Geometric Unsupervised Domain Adaptation for Semantic Segmentation

Vitor Guizilini  Jie Li  Rareș Ambruș  Adrien Gaidon
Toyota Research Institute (TRI), Los Altos, CA
{first.lastname}@tri.global

Abstract

Simulators can efficiently generate large amounts of labeled synthetic data with perfect supervision for hard-to-label tasks like semantic segmentation. However, they introduce a domain gap that severely hurts real-world performance. We propose to use self-supervised monocular depth estimation as a proxy task to bridge this gap and improve sim-to-real unsupervised domain adaptation (UDA). Our Geometric Unsupervised Domain Adaptation method (GUDA) learns a domain-invariant representation via a multi-task objective combining synthetic semantic supervision with real-world geometric constraints on videos. GUDA establishes a new state of the art in UDA for semantic segmentation on three benchmarks, outperforming methods that use domain adversarial learning, self-training, or other self-supervised proxy tasks. Furthermore, we show that our method scales well with the quality and quantity of synthetic data while also improving depth prediction.

1. Introduction

Self-supervised learning from geometric constraints is used to learn tasks like depth and ego-motion directly from unlabeled videos [13, 22, 24, 51, 76]. However, tasks like semantic segmentation and object detection inherently require human-defined labels. A promising alternative to expensive manual labeling is to use synthetic datasets [4, 9, 14, 45, 46]. Simulators can indeed be programmed to generate large quantities of diverse data with accurate labels (cf. Fig. 1), including for optical and scene flow [30, 44], object detection [42], tracking [14], action recognition [9], and semantic segmentation [45, 46]. However, no simulator is perfect. Hence, effectively using synthetic data requires overcoming the sim-to-real domain gap, a distribution shift between a source synthetic domain and a target real one due to differences in content, scene geometry, physics, appearance, or rendering artifacts.

The goal of Unsupervised Domain Adaptation (UDA) is to improve generalization across this domain gap without any real-world labels. Most methods use adversarial learning for pixel or feature-level adaptation [3, 15, 27, 35, 54, 59, 69] or self-training by refining pseudo-labels [31, 32, 73, 77, 78, 50]. These methods yield clear improvements, but require learning multiple networks beyond the target one, are hard to train (adversarial learning), or limited to semantically close domains (iterative diffusion of high-confidence pseudo-labels). Alternatively, few works [53, 67] have explored simple image-level self-supervised proxy tasks [20, 40, 33] to improve generalization across domains, but with only limited success for UDA of semantic segmentation.

In this work, we introduce self-supervised monocular depth as a proxy task for UDA in semantic segmentation. We propose a multi-task mixed-batch training method combining synthetic supervision with a real-
world self-supervised depth estimation objective to learn a domain-invariant encoder. Although it is not obvious that geometric constraints on videos can help overcome a semantic gap on images, our method, called GUDA for Geometric Unsupervised Domain Adaptation, outperforms other UDA methods for semantic segmentation. Furthermore, we can directly combine our method with self-trained pseudo-labels, leading to a new state of the art on the standard SYNTHIA-to-Cityscapes benchmark. In addition, we show on Cityscapes [7], KITTI [18], and DDAD [24] that our method scales well with both the quantity and quality of synthetic data (cf. Fig. 1), from SYNTHIA [46] to VKITTI2 [4], and a new large-scale high-quality dataset [1]. Finally, we show that GUDA is also capable of state-of-the-art monocular depth estimation in the real-world domain.

2. Related Work

2.1. Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) is an active research area in Computer Vision [8, 62, 64]. Its main goal is to learn a model on a labeled source dataset and a related but statistically different unlabeled target dataset where the model is expected to generalize. Common approaches rely on domain-invariant learning or statistical alignment between domains [15, 56, 68]. In this work, we consider UDA for semantic segmentation, for which the labeling process is tedious and expensive. Several synthetic datasets have been proposed to reduce the need for real-world labels, such as SYNTHIA [46], GTA5 [45], and Virtual KITTI [14].

