International Journal of Computer Vision
https://doi.org/10.1007/s11263-023-01799-6

Improving Semi-Supervised and Domain-Adaptive Semantic Segmentation with Self-Supervised Depth Estimation

Lukas Hoyer[1] · **Dengxin Dai**[2] · **Qin Wang**[1] · **Yuhua Chen**[1] · **Luc Van Gool**[1,3]

Received: 27 August 2021 / Accepted: 12 April 2023

Abstract
Training deep networks for semantic segmentation requires large amounts of labeled training data, which presents a major challenge in practice, as labeling segmentation masks is a highly labor-intensive process. To address this issue, we present a framework for semi-supervised and domain-adaptive semantic segmentation, which is enhanced by self-supervised monocular depth estimation (SDE) trained only on unlabeled image sequences. In particular, we utilize SDE as an auxiliary task comprehensively across the entire learning framework: First, we automatically select the most useful samples to be annotated for semantic segmentation based on the correlation of sample diversity and difficulty between SDE and semantic segmentation. Second, we implement a strong data augmentation by mixing images and labels using the geometry of the scene. Third, we transfer knowledge from features learned during SDE to semantic segmentation by means of transfer and multi-task learning. And fourth, we exploit additional labeled synthetic data with Cross-Domain DepthMix and Matching Geometry Sampling to align synthetic and real data. We validate the proposed model on the Cityscapes dataset, where all four contributions demonstrate significant performance gains, and achieve state-of-the-art results for semi-supervised semantic segmentation as well as for semi-supervised domain adaptation. In particular, with only 1/30 of the Cityscapes labels, our method achieves 92% of the fully-supervised baseline performance and even 97% when exploiting additional data from GTA. The source code is available at https://github.com/lhoyer/improving_segmentation_with_selfsupervised_depth.

Keywords Semantic segmentation · Self-supervised depth estimation · Semi-supervised learning · Domain adaptation

1 Introduction

Convolutional Neural Networks (CNNs) (LeCun et al., 1998) have achieved state-of-the-art results for various computer vision tasks including semantic segmentation (Long et al., 2015; Chen et al., 2017). However, training CNNs typically requires large-scale annotated datasets, due to millions of learnable parameters involved. Collecting such training data relies primarily on manual annotation. For semantic segmentation, the process can be particularly costly, due to the required dense annotations. For example, annotating a single image of the Cityscapes dataset took on average 1.5 h (Cordts et al., 2016). For the training set, this sums up to 4460 working hours only for the annotation. For more challenging environmental conditions such as fog, snow, or nighttime, the annotation can be even more expensive. For instance, the annotation of one image of the ACDC dataset (Sakaridis et al., 2021) took 3.3 h on average.

Recently, self-supervised learning (Doersch et al., 2015; Gidaris et al., 2018; He et al., 2020) has shown to be a promising replacement for manually labeled data. It aims to learn representations from the structure of unlabeled data, instead of relying on a supervised loss, which requires manual labels. In particular, the principle has successfully been applied in

Communicated by Karteek Alahari.

