Towards Unsupervised Online Domain Adaptation for Semantic Segmentation

Yevhen Kuznietsov[1] Marc Proesmans[1] Luc Van Gool[1,2]
[1]KU Leuven/ESAT-PSI [2]ETH Zurich/CVL
{yevhen.kuznietsov, marc.proesmans, luc.vangool}@esat.kuleuven.be

Abstract

In recent years, there has been significant progress in overcoming the negative effects of domain shift in semantic segmentation. Yet, existing unsupervised domain adaptation methods operate in an offline fashion, which imposes multiple restrictions on their deployment in real world scenarios. In this paper, we introduce a problem of online domain adaptation for semantic segmentation, which involves producing predictions for and, at the same time, continuously adapting a model to new frames of target domain videos. To tackle this problem, we propose a novel method which utilizes unsupervised structure-from-motion cues as the primary source of domain adaptation. By optimizing online the representation shared between depth and semantics networks, our geometry-guided algorithm achieves semantic segmentation performance comparable to state-of-the-art offline methods, without using target domain training data whatsoever.

1. Introduction

Semantic segmentation is an essential task for various industry fields, such as autonomous driving or robotics. Unfortunately, semantic segmentation models do not perform well under the conditions of domain shift – a change in the distribution between the sets of data a model was trained on (source) and is deployed on (target). Such a difference in the data distribution can be caused, for instance, by transferring from synthetic to real domain, the variation of weather conditions or even the location change.

The goal of domain adaptation is to improve the model performance in the target domain by reducing the consequences of domain shift. Supervised domain adaptation assumes the annotations available in both source and target domains. It allows to benefit from the vast amount of synthetic data with accurate and automatically-derived labels. Nonetheless, the annotations in real domain are expensive or even often unavailable, which is a significant drawback of supervised domain adaptation algorithms.

Unsupervised domain adaptation (UDA) poses a more viable alternative to supervised methods, as it only requires the source domain data to be labeled. Most methods explicitly align source and target domain distributions using adversarial learning or explore the use of self-training by iteratively refining pseudo-labels.

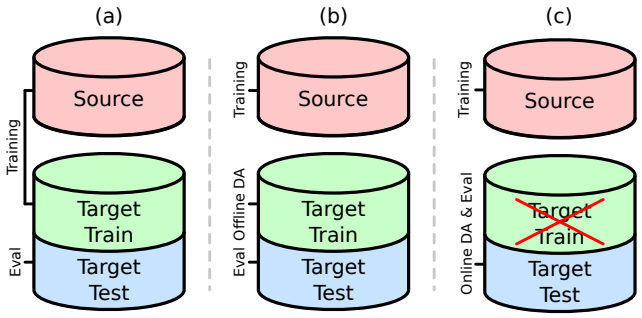

Figure 1. Schematic comparison of UDA setups: (a) offline, (b) source-free offline, (c) ours – source-free online.

One common limitation of existing semantic segmentation UDA (we will often omit "semantic segmentation" from now on) algorithms is offline setting: they rely on curated target domain data to train the model for multiple epochs. Such setting introduces multiple restrictions: First, target domain data has to be acquired and pre-processed in advance, before training. Second, these methods do not benefit from new data available in target domain unless the model is retrained. Third, offline UDA does not allow to adjust the model to dynamically changing domain properties, such as weather conditions. The aforementioned restrictions make offline UDA impractical for many real-world applications desiring online capabilities.

To tackle the shortcomings of offline UDA and to encourage further research on online adaptation, we make the following contributions:

- We introduce a method for joint source-free online UDA of depth and semantic segmentation, which imposes more realistic requirements on model deployment compared to offline algorithms;
- We compose a benchmark for online semantics domain adaptation by complementing KITTI semantic

segmentation [1] annotations with long videos from other KITTI subsets [12, 13]. This benchmark facilitates the evaluation of our method and possible future online UDA works;

- We perform extensive analysis of proposed design choices, such as the use of experience replay, geometry guidance and confidence regularization;
- We demonstrate that online UDA is capable of achieving performance comparable to state-of-the-art offline methods. In particular, our algorithm reaches 57.3% mean intersection over union (IoU) when adapting from Virtual KITTI v2 to KITTI, compared to 59.0% achieved by a well-performing offline baseline.

2. Related Work

In this section, we discuss two lines of research which are most related to our work: offline UDA for semantic segmentation, and unsupervised online adaptation for stereo and monocular depth estimation.

2.1. Offline UDA for Semantic Segmentation

Semantic segmentation UDA methods can roughly be categorized in the following way: distribution alignment and self-training algorithms, and the works exploring various proxy tasks. We will mostly focus on the latter two, since they are more relevant to our research.

Distribution Alignment. Most UDA methods tackle domain shift by explicitly matching source and target domain distributions.

Some algorithms [56, 24, 49, 17, 30, 36, 51, 46, 50] transform target domain images in such a way that they look like those belonging to source domain, or vice versa – source to target. Target to source translation allows to utilize models trained in source domain, while source to target – to train models in "fake" target domain using source domain annotations, assuming that image translation preserves semantics. Other methods try to make learnt representations [17, 19, 30, 51] or model outputs [5, 18, 41, 42, 43, 44, 35, 49] domain indistinguishable.

Most works mentioned in the previous paragraph employ adversarial training. Interesting exceptions are non-adversarial image generation algorithms [51, 46, 50]. For instance, Yang *et al*. [50] showed that style can be transferred by simply swapping the amplitudes of image Fourier transforms.

Self-Training. Self-training UDA methods make use of the assumption that the model trained with source annotations already achieves reasonably good performance on target data. These methods generate pseudo-labels using most confident predictions, and utilize the generated labels as a source of supervision for further model optimization.