However, models trained on such datasets suffer from a significant performance drop when tested on real-world datasets. To overcome this large sim-to-real generalization gap, several works have proposed to use adversarial learning for distribution alignment at the level of pixels [27, 65, 70, 71, 69], features [28, 48, 29], or outputs [55, 60, 49]. Alternatively, self-training (a.k.a. pseudo-labeling) has also been successful for UDA [78, 77, 52, 32, 73, 50]. In these works, models are iteratively trained with both ground truth labels in the source domain and inferred pseudo-labels in the target domain, updated as part of an optimization loop.

Although still underexplored, another promising direction is the use of other modalities in the source domain to help the unsupervised transfer of semantic segmentation to the target domain. SPiGAN [35] uses synthetic depth as privileged information to provide additional regularization during adversarial training. GIO-Ada [6] uses geometric information, including depth and surface normals, during style-transfer in the target domain. DADA [61] predicts depth and semantic segmentation during adversarial training using a shared encoder, and fuses depth-aware features to improve semantic segmentation predictions.

In contrast to [6, 35, 61], we do not only use source depth information as explicit supervision or to enforce additional constraints during the adaptation stage. Instead, we infer and leverage depth in the target domain through self-supervision from geometric video-level cues, and use it as the primary source of domain adaptation. By simultaneously learning depth estimation in both domains, we produce features that are discriminative enough to perform this task while being robust to general differences in distribution between target and source domains.

2.2. Self-Supervised Learning

Self-supervised learning (SSL) has recently shown promising results in feature extraction through the definition of auxiliary tasks, using only unlabeled data as input [20, 33, 40, 57]. Typical auxiliary tasks look at different ways of reconstructing input data, such as rotation [20], patch jigsaw puzzles [40], or image colorization [33]. Only few works have used SSL as a tool for domain adaptation [19, 5, 53, 67], reporting results far from the state of the art (cf. Tab. 1). [19] proposes an auxiliary image reconstruction task on the target domain for image classification [19], whereas [5, 53, 67] explore different image-level pretext tasks to improve standard adversarial domain adaptation.

In this paper, we build upon recent developments in self-supervised learning in videos for monocular depth and egomotion estimation [22, 24, 51, 74]. We show that these SSL tasks help UDA, leveraging strong geometric priors from videos to adapt features in a multi-task setting.

3. Geometric Unsupervised Adaptation

A diagram of our proposed architecture for geometric unsupervised domain adaptation (GUDA) is shown in Fig. 2. It is composed of three networks: Depth $f_D : I \rightarrow \hat{D}$, that takes an input image $I$ and outputs a predicted depth map $\hat{D}$; Semantic $f_S : I \rightarrow \hat{S}$, that takes the same input image and outputs a predicted semantic map $\hat{S}$; and Pose $f_{fr} : \{I_a, I_b\} \rightarrow T_{ab}$, that takes a pair of images and outputs the rigid transformation $T$ between them. The depth and semantic networks share the same encoder $f_E : I \rightarrow \hat{F}$, such that $f_D : f_E(I) \rightarrow \hat{D}$ and $f_S : f_E(I) \rightarrow \hat{S}$ both decode the same latent features $\hat{F}$ into their respective tasks. See Sec. 4.2 for architecture details.

During training we employ a mixed-batch approach, in which at each iteration real $B_R$ and virtual $B_V$ batches are received and processed to generate corresponding real $L_R$ (Sec. 3.1) and virtual $L_V$ (Sec. 3.2) losses, depending on the available information. The final loss is defined as:

$$L = L_R + \lambda_V L_V$$

(1)

where $\lambda_V$ is a coefficient used to balance real and virtual losses during optimization. The next sections detail how each of these losses is calculated.
Figure 2: Diagram of our proposed multi-task multi-domain GUDA architecture for geometric unsupervised domain adaptation using mixed-batch training of real (Sec. 3.1) and virtual (Sec. 3.2) samples. The common paths during training (self-supervised) are in blue, whereas other ones (gray) use synthetic supervision.

