



Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational  
Mathematics (IMACM)

Preprint BUW-IMACM 22/26

A. Mütze und M. Rottmann und H. Gottschalk

**Semi-supervised domain adaptation with  
CycleGAN guided by a downstream task loss**

January 4, 2023

<http://www.imacm.uni-wuppertal.de>

# Semi-supervised domain adaptation with CycleGAN guided by a downstream task loss

Annika Mütze  
IZMD & School of Mathematics  
and Natural Sciences  
University of Wuppertal  
Wuppertal, Germany  
muetze@uni-wuppertal.de

Matthias Rottmann  
IZMD & School of Mathematics  
and Natural Sciences  
University of Wuppertal  
Wuppertal, Germany  
rothmann@uni-wuppertal.de

Hanno Gottschalk  
IZMD & School of Mathematics  
and Natural Sciences  
University of Wuppertal  
Wuppertal, Germany  
hanno.gottschalk@uni-wuppertal.de

**Abstract**—Domain adaptation is of huge interest as labeling is an expensive and error-prone task, especially when labels are needed on pixel-level like in semantic segmentation. Therefore, one would like to be able to train neural networks on synthetic domains, where data is abundant and labels are precise. However, these models often perform poorly on out-of-domain images. To mitigate the shift in the input, image-to-image approaches can be used. Nevertheless, standard image-to-image approaches that bridge the domain of deployment with the synthetic training domain do not focus on the downstream task but only on the visual inspection level. We therefore propose a “task aware” version of a GAN in an image-to-image domain adaptation approach. With the help of a small amount of labeled ground truth data, we guide the image-to-image translation to a more suitable input image for a semantic segmentation network trained on synthetic data (synthetic-domain expert). The main contributions of this work are 1) a modular semi-supervised domain adaptation method for semantic segmentation by training a downstream task aware CycleGAN while refraining from adapting the synthetic semantic segmentation expert 2) the demonstration that the method is applicable to complex domain adaptation tasks and 3) a less biased domain gap analysis by using from scratch trained networks. We evaluate our method on a classification task as well as on semantic segmentation. Our experiments demonstrate that our method outperforms CycleGAN – a standard image-to-image approach – by 7 percent points in accuracy in a classification task using only 70 (10%) ground truth images. For semantic segmentation we can show an improvement of about 4 to 7 percent points in mean Intersection over union on the Cityscapes evaluation dataset with only 14 ground truth images during training.

**Index Terms**—Generative Adversarial Networks, Domain adaptation, semantic segmentation, domain shift, semi-supervised learning, regularization, image-to-image translation

## I. INTRODUCTION

For automatically understanding complex visual scenes from RGB images, semantic segmentation – pixel-wise classification of the image – is a common but challenging method. The state-of-the-art results are achieved by deep neural networks (DNNs) [1]–[5]. These models need plenty of labeled images to generalize well on unseen scenes. However, a manual label process on pixel level detail is time and cost consuming and usually error-prone [6], [7]. To reduce the labeling cost, methods which make use of weak or coarse labels like bounding boxes for the semantic segmentation

task were proposed [8], [9]. Other approaches like semi-supervised learning [10] reduce the amount of labeled ground truth needed. Thereby a large amount of unlabeled data is available but only a small subset has detailed labels [11]–[13]. With active learning one tries to query only the most informative images from a pool of unlabeled data to label and therefore reduce the label amount [14], [15]. The combination of both ideas was studied by Rottmann et al. [16] for classification tasks. However, these methods are limited to scenarios captured in the real world. Furthermore, the annotation cost of weak labels still might be intractable [17]. On the other hand, in the recent years simulations of urban street scenes were significantly improved [18]–[20]. The advantage of synthetic data is that images generated by a computer simulation often come with labels for the semantic content for free. To enlarge the real-world labeled dataset, synthetic data can be incorporated [20]. 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, switching the domain confuses the network [21]. 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/images from simulation) [20] or vice versa. Domain adaptation (DA) is used to mitigate the so-called domain shift between one domain and another [22]. It aims at improving the model’s performance on a target domain by transferring knowledge learned from a labeled source domain. DA has become an active area of research in the context of deep learning [23] ranging from adaptation on feature level [17], [24], [25], adaptation on input level [21], [26]–[30], self-training [31]–[34], a combination thereof [33] to semi-supervised approaches [35]. 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) [23]. Adapting on input level to the style of the target domain disregarding the downstream task at hand but preserving the overall scene was firstly motivated by Hertzmann et al. [36] and is referred to image-to-image translation (I2I) [37].

Taking advantage of the synthetic domain we train a down-

stream task expert therein. We then shift the out-of-domain input (real world) closer to the synthetic domain via a semi-supervised image-to-image approach for mitigating the domain gap to the more abstract domain (e.g. simulation). We thereby refrain from changing the expert which leads to a modular DA method. Based on CycleGAN [37], we combine this GAN-based image-to-image method with a downstream task awareness in a second stage to adapt to the needs of the downstream task network. With the help of a very small contingent of ground truth (GT) in the real domain, we introduce the downstream task awareness of the generator.

Our main contributions:

- we present a novel modular semi-supervised domain adaptation method for semantic segmentation under domain shift by guiding the generator of an image-to-image domain adaptation approach to a semantic segmentation task awareness. Thereby we refrain completely from retraining our downstream task network.
- 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 modular 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 abstract representation of real world data thereby has the potential to lead to more powerful DNNs.

The remainder of this paper is organized as follows. In section II we review existing approaches in the context of domain adaptation, particularly semi-supervised domain adaptation and image-to-image approaches. It follows a detailed description of our method in section III. We then evaluate our method on two different tasks and three different datasets in section IV, showing considerable improvements with only a small number of GT data. A classification task on a real-to-sketch domain shift demonstrates the capacity of the method in terms of domain shift complexity and our detailed experiments on semantic segmentation of urban street scenes illustrate the potential of the method on more complex downstream tasks. For evaluation metrics we use the well established mean Intersection over Union [38], [39] for semantic segmentation and report accuracy for classification experiments. Finally, we conclude and discuss our results as well as giving an outlook to future work in section V.

## II. RELATED WORK

Our work bases on two main concepts: domain adaptation via image-to-image translation and semi-supervised learning in

the context of GANs and domain adaptation. In the following we will review the most recent literature on these topics.