Zou *et al*. [57] introduced iterative self-training (IST) - a procedure which alternates between generating pseudo-labels for target domain and retraining the model using the generated labels. As predictions for some classes are naturally more confident than for others, pseudo-labels may be dominated by high-confidence classes, whereas low-confident ones remain underrepresented. To avoid this, [57] proposed to generate pseudo-labels taking into account class confidence distributions in target domain. Alternatively, [29] tackled this issue by utilizing class- and image-adaptive confidence regions, [31] and [20] added weak labeling loss indicating class presence in the image, and [37] incorporated focal loss [27].

Another way to enhance self-training is the use of additional regularizations or constraints. For instance, in order to avoid over-fitting to noisy pseudo-labels, [58] penalize overconfident predictions similarly to [56]. [57] constrain self-training using source domain spatial priors. [29] proposed to sharpen low confidence regions via direct entropy minimization likewise [43]. [37] and [20] enforce consistency between model predictions at different scales and locations respectively.

Proxy Tasks. Another interesting line of UDA research explores the use of proxy tasks as additional means of domain adaptation.

[38, 48] showed that solving simple problems such as rotation, flip, patch location prediction and jigsaw puzzle completion improves segmentation quality if combined with existing UDA algorithms.

Alternatively, several methods incorporated a more complex task of monocular depth estimation. For instance, [24] proposed to utilize source domain depth for generating geometrically correct target-style images. This is achieved by making the model produce consistent depth predictions for source and generated target-style images. In addition to enforcing consistency on depth outputs, [4] use source domain depth as an extra input to image generator network. Differently from [24] and [4], [44] and [35] do not utilize depth for image transfer. Instead, they proposed to align semantics fused with continuous and discrete depth respectively.

In contrast to previous works, which only try to make target domain depth predictions indistinguishable from source domain depth, our method and the concurrent works by Guizilini *et al*. [15] and Wang *et al*. [45] try to make them consistent with the target scene geometry. This is achieved by optimizing depth predictions using target domain self(un)-supervised structure-from-motion (SfM) cues.

2.2. Online UDA for Depth Estimation

Recently, several works leveraged geometry cues for self-supervised online adaptation of stereo and monocular depth estimation models. Instead of relying on a dedicated target domain training set, these methods either perform per-image adaptation utilizing only the data from a small

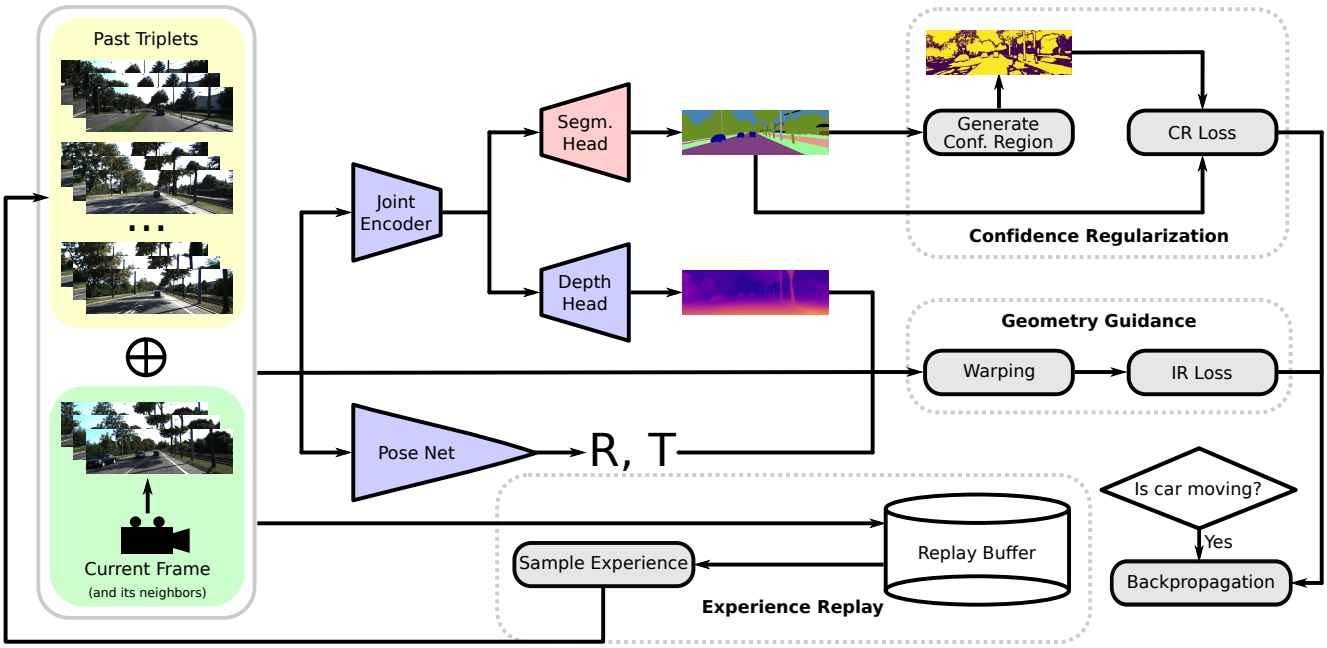

Figure 2. Our online UDA pipeline. We keep segmentation head frozen (pink), whereas the parts of the model depicted in blue are adapted.

neighborhood of an input image [6, 3] or adapt a model on videos in online fashion [53, 25, 23, 39, 40, 52, 54].

