Semi-Supervised Domain Adaptation with CycleGAN Guided by Downstream Task Awareness

Annika Mütze[1], Matthias Rottmann[1,2] and Hanno Gottschalk[1]
[1]*IZMD & School of Mathematics and Natural Sciences, University of Wuppertal, Wuppertal, Germany*
[2]*School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland*

Keywords: Domain Adaptation, Image-to-Image Translation, Generative Adversarial Networks, Semantic Segmentation, Semi-Supervised Learning, Real2Sim.

Abstract: Domain adaptation is of huge interest as labeling is an expensive and error-prone task, especially on pixel-level like for semantic segmentation. Therefore, one would like to train neural networks on synthetic domains, where data is abundant. However, these models often perform poorly on out-of-domain images. Image-to-image approaches can bridge domains on input level. Nevertheless, standard image-to-image approaches do not focus on the downstream task but rather on the visual inspection level. We therefore propose a "task aware" generative adversarial network in an image-to-image domain adaptation approach. Assisted by some labeled data, we guide the image-to-image translation to a more suitable input for a semantic segmentation network trained on synthetic data. This constitutes a modular semi-supervised domain adaptation method for semantic segmentation based on CycleGAN where we refrain from adapting the semantic segmentation expert. Our experiments involve evaluations on complex domain adaptation tasks and refined domain gap analyses using from-scratch-trained networks. We demonstrate that our method outperforms CycleGAN by 7 percent points in accuracy in image classification using only 70 (10%) labeled images. For semantic segmentation we show an improvement of up to 12.5 percent points in mean intersection over union on Cityscapes using up to 148 labeled images.

1 INTRODUCTION

For automatically understanding complex visual scenes from RGB images, semantic segmentation (pixel-wise classification) is a common but challenging task. The state-of-the-art results are achieved by deep neural networks (Chen et al., 2019; Tao et al., 2020; Liu et al., 2021). These models need plenty of labeled images to generalize. However, a manual label process on pixel level detail is time and cost consuming and usually error-prone (Cordts et al., 2016; Rottmann and Reese, 2022). To reduce the labeling cost, weakly- and semi-supervised methods were proposed (Dai et al., 2015; van Engelen and Hoos, 2020). These methods use weak labels like bounding boxes for segmentation tasks or fewer labels as they can benefit from a pool of unlabeled data. However, they are limited to scenarios captured in the real world and the annotation cost of weak labels still might be intractable (Tsai et al., 2018). On the other hand, in recent years simulations, especially of urban street scenes, were significantly improved (Dosovitskiy et al., 2017; Wrenninge and Unger, 2018). The advantage of synthetic data is that images generated by a computer simulation often come with labels for the semantic content for free. Training on synthetic data has the potential to build a well-performing network as plenty of data is available and diverse scenarios can be generated which are rare or life-threatening in the real world. However, neural networks do not generalize well to unseen domains (Hoffman et al., 2016). Even if the model learns to generalize well on one domain (e.g., real world) it can fail completely on a different domain (e.g., synthetic) (Wrenninge and Unger, 2018) or vice versa. Domain adaptation (DA) is used to mitigate the so-called domain shift (Csurka, 2017) between one domain and another. DA aims at improving the model's performance on a target domain by transferring knowledge learned from a labeled source domain. It has become an active area of research in the context of deep learning (Toldo et al., 2020) ranging from adaptation on feature level (Tsai et al., 2018), adaptation on input level (Hoffman et al., 2018; Dundar et al., 2018; Brehm et al., 2022), self-

Mütze, A., Rottmann, M. and Gottschalk, H.
Semi-Supervised Domain Adaptation with CycleGAN Guided by Downstream Task Awareness.
DOI: 10.5220/0011630900003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 80-90
ISBN: 978-989-758-634-7; ISSN: 2184-4321

training (Mei et al., 2020; Zhang et al., 2021), a combination thereof (Kim and Byun, 2020) to semi-supervised approaches (Chen et al., 2021). Depending on the amount of labels available in the target domain the DA is unsupervised (UDA; no labels available), semi-supervised (SSDA; a few labels available) or supervised (SDA; labels exist for all training samples in the target domain) (Toldo et al., 2020). Adapting on input level to the style of the target domain disregarding the downstream task at hand but preserving the overall scene is referred to image-to-image translation (I2I) and is often realized by generative adversarial networks (GANs) (Goodfellow et al., 2014).

Taking advantage of the synthetic domain we train a downstream task expert therein. We then shift the out-of-domain input (real world) closer to the synthetic domain via a semi-supervised I2I approach based on CycleGAN (Zhu et al., 2017) for mitigating the domain gap. We thereby refrain from changing the expert which leads to a modular DA method. We combine the unsupervised GAN-based I2I method from CycleGAN with a downstream task awareness in a second stage with the help of a relatively small contingent of ground truth (GT) in the real domain to adapt to the needs of the downstream task network. Our main contributions are:

- we present a novel modular SSDA method for semantic segmentation guiding the generator of an I2I domain adaptation approach to a semantic segmentation task awareness. Thereby, our downstream task network does not need to be retrained.
- we demonstrate that our method is applicable to multiple complex domain adaptation tasks.
- we consider a pure domain separation in our analysis by using from-scratch-trained neural networks leading to a less biased domain gap.

Based on our knowledge this is the first time the generator of a GAN setup is guided with the help of a semantic segmentation network to focus on the downstream task. Furthermore, the composition of generator and semantic segmentation network can be understood as a method to establish an abstract intermediate representation in a data-driven manner. We study how well the generator can adapt to its tasks of generating the abstract representation and supporting the downstream task.

The remainder of this paper is organized as follows: In Sec. 2 we review related approaches, particularly in the context of semi-supervised domain adaptation. It follows a detailed description of our method in Sec. 3. We evaluate our method on two different tasks and three different datasets in Sec. 4, showing considerable improvements with only a few GT data samples. Finally, we conclude and give an outlook to future work in Sec. 5.