Since its first motivation by Hertzmann et al. [36], image-to-image translation is nowadays dominated by generative adversarial networks (GANs) [40] conditioned on an image and has various applications in computer vision and computer graphics [41]. The concept of GANs is to model the data generating distribution implicitly by training two adversary networks, a generator generating images and a discriminator predicting the input being generated or a 'real' training sample. Formerly, paired data was needed to adapt to the new style [42]. But as paired data is hard to get or generate, unsupervised methods were developed and published like CycleGAN [37], where a cycle consistency loss leads to a consistent mapping of an image from one domain to the other.

Image-to-image in general can be evaluated by different methods [41], e.g., image structural similarity index (SSIM) [43], Fréchet inception distance (FID) [44] or LPIPS [45]. For I2I in the context of domain adaptation, as a fixed task is considered, the downstream task performance is used as performance measure for the image translation. Nevertheless, our generator output could also be evaluated with other metrics.

Our idea to use I2I for domain adaptation while generating more semantically consistent images with the help of the downstream task loss is also tackled, for example, in Cycada [26], SUIT [46], the work of Guo et al. [47] and Brehm et al. [30] but in an unsupervised manner.

In contrast to unsupervised approaches, SSDA methods can make use of a few labeled images in the target domain. There are several SSDA approaches in the context of classification [48]–[52] and Singh et al. [53] proposes an active learning-based method to determine which data should be labeled in the target domain.

SSDA for semantic segmentation tasks is considered less. For example, Wang et al. [54] propose to adapt simultaneously on a semantic and global level using adversarial training. Chen et al. [35] propose 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 domain adaptation by [50].

Training in a two stage manner, starting with an unsupervised training (also referred to 'pre-training') to initialize good network parameters and then retrain in a second stage with a small amount of labeled data is a common semi-supervised learning technique [10]. We transfer this concept to GANs of an image-to-image method. In the context of general GANs, using pre-training is not new. Wang et al. [55] for example, analyzed pre-training for Wasserstein GANs with gradient penalty [56] in the context of image generation when limited data is available. In the work of Zhao et al. [57], the usefulness of pre-training GANs is shown based on the GP-GAN [58]. Grigoryev et al. [59] gives an overview over when, why and which pre-trained GANs are useful. We follow their suggestion of pre-training both the generator and the discriminator but refrain from the suggestion of using ImageNet [60] pre-trained GANs to not distort the domain gap analysis.

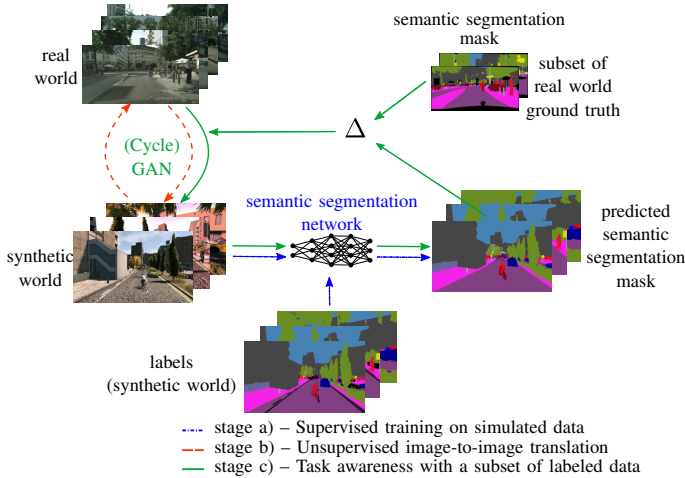


Fig. 1. Concept of our method: **Stage a)** – Training of a downstream task model (e.g. semantic segmentation network) on the abstract/synthetic domain. **Stage b)** – Training a CycleGAN based on unpaired data to transfer real data into the synthetic domain. **Stage c)** – We freeze the downstream task 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.

For the downstream task model we also train completely from scratch for a pure domain separation and keep the task network fixed after it is trained once which makes it reusable when switching to a different domain. This is in contrast to the other above-mentioned UDA approaches which use ImageNet pre-trained downstream task networks and adapt the downstream task network to mitigate the domain gap. In general, UDA methods lack important information of the target domain compared to their supervised trained counterparts [35]. Furthermore, pure image-to-image methods are task agnostic and therefore may lack semantic consistency [23], [47]. For this reason we propose a SSDA method with a task aware image-to-image component. Independently of our approach, Mabu et al. [52] published recently a similar approach based on classification in a medical context with domain gaps primarily in image intensity and contrast. On the contrary, our method aims 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 addition, we consider image-to-image for real images to the synthetic domain to retain the benefits of a synthetic expert, whereas the other approaches consider the opposite direction.

In the following we explain our method in more detail.

### III. 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 labeled training set  $\mathcal{T}_{\mathcal{S}} = \{(x_i^s, y_i^s) \in \mathcal{S} \times \mathcal{Y} : i = 1, \dots, N\}$ , 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” [29], we do not use ImageNet pre-trained weights for the downstream task 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 model fixed.

*b) Unsupervised Image-to-Image Translation:* To mitigate the domain gap between real and synthetic data, we build on the established image-to-image method CycleGAN [37] – a GAN approach which can deal with unpaired data by enforcing a cycle consistency between two generators, one for each domain ( $G_{\mathcal{S} \rightarrow \mathcal{R}}$ ,  $G_{\mathcal{R} \rightarrow \mathcal{S}}$ ). The domain discriminators to classify whether the sample is generated or an in-domain sample are denoted with  $D_{\mathcal{S}}$  and  $D_{\mathcal{R}}$ . Let  $x^r \sim p_{\text{data}}$  denote the data distribution in the real domain, then we can define the four loss components of the image-to-image translation of CycleGAN:

- 1) *adversarial loss for the generators*

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

For  $\mathcal{L}_{G_{\mathcal{S} \rightarrow \mathcal{R}}}$  substitute  $\mathcal{R}$  resp.  $r$  with  $\mathcal{S}$  resp.  $s$  in eq. (1).