✉ Lukas Hoyer
lhoyer@vision.ee.ethz.ch

Dengxin Dai
ddai@mpi-inf.mpg.de

Qin Wang
qwang@ethz.ch

Yuhua Chen
yuhua.chen@vision.ee.ethz.ch

Luc Van Gool
vangool@vision.ee.ethz.ch

[1] ETH Zurich, Zurich, Switzerland

[2] MPI for Informatics, Saarbrücken, Germany

[3] KU Leuven, Leuven, Belgium

Published online: 11 May 2023

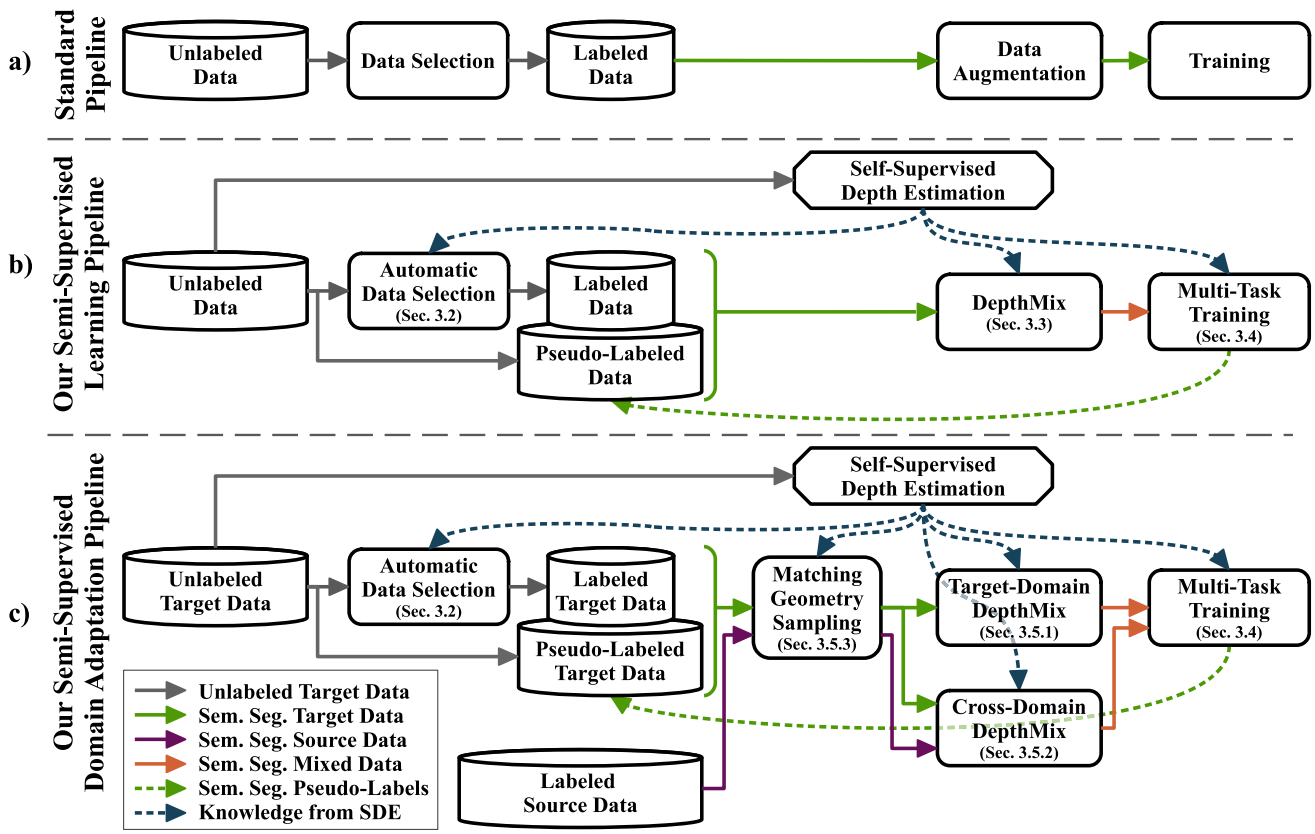


Fig. 1 Our method utilizes self-supervised depth estimation (SDE) in order to improve the holistic learning process of semantic segmentation. In comparison to the standard learning pipeline (**a**), we learn SDE from unlabeled image sequences and use it to improve the data selection, data augmentation, and training process (**b**). Further, we extend our framework to semi-supervised domain adaptation (SSDA), where SDE is used to align domains by Matching Geometry Sampling and Cross-Domain DepthMix (**c**)

depth estimation for stereo pairs (Godard et al., 2017) or image sequences (Zhou et al., 2017). Additionally, semantic segmentation is known to be tightly coupled with depth. For example, sky is always far away, traffic lights are usually closer than their surrounding, and depth discontinuities often coincide with semantic segmentation borders. Several works (Vandenhende et al., 2021; Xu et al., 2018; Liu et al., 2019; Chen et al., 2019b) have reported that jointly learning segmentation and *supervised* depth estimation can benefit the performance of both tasks. Motivated by these observations, we investigate the question: *How can we leverage self-supervised depth estimation to improve semantic segmentation?*

In this work, we propose to use self-supervised monocular depth estimation (SDE) (Godard et al., 2017; Zhou et al., 2017; Godard et al., 2019) to improve the performance of semantic segmentation and to reduce the number of necessary annotations. For this purpose, we consider the holistic learning process covering data selection for annotation, data augmentation, domain adaptation, and multi-task learning. For each step, we show how SDE can effectively be utilized to improve the semantic segmentation performance. In contrast to most previous works, which only exploit *supervised* depth information during the multi-task learning (Vandenhende et al., 2021), we resort to *self-supervised* depth estimation as an auxiliary task comprehensively across the entire learning pipeline and show that it is critical to effectively improve the segmentation performance.

We apply our framework to the semi-supervised learning (SSL) and the semi-supervised domain adaptation (SSDA) setting. In SSL, only a part of the underlying dataset is labeled for semantic segmentation, while in SSDA additional labeled data from another (often synthetic) domain is provided. Figure 1 compares the standard learning pipeline (Fig. 1a) with our SDE-enhanced framework for SSL (Fig. 1b) and our method for SSDA (Fig. 1c).

In our SSL framework (see Fig. 1b), we utilize SDE learned on unlabeled image sequences, to improve the learning pipeline at three places.

First, we propose an *automatic data selection for annotation*, which selects the most useful samples to be annotated in order to maximize the gain. The selection is iteratively driven

by two criteria: *diversity* and *uncertainty*. Both of them are conducted by a novel use of SDE as a proxy task in this context. While our method follows the active learning cycle (i.e. model training → query selection → annotation → model training) (Settles, 2009), it does not require a human in the loop to provide semantic segmentation labels as the human is replaced by a proxy-task SDE oracle. This greatly improves flexibility, scalability, and efficiency, especially considering using crowdsourcing platforms for annotation.

Second, we propose a strong data augmentation strategy, *DepthMix*, which blends images as well as their labels according to the geometry of the scenes obtained from SDE. In comparison to previous methods (Yun et al., 2019; Olsson et al., 2021), DepthMix explicitly respects the geometric structure of the scenes and generates realistic occlusions as the distance of objects to the camera is considered.

And third, we deploy SDE as an auxiliary task for semantic image segmentation under a transfer learning and multi-task learning framework and show that it noticeably improves the performance of semantic segmentation, especially when semantic supervision is limited. Previous works focus on improving SDE instead of semantic segmentation (Chen et al., 2019c; Guizilini et al., 2020b) or only consider the special cases of full supervision (Klingner et al., 2020b) and pretraining (Jiang et al., 2018).

Furthermore, we extend the contributions from SSL to SSDA in order to take advantage of additional synthetic (source) training data (see Fig. 1c). As synthetic data can often be annotated automatically for semantic segmentation, it is a valuable source of supervision and can further reduce the annotation effort for the real (target) data. We demonstrate that the previous contributions are effective for SSDA as well. In order to better bridge the domain gap between source data and target data, we combine the previous *Target-Domain DepthMix* (i.e. the single-domain DepthMix of our SSL method applied to the target domain) with an additional *Cross-Domain DepthMix*, which mixes samples from the source domain and the target domain. In that way, our framework is able to align the distribution of labeled source data with labeled target data (Cross-Domain DepthMix) and unlabeled target data with labeled target data (Target-Domain DepthMix). As the geometric distribution of the source domain is not aligned with the target domain and the Cross-Domain DepthMix can suffer from blending samples with different geometric distributions, we further introduce a *Matching Geometry Sampling* based on SDE to better align the camera pose and scene geometry of the source samples with the target domain.

The main advantage of our method is that we can learn from a large base of easily accessible unlabeled image sequences and use the learned knowledge from SDE to improve semantic segmentation performance over the entire training process. This largely alleviates the need for expensive semantic segmentation annotations. In our experimental evaluation on Cityscapes (Cordts et al., 2016), we demonstrate significant performance gains of all four components and improve the previous state of the art for SSL as well as for SSDA by a considerable margin. Importantly, our contributions are complementary and yield even higher improvements when they are combined in a unified framework. Specifically, in an SSL setting, our method achieves 92% of the fully-supervised model performance with only 1/30 available labels and even slightly outperforms the fully-supervised model with only 1/8 labels. In the SSDA setting with additional supervision from the synthetic GTA5 dataset (Richter et al., 2016), our method achieves even 97% of the fully-supervised model performance with only 1/30 of the target labels.