3.1. Real (Target) Sample Processing

Real-world samples are assumed to contain only unlabeled image sequences \( I_t \), in the form of the current frame \( I_t \) and a temporal context \( \{I_{t-s}, \ldots, I_{t+s}\} \). In all experiments we considered a temporal context of \( s = 1 \), resulting in \( I_t = \{I_{t-1}, I_t, I_{t+1}\} \). For simplicity, we also assume known and constant camera intrinsics \( K \) for all frames, however this assumption can be relaxed to include the simultaneous learning of the projection model [23, 58]. Following [53], we use an auxiliary self-supervised task in the target domain to help adapt features learned in the source domain. Specifically, depth and ego-motion learning via self-supervised photometric consistency in videos has been shown to produce results competitive with supervised learning in some domains [24, 74]. Leveraging this insight, we define our target domain loss as:

\[
L_R = L_P + \lambda_{PL}L_{PL} \tag{2}
\]

where \( L_P \) is the self-supervised photometric loss described in Section 3.1.1, and \( L_{PL} \) is an optional pseudo-label loss described in Section 3.1.2, with weight coefficient \( \lambda_{PL} \).

3.1.1 Self-Supervised Photometric Loss

Following previous works [16, 74], the self-supervised depth and ego-motion objective can be formulated as a novel view synthesis problem, in which a target image \( I_t \) is reconstructed using information from a reference image \( I_{t'} \) given a predicted depth map \( \hat{D}_t \) and relative transformation \( T_{t' t}^{r} \) between images:

\[
\hat{I}_t = I_{t'} \langle \pi(\hat{D}_t, \hat{T}_{t' t}^{r}, K) \rangle \tag{3}
\]

where \( \pi \) is the projection operation determined by camera geometry and \( \langle \rangle \) is the bilinear sampling operator, that is locally sub-differentiable and thus can be used as part of an optimization pipeline. To measure the reconstruction error we use the standard photometric loss [72], with a structural similarity (SSIM) component [63] and the L1 distance in pixel space, weighted by \( \alpha = 0.85 \):

\[
L_P(I_t, \hat{I}_t) = \alpha \left(1 - \text{SSIM}(I_t, \hat{I}_t)\right) + (1 - \alpha)||I_t - \hat{I}_t||_1 \tag{4}
\]

This loss is calculated for every image \( I_{t'} \in I_t \) and averaged for all pixels between multiple scales, after upsampling to the highest resolution. Following [22], we use auto-masking and minimum reprojection error to mitigate effects caused by occlusions and dynamic objects.

3.1.2 Pseudo-Label Distillation

Self-training methods [32, 73, 78] are currently the dominant framework to address unsupervised domain adaptation for several different tasks [47]. They work by iteratively refining high-confidence pseudo-labels in the target domain using supervision in the source domain. This source of domain adaptation can in principle be used to augment our proposed domain adaptation via geometric self-supervision.
Here we propose a simple yet effective way to distill information from self-training methods (or any other UDA method) into GUDA, by using pre-calculated pseudo-labels as supervision in the target domain. The resulting loss is similar to the supervised semantic loss described in Sec. 3.2.1, using the predicted semantic map $\hat{S}$ from the real sample and the pseudo-label $S_{PL}$ pre-calculated from the same input image $I$ as ground-truth:

$$L_{PL} = L_S(\hat{S}, S_{PL})$$  \hspace{1cm} (5)

In our ablation analysis (Tab. 2), we show that the combination of GUDA with pseudo-label supervision from a self-training method [73] achieves the best results, surpassing other methods and establishing a new state of the art in unsupervised domain adaptation for semantic segmentation.