Most depth adaptation methods focus on improving the adaptation stability or speed. [53, 25, 39, 52] employ meta-learning [9] to obtain better adaptor (optimizer) parameters. Other works regularize adaptation using past experience. For instance, Chen *et al*. [6] and Zhang *et al*. [53] force the predictions of the adapted model to be similar to the predictions generated by the model at some previous state. Kuznietsov *et al*. [23], in turn, utilize experience replay [26], optimizing the model estimates for past samples, randomly drawn from replay buffer, the same way as the predictions for the current frame.

Due to the inherent similarity of stereo and monocular depth estimation tasks to the respective self-supervised optimization objectives, these methods yield clear improvements under the conditions of domain shift. Nonetheless, there is no research on online UDA for such an important task as semantic segmentation.

3. Methodology

In this section, we describe the key components of our method – supervised pre-training in source domain and the proposed source-free online UDA. Additionally, we introduce a source-free offline baseline in order to compare to state-of-the-art offline UDA methods.

3.1. Problem Setting

Online UDA. Given the model $\mathcal{M}$ trained with annotated source data D_S and a set of target domain videos without annotations V_T, online UDA setting assumes that in order to produce predictions for frame I_t at time t of video $v \in V_T$ only the following data can be accessed:

- Any data from D_S;
- $\{I_{t-i} | I_{t-i} \in v \wedge i \geq 0\}$.

Source-Free UDA. While online UDA does not prohibit the utilization of source data, we voluntarily waive this privilege likewise other source-free methods [22, 28]. Consequently, our method is also applicable to scenarios when source data is private.

Fig. 1 illustrates the differences in data usage between offline, source-free offline and source-free online UDA settings. We employ the latter in all our "online" experiments.

3.2. Supervised Pre-Training in Source Domain

Since we rely on self-supervised depth and ego-motion estimation as a proxy task for source-free domain adaptation, it is important that source data allows to learn ego-motion.

Given a source dataset containing (possibly very short) videos recorded by a moving camera and ground truth semantic labels for at least some of the video frames, we train semantics, depth and ego-motion estimators jointly using the following loss for supervision:

$$\mathcal{L}_{train} = \mathcal{L}_{IR} + \alpha \mathcal{L}_S, \tag{1}$$

where $\mathcal{L}_{IR}$ is the image reconstruction loss as defined in Monodepth2 [14], and $\mathcal{L}_S$ is supervised semantic segmentation loss. $\mathcal{L}_{IR}$ allows to optimize the parameters θ of depth and ego-motion estimators in self-supervised way. Given

three neighboring frames I_{-1}, I_0 and I_1, $\mathcal{L}_{IR}$ can be obtained as follows. First, the synthetic views $\hat{I}_{-1}$ and $\hat{I}_1$ of I_0 are generated by warping its neighbors I_{-1} and I_1 into I_0. The warping (re-projection) is performed using the respective depth $\hat{d}_0^\theta$ and ego-motion $\hat{m}_{-1}^\theta$, $\hat{m}_1^\theta$ predictions. Then, $\mathcal{L}_{IR}$ is computed as the difference between real I_0 and synthesized $\hat{I}_{-1}$, $\hat{I}_1$. For more details please refer to [14].

For semantic segmentation loss $\mathcal{L}_S$, we employ the modification of bootstrapped cross-entropy [47]:

$$\mathcal{L}_S = -\frac{1}{|\Omega| \cdot |\mathcal{C}|} \sum_{p \in \mathcal{H}} \sum_{c \in \mathcal{C}} y(p,c) log \hat{y}(p,c), \qquad (2)$$

where $y(p,c)$ is 1 if ground truth at pixel p belongs to class c and 0 otherwise, $\hat{y}(p,c)$ is the soft-max score predicted by model for class c at pixel p, Ω is the set of all pixels in the batch, and the set of hard pixels $\mathcal{H}$ is defined as follows:

$$\mathcal{H} = \left\{ p \;\middle|\; -\frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} y(p,c) log \hat{y}(p,c) > \tau \right\}, \qquad (3)$$

with τ being an adaptive cross-entropy value threshold set at top k-th cross-entropy score within a batch.

The described hard pixel mining is known to enhance prediction sharpness and effectively contributes to class balancing, demonstrating dramatic improvements for small but frequent classes such as traffic light, sign or poll.

3.3. Geometry-Guided Online UDA

While devising our method for online UDA, we considered various offline UDA procedures described in Section 2. However, the most widely used algorithms are seemingly hard to integrate into online setup. For instance, the use of both adversarial distribution alignment and IST is not feasible during early stages of adaptation (e.g., when less than 100 target domain frames are available). The reason for this is the need to train a domain discriminator from scratch or re-train a semantic segmentation network for adversarial alignment and for IST respectively. Employing IST at later stages of online adaptation is also debatable, as it is a very time consuming procedure, and, in online setting, the amount of new data acquired while re-training a semantic segmentation network may be significantly bigger than what was originally used for re-training.

Utilizing self-supervised monocular depth and ego-motion adaptation as a proxy task, on the other hand, does not have obvious online restrictions, and can be performed very close to real time [23, 40]. However, in order to be able to use SfM cues as the primary supervision source for semantic segmentation UDA, our method must exhibit following characteristics: 1) features used to predict semantics and geometry are correlated and 2) improving geometrical correctness of learned representation should not make it less discriminative for semantic segmentation.

The first characteristic is encouraged by employing a shared encoder for depth and semantics networks. For the second, we introduce confidence regularization $\mathcal{L}_{CR}$, which forces the model to not deviate significantly from already confident predictions:

$$\mathcal{L}_{CR} = -\frac{1}{|\Omega| \cdot |\mathcal{C}|} \sum_{p \in \mathcal{X}} log \left(\max_{c \in |\mathcal{C}|} \hat{y}(p,c) \right), \qquad (4)$$

where $\mathcal{X}$ is the set of pixels for which the model predicted highly-confident labels. We set $p \in \mathcal{X}$ if $\max_{c \in |\mathcal{C}|} \hat{y}(p,c) > \gamma_c$, where γ_c is the minimum of 0.99 and top $l\%$ softmax score for class c within a batch.