Our contributions summarize as follows:

(1) We propose a novel *automatic data selection for annotation* based on SDE to improve the flexibility of active learning for semantic segmentation. It replaces the human annotator with an SDE oracle and lifts the requirement of having a human in the loop of active learning.
(2) We propose *DepthMix*, a strong data augmentation strategy based on self-supervised depth estimation, which respects the geometry of the scene.
(3) We utilize SDE as an auxiliary task to exploit depth features learned on unlabeled image sequences to significantly improve the performance of semantic segmentation by transfer and multi-task learning. In combination with (1) and (2), we achieve state-of-the-art results for semi-supervised semantic segmentation on Cityscapes.
(4) We propose a novel semi-supervised domain adaptation method, which combines *Target-Domain DepthMix* with *Cross-Domain DepthMix*. Further, *Matching Geometry Sampling* aligns the camera pose and scene geometry during the mixing process towards the target domain. We show that our method achieves state-of-the-art results for SSDA on GTA5→Cityscapes and Synthia→Cityscapes.

This work is an extension of our IEEE Conference on Computer Vision and Pattern Recognition 2021 paper (Hoyer et al., 2021), which focuses on the contributions (1–3). This article further introduces SSDA utilizing SDE both using the previous contributions for SSL as well as the newly proposed combined Cross-Domain/Target-Domain DepthMix and the Matching Geometry Sampling. Also, we extend the ablation studies, detail the analysis (e.g. by class-wise performance insights and by a class frequency analysis of the data selection), and improve the presentation of the unified SDE-enhanced learning framework.

2 Related Work

2.1 Self-supervised Depth Estimation (SDE)

Self-supervised depth estimation (SDE) aims to learn depth estimation from the geometric relations of stereo image pairs (Garg et al., 2016; Godard et al., 2017) or monocular videos (Zhou et al., 2017). Due to the better availability of videos, we use the latter approach, where a neural network estimates the depth and the camera motion of two subsequent images and a photometric loss is computed after a differentiable warping. If the camera intrinsics are not known, their estimation can be incorporated into the learning process as well (Gordon et al., 2019). Follow-up works propose improvements of the loss function (Godard et al., 2019; Gonzalez Bello & Kim, 2020; Shu et al., 2020), network architecture (Wang et al., 2019; Guizilini et al., 2020a), and training scheme (Pilzer et al., 2018, 2019; Casser et al., 2019). To handle dynamic objects, several works (Yin & Shi, 2018; Chen et al., 2019c; Ranjan et al., 2019) extend the projection model and combine depth estimation with optical flow estimation.

2.2 Active Learning

Active learning methods iteratively select the most informative samples to be annotated. Two main directions for the selection heuristic can be differentiated. On the one side, uncertainty-based approaches select samples with a high uncertainty estimated based on, e.g., entropy (Hwa, 2004; Settles & Craven, 2008) or ensemble disagreement (Seung et al., 1992; McCallumzy & Nigamy, 1998). However, this can be prone to querying outliers. On the other side, diversity-based approaches select samples, which most increase the diversity of the labeled set (Sener & Savarese, 2018; Sinha et al., 2019). For segmentation, active learning is typically based on uncertainty measures such as MC dropout (Gal & Ghahramani, 2016; Yang et al., 2017; Mackowiak et al., 2018), entropy (Kasarla et al., 2019; Xie et al., 2020), or multi-view consistency (Siddiqui et al., 2020). In contrast to these works, we perform automatic data selection for annotation by replacing the human with an SDE model as oracle. Therefore, we do not require human-in-the-loop annotation during the active learning cycle. Previous works performing data selection without a human in the loop are restricted to shallow models (Yu et al., 2006; Nie et al., 2013; Li et al., 2018), classification with low-dimensional inputs (Li et al., 2020a), or do not perform an iterative data selection (Zheng et al., 2019) to dynamically adapt to the uncertainty of the model trained on the currently labeled set.

2.3 Semi-supervised Semantic Segmentation

Semi-supervised semantic segmentation makes use of additional unlabeled data during training. An early line of work (Souly et al., 2017; Hung et al., 2018; Mittal et al., 2019) applies generative adversarial networks (Goodfellow et al., 2014) in order to include the unlabeled data into the training.

Another increasingly popular direction is self-training with pseudo-labels (Lee, 2013), which alternates between prediction of pseudo-labels for unlabeled data and model retraining on the (pseudo-)labeled data. To construct the pseudo-labels, a popular approach is the mean teacher framework (Tarvainen & Valpola, 2017). It constructs the teacher network for pseudo-label generation from the exponential moving average of the weights of the student network. In order to avoid lazily mimicking the teacher's predictions and resisting updates, ATSO (Huo et al., 2021) splits the dataset into two parts, trains a model on each, and uses the model trained on one dataset to label the other. Similarly, CPS (Chen et al., 2021b) utilizes two networks with different initialization to generate the pseudo-labels for each other. Further extensions for self-training include training an additional error correction network (Mendel et al., 2020) and dynamically weighing pseudo-labels according to the agreement between two models (Feng et al., 2020b).

Self-training is often combined with consistency training, where perturbations are applied to unlabeled images or their intermediate features and a loss term enforces consistency of the predictions. For instance, Ouali et al. (2020) study perturbation of encoder features, Lai et al. (2021) enforce consistency of overlapping regions of two crops of the same image with different context, and Sohn et al. (2020) train the model on strongly augmented images while the pseudo-labels were generated only with weak augmentation. This general framework is extended by several strong augmentation strategies designed for semantic segmentation. CutMix (Yun et al., 2019; French et al., 2020) mixes crops from images and their pseudo-labels to generate additional training data, ClassMix (Olsson et al., 2021) uses class segments of pseudo-labels to build the mix mask, and Dvornik et al. (2019) paste instance crops into matching context regions of other images. Our proposed DepthMix module is inspired by these methods but it further respects the geometry of the scene when mixing samples in order to produce realistic occlusions.