### 3.2. Virtual (Source) Sample Processing

Virtual samples consist of input images $I_i$ with corresponding dense annotations for all considered tasks, i.e. depth maps $D_i$ and semantic labels $S_i$. If sequential data is available we also assume a temporal context $I_i = \{I_{i-1}, I_i, I_{i+1}\}$, corresponding ground-truth rigid transformation between frames $T_i = \{T_{i-1}^t, T_{i+1}^t\}$, and constant camera intrinsics $K$. The availability of supervision allows the learning of both semantic and depth tasks in the source domain, encoding this information into the shared encoder $f_E$ and the respective decoders $f_D$ and $f_S$. We define our source domain loss as follows:

$$L_V = L_D + \lambda_S L_S + \lambda_N L_N + \lambda_{PP} L_{PP}$$  \hspace{1cm} (6)

where $L_S$ is a supervised semantic loss (Sec. 3.2.1), $L_D$ is a supervised depth loss (Sec. 3.2.2), $L_N$ is a surface normal regularization term (Sec. 3.2.3), and $L_{PP}$ is an optional partially-supervised photometric loss (Sec. 3.2.4), each weighted by their corresponding coefficient.

#### 3.2.1 Supervised Semantic Loss

Following [66, 43], we supervise semantic segmentation in the source domain using a bootstrapped cross-entropy loss between predicted $\hat{S}$ and ground-truth $S$ labels:

$$L_S = -\frac{1}{K} \sum_{u=1}^{H} \sum_{v=1}^{W} \sum_{c=1}^{C} \mathbb{1}_{[c=y_{u,v}, p_{u,v}^c < t]} \log \left( p_{u,v}^c \right)$$  \hspace{1cm} (7)

where $p_{u,v}^c$ denotes the predicted probability of pixel $(u, v)$ belonging to class $c$. The term $t$ is a run-time threshold so that only the worst $K$ performing predictions are considered. We adopt $K = 0.3 \times H \times W$ in our experiments.

#### 3.2.2 Supervised Depth Loss

Our supervised objective loss is the Scale-Invariant Logarithmic loss (SILog) [12], composed by the sum of the variance and the weighted squared mean of the error in log space $\Delta d = \log d - d$:

$$L_D = \frac{1}{P} \sum_{d \in D} \Delta d^2 - \lambda \left( \sum_{d \in D} \Delta d \right)^2$$  \hspace{1cm} (8)

where $P$ is the number of pixels $d \in D$ with valid depth information. The coefficient $\lambda$ balances variance and error minimization, and following previous works [34] we use $\lambda = 0.85$ in all experiments.

#### 3.2.3 Surface Normal Regularization

Because depth estimates are produced on a per-pixel basis, it is common to enforce an additional smoothness loss [21] to maintain local consistency. Here we propose an alternative to the smoothness loss, that leverages the dense depth supervision available in synthetic datasets and minimizes instead the difference between surface normal vectors produced by ground-truth and predicted depth maps. Note that, differently from other methods [11, 39], we are not explicitly predicting surface normals as an additional task, but rather using them as regularization to enforce certain structural properties in the predicted depth maps. For any pixel $p \in D$, its surface normal vector $n \in \mathbb{R}^3$ is calculated as:

$$n = \left( P_{u+1,v} - P_{u,v} \right) \times \left( P_{u,v+1} - P_{u,v} \right)$$  \hspace{1cm} (9)

where $P = \phi(p, d, K)$ is the point obtained by unprojecting $p$ from the camera frame of reference into 3D space, given its depth value $d$ and intrinsics $K$. As a measure of proximity between two surface normal vectors we use the cosine similarity metric, defined as:

$$L_N = \frac{1}{2P} \sum_{p \in D} \left( 1 - \frac{n \cdot \hat{n}}{||n|| \cdot ||\hat{n}||} \right)$$  \hspace{1cm} (10)

where $n$ and $\hat{n}$ are unitary vectors representing respectively ground-truth and predicted surface normals for each pixel.
epochs with the introduction of 10 pseudo-labels are available, the model is refined for another pact in final model performance, as shown in Tab. 2. If to increase convergence speed and has no meaningful im-
10 by 20 one order of magnitude.

stable, starting to vary with coefficient changes of around λ determined the loss coefficients to be
5