2 RELATED WORK

Our work is based on two main concepts: DA with I2I, and semi-supervised learning in the context of GANs and DA. For I2I, GANs have shown excellent performance. Formerly, paired data was needed to adapt to the new style (Isola et al., 2017). But as paired data is sparse, unsupervised methods were developed like CycleGAN (Zhu et al., 2017), where a composition of GANs and a cycle consistency loss leads to a consistent mapping between the domains. Depending on whether and how much data is available in a paired manner, I2I is called supervised, semi-supervised or unsupervised. For example, (Shukla et al., 2019) propose a semi-supervised I2I approach in the context of semantic segmentation via image to label transformation. When using I2I in the context of domain adaptation this taxonomy is used for the amount of labeled data in the target domain as explained in the introduction. In the following we will always refer to the latter taxonomy. Independent of the label amount, for I2I in DA, a semantic consistency is pursued, and the performance is measured via the downstream task performance. To this aim (Hoffman et al., 2018) and (Brehm et al., 2022) make use of the task loss in an unsupervised manner to adapt the task network to the real domain. In addition, there are several SSDA approaches in the context of classification (Wu et al., 2018; Saito et al., 2019; Kim and Kim, 2020; Jiang et al., 2020; Mabu et al., 2021). SSDA for semantic segmentation tasks is considered less. (Wang et al., 2020) propose to adapt simultaneously on a semantic and global level using adversarial training. A student-teacher approach aligning the cross-domain features with the help of the intra-domain discrepancy of the target domain, firstly considered in the context of DA by (Kim and Kim, 2020), is proposed by (Chen et al., 2021).

Training in a two stage manner where the first stage (pre-training) aims to initialize good network parameters for the second stage is a common semi-supervised learning technique. We transfer this concept to GANs of an I2I method. In the context of general GANs, using pre-training is not new. (Wang et al., 2018) for example, analyzed pre-training for Wasserstein GANs with gradient penalty((Gulrajani et al., 2017)) in the context of image generation. An overview over when, why and which pre-trained GANs are useful is given by (Grigoryev et al., 2022). We follow their suggestion of pre-training both the

generator and the discriminator but refrain from the suggestion of using ImageNet pre-trained GANs to not distort the domain gap analysis.

In general, UDA methods tend to lack important information of the target domain compared to their counterparts trained in a supervised manner (Chen et al., 2021). On the other hand for I2I it is unlikely to have paired data between real world images and the abstract (e.g., synthetic) domain like assumed in (Shukla et al., 2019). Furthermore, pure I2I methods are task agnostic and therefore may lack semantic consistency (Toldo et al., 2020). For this reason we propose a SSDA method with a task aware I2I component. Independently of our approach, recently a similar approach was published based on classification in a medical context with domain gaps primarily in image intensity and contrast (Mabu et al., 2021). Unlike this publication, we aim at more demanding tasks such as semantic segmentation on much broader domain gaps like realistic to abstract domains leading to potentially broader applications. Furthermore, our method allows an analysis of the influence of the task awareness compared to the standard loss. In contrast to other above-mentioned approaches which use ImageNet pre-trained networks, we train completely from scratch for a pure domain separation. Furthermore, they adapt the downstream task network to mitigate the domain gap, whereas we keep the task network fixed. This leads to a modular approach where the real world domain can be exchanged without the need of retraining the synthetic expert. Besides, we consider I2I from real images to the synthetic domain to retain the benefits of a synthetic expert. This includes the possibility to train and test on a variety of scenarios which are rare and life-threatening in the real world. Testing is more challenging for the other approaches as they consider the opposite direction. In the following we explain our method in more detail.

3 METHODOLOGY

Our method consists of three stages which are depicted in Fig. 1 and explained in detail in this section.

a) Training of Downstream Task Network: We assume that we have full and inexhaustible access to labeled data in the synthetic domain $\mathcal{S}$. Based on a training set (X, y) with $X \subset \mathcal{S}$, we train a neural network f in a supervised manner on the synthetic domain solving the desired task (e.g., semantic segmentation or classification). In contrast to the "common practice" (Kang et al., 2020), we do not use ImageNet (Deng et al., 2009) pre-trained weights for the downstream task

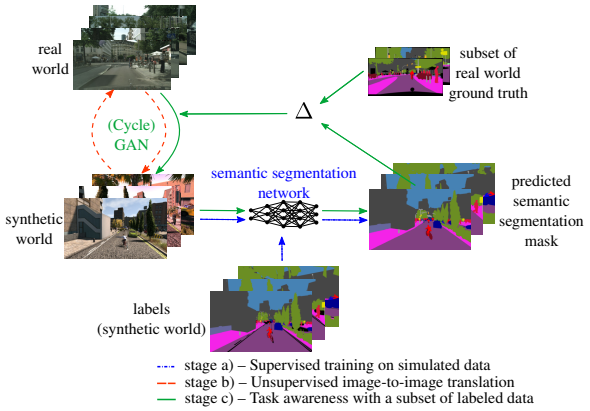

Figure 1: Concept of our method: Stage a) – Training of a downstream task model (e.g., semantic segmentation network) on the abstract/synthetic domain. Stage b) – Training a CycleGAN based on unpaired data to transfer real data into the synthetic domain. Stage c) – We freeze the downstream task network and tune the generator with the help of a few labeled data points by guiding it based on the loss of the downstream task network.

network backbone. To ensure a pure domain separation and a non-biased downstream task network, we train f completely from scratch. This ensures that the network learns only based on the synthetic data, and we prevent a bias towards the real world. As a consequence, we accept a reduction of the total accuracy when evaluating the model on the real domain (out-domain accuracy). However, with the help of an independent validation set, we measure our in-domain accuracy to ensure appropriate performance in the synthetic domain. After the model has reached the desired performance, we freeze all parameters and keep our synthetic domain expert fixed.