2.4 Multi-task Learning of Semantic Segmentation and Self-supervised Depth Estimation

Jointly learning semantic segmentation and SDE was studied in previous works with the goal of improving *depth* estimation. Several works (Ramirez et al., 2018; Jiao et al., 2018; Yang et al., 2018; Chen et al., 2019a; Klingner et al., 2020b) learn both tasks jointly with a single network. Another line of work (Casser et al., 2019; Guizilini et al., 2020b; Jiang et al., 2019) distills knowledge from a teacher semantic segmentation network to guide SDE. To further promote coherence between semantic segmentation and SDE, Ramirez et al. (2018) and Chen et al. (2019a) propose a loss term to encourage spatial proximity between depth discontinuities and segmentation contours. As moving objects break the static world assumption of the SDE warping process, Casser et al. (2019) and Klingner et al. (2020b) incorporate dynamic object segmentations into the SDE loss calculation.

In contrast to these works, we do not aim to improve SDE but rather semi-supervised semantic segmentation. The closest to our approach are Jiang et al. (2018), Novosel et al. (2019), and Klingner et al. (2020b). Jiang et al. (2018) utilize relative depth computed from optical flow to replace ImageNet pretraining for semantic segmentation. In contrast, we additionally study multi-task learning of SDE and semantic segmentation and show that combining SDE with ImageNet features can further boost performance. Novosel et al. (2019) and Klingner et al. (2020b) improve the semantic segmentation performance by jointly learning with SDE. However, they focus on the fully-supervised setting, while our work explicitly addresses the challenges of semi-supervised semantic segmentation by using the depth estimates to generate additional training data and an automatic data selection mechanism based on SDE. Another work (Klingner et al., 2020a) supports the usefulness of SDE by improving the robustness of semantic segmentation.

2.5 Domain Adaptive Semantic Image Segmentation

A special kind of semi-supervised semantic segmentation is domain adaptation, where the unlabeled and labeled data originate from different domains. Different domains can be, for instance, real and synthetic data (Hoffman et al., 2016) or data captured under different conditions such as daytime/nighttime (Dai & Van Gool, 2018) or weather (Sakaridis et al., 2018). Further, it can be distinguished between unsupervised domain adaptation (UDA), if no labeled target data is available, and semi-supervised domain adaptation (SSDA), if a small number of annotations is available for the target domain.

For semantic segmentation, the better-studied scenario is UDA. In order to overcome the domain shift from the source to the target domain, adversarial training can be applied to the input (Hoffman et al., 2018), feature (Tsai et al., 2018), and output space (Tsai et al., 2018; Vu et al., 2019a). Also, non-adversarial input style transfer methods can be utilized (Yang & Soatto, 2020; Kim & Byun, 2020). An increasingly popular approach for UDA is self-training (Chapelle et al., 2009), where high-confidence predictions of a trained model are used to generate pseudo-labels for unlabeled data to iteratively improve the model (Zou et al., 2018; Wei et al., 2018). DACS (Tranheden et al., 2021) shows that ClassMix (Olsson et al., 2021) can also be applied to images from different domains. In contrast to DACS, our method uses the proposed DepthMix strategy, which respects the geometry of the scene during mixing to avoid geometric artifacts, and it combines Cross-Domain DepthMix with Target-Domain DepthMix for effective SSDA. Furthermore, we propose Matching Geometry Sampling to align the scene geometry and camera perspective for Cross-Domain DepthMix. A similar approach has been developed by Li et al. (2020b) by sampling images from the source domain with a similar semantic layout as the target domain. However, they do not perform data mixing, do not consider the geometry of the scene, and rely on the generalization from the semantic segmentation network trained on the source domain to the target domain in order to perform the semantic layout matching. As we use SDE, which can be trained on both the source and the target domain, our Matching Geometry Sampling lifts this assumption. Further self-training extensions include curriculum learning (Dai & Van Gool, 2018; Zhang et al., 2019; Lian et al., 2019), refining pseudo-labels using uncertainties (Zheng and Yang, 2021), augmentation consistency (Araslanov & Roth, 2021), and class prototypes (Zhang et al., 2021).

In contrast to UDA, semi-supervised domain adaptation (SSDA), where a few annotations are also available for the target domain, is less studied. Kalluri et al. (2019) propose a framework with a domain-shared encoder and a domain-specific decoder with additional entropy minimization in a separate embedding space. Wang et al. (2020) extend adversarial domain alignment from UDA (Tsai et al., 2018) and utilizes the additional target labels by applying feature-level adversarial domain alignment between labeled source and labeled target samples. For that, a spatial and a class-wise discriminator are introduced to mitigate inter-class confusions. To produce a better feature representation, Alonso et al. (2021) extend self-training with a student-teacher framework by contrastive learning (Hadsell et al., 2006). Concurrent to our work, Chen et al. (2021a) propose to train one teacher model on domain-mixed batches and one teacher model on CutMix (Yun et al., 2019; French et al., 2020) batches. A student model is trained on an ensemble of the two teachers and iterative pseudo-labeling is applied to the training of teachers and students. In contrast to these works, our method requires neither sensitive adversarial training nor costly ensemble

training. Also, instead of CutMix, we resort to our DepthMix algorithm, which produces geometrically valid synthesized samples. Further, we propose a combined Cross-Domain and Target-Domain DepthMix as well as a Matching Geometry Sampling, which leads to more effective SSDA.

2.6 Auxiliary Depth Estimation for Domain Adaptation

For UDA, depth estimates can be another valuable source of supervision to align the domains. For that purpose, SPIGAN (Lee et al., 2018) and DADA (Vu et al., 2019b) extend domain adversarial learning with privileged depth information from the source domain. GIO-Ada (Chen et al., 2019b) additionally uses the depth information for input style transfer. By providing depth information from the target domain as well, ATDT (Ramirez et al., 2019) learns a bottleneck feature domain transfer network with depth supervision on both domains, which generalizes to semantic segmentation. In contrast to our work, these approaches require depth ground truth, which can be difficult to acquire.

Concurrently to this work, SDE has been studied as an auxiliary task for *unsupervised* domain adaptation. Guizilini et al. (2021) utilize multi-task learning of semantic segmentation and SDE to learn a more domain-invariant representation. Instead of applying the view synthesis loss from SDE directly, Wang et al. (2021) use depth pseudo-labels from an SDE teacher network to learn depth estimation and semantic segmentation in a multi-tasking framework. To better transfer knowledge between both domains and tasks, the correlation of depth and semantic segmentation features is explicitly transferred from the source to the target domain and the depth adaptation difficulty is transferred to semantic segmentation to weigh the trust in the semantic segmentation pseudo-labels. Using (self-supervised) depth estimation for *semi-supervised* domain adaptation, however, has not been studied so far.