While confidence regularization is very similar to self-training using online-generated pseudo-labels, its purpose is to prevent the model updates which improve depth estimation, but worsen segmentation quality. And, as shown in Section 4.3, confidence regularization does not yield noticeable segmentation improvements when used without geometry guidance.

As the result of our previous considerations, we propose online UDA based on the combination of confidence regularization and geometry guidance. In particular, for every frame I_t (also using I_{t-1} and I_{t+1} for image reconstruction) of a target domain video, model parameters are optimized by minimizing the following loss:

$$\mathcal{L}_{adapt} = \mathcal{L}_{IR} + \alpha \mathcal{L}_{CR}. \qquad (5)$$

In practice, monocular depth adaptation methods often employ additional mechanisms to avoid catastrophic forgetting and overfitting. Similarly to [23], we employ experience replay [26], adapting our model using several past samples (frame triplets), drawn from replay buffer, in addition to the current frame sample. Past samples are drawn randomly, and experience in the buffer is distributed uniformly over time. However, in contrast to [23], new samples are added to our replay buffer during adaptation in target domain, and the buffer does not contain source domain samples unless stated otherwise. Together with experience replay, our method is summarized by Fig. 2 and Algorithm 1. The latter follows the notation from this section and also formalizes our online evaluation protocol.

3.4. Offline UDA Baseline

For source-free offline baseline, we fine-tune the source-trained model on a dedicated target domain training set. First, the model is optimized using the loss from Eq. 5 for $n = 4$ epochs. After this, we continue to train our model with self-training loss instead of confidence regularization. In particular, in the beginning of each epoch, pseudo-labels are generated using the confidence region $\mathcal{X}$ selection procedure described in Section 3.3. Then, self-training loss is computed as cross-entropy between the predicted semantics and the generated pseudo-labels.

Algorithm 1 Proposed Online UDA

Require: Set of target videos V_T, model $\mathcal{M}$ pre-trained in source domain D_S, batch size b, α, other hyper-parameters.
Ensure: IoU for every class

$I\&U \leftarrow [[0...0], [0...0]]$ ▷ *Intersections and unions*
for $v \in V_T$ **do**
  $\mathcal{E} \leftarrow \emptyset$ ▷ *Initialize replay buffer*
  $\mathcal{M}^* \leftarrow \mathcal{M}$ ▷ *Reset model parameters for every video*
  for $t \in [2..\textbf{end}(v)]$ **do**
    $\mathcal{E} \leftarrow$ **add_sample**$(I^v_{t\text{-}2}, I^v_{t\text{-}1}, I^v_t)$
    $\mathcal{B} \leftarrow (I^v_{t\text{-}2}, I^v_{t\text{-}1}, I^v_t) \oplus$ **sample**$(\mathcal{E}, b - 1)$ ▷ *Compose a batch*
    ▷ *consisting of the current and $b-1$ past frame triplets from $\mathcal{E}$*

    ▷ *Model optimization w.r.t. middle frames of [-1, 0, 1] triplets*
    $\hat{m}_{\text{-}1}, \hat{m}_1, \hat{d}_0, \hat{y}_0 \leftarrow \mathcal{M}^*(\mathcal{B})$ ▷ *Inference*
    if car is moving then
      $\hat{I}_{\text{-}1}, \hat{I}_1 \leftarrow$**do_warping**$(\hat{m}_{\text{-}1}, \hat{m}_1, \hat{d}_0, \mathcal{B}[-1, 1])$
      compute $\mathcal{L}_{IR}(\hat{I}_{\text{-}1}, \hat{I}_1, \mathcal{B}[0])$
      $\mathcal{X} \leftarrow$ **generate_cr** $(\hat{y}_0)$ ▷ *Confidence region*
      compute $\mathcal{L}_{CR}(\hat{y}_0, \mathcal{X})$
      $\mathcal{L}_{adapt} \leftarrow \mathcal{L}_{IR} + \alpha\mathcal{L}_{CR}$
      $\mathcal{M}^* \leftarrow$**optimize**$(\mathcal{M}^*, \mathcal{L}_{adapt})$
    if y^v_t exists **then** ▷ *Evaluation*
      $\hat{y}^v_t \leftarrow \mathcal{M}^*(I^v_t)$
      $I\&U \leftarrow$ **update_I&U**$(\hat{y}^v_t, y^v_t)$
compute_IoU$(I\&U)$

4. Experiments

In this section, we provide experimental evaluation of our method as well as comparison to prior art.

4.1. Datasets

Most of our experiments are evaluated using simulated to real UDA setup. Thus, we chose one synthetic and two real datasets for source and target domains respectively.

Virtual KITTI v1/v2. Virtual KITTI [2, 10] is a synthetic dataset simulating the scenes from KITTI. It contains five daytime videos with normal weather conditions, resulting in a total of 2126 frames for left camera. Each frame has automatically generated dense semantic annotations for 12 classes. Unlike other datasets widely used in sim-to-real setup (Synthia [33] and GTA5 [32]), Virtual KITTI consists of videos and allows to learn ego-motion in source domain, which is crucial for our method.

There are two versions of Virtual KITTI available. Both cover the same scenes and are almost identical, but more recent v2 is supposedly more photo-realistic. We use both v1 and v2 to compare our offline baseline to other methods, but perform all other experiments utilizing only v2, since the annotations of v1 for classes such as pole, traffic light and traffic sign are inconsistent with the annotation policy of KITTI for these classes. More details can be found in the supplementary materials.