8541

epochs at full resolution, halving the learning rate
for the final
5

epochs. This multi-scale schedule is used

4.2. Networks

Unless noted otherwise, we use a ResNet101 [26] with ImageNet [10] pre-trained weights as the shared backbone for the semantic and depth decoders. The depth decoder follows [22] and outputs inverse depth maps at four different resolutions. The semantic decoder is similar and outputs semantic logits at a single resolution, obtained by concatenating the four output scales (upsampled to the highest resolution) followed by a final convolution layer. Our pose network also follows [22] and uses a ResNet18 encoder (also pre-trained on ImageNet), followed by a series of convolutions that output a 6-dimensional vector containing translation and rotation in Euler angles. For more details we refer the reader to the supplementary material.

4.3. Datasets

4.3.1 Real Datasets

Cityscapes [7] The Cityscapes dataset is a widely used benchmark for semantic segmentation evaluation. For self-supervision on the target domain, following [24], we use the 2975 training images (without the labels) with their corresponding 30-frame sequences, for a total of 2975 × 30 = 89250 images. In Tab. 2 we ablate the impact of training with fewer frames, such as 2-frame sequences (the minimum required for monocular self-supervised learning). We evaluate our semantic segmentation performance on the official 500 annotated validation set.

KITTI [17] The KITTI dataset is considered the standard benchmark for depth evaluation. We use the Eigen split filtered according to [75], resulting in 39810 training, 888 validation and 697 test images, with corresponding LiDAR-projected depth maps (used only for evaluation). To evaluate semantic segmentation we use the 200 annotated frames found in [2], mapped to the Cityscapes ontology [7].

DDAD [24] The Dense Depth for Automated Driving dataset is a challenging depth evaluation benchmark, with denser ground-truth depth maps and longer ranges of up to 250m. We consider only the front camera, resulting in 150 training sequences with 12560 images and 50 validation sequences with 3950 images, from which 50 are semantically labeled (the middle frame in each sequence).

4.3.2 Synthetic Datasets

SYNTHIA [46] The SYNTHIA dataset contains scenes generated from an autonomous driving simulator of urban scenes. For a fair comparison with other methods we used the SYNTHIA-RAND-CITYSCAPES subset, with 9400 images and semantic labels compatible with Cityscapes.
### 5. Experimental Results

#### 5.1. Semantic Segmentation

First, we evaluate our proposed GUDA framework on the task of unsupervised domain adaptation for semantic segmentation using the Cityscapes dataset. We consider three different scenarios, with results shown in Tab. 1. In (a) we use the SYNTHIA dataset as source and compare against other methods that use depth in the source domain as additional supervision, either as regularization [35], to reduce domain shift at different feature levels [6], or by sharing a lower-level representation like us. From these results, we see that **GUDA outperforms all previous methods**, even though it does not leverage additional translation networks or adversarial training. These results confirm that training the depth network in both domains thanks to self-supervised geometric constraints on target videos improves the generalization of the shared intermediate representation.

A more detailed analysis (available in the supplementary material) shows that **GUDA excels in classes with well-defined geometries**, such as road, sidewalk, and building. Interestingly, it also performs well on sky, most likely due to our proposed surface normal regularization (Fig. 3). We also note that GUDA's smallest improvements are on rarer dynamic classes (e.g., motorcycle). This stems from the photometric loss being unable to model dynamic object motion due to a static world assumption [22, 25].

To overcome this limitation, we introduce pseudo-label supervision (Sec. 3.1.2) from USAMR [73], obtained by evaluating the official pre-trained model (mIoU 46.5) on our 89250 training images. In this configuration, **GUDA achieves state-of-the-art results**, outperforming UDA methods relying on style-transfer [70, 69], adversarial learning [38], self-training [78, 32, 36, 50], or other forms of (non-geometric) self-supervision [67, 53]. Fig. 4 analyzes the interplay between geometric self-supervision (GS) and pseudo-labels (PL), indicating an optimal pseudo-labeling loss weight $\lambda_{PL} = 0.01$.