3 Methods

In this chapter, we present our four approaches to improve the performance of semantic segmentation with self-supervised depth estimation (SDE). They focus on four different aspects of the training process, covering data selection for annotation, data augmentation, multi-task learning, and domain adaptation. Given N images and M image sequences from the same domain, our first method, *automatic data selection for annotation*, uses SDE learned on the M (unlabeled) sequences to select N_A images out of the N images for human annotation (see Sect. 3.2. Our second approach, termed *DepthMix*, leverages the learned SDE to create geometrically-sound 'virtual' training samples from pairs of labeled images and their annotations (see Sect. 3.4). Our third method learns semantic segmentation with SDE as an auxiliary task under a multi-tasking framework (see Sect. 3.3). The learning is reinforced by a multi-task pretraining process combining SDE with image classification. And fourth, we extend our method to semi-supervised domain adaptation (SSDA) in order to utilize additional synthetic data, which has a low labeling effort (see Sect. 3.5). To address the domain gap, we propose a combined *Cross-Domain* and *Target-Domain DepthMix* strategy, which is enhanced by *Matching Geometry Sampling*.

3.1 Self-supervised Depth Estimation (SDE)

For self-supervised depth estimation (SDE), we follow the method of Godard et al. (2019), which we briefly introduce in the following. We first train a depth estimation network to predict the depth of a target image and a pose estimation network to estimate the camera motion from the target image and the source image. Depth and pose are used to produce a differentiable warping to transform the source image into the target image. The photometric error between the target image and multiple warped source frames is combined by a pixel-wise minimum. Besides, stationary pixels are masked out and an edge-aware depth smoothness term is applied resulting in the final SDE loss L_D. We refer the reader to the original paper (Godard et al., 2019) for more details.

3.2 Automatic Data Selection for Annotation

We use SDE as a proxy task for selecting N_A samples out of a set of N unlabeled samples for a human to create semantic segmentation labels. The selection is conducted progressively in multiple steps, similar to the standard active learning cycle (model training → query selection → annotation → model training). However, our data selection is fully automatic and does not require a human in the loop as the annotation is done by a proxy-task SDE oracle as visualized in Fig. 2.

Let's denote by $\mathcal{G}$, $\mathcal{G}_A$, and $\mathcal{G}_U$, the whole image set, the selected subset for annotation, and the unselected subset. Initially, we have $\mathcal{G}_A = \emptyset$ and $\mathcal{G}_U = \mathcal{G}$. The selection is driven by two criteria: *diversity* and *uncertainty*. Diversity sampling encourages the selected images to be diverse and cover different scenes. Uncertainty sampling favors adding unlabeled images that are near a decision boundary (with high uncertainties) of the model trained on the current $\mathcal{G}_A$. For uncertainty sampling, we need to train and update the model with $\mathcal{G}_A$. Specifically, the trained model f_{SIDE} solves the proxy task of single-image depth estimation (SIDE) on $\mathcal{G}_A$ with supervision from the SDE oracle. It is inefficient to repeat this every time a new image is added. For the sake of efficiency, we divide the selection into T steps and only train the model T times. In each step t, n_t images are selected and

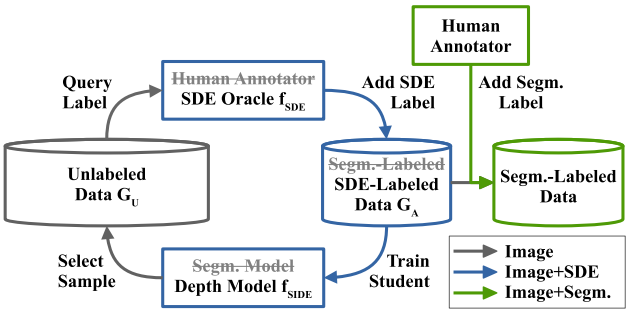


Fig. 2 The automatic data selection for annotation process selects the most useful samples from the set of unlabeled data $\mathcal{G}_U$ to be annotated. In contrast to active learning, the human annotator is replaced by an SDE oracle, and the samples are selected according to depth estimation as proxy-task. This lifts the requirement of a human in the loop. Samples are selected according to SDE feature diversity (Sect. 3.2.1) and depth student uncertainty (Sect. 3.2.2). The depth student is a single-image depth estimation network f_{SIDE}, which is trained with supervision from the SDE oracle

Algorithm 1 Automatic Data Selection

$t = 1$
$i \leftarrow \text{uniform}(1, N)$
$\mathcal{G}_A = \{I_i\}$ and $\mathcal{G}_U = \mathcal{G}_U \setminus \{I_i\}$
for $k = 2$ **to** N_A **do**
  if $k == \sum_{t'=1}^{t} n_{t'}$ **then**
    Train depth student Φ_{SIDE} on $\mathcal{G}_A$
    Calculate $E(i)\ \forall I_i \in \mathcal{G}_U$
    $t = t + 1$
  end if
  if $t == 1$ **then**
    Obtain index i according to Eq. (2)
  else
    Obtain index i according to Eq. (4)
  end if
  $\mathcal{G}_A = \mathcal{G}_A \cup \{I_i\}$ and $\mathcal{G}_U = \mathcal{G}_U \setminus \{I_i\}$
end for

moved from $\mathcal{G}_U$ to $\mathcal{G}_A$, so we have $\sum_{t=1}^{T} n_t = N_A$. After each step t, a model is trained on $\mathcal{G}_A$ and evaluated on $\mathcal{G}_U$ to get updated uncertainties for step $t + 1$.