KITTI. KITTI [12] is a huge driving dataset consisting of videos taken in urban, residential and road environments. For offline evaluation, we follow the protocol of [4]. In particular, 200 unannotated frames from KITTI semantic segmentation [1] test set are used as the training set for UDA, whereas 200 annotated images from the official training set are used for evaluation. For our online protocol, the annotated images from the segmentation benchmark are mapped to the videos from KITTI raw [12] (as in development kit) and odometry set [13] (manually). This resulted in 30 videos with 9070 frames total, 148 of which are semantically annotated. The length of obtained videos ranges from 35 to 943 frames, and the number of annotated frames per video varies from 1 to 12. On average, the annotated frames have approximately 200 preceding frames. The complete mapping is provided in the supplementary materials.

Cityscapes. Cityscapes [7] is an urban driving dataset with big amount of high quality semantic annotations. Annotated images are surrounded by 30-frames video snippets, with 18 preceding frames each. This, however, is not sufficient for online adaptation setup. For online protocol, we selected the video from Frankfurt – the only long video from Cityscapes with semantic annotations available. This video consists of 106917 frames, 267 of which are annotated and used for evaluation. Since the video is huge, we perform adaptation step once every 10 frames.

4.2. Model Setup

For shared depth-semantics and for pose encoders, we employ ResNet18 [16]. We use depth and pose heads from Monodepth2 [14]. For segmentation head, we use the depth decoder architecture with the last layer modified: the number of output channels is set to be equal the number of classes used for training, and sigmoid activation is replaced with softmax. Our training configuration in source domain is following. We keep the parameters of the image reconstruction loss as proposed in [14]. Semantic segmentation loss weight α is set to 0.1, and the warm-up schedule is used for the bootstrapped cross-entropy parameter k:

$$k = h \cdot w \cdot b \cdot \left(0.15 + 0.85 \cdot max\left(1 - \frac{epoch}{10}, 0\right)\right). \quad (6)$$

We resize Virtual KITTI images to height $h = 320$ and width $w = 1024$. Our model is initialized with ImageNet [34] pre-trained parameters, and is trained for 80 epochs with Adam optimizer [21], learning rate 0.0001 and batch size $b = 6$. We use same data augmentation strategy as [14]. During both online and offline UDA we keep most model parameters (including α) unchanged. We resize KITTI and Cityscapes images to 1024×320 and 1024×512 px respectively. Offline baseline is trained for 40 epochs in target domain. Horizontal flip augmentation is used at test time. Due to geometric guidance relying on camera motion cues, we do not perform back-propagation when camera translation is not sufficient. Car movement can be determined either by thresholding translation predicted by pose network, or using CAN bus.

Model	Road	Building	Pole	T.light	T.sign	Vegetation	Terrain	Sky	Car	Truck	Mean
Source	58.0	53.0	42.6	26.9	25.4	61.5	16.4	87.4	68.1	10.4	45.0
Baseline (offline)	89.7	66.0	40.1	34.1	30.4	83.1	62.1	90.6	83.4	9.9	59.0
Online (GG + CR)	**87.8**	**51.2**	40.3	35.7	**30.4**	**76.6**	**55.5**	**88.4**	**81.6**	26.9	**57.3**
Online (GG)	86.7	48.5	**41.0**	**36.6**	28.4	74.4	50.6	86.9	80.8	23.2	55.7
Online (CR)	79.4	32.6	17.6	25.2	8.7	73.6	52.3	87.8	61.3	**31.5**	47.0

Table 1. Semantic segmentation results for UDA from Virtual KITTI v2 to KITTI online semantics benchmark.

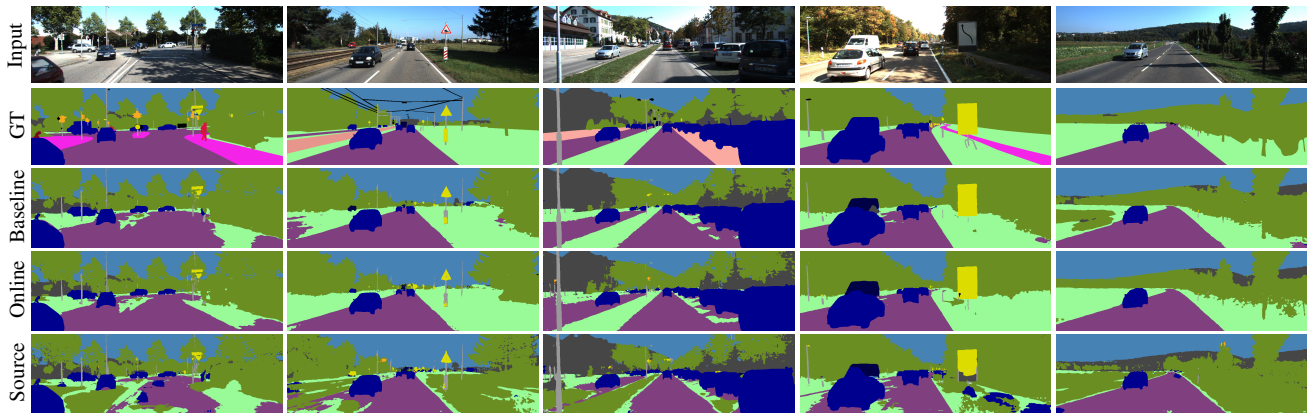

Figure 3. Semantic segmentation predictions for Virtual KITTI v2 → KITTI UDA. Cityscapes color coding is applied.