- 2) *cycle consistency loss:*

$$\begin{aligned} \mathcal{L}_{\text{cyc}}(G_{\mathcal{R} \rightarrow \mathcal{S}}, G_{\mathcal{S} \rightarrow \mathcal{R}}) &= \mathbb{E}_{x^r \sim p_{\text{data}}(x^r)} \left[ \left\| G_{\mathcal{S} \rightarrow \mathcal{R}} \left( G_{\mathcal{R} \rightarrow \mathcal{S}}(x^r) \right) - x^r \right\|_1 \right] \quad (2) \\ &+ \mathbb{E}_{x^s \sim p_{\text{data}}(x^s)} \left[ \left\| G_{\mathcal{R} \rightarrow \mathcal{S}} \left( G_{\mathcal{S} \rightarrow \mathcal{R}}(x^s) \right) - x^s \right\|_1 \right]. \end{aligned}$$

- 3) *identity preserving loss*

$$\begin{aligned} \mathcal{L}_{\text{identity}}(G_{\mathcal{R} \rightarrow \mathcal{S}}, G_{\mathcal{S} \rightarrow \mathcal{R}}) &= \mathbb{E}_{x^s \sim p_{\text{data}}(x^s)} \left[ \left\| G_{\mathcal{R} \rightarrow \mathcal{S}}(x^s) - x^s \right\|_1 \right] \quad (3) \\ &+ \mathbb{E}_{x^r \sim p_{\text{data}}(x^r)} \left[ \left\| G_{\mathcal{S} \rightarrow \mathcal{R}}(x^r) - x^r \right\|_1 \right] \end{aligned}$$

- 4) *adversarial loss for the discriminators*

$$\begin{aligned} \mathcal{L}_{D_{\mathcal{S}}} &= \mathbb{E}_{x^s \sim p_{\text{data}}(x^s)} \left[ \left( D_{\mathcal{S}}(x^s) - 1 \right)^2 \right] \\ &+ \mathbb{E}_{x^r \sim p_{\text{data}}(x^r)} \left[ \left( D_{\mathcal{S}} \left( G_{\mathcal{R} \rightarrow \mathcal{S}}(x^r) \right) \right)^2 \right] \quad (4) \end{aligned}$$

For  $\mathcal{L}_{D_{\mathcal{R}}}$  substitute  $\mathcal{R}$  resp.  $r$  with  $\mathcal{S}$  resp.  $s$  in eq. (4). As adversarial losses  $\mathcal{L}_{G_*}$  we use the least-squares loss proposed by Mao et al. [61] which has been used in the implementation of CycleGAN and leads to a more stable

training according to Zhu et al. [37]. With the help of the adversarial losses [61] (eq. (1) and eq. (4)), the general aim of matching the target data distribution with the generated data is pursued. Furthermore, with the help of the cycle consistency loss the method enforces an approximate inverse mapping between the two domains. In addition, the identity loss is used to prevent the generator to modify an image if it already lies within the desired domain. The overall generator loss is defined as the weighted sum of the losses eqs. (1) to (3):

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

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  $\mathcal{R}$ , where  $N_L$  denotes the number of labeled samples.

We achieve the downstream task awareness by extending the adversarial loss eq. (1) for the generator  $G_{\mathcal{R} \rightarrow \mathcal{S}}$  based on the loss of the downstream task network  $f$ . Given a problem with  $C$  different classes and an input  $x$ , we calculate the loss between the prediction  $f(x) = (f(x)_1, \dots, f(x)_C)$  and the corresponding ground truth class label  $y$  with the categorical cross entropy loss (also known as softmax loss):

$$\begin{aligned} \mathcal{L}_{\text{task}}(x, y) &= \mathcal{L}_{\text{CE}}(f(x), y) \\ &= -\log \left( \frac{\exp(f(x)_y)}{\sum_{c=1}^C \exp(f(x)_c)} \right), \end{aligned} \quad (5)$$

where  $f(x)_y$  is the value assigned to the true class  $y$  by the neural network. Depending on the task we additionally average the result. For example, in the case of semantic segmentation we calculate the mean over all pixel. When using a batch size greater than 1, the mean is additionally computed with respect to the batch size.

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

$$\begin{aligned} \tilde{\mathcal{L}}_{G_{\mathcal{R} \rightarrow \mathcal{S}}}(t_i) &= (1 - \alpha) \underbrace{\left( D_s(G_{\mathcal{R} \rightarrow \mathcal{S}}(x_i^r)) - 1 \right)^2}_{\text{adversarial loss as in eq. (1)}} \\ &\quad + \alpha \underbrace{\left( \mathcal{L}_{\text{CE}}(f(G_{\mathcal{R} \rightarrow \mathcal{S}}(x_i^r)), y_i^r) \right)}_{\text{task loss}}. \end{aligned} \quad (6)$$

We use  $\alpha$  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:

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

The discriminator losses (cf. eq. (4)) are kept identically.



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

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  $G_{\mathcal{R} \rightarrow \mathcal{S}}$  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  $G_{\mathcal{R} \rightarrow \mathcal{S}}$  could also be evaluated with respect to other metrics described in [41].

#### IV. NUMERICAL EXPERIMENTS

We evaluate our method on two different downstream tasks: classification and semantic segmentation. For the first mentioned we consider the domain shift between real objects and their sketches and for the semantic segmentation we examine experiments on real world urban scenes transferred to two different simulations.

##### A. 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 [62]. 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. Per photograph, five participants were asked to draw a corresponding sketch to capture the diversity of sketching and drawing skills. A detailed description of the dataset generation process is given in [62]. Fig. 2 shows examples from the dataset. 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” [62], we removed sketches which we could not identify as the labeled class. For validation, we randomly chose 50 sketches per class for validation for the sketch data. 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 a ResNet18 [63] as classifier which is our downstream task network. For the I2I based on CycleGAN (stage b) and c)) we used the implementation of [37] and extended it according to our method described in section III. We fix the amount of GT data used in the stage c) training to 70 images (10% of the data) for our experiment and use the categorical cross entropy loss as task loss. We

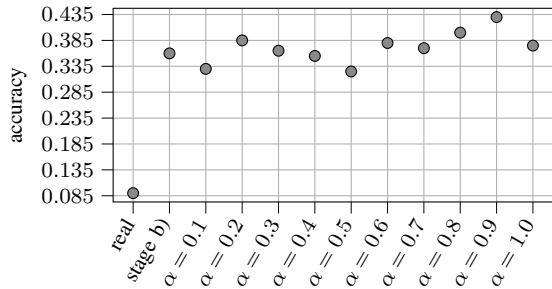


Fig. 3. Classification results of the sketch expert network with different types of input images. The performance when an RGB photograph is given as input, is indicated by “real”. CycleGAN-only performance is depicted with “stage b)” (our method when no task awareness is added). With increasing weighting factor  $\alpha$  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.