b) Unsupervised Image-to-Image Translation: To mitigate the domain gap between real ($\mathcal{R}$) and synthetic ($\mathcal{S}$) data, we build on the established I2I method CycleGAN (Zhu et al., 2017) – a GAN approach which can deal with unpaired data by enforcing a cycle consistency between two generators ($G_{\mathcal{S}\to\mathcal{R}}$ and $G_{\mathcal{R}\to\mathcal{S}}$). The domain discriminators to classify whether the sample is generated or an in-domain sample are denoted with $D_{\mathcal{S}}$ and $D_{\mathcal{R}}$. The generator loss consists of four loss components ($\mathcal{L}_{G_{\mathcal{R}\to\mathcal{S}}}, \mathcal{L}_{G_{\mathcal{S}\to\mathcal{R}}}, \mathcal{L}_{\text{cyc}}, \mathcal{L}_{\text{identity}}$) which are described in detail in (Zhu et al., 2017). As adversarial losses $\mathcal{L}_{G_*}$ we use the least-squares loss (Mao et al., 2017) which has already been used for CycleGAN and leads to a more stable training according to (Zhu et al., 2017). Let $x^r \sim p_{\text{data}}$ denote the data distribution in the real domain then the loss for $G_{\mathcal{R}\to\mathcal{S}}$ is given by

$$\mathcal{L}_{G_{\mathcal{R}\to\mathcal{S}}} = \mathbb{E}_{x^r\sim p_{\text{data}}(x^r)}\left[\left(D_{\mathcal{S}}\left(G_{\mathcal{R}\to\mathcal{S}}(x^r)\right)-1\right)^2\right] \quad (1)$$

The overall generator loss is defined as the weighted sum:

$$\mathcal{L}_{\text{Gen}} = \mathcal{L}_{G_{\mathcal{R}\to\mathcal{S}}} + \mathcal{L}_{G_{\mathcal{S}\to\mathcal{R}}} + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \lambda_{\text{cyc}}\lambda_{\text{id}}\mathcal{L}_{\text{identity}},$$

with weighting factors $\lambda_{\text{cyc}} > 0$ and $\lambda_{\text{id}} > 0$. This leads to a solid image-to-image translation. However, this translation is still task agnostic and therefore potentially misses important features when transferring the style from one domain to another.

c) Downstream Task Awareness: We use the unsupervised models from stage b) as initialization for stage c) where we extend the model training and guide the generator with the help of a small amount of labeled data to the downstream task. Let $\mathcal{T}_{\mathcal{R}} = \{(x_i^r, y_i^r) \in \mathcal{R} \times \mathcal{Y} : i = 1,\dots,N_L\}$ be a labeled subset from domain $\mathcal{R}$ with a label set $\mathcal{Y}$, where N_L denotes the number of labeled samples. We achieve the downstream task awareness by extending the adversarial loss in Eq. (1) for the generator $G_{\mathcal{R}\to\mathcal{S}}$ based on the loss of the downstream task network f. As task loss we consider the (pixel-wise) cross entropy (CE) between the prediction $f(x)$ and the label y denoted by $\mathcal{L}_{task}(x,y) = \mathcal{L}_{CE}(f(x),y)$.

For a labeled training sample $t_i = (x_i^r, y_i^r) \in \mathcal{T}_{\mathcal{R}}$ we define the extended generator loss $\hat{\mathcal{L}}_{G_{\mathcal{R}\to\mathcal{S}}}$ with the help of a weighting factor $\alpha \in [0,1]$ as follows:

$$\hat{\mathcal{L}}_{G_{\mathcal{R}\to\mathcal{S}}}(t_i) = (1-\alpha)\underbrace{\left(D_{\mathcal{S}}\left(G_{\mathcal{R}\to\mathcal{S}}(x_i^r)\right)-1\right)^2}_{\text{adversarial loss as in Eq. (1)}} + \alpha\underbrace{\left(\mathcal{L}_{CE}\left(f(G_{\mathcal{R}\to\mathcal{S}}(x_i^r)), y_i^r\right)\right)}_{\text{task loss}}. \quad (2)$$

We use α for a linear interpolation between the two loss components to control the influence of one or the other loss during training. The overall generator loss therefore becomes:

$$\hat{\mathcal{L}}_{\text{Gen}} = \hat{\mathcal{L}}_{G_{\mathcal{R}\to\mathcal{S}}} + \mathcal{L}_{G_{\mathcal{S}\to\mathcal{R}}} + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \lambda_{\text{cyc}}\lambda_{\text{id}}\mathcal{L}_{\text{identity}}$$

The discriminator losses are kept identically.

The combination of stage b) and c) leads to our semi-supervised learning strategy for the GAN training. For the DA the images generated by $G_{\mathcal{R}\to\mathcal{S}}$ are fed to f. In principle our approach is independent of the chosen architecture as the general concept is transferable, and we make no restriction to the underlying domain expert as long as a task loss can be defined. Furthermore, due to our modular composition, the intermediate representation generated by $G_{\mathcal{R}\to\mathcal{S}}$ could also be used for additional tasks/analyses and could be evaluated with respect to other metrics such as those described by (Pang et al., 2021).

Figure 2: Examples of the Sketchy dataset. Top row: real photos. Bottom row: one of the corresponding sketches.

4 NUMERICAL EXPERIMENTS

We evaluate our method on two different downstream tasks: classification and semantic segmentation. For the first one mentioned we consider the domain shift between real objects and their sketches and for the semantic segmentation we examine experiments on real world urban street scenes transferred to two different simulations. As evaluation metrics we use the well established mean Intersection over Union (mIoU) (Jaccard, 1912) for semantic segmentation and report accuracy for classification experiments.