Due to experience sampling being a stochastic process, we report semantic segmentation results averaged over 10 runs for every online experiment. For most online experiments we draw 5 past samples for experience replay at every step, so that the total batch size remains unchanged. The only exceptions are experience replay ablation studies and online adaptation on Cityscapes, where we had to reduce batch size to 4 in order to fit into Nvidia GTX 1080Ti GPU memory. Maximum replay buffer size is set to 2000 samples, which easily fit into RAM. The procedure for preventing buffer overflow (only relevant for Cityscapes) is described in the supplementary materials.

4.3. Results

In this section, we discuss the performance of our method for both semantic segmentation and depth estimation, and assess the effects of its individual components using the benchmark proposed in Section 4.1. We also explore how our method behaves over time, adapts to Cityscapes Frankfurt, and to changing weather conditions. In addition to this, we demonstrate another interesting scenario (post-adaptation) in the supplementary materials.

Performance Analysis. As shown in Table 1 and Fig. 3, online adaptation (**Online**) significantly improves over the non-adapted model (**Source**) on all classes except building and pole. The most dramatic performance gain is achieved for big classes: 39.1% IoU for terrain, 29.8%, 16.5%, 15.4%, 13.5% – for "road", "truck", "vegetation" and "car" respectively. Despite using a curated target domain training set, offline **Baseline** surpasses our online method by only 1.7% mean IoU. Interestingly, online adaptation performs better on hard classes. For instance, online adaptation demonstrates substantially better performance for "truck" compared to offline baseline. This presumably happens because the adaptation is performed separately for each video, and, if a truck is present in a video, the frequency of the truck appearance in its frames is noticeably higher than in the training set.

Both confidence regularization (**CR**) and geometry guidance (**GG**) are important for our method. While geometry guidance alone allows to obtain mean IoU gain of 10.7% compared to the non-adapted model, adding confidence regularization to it further improves performance by 1.6% mean IoU. In contrast to geometry guidance, confidence regularization does not perform well as the primary source of domain adaptation. When used alone, it demonstrates substantially lower IoU for four classes – "building", "pole", "sign" and "car".

Effect of Experience Replay. Fig. 4 shows the effects of utilizing **Random** and **Sequential** sampling for experience replay compared to no experience replay (**No ER**). While random sampling was previously explained in Section 3.3, sequential sampling refers to the use of past samples (triplets) at times $t-1, t-2, ...$ when adapting at t. Increasing the number of past samples used for every adaptation step improves performance at seemingly same rate

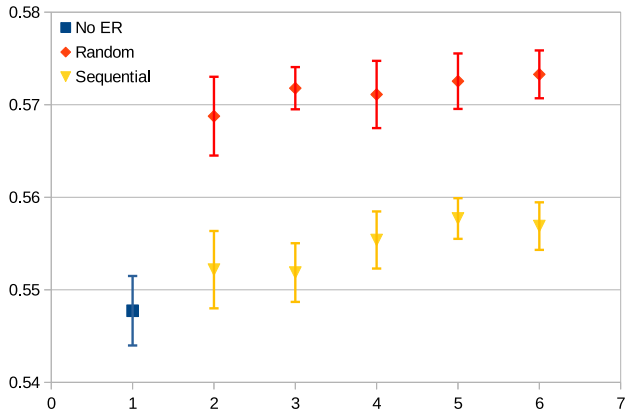

Figure 4. Semantic segmentation results (mean IoU) for online UDA from Virtual KITTI v2 to KITTI online semantics benchmark wrt experience sampling strategy and the number of past samples drawn from replay buffer. Horizontal axis shows total adaptation batch size, and candlesticks define regions within $\mu \pm \sigma$.

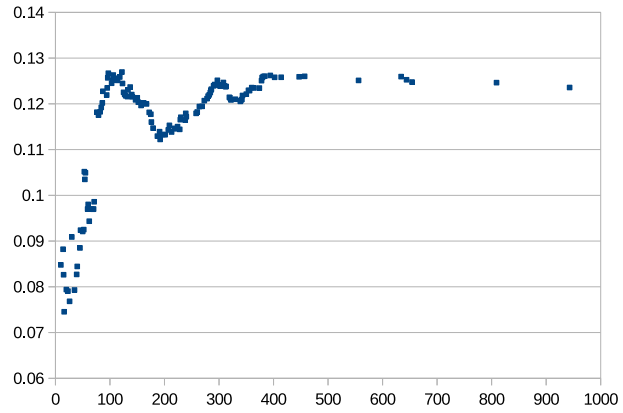

Figure 5. Running mean IoU gain (y axis) of our online UDA compared to non-adapted model when transferring from Virtual KITTI v2 to KITTI online semantics benchmark.

for both random and sequential experience replay. However, the addition of only one random sample noticeably increases mean IoU, which is not the case with sequential sampling. Thus, we expect the performance gap of about 1.5% IoU between random and sequential experience replay to hold if the number of samples per batch is further increased. We also expect further mean IoU growth for both sampling strategies, but, unfortunately, could not test this hypothesis due to the GPU memory limitations.

Adaptation Performance over Time. Since online adaptation is a continuous process, we explore how the adapted model behaves over time. For this, we depict the running mean IoU difference between online-adapted and non-adapted models (Fig. 5). Running mean IoU at time t is computed in the following way: First, annotations from all videos and the corresponding predictions are sorted according to their timestamp f relative to the start of a video. Then, the mean IoU is obtained for the set of annotated

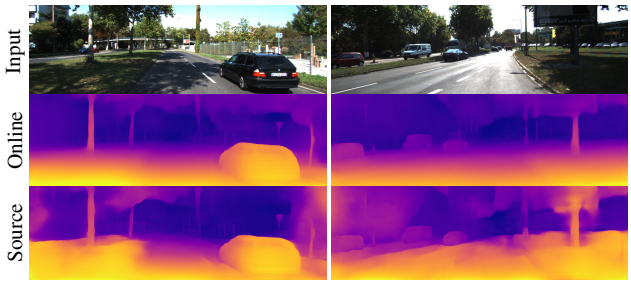

Figure 6. Depth maps for Virtual KITTI v2 $\rightarrow$ KITTI UDA. Brighter pixels correspond to closer regions.