A detailed ablation study of our proposed architecture can be found in Tab. 2. It shows that (i) geometric supervision by itself improves performance, (ii) all components help, and (iii) the benefits of our method are not due simply to using more target data (video frames), although GUDA can benefit from them in contrast to other approaches.

Finally, in Tab. 1 (b) we present results considering different source datasets. Because GTA5 [45] does not provide depth labels, we instead report results using the photorealistic Parallel Domain dataset, with GTA5 pseudo-labels (mIoU 53.1) from USAMR [73]. In this configuration, **GUDA once more outperforms other considered methods**, improving upon the state of the art when different source datasets are considered.

---

**Table 1: Semantic segmentation results on Cityscapes using different unsupervised domain adaptation (UDA) methods and synthetic datasets. mIoU considers all 16 classes, and mIoU* only the 13 SYNTHIA classes. Source shows results without any adaptation, and Target shows results with semantic supervision on the target domain. Synthetic datasets include: SYNTHIA (SY), Parallel Domain (PD), and GTA5 (G5). Detailed per class results are given in the supplementary material.**

<table>
<thead>
<tr>
<th>Method</th>
<th>Split-Trans.</th>
<th>Advers.</th>
<th>Depth</th>
<th>Self-Sup.</th>
<th>Ps.-Label</th>
<th>mIoU</th>
<th>mIoU*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source (SY)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>31.7</td>
<td>36.7</td>
</tr>
<tr>
<td>Source (PD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>38.1</td>
<td>44.0</td>
</tr>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>72.9</td>
<td>77.8</td>
</tr>
<tr>
<td>(a) SYNTHIA → Cityscapes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>36.8</td>
<td>42.4</td>
</tr>
<tr>
<td>SPIGAN [35]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>37.3</td>
<td>43.0</td>
</tr>
<tr>
<td>GIO-Ada [6]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>42.6</td>
<td>49.8</td>
</tr>
<tr>
<td>DADA [61]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>42.5</td>
<td>47.8</td>
</tr>
<tr>
<td>GUDA</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>44.5</td>
<td>50.9</td>
</tr>
<tr>
<td>CLAN [38]</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>43.8</td>
<td>50.1</td>
</tr>
<tr>
<td>Xu et al. [67]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>43.5</td>
<td>50.7</td>
</tr>
<tr>
<td>CBST [78]</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>42.5</td>
<td>48.4</td>
</tr>
<tr>
<td>CSRST [77]</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>43.8</td>
<td>50.1</td>
</tr>
<tr>
<td>ESL [50]</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>43.5</td>
<td>50.7</td>
</tr>
<tr>
<td>FDA [70]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>52.5</td>
<td></td>
</tr>
<tr>
<td>CCMD [36]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>45.2</td>
<td>52.6</td>
</tr>
<tr>
<td>Yang et al. [69]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>46.5</td>
<td>53.8</td>
</tr>
<tr>
<td>USAMR [73]</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49.8</td>
<td>57.0</td>
</tr>
<tr>
<td>IAST [32]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>51.0</td>
<td>57.9</td>
</tr>
<tr>
<td>GUDA+PL</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>44.3</td>
<td>49.2</td>
</tr>
<tr>
<td>(b) Varying Sources (G5, PD) → Cityscapes</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>53.1</td>
<td>58.0</td>
</tr>
<tr>
<td>USAMR (G5) [73]</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>55.6</td>
<td>61.2</td>
</tr>
<tr>
<td>IAST (G5) [32]</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>57.2</td>
<td>63.2</td>
</tr>
</tbody>
</table>

**VKITTI2 [4]** The VKITTI2 dataset was recently released as a more photo-realistic version of [14], containing reconstructions of five sequences found in the KITTI odometry benchmark [18] for a total of 2156 samples.

**Parallel Domain [1]** The Parallel Domain dataset\(^2\) is procedurally-generated and contains fully annotated photorealistic renderings of urban driving scenes, including multiple cameras and LiDAR sensors. It contains 5000 10-frame sequences, with 4200/800 training and validation splits. More details and examples can be found in the supplementary material.