4.1 Classification on Real and Sketch Data

For the classification experiments we choose sketches as abstract representation of real world objects. Therefore, we consider a subset of the Sketchy dataset (Sangkloy et al., 2016). The original dataset comprises 125 categories – a subset of the ImageNet classes – and consists of 12,500 unique photographs of objects as well as 75,471 sketches drawn by different humans. Figure 2 shows examples from the dataset. A detailed description of the dataset generation process is given in (Sangkloy et al., 2016). For our experiments we limit our dataset to the 10 classes *alarm clock*, *apple*, *cat*, *chair*, *cup*, *elephant*, *hedgehog*, *horse*, *shoe* and *teapot*. As the original dataset includes sketches which are "incorrect in some way" (Sangkloy et al., 2016), we removed sketches which we could not identify as the labeled class. For validation, we randomly chose 50 sketches per class. As the number of real photos is more limited we chose 10 random photos per class for validation. This results in a remaining training set of 4,633 sketch images and 700 real photos.

For the stage a) training we use as classifier a ResNet18 (He et al., 2016) which is our downstream task network. For the I2I approach based on CycleGAN (stage b) and c)) we used the implementation of (Zhu et al., 2017) and extended it according to our method described in Sec. 3. We fix the amount of GT data used in stage c) to 70 images (10% of the data)

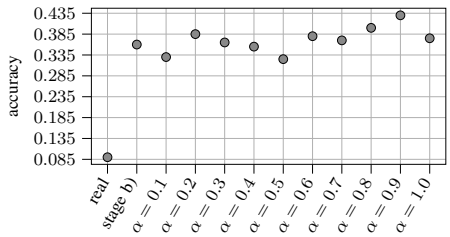

Figure 3: Classification results of the sketch expert with different types of input. The performance when an RGB photograph is given as input, is indicated by "real". CycleGAN-only performance is denoted with "stage b)" (our method when no task awareness is added). With increasing weighting factor α the generator which generates the inputs was trained with more emphasis on the task loss $\mathcal{L}_{task}$. The results are based on 70 GT (10%) images during stage c) training.

for our experiment and use the categorical cross entropy loss as task loss.

After stage a) training on sketches, the classification network f achieves an in-domain accuracy of 94.11%. When evaluating f on the real domain we see a drop to 9% accuracy. For a 10-class problem, this performance is slightly below the performance when predicting the classes uniformly at random. This confirms that the domain gap between gray scale sketches and RGB photographs is notably bigger than the domain gap considered in (Mabu et al., 2021).

Feeding f with images generated by $G_{\mathcal{R}\to\mathcal{S}}$ after stage b) training, already improves the accuracy substantially by 27 percent points (pp) reaching an absolute accuracy of 36%. When continuing training $G_{\mathcal{R}\to\mathcal{S}}$ with stage c), we achieve up to 43% accuracy of f depending on how much we weight the task loss component in Eq. (2). A quantitative comparison of the network performance with respect to different inputs is shown in Fig. 3. For values of $\alpha \leq 0.5$, we observe no clear trend compared to a CycleGAN-only training (i.e., task agnostic). Whereas, we improve the accuracy using our method when $\mathcal{L}_{task}$ dominates ($\alpha > 0.5$) the adversarial loss yielding a relative increase of up to 7 pp.

When we only use the task loss in the generator training ($\alpha = 1.0$), the performance of f drops again. This is expected as we remove the adversarial loss completely and therefore do not get notable feedback from the discriminator. We also investigate the change of the performance with respect to α in our semantic segmentation experiments where we observe this trend more clearly in the CARLA setup (cf. Fig. 8).

Exemplary, we show the results of different generators trained with different α weighting in Fig. 4.

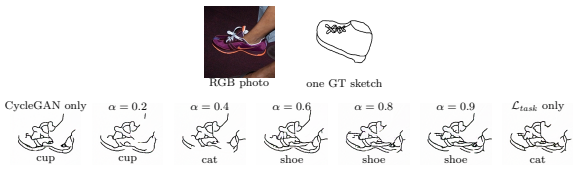

Figure 4: Qualitative results of the output of the generators trained with different weighting (α) of the downstream task loss. Top row: Real domain RGB photo (input of generator) and one of the ground truth sketches of the RGB photo. Bottom row: Generated sketches and the prediction results of the downstream task network.

The classification results of the network trained on sketches are reported underneath the images. Even though f classifies the shoe correctly for $\alpha = 0.6$, $\alpha = 0.8$ and $\alpha = 0.9$, we as human can barely see a difference between the generated images. Nevertheless, these results show that with emphasis on the task loss the generator learned to support the downstream task.

In the next paragraph we consider semantic segmentation – a distinct more challenging task – on the domain gap between real and simulated street scenes.

4.2 Semantic Segmentation on Simulated Street Scenes

a) Dataset: For the semantic segmentation task we use the established dataset Cityscapes (Cordts et al., 2016) for the real domain. The dataset contains images which were taken in multiple cities and have a resolution of $2{,}048 \times 1{,}024$ pixels. For our experiments we use the 2,975 images of the train split as well as the 500 validation images where the fine annotations are publicly available.

For the synthetic domain we conduct our experiments on two different datasets. In the first experiment we use one of the standard dataset in domain adaptation experiments: SYNTHIA-RAND-CITYSCAPES (Synthia) (Ros et al., 2016). It consists of 9,000 images with a resolution of $1{,}280 \times 760$, randomly taken in a virtual town from multiple view points. To have coincided classes in both domains, we restrict the classes to the commonly used 16 for domain adaptation which are the Cityscapes training IDs except for *train*, *truck* and *terrain* (Brehm et al., 2022). As no fixed validation set is given, we leave out the last 1400 images during training. Using the first 700 thereof for validation.