Model	Abs Rel↓	Sq Rel↓	RMSE↓	δ<1.25 ↑
Source	0.246	2.420	8.205	0.585
Baseline	0.183	2.495	7.884	0.783
Online-CR	0.151	2.035	7.089	0.835
Online	**0.145**	**1.878**	**6.907**	**0.842**

Table 2. Depth estimation performance for UDA from Virtual KITTI v2 to KITTI online benchmark. Metrics definition can be found in [8], ↓ and ↑ indicate whether lower or higher values are better. "-CR" denotes no confidence regularization.

frames with $f \leq t$. Even though IoU is a very sensitive metric and can be significantly influenced by only one frame, Fig. 5 shows that our method benefits from adapting for at least 100 frames. In particular, Pearson correlation coefficient between mean IoU gain and frame "timestamp" in video is 91.8 for first 100 video frames. The performance drop after approximately 100 frames can be explained by the sensitivity of IoU and the influence of videos, for which online adaptation improves the model slower. E.g., there are videos with car being static in the beginning and the annotations only available after 100 frames.

Depth Estimation Results. Table 2 and Fig. 6 demonstrate that our semantic segmentation method also improves depth. Table 2 also illustrates two interesting findings: 1) Online-adapted model achieves significantly better depth estimation performance compared to the model trained offline in target domain; 2) Unsupervised semantic cues such as confidence regularization (**CR**) may be useful for online depth adaptation.

Adaptation to Cityscapes. The results for our method adapting to Cityscapes Frankfurt video are shown in Table 3. On average, online-adapted model (**Online**) achieves 10.7% IoU higher performance for big classes compared to the non-adapted one (**Source**), and 7.6% – for all classes respectively. While the observed performance gain is less impressive compared to online adaptation to KITTI, we point out that the adaptation was only performed for every 10-th frame. Thus, when evaluating the predictions for a particular frame, the last image used for adaptation can be up to 10 frames behind. Besides, the domain shift from Virtual

Model	Road	Building	Pole	T.light	T.sign	Vegetation	Terrain	Sky	Car	Truck	Mean	Mean*
Source	59.2	66.3	**20.3**	5.4	**21.7**	65.8	13.6	73.6	62.4	4.2	39.3	49.3
Online	**85.0**	**72.6**	19.2	**9.1**	20.8	**67.5**	**29.3**	**83.9**	**72.1**	**9.4**	**46.9**	**60.0**

Table 3. Semantic segmentation results (IoU) for online adaptation (Online) from Virtual KITTI v2 to Cityscapes Frankfurt video. Mean* IoU is computed excluding "pole", "light" and "sign". For labeling consistency reasons,"sidewalk" class from Cityscapes is also considered to be "terrain" for the experiments in this table.

	Method	Road	Building	Pole	T.light	T.sign	Veg.	Terrain	Sky	Car	Truck	Mean
V1	DADA [44]*	90.9	76.2	12.4	30.3	30.8	73.5	24.1	**88.4**	**86.8**	**17.2**	53.0
	Chen *et al*. [4]	81.4	71.2	11.3	26.6	23.6	**82.8**	56.5	**88.4**	80.1	12.7	53.5
	Saha *et al*. [35]	90.9	**78.9**	18.1	**32.2**	**38.9**	73.7	22.0	88.2	86.2	16.7	**54.6**
	Baseline	**91.1**	67.6	**18.4**	24.5	24.5	80.0	**59.0**	87.5	81.9	8.5	54.3
V2	DANN [11]*	70.3	49.4	39.5	28.0	22.2	67.0	23.1	82.0	69.4	5.1	45.6
	GUDA [15]	86.8	**72.7**	**46.2**	**41.4**	**44.6**	77.3	29.1	88.5	**86.1**	9.8	58.25
	Baseline	**90.1**	72.4	37.8	31.3	34.9	**83.9**	**58.9**	**89.6**	84.0	**10.9**	**59.4**

Table 4. Semantic segmentation results (IoU) for UDA from Virtual KITTI to KITTI. First column indicates the version of Virtual KITTI used. * Results for [44] and [11] are taken from [35] and [15] respectively.

Weather			mIoU		
Source		Target	NA		OA
Normal	→	Fog	57.8	→	71.4
Normal	→	Rain	72.2	→	82.2

Table 5. Semantic segmentation results under conditions of weather changes in Virtual KITTI v2. NA and OA denote non-adapted and online-adapted models respectively.

KITTI to Cityscapes is larger than to KITTI.

Weather Variations. Weather conditions are an interesting special case of domain change, so we performed a few specific experiments in this context. We utilize Virtual KITTI v2 videos rendered with different weather settings. In particular, normal (sunny) videos are used as source domain, while foggy and rainy – as target respectively. In this videos, every 10-th frame is evaluated while car is moving. Table 5 shows that online adaptation allows to increase mean IoU by 13.6% and 10.0% for foggy and rainy conditions respectively.

4.4. Baseline Comparison

Unfortunately, it is not possible to compare our online method to existing offline approaches using the frames selected for KITTI online semantics benchmark, since neither predictions nor code (or KITTI configuration for [35]) are available for these works. To indicate how our method performs relative to existing methods, it is first compared to our offline baseline in Section 4.3, and the baseline is further evaluated against other offline works on complete KITTI semantics in this section.