3.2.1 Diversity Sampling

To ensure that the chosen annotated samples are diverse enough to represent the entire dataset well, we use an iterative farthest point sampling based on the L2 distance over features Φ^{SDE} computed by an intermediate layer of the SDE network. At step t, for each of the n_t samples, we choose the one in $\mathcal{G}_U$ with the largest distance to the current annotation set $\mathcal{G}_A$. The set of selected samples $\mathcal{G}_A$ is iteratively extended by moving one image at a time from $\mathcal{G}_U$ to $\mathcal{G}_A$ until the n_t images are collected:

$$\mathcal{G}_U = \mathcal{G}_U \setminus \{I_i\} \text{ and } \mathcal{G}_A = \mathcal{G}_A \cup \{I_i\}, \tag{1}$$

$$i = \text{argmax}_{I_i \in \mathcal{G}_U} \min_{I_j \in \mathcal{G}_A} ||\Phi_i^{\text{SDE}} - \Phi_j^{\text{SDE}}||_2. \tag{2}$$

3.2.2 Uncertainty Sampling

While diversity sampling is able to select diverse new samples, it is unaware of the uncertainties of a semantic segmentation model over these samples. Our uncertainty sampling aims to select difficult samples, i.e., samples in $\mathcal{G}_U$ that the model trained on the current $\mathcal{G}_A$ cannot handle well. In order to train this model, active learning typically uses a human-in-the-loop strategy to add annotations for selected samples. In this work, we use a proxy task based on self-supervised annotations, which can run automatically, to make the method more flexible and efficient. Since our target task is single-image semantic segmentation, we choose to use single-image depth estimation (SIDE) as the proxy task. Importantly, due to our SDE framework, depth pseudo-labels are available for $\mathcal{G}$. Using these pseudo-labels, we train a SIDE method on $\mathcal{G}_A$ and measure the uncertainty of its depth predictions on $\mathcal{G}_U$. Due to the high correlation of single-image semantic segmentation and SIDE, the generated uncertainties are informative and can be used to guide our sampling procedure. For example, if the depth model fails to correctly estimate the depth of a truck because trucks were underrepresented in $\mathcal{G}_A$, the semantic segmentation model will probably also struggle to recognize the truck. As the depth student model is trained only on $\mathcal{G}_A$, it can specifically approximate the difficulty of candidate samples with respect to the already selected samples in $\mathcal{G}_A$. The student is trained from scratch in each step t, instead of being fine-tuned from $t - 1$, to avoid getting stuck in the previous local minimum. Note that the SDE method is trained on a much larger unlabeled dataset, i.e., the M image sequences, and can provide good guidance for the SIDE method.

In particular, the uncertainty is signaled by the disparity error between the student network f_{SIDE} and the teacher network f_{SDE} in the log-scale space under L1 distance:

$$E(i) = ||\log(1 + f_{\text{SDE}}(I_i)) - \log(1 + f_{\text{SIDE}}(I_i))||_1. \tag{3}$$

As the disparity difference of far-away objects is small, the log-scale is used to avoid the loss being dominated by close-range objects. This criterion can be added into Eq. (2) to also select samples with higher uncertainties for the dataset update in Eq. (1):

$$i = \text{argmax}_{I_i \in \mathcal{G}_U} \min_{I_j \in \mathcal{G}_A} ||\Phi_i^{\text{SDE}} - \Phi_j^{\text{SDE}}||_2 + \lambda_{\text{E}} E(i), \tag{4}$$

where λ_E is a parameter to balance the contribution of the two terms. For diversity sampling, we still use SDE features instead of SIDE student features as SDE is trained on the entire dataset, which provides better features for diversity estimation. When n_t images have been selected according to Eqs. (1) and (4) at step t, a new SIDE model will be trained

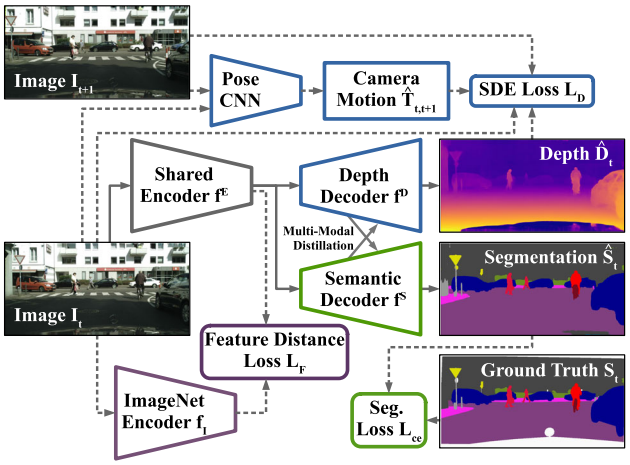


Fig. 3 Architecture for learning semantic segmentation with SDE as auxiliary task according to Sect. 3.3. The dashed paths are only used during training and only if image sequences and/or segmentation ground truth are available for a training sample

on the current $\mathcal{G}_A$ in order to continue further. As presented previously, our selection proceeds progressively in T steps until we collect all N_A images. The algorithm of this selection is summarized in Algorithm 1, where $\sum_{t'=1}^{t} n_{t'}$ describes the desired size of $\mathcal{G}_A$ at the end of step t.

3.3 Learning with Auxiliary Self-supervised Depth Estimation

In this section, we resort to features learned by SDE from unlabeled image sequences to improve the performance of semantic segmentation through transfer and multi-task learning. For that purpose, we use a network with a shared encoder f_θ^E, a separate depth decoder f_θ^D, and a separate segmentation decoder f_θ^S (see Fig. 3). For effective multi-task learning, useful intermediate features are exchanged between both task-specific decoders. In particular, we use the attention-guided multi-modal distillation module proposed by Xu et al. (2018). Guided by a learned attention map, this module distills features from the depth decoder, which are relevant for semantic segmentation decoder, and induces them into the semantic segmentation decoder. Vice versa, also features from semantic segmentation are distilled and induced in the depth decoder. The depth branch $g_\theta^D = f_\theta^D \circ f_\theta^E$ is trained using the SDE loss L_D and the segmentation branch $g_\theta^S = f_\theta^S \circ f_\theta^E$ is trained using semi-supervise semantic segmentation loss L_S, which we will introduce in Eq. (13) of the next section

$$L_{MTL} = L_D + L_S. \tag{5}$$

In order to initialize the pose estimation network and the depth branch $g_\theta^D = f_\theta^D \circ f_\theta^E$ properly, the architecture is first only trained on M unlabeled image sequences for SDE. As a common practice, we initialize the encoder with ImageNet weights as they provide useful semantic features learned during image classification. To avoid forgetting these semantic features during the SDE pretraining, we utilize a feature distance loss between the current bottleneck features f_θ^E and the bottleneck features generated by the encoder with ImageNet weights f_I^E

$$L_F = ||f_\theta^E - f_I^E||_2. \tag{6}$$

The loss for the depth pretraining is the weighted sum of the SDE loss and the ImageNet feature distance loss

$$L_{D,pretrain} = L_D + \lambda_F L_F. \tag{7}$$

To exploit the features from SDE for semantic segmentation by transfer learning, the weights from SDE g_θ^D are used to initialize the semantic segmentation branch g_θ^S.