As a second setup we generated a dataset with the help of the open-source simulator CARLA (Dosovitskiy et al., 2017) which allows for the extraction of a strongly controlled dataset to realize our hypothesis of unlimited data in the synthetic domain. To showcase

this we restrict our data to town 1 of CARLA with fixed environmental settings like weather, wind etc. We generated 3,900 images for training and 1,200 images for validation with a resolution of $1{,}920 \times 1{,}080$ by randomly spawning the ego vehicle on the map. Furthermore, we spawned each time a random number of road users for a diverse scenery. Similar to Synthia not all Cityscapes training classes exist in CARLA. In particular, there is no distinction between different vehicle and pedestrian types. To this end, we fuse them into a vehicle and a pedestrian metaclass. Therefore, we consider only 13 classes: *road*, *sidewalk*, *building*, *wall*, *fence*, *pole*, *traffic light*, *traffic sign*, *vegetation*, *terrain*, *sky*, *pedestrian*, *vehicles*.

In contrast to manually labeling, the segmentation mask of CARLA is comparably fine detailed. To adapt the more coarse labeling of a human annotator and therefore generate more comparable semantic segmentation masks we smooth the label and the RGB images in a post-processing step according to the method from (Rottmann and Reese, 2022).

b) Implementation & Results – Synthetic Domain Expert: For the semantic segmentation network f, we use a DeepLabv3 with ResNet101 backbone (Chen et al., 2017) ranging under the top third of semantic segmentation models on Cityscapes with respect to the comparison of (Minaee et al., 2021). We train with Adam (Kingma and Ba, 2014) with class weighting, polynomial learning rate and from scratch without pre-training to evaluate the domain gap accurately. To range the results, we trained and evaluated f once on Cityscapes to state the oracle performance of the from-scratch-trained network independently of our experiments. This led to a mIoU of 62.74% on the validation set. This model is only used as reference and therefore we refrained from hyperparameter tuning.

For the experiments with Synthia as synthetic domain, we trained f for 3 days with a batch size of 2 due to GPU memory capacity which led to 107 epochs of training on the training dataset. During training, we crop patches of size $1{,}024 \times 512$ and flip horizontally with a chance of 50%. On the in-domain validation set we achieve a mIoU of 64.83%.

For the experiments with CARLA as synthetic domain, we trained our network for 200 epochs with random quadratic crops of size 512. The best mIoU achieved on the validation set during training is 91.89%. Benefiting from the simulation we constructed a meaningful in-domain expert with this. As the image resolution of Synthia and CARLA images, differ from the resolution of Cityscapes, a resizing is necessary. Depending on the scaling and aspect ratio the network's prediction performance differs. We chose the scaling with the best performance, which we found for $1{,}024 \times 512$. For a fair comparison we let the GANs generate the same resolution.

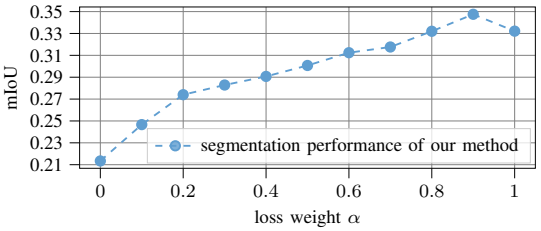

Figure 5: Influence of the task loss based on the Synthia experiment setup. The weighting represents a linear interpolation between the adversarial generator loss and the task loss (cf. Eq. (2)), resulting in the original CycleGAN implementation for $\alpha = 0$ and the pixel-wise cross entropy loss for $\alpha = 1$.

c) Implementation Details – Domain Shift: When not denoted otherwise we used 175 epochs for the stage b) training (task agnostic training) and additionally 50 epochs for the stage c) training where labeled data is available. We use the pixel-wise cross entropy as task loss. To balance the scale of the task loss $\mathcal{L}_{task}$ with respect to the adversarial generator loss $\mathcal{L}_{G_{\mathcal{R}\to\mathcal{S}}}$ we include an additional scaling factor γ. Multiplying the task loss with γ leads to more balanced loss components and therefore a better interpretability.

d) Experiment Setup and Results: First experiments were done on a mixture of labeled and unlabeled data, but we experienced an unstable training when alternating between the corresponding loss functions $\hat{\mathcal{L}}_{\text{Gen}}$ and $\mathcal{L}_{\text{Gen}}$. Splitting the generator training into two stages as described in Sec. 3, led to a more stable training and therefore better results.

As explained in Sec. 2, due to the pure domain separation we are considering, a direct comparison to other DA methods is barely meaningful. Hence, for evaluation we compare our approach with the same types of methods as done in (Mabu et al., 2021):

M1. Synthetic domain expert f fed with images generated by $G_{\mathcal{R}\to\mathcal{S}}$ based on CycleGAN-only training (stage b) only; equaling $\alpha = 0.0$)

M2. Synthetic domain expert f fed with real images without domain transformation (original Cityscapes images)

M3. Semantic segmentation network $f_{\mathcal{R}}$ trained from scratch in a supervised manner on the same amount of labeled real-world images as available at stage c).

During the GAN-training we evaluate f on the (GAN-transformed) Cityscapes validation set and report the best mIoU during training.

Table 1: Domain gap comparison of networks trained from scratch vs. ImageNet pre-trained (Dundar et al., 2018) with Cityscapes as out-of-domain (ood) evaluation.

Synthia → Cityscapes (mIoU in %)			
method	ood	oracle	gap
ImageNet pre-trained	31.8	75.6	43.8
from scratch (ours)	9.9	62.7	52.8

To analyze the impact of the different loss components in the Synthia setup, we set the scaling parameter γ empirically to 0.25, we fix the GT amount to 5% (148 labeled training images) and vary the weighting parameter α between 0 and 1. The corresponding results are shown in Fig. 5. We see the positive impact of the task awareness in the growing mIoU values. Using a weighting of $\alpha = 0.9$ for $\mathcal{L}_{task}$, we achieve 34.75% mIoU which is a performance increase of 13.41 pp compared to M1 (task agnostic training).