Table 4 shows that our source-free baseline achieves higher mean IoU than Chen *et al*. [5] and DADA [44], demonstrating best performance for "road", "pole" and "terrain" classes. State-of-the-art method by Saha *et al*. [35] reaches 0.3% higher mean IoU score compared to our baseline, but they employ a bigger backbone (ResNet101 [16] vs. ResNet18).

Similarly to Saha *et al*. [35], GUDA, which is a concurrent work, also employ ResNet101 encoder. Despite this, and the possible use of the entire Eigen train set [55] (39810 images) for offline UDA to KITTI, our source-free baseline achieves 1.15% higher mean IoU compared to their method. GUDA demonstrates better performance for such classes as "pole", "light" and "sign", presumably due to the explicit use of source data and Eigen train set during UDA, or surface normal regularization. However, our baseline compensates for this by achieving 31.5% higher IoU for "terrain".

5. Conclusion

In this paper, we proposed a new framework – online UDA for semantic segmentation, which poses a more applied alternative to offline approaches. We show that the presented pipeline is competitive with state-of-the-art offline UDA methods when transferring from simulated (Virtual KITTI) to real (KITTI) environment. Extensive experimental evaluation demonstrates the importance of individual components of our algorithm and the effectiveness of proposed design choices in various online UDA scenarios. Finally, in order to encourage further research on the presented topic, we proposed the online semantic segmentation benchmark composed of KITTI subsets.

Acknowledgment. The authors thankfully acknowledge the support by Toyota via the TRACE project.

References

[1] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*, 2018.

[2] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020.

[3] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8001–8008, 2019.

[4] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1841–1850, 2019.

[5] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2018.

[6] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, 27:2366–2374, 2014.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[10] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2016.

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[15] Vitor Guizilini, Jie Li, Rares Ambrus, and Adrien Gaidon. Geometric unsupervised domain adaptation for semantic segmentation. *arXiv preprint arXiv:2103.16694*, 2021.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[18] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[19] Haoshuo Huang, Qixing Huang, and Philipp Krahenbuhl. Domain transfer through deep activation matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 590–605, 2018.

[20] Javed Iqbal and Mohsen Ali. Mlsl: Multi-level self-supervised learning for domain adaptation with spatially independent and semantically consistent labeling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1864–1873, 2020.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Marvin Klingner, Jan-Aike Termöhlen, Jacob Ritterbach, and Tim Fingscheidt. Unsupervised batchnorm adaptation (ubna): A domain adaptation method for semantic segmentation without using source domain representations. *arXiv preprint arXiv:2011.08502*, 2020.

[23] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. Comoda: Continuous monocular depth adaptation using past experiences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2907–2917, 2021.

[24] Kuan-Hui Lee, German Ros, Jie Li, and Adrien Gaidon. Spigan: Privileged adversarial learning from simulation. In *International Conference on Learning Representations*, 2018.

[25] Shunkai Li, Xin Wang, Yingdian Cao, Fei Xue, Zike Yan, and Hongbin Zha. Self-supervised deep visual odometry with online adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6339–6348, 2020.

[26] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321, 1992.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[28] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1215–1224, 2021.

[29] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 415–430. Springer, 2020.

[30] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018.

[31] Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K Roy-Chowdhury, and Manmohan Chandraker. Domain adaptive semantic segmentation using weak labels. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 571–587. Springer, 2020.

[32] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016.

[33] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[35] Suman Saha, Anton Obukhov, Danda Pani Paudel, Menelaos Kanakis, Yuhua Chen, Stamatios Georgoulis, and Luc Van Gool. Learning to relate depth and semantics for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8197–8207, 2021.

[36] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Unsupervised domain adaptation for semantic segmentation with gans. *arXiv preprint arXiv:1711.06969*, 2(2):2, 2017.

[37] M Naseer Subhani and Mohsen Ali. Learning from scale-invariant examples for domain adaptation in semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 290–306. Springer, 2020.

[38] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.

[39] Alessio Tonioni, Oscar Rahnama, Thomas Joy, Luigi Di Stefano, Thalaiyasingam Ajanthan, and Philip HS Torr. Learning to adapt for stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9661–9670, 2019.

[40] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Real-time self-adaptive deep stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 195–204, 2019.

[41] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018.

[42] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1456–1465, 2019.

[43] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.

[44] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7364–7373, 2019.

[45] Qin Wang, Dengxin Dai, Lukas Hoyer, Luc Van Gool, and Olga Fink. Domain adaptive semantic segmentation with self-supervised depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8515–8525, 2021.

[46] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–534, 2018.

[47] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Bridging category-level and instance-level semantic image segmentation. *arXiv preprint arXiv:1605.06885*, 2016.

[48] Jiaolong Xu, Liang Xiao, and Antonio M López. Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 7:156694–156706, 2019.

[49] Jinyu Yang, Weizhi An, Sheng Wang, Xinliang Zhu, Chaochao Yan, and Junzhou Huang. Label-driven reconstruction for domain adaptation in semantic segmentation. In *European Conference on Computer Vision*, pages 480–498. Springer, 2020.

[50] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020.

[51] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6810–6818, 2018.

[52] Zhenyu Zhang, Stéphane Lathuilière, Andrea Pilzer, Nicu Sebe, Elisa Ricci, and Jian Yang. Online adaptation through

meta-learning for stereo depth estimation. *arXiv preprint arXiv:1904.08462*, 2019.

[53] Zhenyu Zhang, Stéphane Lathuilière, Elisa Ricci, Nicu Sebe, Yan Yan, and Jian Yang. Online depth learning against forgetting in monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4494–4503, 2020.

[54] Yiran Zhong, Hongdong Li, and Yuchao Dai. Open-world stereo video matching with deep rnn. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–116, 2018.

[55] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[56] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 568–583, 2018.

[57] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.

[58] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019.