3.4 DepthMix Data Augmentation

Inspired by the recent success of data augmentation approaches that mixup pairs of images and their (pseudo) labels to generate more training samples for semi-supervised semantic segmentation (Yun et al., 2019; French et al., 2020; Olsson et al., 2021), we propose an algorithm, termed DepthMix, to utilize self-supervised depth estimates to maintain the integrity of the scene structure during mixing.

Given two images I_i and I_j of the same size, we would like to copy some regions from I_i and paste them directly into I_j to get a virtual sample I'. The copied regions are indicated by a binary mask M, which has the same size as the two images. The image creation is done as

$$I' = M \odot I_i + (1 - M) \odot I_j, \tag{8}$$

where $\odot$ denotes the element-wise product. The semantic segmentation labels of the two images S_i and S_j are mixed up with the same mask M to generate the corresponding mixed semantic segmentation

$$S' = M \odot S_i + (1 - M) \odot S_j. \tag{9}$$

The mixing can be applied to labeled data and unlabeled data using human ground truths or pseudo-labels, respectively. Existing methods generate this mask M in different ways, e.g., randomly sampled rectangular regions (Yun et al., 2019; French et al., 2020) or randomly selected class segments (Olsson et al., 2021). In those methods, the structure of the scene is not considered and foreground and background are not distinguished. We find images synthesized by these methods often violate the geometric relationships between

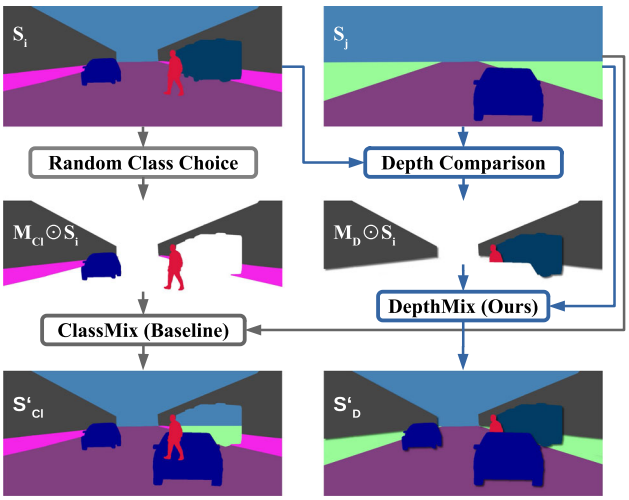


Fig. 4 Concept of the proposed DepthMix data augmentation (refer to Sect. 3.4) and its baseline ClassMix (Olsson et al., 2021) shown for the mixing of the semantic segmentation labels. By utilizing SDE, DepthMix mitigates geometric artifacts such as missing occluders (bus-shaped hole in the building) or missing occlusion (legs of the person). The corresponding images are mixed in the same way

objects. For instance, a distant object can be copied onto a close-range object or only unoccluded parts of mid-range objects are copied onto the other image. Imagine how strange it is to see a pedestrian standing on top of a car or to see the sky through a hole in a building (just as shown in Fig. 4 left).

Our DepthMix is designed to mitigate this issue. It uses the self-supervised depth estimates $\hat{D}i$ and $\hat{D}j$ of the two images to generate the mask M, which respects the notion of geometry. It is implemented by selecting only pixels from I_i whose depth values are smaller than the depth values of the pixels at the same locations in I_j:

$$M(a,b) = \begin{cases} 1 \text{ if } \hat{D}_i(a,b) < \hat{D}_j(a,b) + \epsilon \\ 0 \text{ otherwise} \end{cases} \tag{10}$$

where a and b are pixel indices, and ϵ is a small value to avoid conflicts of objects that are naturally at the same depth plane such as road or sky. By using this M, DepthMix respects the depth of objects in both images, such that only closer objects can occlude further-away objects. We illustrate this advantage of DepthMix with an example in Fig. 4.

In order to further take advantage of the unlabeled dataset $\mathcal{G}_U$ for DepthMix, we generate pseudo-labels using the mean teacher algorithm (Tarvainen & Valpola, 2017), which is commonly deployed in SSL (Berthelot et al., 2019; Verma et al., 2019; French et al., 2020; Olsson et al., 2021). For that purpose, an exponential moving average is applied to the weights of the semantic segmentation model g_θ^S to obtain the weights of the mean teacher θ_T:

$$\theta'_T = \alpha\theta_T + (1-\alpha)\theta. \tag{11}$$

To generate the pseudo-labels, an argmax over the classes C is applied to the prediction of the mean teacher:

$$S_U = \text{argmax}_{c\in C}(g^S_{\theta_T}(I_U)). \tag{12}$$

The mean teacher can be considered as a temporal ensemble, resulting in stable predictions for the pseudo-labels, while the argmax promotes confident predictions (Olsson et al., 2021).