In Fig. 6 we show for one example the differently generated images as well as their predictions by the synthetic expert. The column "Cityscapes" in Fig. 6 illustrates the low prediction performance of a synthetic domain expert when never having seen real-world images (M2). The network's performance drops to roughly 10% when real world images are used as input for the domain expert. Here we see a significant difference to results reported by other domain adaptation methods which use ImageNet pre-trained networks, e.g.,(Dundar et al., 2018). The domain gap is summarized in Tab. 1 where we compare ImageNet pre-trained network performance to ours evaluated on Cityscapes. We state out-of-domain performance (i.e., trained on Synthia; second column), oracle performance (i.e., trained on the full Cityscapes training dataset; third column), and the domain gap between them, measured as difference in performance (last column). The results indicate that the ImageNet pre-training already induces a bias towards the real domain distorting a pure domain separation which should be avoided when analyzing domain gaps. Based on our training-from-scratch setup, using task agnostic generated images (M1) improves already significantly the performance (11.44 pp) whereas our approach (task aware GAN) can lead to a relative improvement of up to 24.85 pp when 5% GT images are available.

Moreover, we analyze the capacity of the method based on the amount of GT available. Therefore, we fix $\alpha = 0.8$ and vary the GT amount for the stage c) training. We randomly sample images from the Cityscapes training dataset for each percentage but fix the set of labeled data for the experiments with CARLA and "Cityscapes-only" training (M3) for the sake of comparison. Results are shown in Fig. 7 (blue curve). The dotted horizontal line is the mIoU achieved by f when exclusively feeding images generated by the task agnostic GAN after finishing stage b) training. For a fair comparison we trained the task agnostic GAN for another 125 epochs which results in a better mIoU of 21.34% which we use as result for GT $= 0$ (equaling $\alpha = 0.0$). For our experiment we compare 0.5% (14 images), 1% (29 images), 2% (59 images), 5% (148 images) and 10% (297 images) of GT data for the stage c) training. Triggering the task awareness with only 14 images already improves the network accuracy by 6.75 pp. The results show that training $G_{\mathcal{R}\to\mathcal{S}}$ with the task loss on negligible few GT data, improves the network's understanding of the scene without retraining the network itself.

Additionally, we compare our method with results of $f_{\mathcal{R}}$ (M3). Having no labeled data, a supervised method can barely learn anything. Therefore, we set the value to the same as for 0.5% GT which most likely overestimates the performance. The results are visualized by the orange curve in Fig. 7. The results show that our method outperforms M3 by a distinct margin when only a few labels are available. However, when we have access to more than 297 (10%) fine labeled images of Cityscapes, a direct supervised training should be taken into consideration.

For the CARLA experiments we set $\gamma = 1$, as the losses are already in the same scale. We repeat the three experiments on our CARLA dataset. The results of varying α are visualized by the blue curve in Fig. 8. Also, on the CARLA dataset our method shows a notable improvement over CycleGAN-only training (M1; $\alpha = 0$) when choosing a balanced weighting between the adversarial and the task loss. These experiments confirm that the task awareness improves the performance, but the task loss should be used in addition and not as a stand-alone concept.

The results of the GT amount variation are shown in Fig. 9 for $\alpha = 0.4$ where the blue curve represents the best mIoU results achieved with our method and the orange curve shows the results of $f_{\mathcal{R}}$ (M3) given different amount of GT. As before our method outperforms M1 as well as M3 when less than 5% GT is available. Above that, the supervised method is superior. The distinct improved semantic segmentation of the street scene can also be seen in the qualitative results shown in Fig. 10 (bottom row). As in the previous experiment visual differences recognizable by humans of the generated images with CycleGAN (top row mid) and our method (top row right) are limited. Furthermore, we see again the low performance caused by the domain gap when feeding real images to our from-scratch-trained synthetic expert f. On the untranslated images (M3), f yields an mIoU of 9%. Hence, the observed results achieved by our method

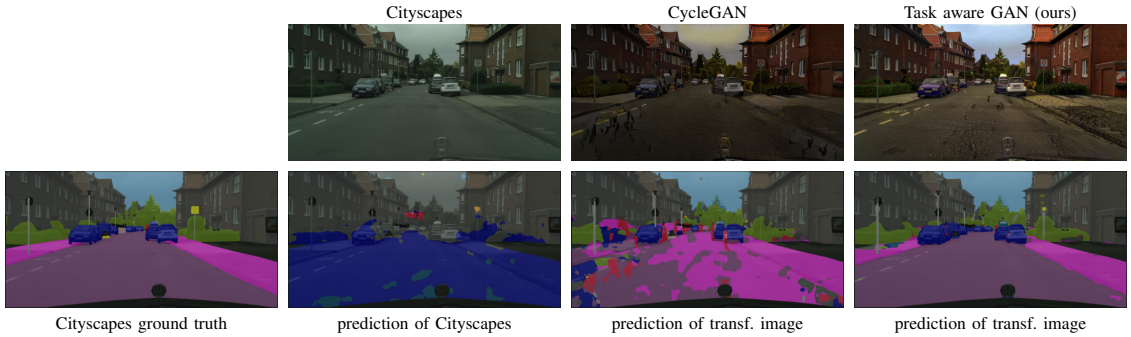

Figure 6: Comparison of prediction results of an untranslated Cityscapes image (left), task agnostic I2I (mid) and our approach (right) based on a semantic segmentation network trained on Synthia.

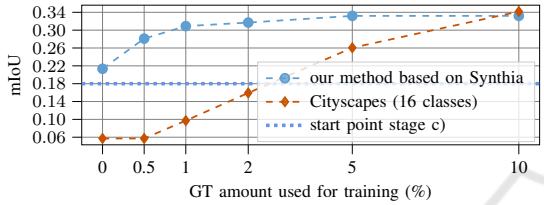

Figure 7: Performance comparison of our method based on Synthia setup with different amount of ground truth (blue) and a from scratch supervised training on Cityscapes with the same amount of data (orange).