measure the performance of the classifier in terms of accuracy on a validation set  $\mathcal{V} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i = 1, \dots, N_v\}$  where  $\mathcal{X}$  is a subset of  $\mathcal{S}$  or  $\mathcal{R}$ .

$$acc_f(\mathcal{V}) = \frac{|\{(x, y) \in \mathcal{V} \mid f(x) = y\}|}{|\{(x, y) \in \mathcal{V}\}|}.$$

After stage a) training on sketch images, the classification network  $f$  achieves an in-domain accuracy of 94.11% so that we can call it a sketch expert. When evaluating the performance of the network on the real domain we see a drop to 9% accuracy. For a 10-class problem, this performance is slightly below when predicting the classes uniformly at random. This confirms that the domain gap between black and white sketch images and RGB photographs is notably bigger than the domain gap considered in [52].

Feeding  $f$  with images generated by the generator  $G_{\mathcal{R} \rightarrow \mathcal{S}}$  after stage b) training, already improves the accuracy substantially by 27 percent points (pp) reaching an absolute accuracy of 36%. When continuing training the generator with stage c), and feeding the images generated by that generator to the downstream task network  $f$ , we achieve up to 43% accuracy depending on how much we weight the task loss component in eq. (6). A qualitative 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 (only stage b), 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 of the discriminator. We will also investigate the change of the performance with respect to  $\alpha$  in our semantic segmentation experiments (see section IV-B) where we observe this trend more clearly in the CARLA setup (cf. fig. 9).

Exemplary, we show the results of different generators trained with different  $\alpha$  weighting in fig. 4. The classification

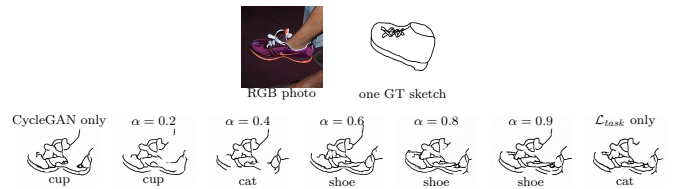


Fig. 4. Quantitative results of the output of the generators trained with different weighting ( $\alpha$ ) of the downstream task loss. Top row: Real domain RGB photo (input of generator) and one of the ground truth sketches of the RGB photo. Bottom row: Generated sketches and the prediction results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

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. It seems the images generated by a generator trained with higher  $\alpha$  values might capture more details. 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 a semantic segmentation task – a distinct more challenging task – on the potentially smaller domain gap between real and simulated street scenes.

#### B. Semantic segmentation on urban and simulated street scenes

a) *Dataset*: For the semantic segmentation task we focus on urban street scenes. We use the established dataset Cityscapes [6] 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) [64], a dataset based on the Unity simulation engine. 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* [30]. As no fixed validation set is given, we leave out the last 1400 images during training. Using the first 700 for validation. This leads to camera view points for evaluation that are independent of those used for training.

As a second setup we generated a dataset with the help of the open-source simulator CARLA [19] which bases on the Unreal Engine. The simulator allows for the extraction of a strongly controlled dataset to realize our hypothesis of unlimited data in the synthetic domain. To showcase this we use only one town map of CARLA (town 1) 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 a random number of road users for a diverse scenery. Similar to Synthia not all Cityscapes training classes exist in CARLA. Especially there is no distinction between different vehicle and pedestrian types. To this end, we fuse them into a vehicle and pedestrian meta-class. 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 very 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 [7].

*b) Implementation details & results – synthetic domain expert:* For the semantic segmentation network  $f$ , we use the PyTorch [65] implementation<sup>1</sup> of a DeepLabv3 with ResNet101 backbone [66] ranging under the top third of semantic segmentation models on Cityscapes with respect to the comparison of Minaee et al. [67]. We train the semantic segmentation network 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 Cityscapes validation set. This model is only used as reference and therefore we refrained from hyperparameter tuning.

For both experiments, we train using Adam [68] with class weighting and normalize the images according to the pixel-wise mean and variance of the training data. During training, we used a horizontal flip with a probability of 50% and a polynomial learning rate (LR) scheduler. 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  $1,024 \times 512$ .

To measure the performance of our method we use the established mean intersection over Union (mIoU) metric which measures the relation between the intersection and the union of the prediction and ground truth as follows:

$$IoU(c) = \frac{TP_c}{TP_c + FP_c + FN_c}, \quad c = 1, \dots, C$$

$$mIoU = \frac{1}{C} \sum_{c=1}^C IoU(c)$$

where  $TP_c$  denotes the amount of pixels correctly classified as class  $c$ ,  $FP_c$  the amount of pixels which were falsely predicted as class  $c$  and  $FN_c$  denotes the amount of omitted predictions, so pixels which are of class  $c$  but were not classified as such. 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

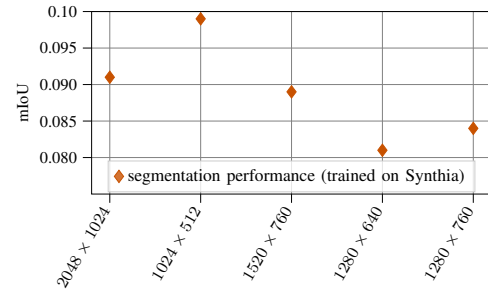


Fig. 5. Prediction performance of the Synthia expert given differently scaled Cityscapes images as input. The resolution of the scaled images is given on the x-axis. Scaling needs to be considered as the aspect ratio and the resolution differ between training (synthetic domain) and inference on Cityscapes images (real domain).

crops of size 512. We yield the best mIoU of 91.89% on the validation set. Benefiting of the simulation we constructed a meaningful in-domain expert.

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  $1,024 \times 512$ . A comparison of the network’s input resolution (x-axis) and network’s performance measured by mIoU (y-axis) for the Synthia expert is shown in fig. 5. For a fair comparison we let the GANs generate the same resolution.