**GTA5 [45]** The GTA5 dataset is a street view synthetic dataset rendered from the GTA5 game engine, containing 24966 images and semantic labels defined on 19 classes compatible with Cityscapes.

\(^2\)https://paralleldomain.com/public-datasets/
Figure 4: Effects of pseudo-label supervision, with and without our proposed geometric self-supervision (GS) for different values of $\lambda_{PL}$. When GS is not used (blue line), there is no benefit in adding virtual supervision, and as $\lambda_{PL}$ increases results converge to those using only pseudo-label supervision (Tab. 2). When GS is used (red line), results are consistently higher and start to degrade after $\lambda_{PL} = 0.01$.

Table 2: Ablation study of our proposed method (SYNTHIA $\rightarrow$ Cityscapes). In the Real columns, GT refers to the use of ground-truth semantic labels, PL to the use of USAMR [73] semantic pseudo-labels (Sec. 3.1.2), and GS to the use of geometric self-supervision (Sec. 3.1.1). In the Virtual columns, D, S, and N refer respectively to semantic, depth, and surface normal supervision (Sec. 3.2). DANN refers to the use of an additional domain adversarial loss [15], and single res. to training only in full resolution. We also provide DADA results trained using 2 and 30 context frames (rather than 1), as well as GUDA results using only the minimum 2 context frames (rather than 30).

Figure 5: Semantic segmentation results on VKITTI2 $\rightarrow$ KITTI, using GUDA and DANN [15]. Detailed numbers are available in the supplementary material.

Figure 6: Semantic segmentation results on Parallel Domain $\rightarrow$ DDAD, using GUDA and DANN [15]. Detailed numbers are available in the supplementary material.

Figure 7: Performance improvement on Cityscapes with increasing data quality (SYNTHIA $\rightarrow$ PD) and quantity.

work (DANN) baseline [15], which uses a gradient reversal layer to learn discriminative features for the main task on the source domain while maximizing domain confusion.

As expected, DANN improves over source-only results by a significant margin (+7.0/ +5.1 mIoU). Nevertheless, the stronger geometric supervision from GUDA still substantially outperforms DANN (+12.65/ +8.58 mIoU), with similar trends as observed in previous experiments. In Fig. 7 we show how GUDA and DANN scale with improvements in data quality (from SYNTHIA to Parallel Domain) and data quantity (different subsets of Parallel Domain). Assuming linear improvement, GUDA would only require 200k synthetic samples to fully overcome the domain gap, whereas DANN would require 350k samples. We also show in Tab. 2 that GUDA and DANN can be combined, resulting in a +0.2/0.3% improvement.
5.2. Depth Estimation

As stated previously, GUDA achieves domain adaptation by jointly learning depth features in both domains, using a combination of dense synthetic supervision and geometric self-supervision on real-world images. To further demonstrate this property, in this section we analyze how GUDA impacts the task of monocular depth estimation itself and improves upon the standard approach of model fine-tuning. Similar to previous experiments, we evaluate on VKITTI2 to KITTI and Parallel Domain to DDAD, noting that each of these combinations have similar sensor configuration (intrinsic and extrinsic), which makes them particularly suitable for domain adaptation experiments. The same training schedule and architecture was used, with the inclusion of experiments using a ResNet18 backbone to facilitate comparison with other methods.

Quantitative results are shown in Tab. 3, with qualitative examples in Fig. 9. The first noticeable aspect is that direct transfer (Source only) not only produces relatively accurate, but also scale-aware results, due to similarities in vehicle extrinsics and camera parameters. As expected, this behavior is not observed when only real-world informa-