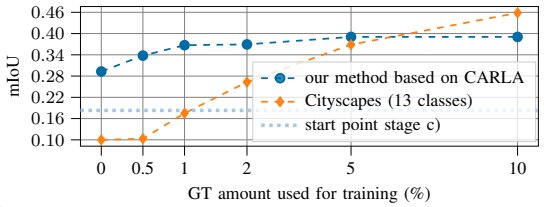

Figure 9: Performance comparison of our method based on CARLA setup with different amount of GT (blue) and of $f_{\mathcal{R}}$ which is trained from scratch in a supervised manner on Cityscapes with the same amount of data (orange).

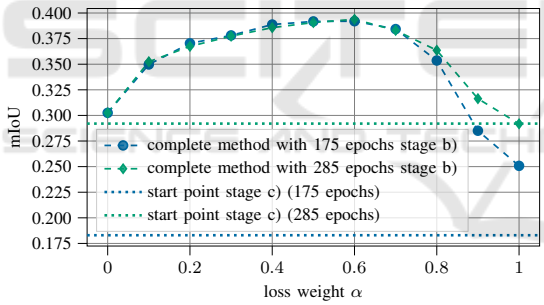

Figure 8: Influence of the task loss for the CARLA experiment setup. The weighting represents a linear interpolation between the adversarial generator loss and the task loss Eq. (2). Results after stage c) based on a 175 epochs unsupervised GAN-training are shown in blue. The green graph shows the method performance when trained with a longer amount of stage b) steps.

demonstrate a significant reduction of the domain gap via generating more downstream task relevant visual features.

Lastly, we consider a longer stage b) training to find out whether a longer training further improves the results. We train in total 285 epochs in stage b) and show the results of the complete method with 5% GT in Fig. 8 visualized by the green curve. The experiments reveal that a moderate number of epochs for stage b) is already enough for a good initialization of stage c). Although we start the stage c) training with a higher mIoU (dotted lines) when trained with stage b) for more steps, the experiments show that we achieve nearly the same absolute mIoU values.

5 CONCLUSION AND OUTLOOK

In this paper, we presented a modular semi-supervised domain adaptation method based on CycleGAN where we guide the generator of the image-to-image approach towards downstream task awareness without retraining the downstream task network itself. In our experiments we showed on a “real to sketch” domain adaptation classification task that the method can cope with large domain gaps. Furthermore, we showed that our method can be applied to more complex downstream tasks like semantic segmentation yielding significant improvements compared to a pure I2I approach and from scratch training when a limited amount of GT is available. Besides, we analyzed the impact of the task awareness and the GT. Contrary to the common practice, all results were produced based on a non-biased domain gap. To this end, we trained all components from scratch. Our achieved results suggest that the commonly used ImageNet pretrained backbone already incorporates real world domain information and therefore distorts the gap analysis. Additionally, we showed that we can achieve very

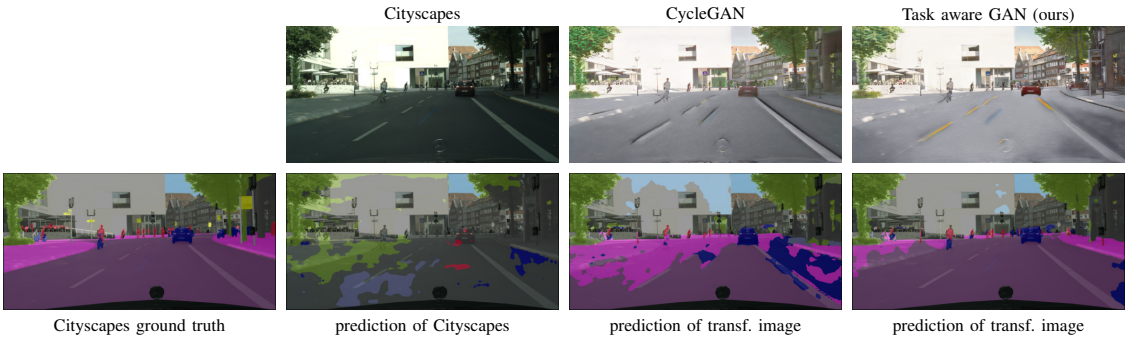

Figure 10: Comparison of prediction results of an untranslated Cityscapes image (left), task agnostic style transfer (mid) and our approach (right) based on a semantic segmentation network trained on our CARLA dataset.

strong models when considering abstract representations (like sketches or modifiable simulations).

For future work, we are interested in elaborating more on the (intermediate) abstract representation, e.g., investigating if the robustness of the model can benefit from it. Additionally, it has potential to help us better understand which visual features are important for a downstream task network. Generating more informative images for a downstream task network might give insights into the network behavior and help generate datasets which are cut down to the most important aspects of the scene for a neural network which is not necessarily what a human would describe as meaningful. Moreover, an uncertainty based data selection strategy for stage c) training, could further improve the method. In addition, the method could be combined with self-training as these models need a good initialization to generate reasonable pseudo labels (Mei et al., 2020). Nevertheless, when training the downstream task network completely from scratch, we have shown that the network performance is questionably low. Therefore, our method can be seen as complementary to the self-training approaches to ensure a reasonable prediction of the network in early stages.

ACKNOWLEDGMENT

This work is funded by the German Federal Ministry for Economic Affairs and Climate Action within the project “KI Delta Learning”, grant no. 19A19013Q. We thank the consortium for the successful cooperation. Moreover, the authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS (Jülich Supercomputing Centre, 2019) at Jülich Supercomputing Centre (JSC). Furthermore, we thank Hannah Lörcks for assisting in setting up the classification experiments.