*c) Implementation details – domain shift:* For the I2I (stage b)), we use the same architecture as described in the CycleGAN publication [37]. 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  $\mathcal{L}_{G_{R \rightarrow S}}$  we include an additional scaling factor  $\gamma$ . Multiplying the task loss with  $\gamma$  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  $\tilde{\mathcal{L}}_{Gen}$  and  $\mathcal{L}_{Gen}$ . Splitting the generator training into two stages as described in section III, led to a more stable training and therefore better results.

To evaluate our approach we set up three experiments mainly differing in the stage c) training:

- 1) Variation of  $\mathcal{L}_{task}$  weighting (cf. figs. 6 and 9)
- 2) Evaluation of the synthetic domain expert on Cityscapes (cf. figs. 7 and 11 and table I)
- 3) Variation of GT amount and comparison to the network performance when trained from scratch on Cityscapes with reduced GT (cf. figs. 8 and 10)

Within the experiments we compare our approach with the same types of methods as done in [52]:

<sup>1</sup><https://github.com/pytorch/vision/blob/main/torchvision/models/segmentation/deeplabv3.py>

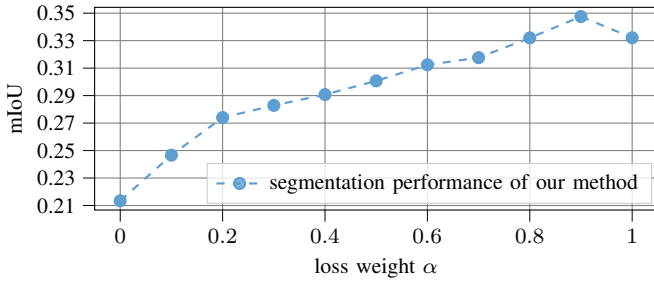


Fig. 6. Influence of the task loss based on the Synthia experiment setup. The weighting factor  $\alpha$  represents a linear interpolation between the adversarial generator loss and the task loss (cf. eq. (6)). Resulting in the original CycleGAN implementation for  $\alpha = 0$  and only the pixel-wise cross entropy loss for  $\alpha = 1$ .

- M1 Synthetic domain expert  $f$  fed with images generated by a GAN with CycleGAN-only training (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 small amount of labeled real-world images available at stage c).

We report the best mIoU of  $f$  evaluated on the output of  $G_{\mathcal{R} \rightarrow \mathcal{S}}$  based on the Cityscapes validation set during GAN training. To analyze the impact of the different loss components in the Synthia setup, we set the scaling parameter  $\gamma$  empirically to 0.25, we fix the GT amount to 5% (148 labeled training images) and vary the weighting parameter  $\alpha$  between 0 and 1. As stated in eq. (6) we use a linear interpolation between the losses. For the choice  $\alpha = 0$ , the training loss equals the original CycleGAN loss. For the other extreme choice  $\alpha = 1$ , the adversarial loss part is omitted entirely. The corresponding results are shown in fig. 6.

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 GAN training). In fig. 7 we show for one image from the Cityscapes validation dataset the differently generated images as well as their predictions by  $f$  trained on Synthia. The column “Cityscapes” in fig. 7 gives an exemplary visualization of 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., [28]. The domain gap is summarized in table I where we compare ImageNet pre-trained network performance to ours evaluated on Cityscapes. We state out-of-domain performance (second column), i.e. trained on Synthia, oracle performance (third column), i.e. trained on the full Cityscapes training dataset, 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

TABLE I  
DOMAIN GAP COMPARISON OF NETWORKS TRAINED FROM SCRATCH VS  
IMAGENET PRE-TRAINED

Synthia $\rightarrow$ Cityscapes (mIoU in %)			
method	out-of-domain	oracle	gap
ImageNet pre-trained [28]	31.8	75.6	43.8
from scratch (ours)	9.9	62.7	52.8

the real domain distorting a pure domain separation which should be pursued 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 training) 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 in our method. 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. 8 (blue curve), the graphs depict the ground truth amount in comparison to the mIoU. 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 175 epochs which we use as start point for stage c) training. For a fair comparison we trained the task agnostic GAN for another 125 epochs which results in a better mIoU of 21.34% in epoch 243 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 graph monotonously increases with the amount of GT available, reaching 33.20% mIoU for 5% GT and 33.76% when the GAN is trained with 10% labeled data in stage c). The results show that training  $G_{\mathcal{R} \rightarrow \mathcal{S}}$  with the task loss on negligible few GT data, improves the neural network’s understanding of the scene without retraining the neural network itself.

Additionally, we compare our method with results of  $f_{\mathcal{R}}$  (M3). We train the semantic segmentation network from scratch using (only) the same images as for the stage c) training of our method. Having no labeled data available, 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. 8. The results show that our method outperforms M3 by a distinct margin when only a few labels are available. However, when more than 297 (10%) fine labeled images of Cityscapes are available, 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 experiments 1) - 3) on our CARLA dataset with the CARLA semantic segmentation



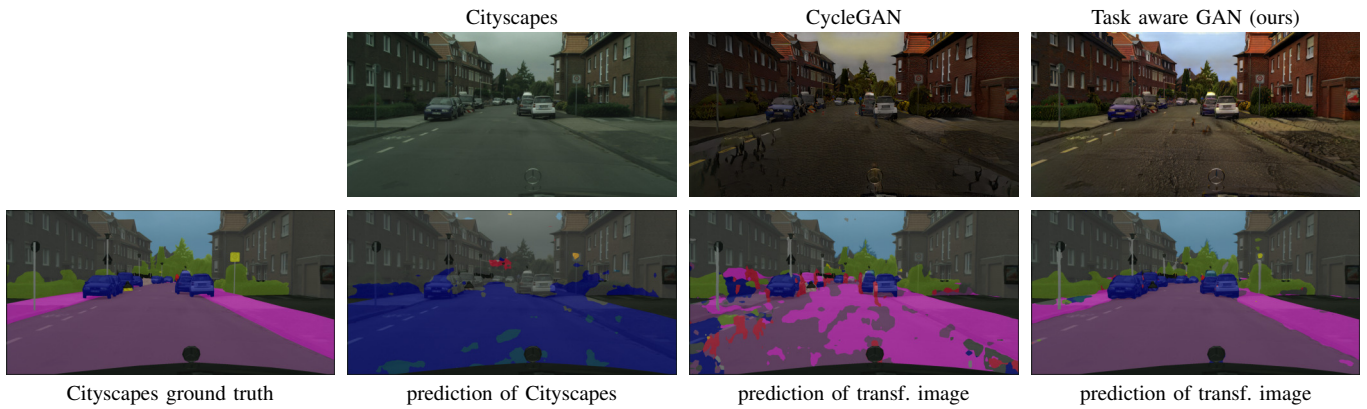


Fig. 7. Comparison of prediction results of an untranslated Cityscapes image (left), task agnostic I2I (mid) and our approach (right) based on a semantic segmentation network trained on Synthia.