<table>
<thead>
<tr>
<th>Method</th>
<th>Abs.Rel↓</th>
<th>Sqr.Rel↓</th>
<th>RMSE↓</th>
<th>δ &lt; 1.25↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source only† (R18)</td>
<td>0.191</td>
<td>2.078</td>
<td>7.233</td>
<td>0.699</td>
</tr>
<tr>
<td>Target only (R18)</td>
<td>0.117</td>
<td>0.811</td>
<td>4.902</td>
<td>0.867</td>
</tr>
<tr>
<td>Fine-tune (R18)</td>
<td>0.114</td>
<td>0.800</td>
<td>4.855</td>
<td>0.871</td>
</tr>
<tr>
<td>GUDA† (R18) - PS</td>
<td>0.114</td>
<td>0.875</td>
<td>4.808</td>
<td>0.871</td>
</tr>
<tr>
<td>GUDA† (R18)</td>
<td>0.109</td>
<td>0.762</td>
<td>4.606</td>
<td>0.879</td>
</tr>
<tr>
<td>GUDA†</td>
<td>0.107</td>
<td>0.714</td>
<td>4.421</td>
<td>0.883</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Abs.Rel↓</th>
<th>Sqr.Rel↓</th>
<th>RMSE↓</th>
<th>δ &lt; 1.25↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source only† (R18)</td>
<td>0.233</td>
<td>7.429</td>
<td>18.498</td>
<td>0.620</td>
</tr>
<tr>
<td>Target only (R18)</td>
<td>0.203</td>
<td>6.999</td>
<td>16.844</td>
<td>0.748</td>
</tr>
<tr>
<td>Fine-tune (R18)</td>
<td>0.173</td>
<td>4.846</td>
<td>16.025</td>
<td>0.747</td>
</tr>
<tr>
<td>GUDA† (R18) - PS</td>
<td>0.166</td>
<td>3.556</td>
<td>16.004</td>
<td>0.769</td>
</tr>
<tr>
<td>GUDA† (R18)</td>
<td>0.158</td>
<td>3.332</td>
<td>15.112</td>
<td>0.778</td>
</tr>
<tr>
<td>GUDA†</td>
<td>0.147</td>
<td>2.922</td>
<td>14.452</td>
<td>0.809</td>
</tr>
</tbody>
</table>

Table 3: Depth estimation results on KITTI and DDAD. R18 indicates a ResNet18 [26] backbone, Source only and Target only indicate only synthetic or real-world, and Fine-tune indicates synthetic pre-training followed by real-world fine-tuning. PS indicates the removal of the partially-supervised photometric loss (Sec. 3.2.4). The symbol † indicates a scale-aware model (no test-time median-scaling).

Figure 8: GUDA semantic segmentation results on Cityscapes, DDAD and KITTI.

Figure 9: GUDA depth estimation results on KITTI and DDAD. Our proposed mixed-batch training schedule produces much sharper and consistent depth maps, especially at longer ranges and in “invalid” areas such as the sky. When fine-tuning (Target only) is used, however it is also not observed when fine-tuning, indicating a catastrophic forgetting of the scale factor. In contrast, and although not our primary goal, GUDA preserves the scale learned from synthetic supervision and also significantly improves depth estimation performance relative to the standard fine-tuning approach. In alignment with recent observations [24], switching to a larger backbone improves results even further.

6. Conclusion

We introduce self-supervised monocular depth estimation as a proxy task for unsupervised sim-to-real transfer of semantic segmentation models. Our Geometric Unsupervised Domain Adaptation method, GUDA, combines dense synthetic labels with self-supervision from real-world unlabeled videos to bridge the sim-to-real domain gap. Although depth estimation is fundamentally a geometric task, we show it improves semantic representation transfer without any real-world semantic labels. Our multi-task self-supervised method outperforms other UDA approaches, while also improving monocular depth estimation. Furthermore, by introducing self-trained pseudo-labels as an extra source of supervision, we establish a new state of the art on this task. Finally, we show that our method scales well with both the quantity and quality of synthetic data, highlighting its potential to eventually close the sim-to-real gap in challenging visual conditions like driving scenes.
References


[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 IEEE conference on computer vision and pattern recognition, pages 3213–3223, 2016. 2, 5


[14] 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. 1, 2, 6


[22] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In ICCV, 2019. 1, 2, 3, 5, 6


[34] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326, 2019. 4
[54] Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In Association for the Advancement of Artificial Intelligence (AAAI), 2020. 1