REFERENCES

Brehm, S., Scherer, S., and Lienhart, R. (2022). Semantically consistent image-to-image translation for unsupervised domain adaptation. *ICAART*, 2:131–141.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv:1706.05587*.

Chen, Y., Li, W., Chen, X., and Van Gool, L. (2019). Learning Semantic Segmentation From Synthetic Data: A Geometrically Guided Input-Output Adaptation Approach. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1841–1850.

Chen, Y., Ouyang, X., Zhu, K., and Agam, G. (2021). Semi-supervised Domain Adaptation for Semantic Segmentation. *arXiv:2110.10639*.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223.

Csurka, G. (2017). Domain Adaptation for Visual Applications: A Comprehensive Survey. *arXiv:1702.05374*.

Dai, J., He, K., and Sun, J. (2015). Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1635–1643.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An Open Urban Driving Simulator. In *Conference on Robot Learning*, pages 1–16. PMLR.

Dundar, A., Liu, M.-Y., Wang, T.-C., Zedlewski, J., and

Kautz, J. (2018). Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation. *arXiv:1807.09384*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Cambridge, MA, USA. MIT Press.

Grigoryev, T., Voynov, A., and Babenko, A. (2022). When, Why, and Which Pretrained GANs Are Useful? *arXiv:2202.08937*.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017). Improved Training of Wasserstein GANs. *arXiv:1704.00028*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *International Conference on Machine Learning*, pages 1989–1998. PMLR.

Hoffman, J., Wang, D., Yu, F., and Darrell, T. (2016). FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation. *arXiv:1612.02649*.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5967–5976.

Jaccard, P. (1912). The Distribution of the Flora in the Alpine Zone. *New Phytologist*, 11(2):37–50.

Jiang, P., Wu, A., Han, Y., Shao, Y., Qi, M., and Li, B. (2020). Bidirectional Adversarial Training for Semi-Supervised Domain Adaptation. In *Twenty-Ninth International Joint Conference on Artificial Intelligence*, volume 1, pages 934–940.

Jülich Supercomputing Centre (2019). JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre. *Journal of large-scale research facilities*, 5(A135).

Kang, G., Wei, Y., Yang, Y., Zhuang, Y., and Hauptmann, A. (2020). Pixel-Level Cycle Association: A New Perspective for Domain Adaptive Semantic Segmentation. In *Advances in Neural Information Processing Systems*, volume 33, pages 3569–3580. Curran Associates, Inc.

Kim, M. and Byun, H. (2020). Learning Texture Invariant Representation for Domain Adaptation of Semantic Segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12975–12984.

Kim, T. and Kim, C. (2020). Attract, Perturb, and Explore: Learning a Feature Alignment Network for Semi-supervised Domain Adaptation. In *European conference on computer vision – ECCV 2020*, pages 591–607, Cham. Springer International Publishing.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022.

Mabu, S., Miyake, M., Kuremoto, T., and Kido, S. (2021). Semi-supervised CycleGAN for domain transformation of chest CT images and its application to opacity classification of diffuse lung diseases. *International Journal of Computer Assisted Radiology and Surgery*, 16(11):1925–1935.

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Paul Smolley, S. (2017). Least Squares Generative Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802.

Mei, K., Zhu, C., Zou, J., and Zhang, S. (2020). Instance Adaptive Self-Training for Unsupervised Domain Adaptation. *arXiv:2008.12197*.

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., and Terzopoulos, D. (2021). Image Segmentation Using Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Pang, Y., Lin, J., Qin, T., and Chen, Z. (2021). Image-to-Image Translation: Methods and Applications. *IEEE Transactions on Multimedia*.

Ros, G., Sellart, L., Materzynska, J., Vázquez, D., and López, A. M. (2016). The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243.

Rottmann, M. and Reese, M. (2022). Automated Detection of Label Errors in Semantic Segmentation Datasets via Deep Learning and Uncertainty Quantification. *arXiv:2207.06104*.

Saito, K., Kim, D., Sclaroff, S., Darrell, T., and Saenko, K. (2019). Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058.

Sangkloy, P., Burnell, N., Ham, C., and Hays, J. (2016). The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics*, 35(4):119:1–119:12.

Shukla, S., Van Gool, L., and Timofte, R. (2019). Extremely Weak Supervised Image-to-Image Translation for Semantic Segmentation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3368–3377. ISSN: 2473-9944.

Tao, A., Sapra, K., and Catanzaro, B. (2020). Hierarchical multi-scale attention for semantic segmentation. *arXiv:2005.10821*.

Toldo, M., Maracani, A., Michieli, U., and Zanuttigh, P. (2020). Unsupervised Domain Adaptation in Semantic Segmentation: A Review. *Technologies*, 8(2):35.

Tsai, Y.-H., Hung, W.-C., Schulter, S., Sohn, K., Yang, M.-H., and Chandraker, M. (2018). Learning to Adapt Structured Output Space for Semantic Segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7472–7481.

van Engelen, J. E. and Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

Wang, Y., Wu, C., Herranz, L., van de Weijer, J., Gonzalez-Garcia, A., and Raducanu, B. (2018). Transferring GANs: generating images from limited data. In *Proceedings of the European Conference on Computer Vision*, pages 218–234.

Wang, Z., Wei, Y., Feris, R., Xiong, J., Hwu, W.-M., Huang, T. S., and Shi, H. (2020). Alleviating Semantic-Level Shift: A Semi-Supervised Domain Adaptation Method for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 936–937.

Wrenninge, M. and Unger, J. (2018). Synscapes: A Photorealistic Synthetic Dataset for Street Scene Parsing. *arXiv:1810.08705*.

Wu, Z., Han, X., Lin, Y.-L., Uzunbas, M. G., Goldstein, T., Lim, S. N., and Davis, L. S. (2018). Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision*, pages 518–534.

Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., and Wen, F. (2021). Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12414–12424.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision (ICCV)*.