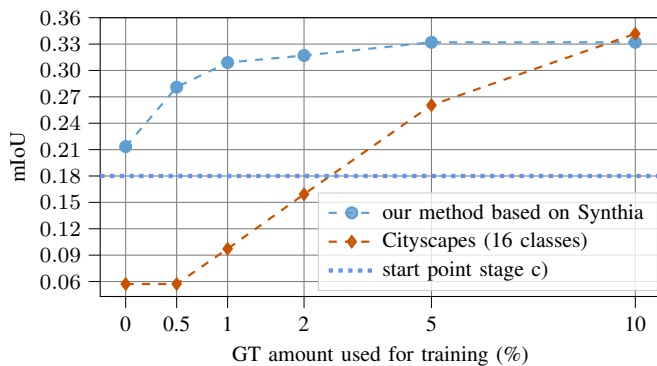


Fig. 8. Performance comparison of our method based on Synthia setup with different amount of ground truth (blue) and a from scratch supervised training on Cityscapes with the same amount of data (orange).

expert. The results of varying  $\alpha$  are visualized by the blue curve in fig. 9. 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 should be used in addition and not as a stand-alone concept.

The results of experiment 2) and 3) are shown in fig. 10 where the blue curve represents the best mIoU results achieved with our method for the given GT amount and the orange curve depicts the results  $f_{\mathcal{R}}$  (M3) given different amount of GT. We see a slightly lower increase (3.52 pp) of the performance between an exclusive stage b) training (0% GT, equaling a CycleGAN-only training) and our method with 0.5% GT. The best mIoU is achieved when using 5% of the dataset for the stage c) training, yielding a performance increase of 8.81 pp and an absolute mIoU of 39.06%. For the direct supervised training we experienced a steeper increase when training only on the 13 classes used for the CARLA experiments. However, As before our method outperforms M1 as well as M3 when less than 5% GT is available. Above that, the supervised method is superior.

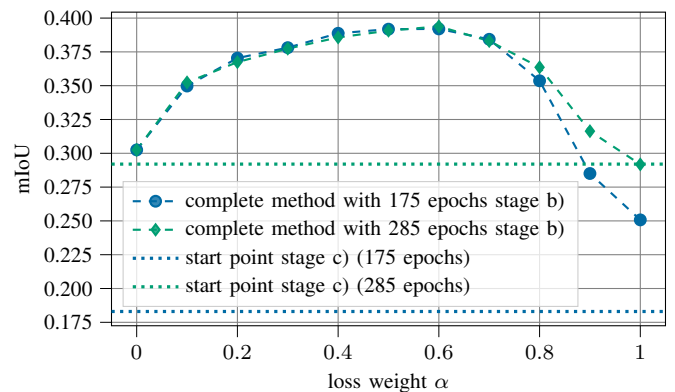


Fig. 9. Influence of the task loss on the network performance for the CARLA experiment setup. On the x-axis the weighting factor ( $\alpha$ ) of the task aware loss is plotted. The weighting represents a linear interpolation between the adversarial generator loss and the task loss eq. (6). Results after 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.

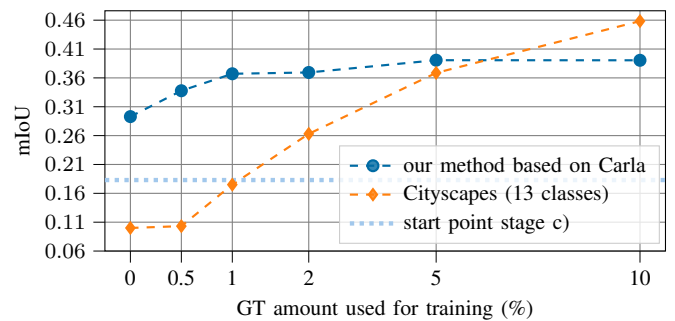


Fig. 10. Performance comparison of our method based on CARLA setup with different amount of GT (blue) and  $f_{\mathcal{R}}$  trained supervised from scratch on Cityscapes with the same amount of data (orange).

The qualitative results shown in fig. 11 reveal a distinct improved semantic segmentation of the street scene (bottom row) on the CARLA dataset. 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 inputting 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 CycleGAN (M1) and our method demonstrate a significant reduction of the domain gap.

Lastly we consider a longer stage b) training to find out whether our unsupervised pre-training was long enough or if 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. 9 visualized by the green curve. The experiments reveal that the choice of longer unsupervised training is not game changing. Although we start the stage c) training with a higher mIoU (dotted lines) when we train with stage b) for more steps, the experiments show that we achieve nearly the same absolute mIoU values, which only diverge when we put too much weight on the task loss ( $\alpha \geq 0.8$ ). We therefore conclude that a moderate number of epochs is enough for stage b).

## V. 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 image-to-image approach and from scratch training when a limited amount of GT is available. In addition, we analyzed the impact of the task awareness and the amount of GT. We showed that with our approach, we significantly reduce the amount of manually labeled data needed on the real domain compared to a from scratch training.

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 pre-trained backbone already incorporates real world domain information and therefore distort the gap analysis. Additionally, we showed that we can achieve very 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 therefrom and an in-depth analysis of how much abstract representations facilitate generalization. Recent work of Harary et al. [69] strengthens our hypothesis of an enlarged generalizability. 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, e.g., active learning-based methods like Singh et al. [53] did for classification. In addition, the method could be combined with self-training as these models need a good initialization to generate reasonable pseudo labels [31]. 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 self-training stages.

## ACKNOWLEDGMENT

This work is funded by the German Federal Ministry for Economic Affairs and Climate Action. The authors gratefully acknowledge the Gauss Center 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 [70] at Jülich Supercomputing Centre (JSC). Furthermore, we thank Hannah Lörcks for assisting in setting up the classification experiments.

