translation. In all cases the segmentation model is DeepLabV2 [5] and the source and target datasets are GTA5 [25] and Cityscapes [9] respectively.

In Table 1 we start with an apples-to-apples comparison that builds on directly on the ADVENT baseline [32]; the first two rows compared 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 - which amount to training with Eq. 3. The forth row reports our stochastic translation-based result, amounting to training with Eq. 4. We observe a substantial improvement, that can be attributed solely to the stochasticity of the translation. The

Method	Output space	Pixel space	mIoU
ADVENT [32]	✓		43.8
ADVENT *	✓		42.9
ADVENT *+ CycleGAN*	✓	✓	45.3
Ours	✓	✓	46.2
Ours w/ L_{sem}	✓	✓	**46.6**

Table 1. GTA to Cityscapes UDA using stochastic translation: We train ADVENT using synthetic images obtained form deterministic translation (CycleGAN) and stochastic translation (Ours); * denotes our retrained models. We observe a clear improvement thanks to pixel-space alignment based on stochastic translation.

last row shows that imposing a semantic consistency constraint as described in Eq. 5 further improves the performance.

F_s, n=1	F_s, n=5	F_s, n=10	F_t, $\sigma^2 = 1$	F_t, $\sigma^2 = 10$	mIoU
✓					43.3
	✓				44.0
		✓			44.4
			✓		46.6
				✓	46.1
		✓	✓		47.7
		✓		✓	47.6
			✓	✓	47.7
		✓	✓	✓	**48.2**

Table 2. Performance of different models and their combinations. The first 3 rows show the performance of the source network F_s when averaging the predictions of multiple translations n, of a target image while rows 4, 5 show the performance of the target networks F_t, trained with different degrees of stochasticity (σ^2) in the translation. Averaging the predictions of multiple translations and combining the three models allows us to obtain better pseudo-labels for the target domain.

Pseudo labeling As discussed in Section 3.3 we translate from the target to the source domain and generate pseudo labels for the target data. The first three rows in Table 2 show the impact of the number of samples n, on performance. Averaging the predictions of multiple translations for a given target image improves the performance and allows to obtain better pseudo labels for the target domain. Our results show that using 10 samples yields better performance. In rows 4, 5 of the same table we report the performance obtained from the two target networks trained with different degrees of stochasticity in the translation as described in Sec. 3.4. Averaging the prediction of the three networks gives the best results, indicating the complementary of the model predictions.

Network ensembling: Table 3 shows the results obtained in three rounds of pseudo-labeling and training, following the approach of [6, 17, 39]. In the first round ($R = 0$) we

Model	mIoU
F_s (R=0)	44.42
F_t, $\sigma^2 = 1$ (R=0)	46.65
F_t, $\sigma^2 = 10$ (R=0)	46.09
Ens (R=0)	48.25
F_s (R=1)	49.13
F_t, $\sigma^2 = 1$ (R=1)	50.14
F_t, $\sigma^2 = 10$ (R=1)	50.68
Ens (R=1)	52.02
F_s (R=2)	51.35
F_t, $\sigma^2 = 1$ (R=2)	52.76
F_t, $\sigma^2 = 10$ (R=2)	52.69
Ens (R=2)	53.62

Table 3. Ablation study on GTA to 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.

train the target and source networks with Eq. 4 and Eq. 9 respectively using the synthetic and real data and average the predictions of the three models to generate pseudo-labels for the target data. In the second round (R=1) we use the generated pseudo-labels as ground-truth labels to train the source and target networks. We observe that the pseudo-labels obtained by ensembling improve the performance of each individual network, as well as the ensemble obtained in the last round (R=2).

Benchmark results We use DeepLabV2 [5] with ResNet101 [11] backbone, and FCN-8s [18] with VGG-16 [29] for the segmentation and compare with [6, 10, 15, 17, 22, 28, 30, 32, 34, 38, 39] which use exactly the same experimental settings. We report both the results obtained using a single target network and the results obtained by ensembling. We provide qualitative results in the supplementary material.

The results for the **GTA-to-Cityscapes** benchmark are summarized in Table 4. Our results show that our methods achieves state-of-the-art performance. When compared with other approaches relying on both deterministic translation and pseudo-labeling [6, 17], our approach performs better while at the same time is simpler. In particular, [17] and [6] train both the image translation and segmentation networks multiple times and use complex warm-up stages [6]. On the other hand we train the image translation network only once and use the same image translation network in all rounds of pseudo-labeling and training.

The results for the **SYNTHIA-to-Cityscapes** benchmark are reported in Table 5. Following the protocol evaluation protocol of previous studies [17, 38, 39] we report the mIoU of our method on 13 and 16 classes. We ob-

Method	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	bicycle	mIoU
VGG backbone																				
AdaptSegNet [30]	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 [30]	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 [17]	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 [15]	**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 [39]	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 [38]	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) [6]	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
ResNet101 backbone																				
AdvEnt [32]	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 [17]	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 [15]	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 [39]	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 [38]	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 [28]	**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. [34]	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 [22]	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) [6]	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
Ours	92.5	48.9	85.9	42.5	34.1	32.7	42.4	36.8	86.6	47.6	84.5	61.3	30.8	87.1	42.7	56.5	10.4	37.4	41.5	52.8
Ours (Ensemble)	93.5	52.3	86.0	**42.5**	**34.8**	33.2	42.4	36.7	**86.8**	**49.4**	84.4	61.4	31.3	**87.6**	45.2	**56.6**	13.6	39.2	41.6	**53.6**

Table 4. Quantitative comparison on GTA5→Cityscapes. We present per-class IoU and mean IoU (mIoU) obtained using VGG and ResNet101 backbones.