## REFERENCES

- [1] Y. Chen, W. Li, X. Chen, and L. Van Gool, “Learning Semantic Segmentation From Synthetic Data: A Geometrically Guided Input-Output Adaptation Approach,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 1841–1850. 1
- [2] Y. Yuan, X. Chen, and J. Wang, “Object-contextual representations for semantic segmentation,” in *European conference on computer vision*. Springer, 2020, pp. 173–190. 1
- [3] A. Tao, K. Sapra, and B. Catanzaro, “Hierarchical multi-scale attention for semantic segmentation,” *arXiv:2005.10821 [cs]*, 2020. 1
- [4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, and S. Gelly, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv:2010.11929 [cs]*, 2020. 1
- [5] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012–10022. 1
- [6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The Cityscapes Dataset for Semantic Urban Scene Understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3213–3223. 1, 5
- [7] M. Rottmann and M. Reese, “Automated Detection of Label Errors in Semantic Segmentation Datasets via Deep Learning and Uncertainty Quantification,” *arXiv:2207.06104 [cs]*, 2022. 1, 6
- [8] J. Dai, K. He, and J. Sun, “Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1635–1643. 1
- [9] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, “Simple Does It: Weakly Supervised Instance and Semantic Segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 876–885. 1
- [10] J. E. van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Machine Learning*, vol. 109, no. 2, pp. 373–440, Feb. 2020. 1, 2
- [11] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, and M.-H. Yang, “Adversarial Learning for Semi-Supervised Semantic Segmentation,” *arXiv:1802.07934 [cs]*, Jul. 2018. 1

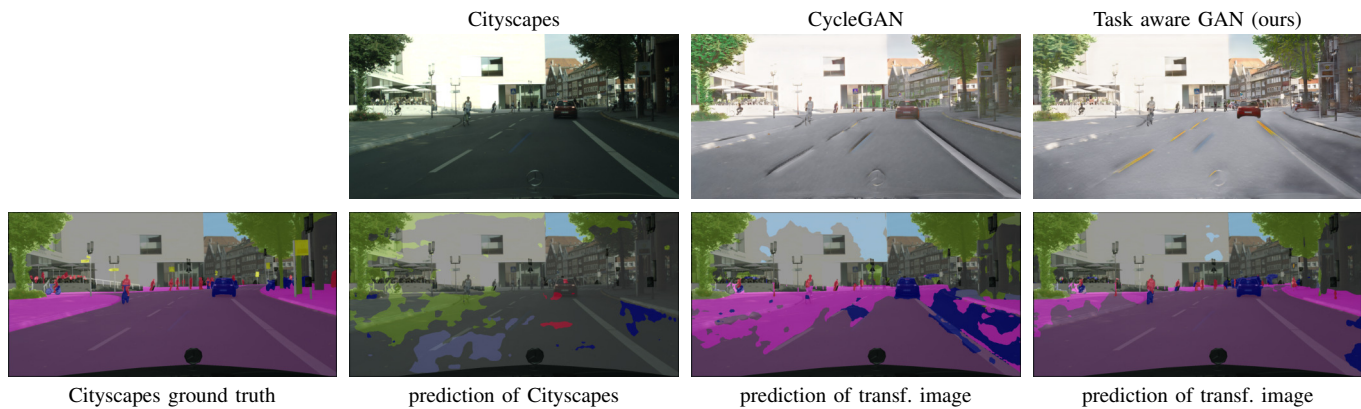


Fig. 11. Comparison of prediction results of an untranslated Cityscapes image (left), task agnostic transfer (mid) and our approach (right) based on a semantic segmentation network trained on our CARLA dataset.

- [12] A. K. Mondal, A. Agarwal, J. Dolz, and C. Desrosiers, “Revisiting CycleGAN for semi-supervised segmentation,” *arXiv:1908.11569 [cs]*, Aug. 2019. 1
- [13] Y. Ouali, C. Hudelot, and M. Tami, “Semi-Supervised Semantic Segmentation With Cross-Consistency Training,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 674–12 684.
- [14] P. Colling, L. Roese-Koerner, H. Gottschalk, and M. Rottmann, “Metabox+: A new region based active learning method for semantic segmentation using priority maps,” *arXiv:2010.01884 [cs]*, 2020. 1
- [15] L. Cai, X. Xu, J. H. Liew, and C. S. Foo, “Revisiting Superpixels for Active Learning in Semantic Segmentation With Realistic Annotation Costs,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 988–10 997. 1
- [16] M. Rottmann, K. Kahl, and H. Gottschalk, “Deep Bayesian Active Semi-Supervised Learning,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2018, pp. 158–164. 1
- [17] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, “Learning to Adapt Structured Output Space for Semantic Segmentation,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 7472–7481. 1
- [18] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for Data: Ground Truth from Computer Games,” in *European Conference on Computer Vision (ECCV)*, ser. LNCS, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9906. Springer International Publishing, 2016, pp. 102–118. 1
- [19] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An Open Urban Driving Simulator,” in *Conference on Robot Learning*. PMLR, Oct. 2017, pp. 1–16. 1, 5
- [20] M. Wrenninge and J. Unger, “Synscapes: A Photorealistic Synthetic Dataset for Street Scene Parsing,” *arXiv:1810.08705 [cs]*, Oct. 2018. 1
- [21] J. Hoffman, D. Wang, F. Yu, and T. Darrell, “FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation,” *arXiv:1612.02649 [cs]*, 2016. 1
- [22] G. Csurka, “Domain Adaptation for Visual Applications: A Comprehensive Survey,” *arXiv:1702.05374 [cs]*, Mar. 2017. 1
- [23] M. Toldo, A. Maracani, U. Michieli, and P. Zanuttigh, “Unsupervised Domain Adaptation in Semantic Segmentation: A Review,” *Technologies*, vol. 8, no. 2, p. 35, Jun. 2020, number: 2 Publisher: Multidisciplinary Digital Publishing Institute. 1, 3
- [24] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Perez, “ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019. 1
- [25] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker, “Domain adaptation for structured output via discriminative patch representations,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1456–1465. 1
- [26] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, “CyCADA: Cycle-Consistent Adversarial Domain Adaptation,” in *International Conference on Machine Learning*. PMLR, Jul. 2018, pp. 1989–1998. 1, 2
- [27] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, “Image to Image Translation for Domain Adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 4500–4509. 1
- [28] A. Dundar, M.-Y. Liu, T.-C. Wang, J. Zedlewski, and J. Kautz, “Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation,” *arXiv:1807.09384 [cs]*, 2018. 1, 7
- [29] G. Kang, Y. Wei, Y. Yang, Y. Zhuang, and A. Hauptmann, “Pixel-Level Cycle Association: A New Perspective for Domain Adaptive Semantic Segmentation,” in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 3569–3580. 1, 3
- [30] S. Brehm, S. Scherer, and R. Lienhart, “Semantically consistent image-to-image translation for unsupervised domain adaptation,” *ICAAART*, vol. 2, pp. 131–141, 2022. 1, 2, 5
- [31] K. Mei, C. Zhu, J. Zou, and S. Zhang, “Instance Adaptive Self-Training for Unsupervised Domain Adaptation,” *arXiv:2008.12197 [cs]*, Aug. 2020. 1, 9
- [32] Y. Yang and S. Soatto, “Fda: Fourier domain adaptation for semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4085–4095. 1
- [33] M. Kim and H. Byun, “Learning Texture Invariant Representation for Domain Adaptation of Semantic Segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 975–12 984. 1
- [34] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, “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*, 2021, pp. 12 414–12 424. 1
- [35] Y. Chen, X. Ouyang, K. Zhu, and G. Agam, “Semi-supervised Domain Adaptation for Semantic Segmentation,” *arXiv:2110.10639 [cs]*, 2021. 1, 2, 3
- [36] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, “Image analogies,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 327–340. 1, 2
- [37] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” in *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 3, 4, 6
- [38] P. Jaccard, “The Distribution of the Flora in the Alpine Zone,” *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912. 2
- [39] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes Challenge: A Retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015. 2
- [40] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proceedings of the 27th International Conference on Neural Information*

- Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, Dec. 2014, pp. 2672–2680. 2
- [41] Y. Pang, J. Lin, T. Qin, and Z. Chen, “Image-to-Image Translation: Methods and Applications,” *IEEE Transactions on Multimedia*, 2021. 2, 4
- [42] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 5967–5976. 2
- [43] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004. 2
- [44] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium,” in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. 2
- [45] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595. 2
- [46] R. Li, W. Cao, Q. Jiao, S. Wu, and H.-S. Wong, “Simplified Unsupervised Image Translation for Semantic Segmentation Adaptation,” *Pattern Recognition*, vol. 105, p. 107343, Apr. 2020. 2
- [47] X. Guo, Z. Wang, Q. Yang, W. Lv, X. Liu, Q. Wu, and J. Huang, “GAN-Based virtual-to-real image translation for urban scene semantic segmentation,” *Neurocomputing*, vol. 394, pp. 127–135, Jun. 2020. 2, 3
- [48] Z. Wu, X. Han, Y.-L. Lin, M. G. Uzunbas, T. Goldstein, S. N. Lim, and L. S. Davis, “Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 518–534. 2
- [49] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, “Semi-supervised domain adaptation via minimax entropy,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8050–8058. 2
- [50] T. Kim and C. Kim, “Attract, Perturb, and Explore: Learning a Feature Alignment Network for Semi-supervised Domain Adaptation,” in *European conference on computer vision – ECCV 2020*. Cham: Springer International Publishing, 2020, pp. 591–607. 2
- [51] P. Jiang, A. Wu, Y. Han, Y. Shao, M. Qi, and B. Li, “Bidirectional Adversarial Training for Semi-Supervised Domain Adaptation,” in *Twenty-Ninth International Joint Conference on Artificial Intelligence*, vol. 1, Jul. 2020, pp. 934–940. 2
- [52] S. Mabu, M. Miyake, T. Kuremoto, and S. Kido, “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*, vol. 16, no. 11, pp. 1925–1935, Nov. 2021. 2, 3, 5, 6
- [53] A. Singh, N. Doraiswamy, S. Takamuku, M. Bhalerao, T. Dutta, S. Biswas, A. Chepuri, B. Vengatesan, and N. Natori, “Improving Semi-Supervised Domain Adaptation Using Effective Target Selection and Semantics,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2709–2718. 2, 9
- [54] Z. Wang, Y. Wei, R. Feris, J. Xiong, W.-M. Hwu, T. S. Huang, and H. Shi, “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*, 2020, pp. 936–937. 2
- [55] Y. Wang, C. Wu, L. Herranz, J. van de Weijer, A. Gonzalez-Garcia, and B. Raducanu, “Transferring GANs: generating images from limited data,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 218–234. 2
- [56] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved Training of Wasserstein GANs,” *arXiv:1704.00028 [cs]*, Mar. 2017. 2
- [57] M. Zhao, Y. Cong, and L. Carin, “On Leveraging Pretrained GANs for Limited-Data Generation,” in *ICML*, 2020. 2
- [58] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for GANs do actually converge?” in *International conference on machine learning*. PMLR, 2018, pp. 3481–3490. 2
- [59] T. Grigoryev, A. Voynov, and A. Babenko, “When, Why, and Which Pretrained GANs Are Useful?” *arXiv:2202.08937 [cs]*, Mar. 2022. 2
- [60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. 2
- [61] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. Paul Smolley, “Least Squares Generative Adversarial Networks,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2794–2802. 3, 4
- [62] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, “The sketchy database: learning to retrieve badly drawn bunnies,” *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 119:1–119:12, Jul. 2016. 4
- [63] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. 4
- [64] G. Ros, L. Sellart, J. Materzynska, D. Vázquez, and A. M. López, “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 (CVPR)*, Jun. 2016, pp. 3234–3243. 5
- [65] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. 6
- [66] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation,” *arXiv:1706.05587 [cs]*, Dec. 2017. 6
- [67] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image Segmentation Using Deep Learning: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. 6
- [68] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv:1412.6980 [cs]*, 2014. 6
- [69] S. Harary, E. Schwartz, A. Arbel, P. Staar, S. Abu-Hussein, E. Amrani, R. Herzig, A. Alfassy, R. Giryes, and H. Kuehne, “Unsupervised Domain Generalization by Learning a Bridge Across Domains,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5280–5290. 9
- [70] Jülich Supercomputing Centre, “JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre,” *Journal of large-scale research facilities*, vol. 5, no. A135, 2019. 9