Method	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	sky	person	rider	car	bus	motocycle	bicycle	mIoU	mIoU*
VGG backbone																		
AdvEnt [32]	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 [17]	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 [39]	**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 [38]	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) [6]	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**
ResNet101 backbone																		
AdvEnt [32]	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 [15]	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 [17]	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 [39]	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 [38]	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 [28].	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. [34]	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 [22]	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) [6]	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
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 5. 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.

serve that our methods outperforms previous state-of-the art methods by a large margin. We note here that the domain gap between SYNTHIA and Cityscapes is much larger compared to the domain gap between GTA and Cityscapes. We attribute the substantial improvements obtained by our method to the stochasticity in the translation which allows us to better capture the range of scenes encountered in the two domains and to generate sharp samples even in cases where there is a large domain gap between the two domains.

5. Conclusions

In this work we have introduced stochastic translation in the context of UDA and showed that we can reap multiple benefits by acknowledging that certain structures are 'lost in translation' across two domains. The networks trained directly through stochastic translation clearly outperforms all comparable counterparts, while we have also shown that we retain our edge when combining our approach with more involved UDA approaches such as pseudo-labeling and ensembling.

References

[1] Amjad Almahairi and et al. Augmented CycleGAN: Learning many-to-many mappings from unpaired data. In *ICML*, 2018. 1, 2

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017. 3

[3] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *CVPR*, 2019. 2

[4] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *AISTATS*, 2005. 3

[5] 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. *TPAMI*, 2017. 6, 7

[6] Yiting Cheng, Fangyun Wei, Jianmin Bao, Dong Chen, Fang Wen, and Wenqiang Zhang. Dual path learning for domain adaptation of semantic segmentation. In *CVPR*, 2021. 1, 2, 5, 7, 8

[7] Eleni Chiou, Francesco Giganti, Shonit Punwani, Iasonas Kokkinos, and Eleftheria Panagiotaki. Harnessing uncertainty in domain adaptation for mri prostate lesion segmentation. In *MICCAI*, 2020. 2

[8] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *ICCV*, 2019. 2

[9] 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 *CVPR*, 2016. 6

[10] Xiaoqing Guo, Chen Yang, Baopu Li, and Yixuan Yuan. Metacorrection: Domain-aware meta loss correction for unsupervised domain adaptation in semantic segmentation. In *CVPR*, 2021. 7

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7

[12] 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 *ICML*, 2018. 1, 2, 3, 5

[13] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 4

[14] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 1, 2, 3, 4, 5, 6

[15] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *CVPR*, 2020. 7, 8

[16] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018. 1, 2

[17] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, 2019. 1, 2, 3, 6, 7, 8

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 6, 7

[19] 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 *CVPR*, 2019. 2

[20] Haoyu Ma, Xiangru Lin, Zifeng Wu, and Yizhou Yu. Coarse-to-fine domain adaptive semantic segmentation with photometric alignment and category-center regularization. In *CVPR*, 2021. 2

[21] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *CVPR*, 2020. 2

[22] Luke Melas-Kyriazi and Arjun K Manrai. Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In *CVPR*, 2021. 2, 7, 8

[23] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *CVPR*, 2020. 2

[24] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 6

[25] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 6

[26] 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 *CVPR*, June 2016. 6

[27] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018. 2

[28] Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In *ECCV*, 2020. 2, 7, 8

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *CoRR, abs/1409.1556*, 2014. 6, 7

[30] 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. 2, 3, 7, 8

[31] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In *CVPR*, 2019. 2

[32] 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. 2, 3, 5, 6, 7, 8

[33] Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *ECCV*, 2020. 2

[34] Yuxi Wang, Junran Peng, and ZhaoXiang Zhang. Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation. In *ICCV*, 2021. 7, 8

[35] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-mei Hwu, Thomas S. Huang, and Honghui Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *CVPR*, 2020. 2

[36] 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 *ECCV*, 2018. 3

[37] Jinyu Yang, Weizhi An, Sheng Wang, Xinliang Zhu, Chaochao Yan, and Junzhou Huang. Label-driven reconstruction for domain adaptation in semantic segmentation. In *ECCV*, 2020. 1

[38] Yanchao Yang, Dong Lao, Ganesh Sundaramoorthi, and Stefano Soatto. Phase consistent ecological domain adaptation. In *CVPR*, 2020. 1, 2, 3, 6, 7, 8

[39] Yanchao Yang and Stefano Soatto. FDA: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020. 2, 3, 5, 6, 7, 8

[40] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *CVPR*, 2021. 2

[41] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *NeurIPS*, 2019. 2

[42] Jun-Yan Zhu and et al. Toward multimodal image-to-image translation. In *NIPS*, 2017. 1, 2

[43] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2, 3

[44] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. 2, 6

[45] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019. 2