In order to utilize the pseudo-labels, we apply DepthMix to two samples (I_i, S_i), (I_j, S_j) from the combined labeled and pseudo-labeled data pool $\mathcal{G}_A \cup \mathcal{G}_U$ to produce a mixed training pair (I', S') according to Eq. (8). The semantic segmentation network is trained using the cross-entropy of labeled samples (I_A, S_A) and the quality-weighted cross-entropy of mixed samples (I', S'):

$$L_S = L_{ce}(g^S_\theta(I_A), S_A) + q' L_{ce}(g^S_\theta(I'), S'), \tag{13}$$

where q' denotes the estimated quality of the mixed pseudo-label. We follow Olsson et al. (2021) and define q' as the fraction of pixels exceeding a threshold τ for the predicted probability of the most confident class P':

$$q' = \frac{\sum_{a,b}[P'(a,b) > \tau]}{W \cdot H}. \tag{14}$$

As the DepthMix segmentation S' consists of labels from two images, we calculate P' as the mix of its sources:

$$P' = M \odot P_i + (1-M) \odot P_j, \tag{15}$$

where P is the predicted probability of the most confident class for unlabeled images and 1 for labeled images:

$$P(a,b) = \begin{cases} \max_{c\in C}(g^S_{\theta_T}(I)(a,b)), & \text{if } I \in \mathcal{G}_U \\ 1, & \text{otherwise} \end{cases} \tag{16}$$

By applying DepthMix to labeled and pseudo-labeled samples, the network is exposed to image regions from both distributions in a single image. This can improve its generalization to the unlabeled data as the context for labeled regions can originate from unlabeled data and vice versa. The improved generalization can lead to better pseudo-labels, which in turn improve the quality of the DepthMix labels.

3.5 Semi-supervised Domain Adaptation (SSDA)

Synthetic data can be another valuable source for low-effort semantic segmentation annotations to reduce the number of expensive target labels. In semi-supervised domain adaptation (SSDA), a neural network is trained to solve a task on the real (target) domain while being trained using a limited

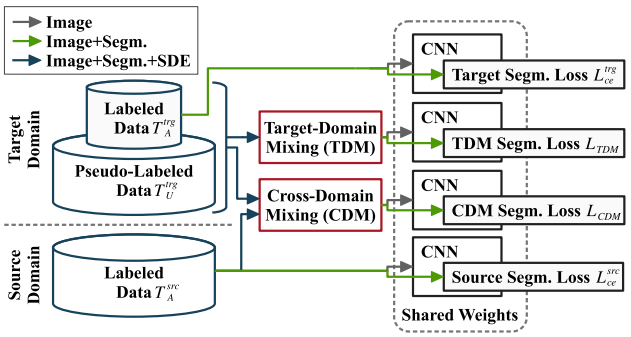


Fig. 5 Semi-supervised domain adaptation (SSDA) pipeline with Cross-Domain DepthMix (CDM) and Target-Domain DepthMix (TDM). While CDM applies DepthMix to samples from source and target domain to align both domains, TDM mixes labeled and pseudo-labeled samples from the target domain to align labeled and unlabeled target data. The network is trained on clean labeled source data, CDM/TDM data, and clean labeled target data for semantic segmentation. The target semantic segmentation pseudo-labels are obtained online using a mean teacher network

number of annotated target samples (I_A^{trg}, S_A^{trg}), further unlabeled target images I_U^{trg}, and additional annotated data from the synthetic (source) domain (I_A^{src}, S_A^{src}).

Naively, the semantic segmentation network branch g_θ^S can be trained on the labeled samples from both source and target domain using a pixel-wise cross-entropy loss

$$L_{ce}^{trg} = L_{ce}(g_\theta^S(I_A^{trg}), S_A^{trg}), \tag{17}$$

$$L_{ce}^{src} = L_{ce}(g_\theta^S(I_A^{src}), S_A^{src}). \tag{18}$$

However, as the labeled data from the target dataset is limited, the vanilla training strategy suffers from the gap between both domains.

In this work, we propose to use SDE to overcome the domain gap of SSDA. Extending the default setup, we augment both the target and the source dataset with self-supervised depth estimates. For that purpose, an SDE network f_D^{trg} is trained on image sequences from the target domain and another SDE network f_D^{src} is trained on image sequences from the source domain. Note that the image sequences can be different from the images labeled for semantic segmentation. After the SDE training, depth pseudo-labels are inferred for the images of the semantic segmentation datasets: $D_A^{src} = f_D^{src}(I_A^{src})$; $D_U^{trg} = f_D^{trg}(I_U^{trg})$; $D_A^{trg} = f_D^{trg}(I_A^{trg})$. Further, pseudo-labels S_U^{trg} are obtained online according to Eq. (12) for the unlabeled target data. The additional depth and semantic segmentation pseudo-labels are added to the SSDA training data.

Based on this data, we propose a combined Cross-Domain and Target-Domain DepthMix in order to facilitate effective self-training across domains as well as across labeled and unlabeled samples, respectively. Further, we enhance the mixing by Matching Geometry Sampling. The training process is visualized in Fig. 5 and described in the following.

3.5.1 Target-Domain DepthMix (TDM)

Target-Domain DepthMix (TDM) applies the DepthMix algorithm to the target dataset. It mixes labeled and unlabeled target samples to improve the generalization from the labeled target to the unlabeled target samples due to the increased variety of objects in different contexts. Therefore, it can favorably affect the quality of the pseudo-labels. Target-Domain DepthMix uses the same procedure as the single-domain SSL DepthMix described in Sect. 3.4. It produces a mixed sample (I'_{TDM}, S'_{TDM}) based on two target samples according to Eqs. (8)–10. The segmentation branch of the network is trained using the pixel-wise cross-entropy on the mixed samples

$$L_{TDM} = q'_{TDM} L_{ce}(g_\theta^S(I'_{TDM}), S'_{TDM}), \tag{19}$$

where q'_{TDM} weighs the loss according to the certainty of the pseudo-label as described in Sect. 3.4.

Mixing within a domain is only applied to the target domain and not to the source domain because the mixing serves the purpose of better generalization from labeled to unlabeled samples during the self-training. The source domain already contains many labeled samples. Therefore, self-training augmented by mixing is not necessary.

3.5.2 Cross-Domain DepthMix (CDM)

As there is only a small number of labeled samples available for the target domain, the trained network will still suffer from the gap between the source and target domain. To further align the domains during training, we propose Cross-Domain DepthMix, which mixes samples from both domains. This allows the network to better generalize across domains as both domains are present within each image.

Cross-Domain DepthMix utilizes one target sample and one source sample. If the target image is unlabeled, a pseudo-label is generated according to Eq. (12). The samples are mixed according to Eqs. (8)–10 to generate the cross-domain mixed sample (I'_{CDM}, S'_{CDM}). The segmentation branch of the network is trained using the pixel-wise cross-entropy on the mixed samples

$$L_{CDM} = q'_{CDM} L_{ce}(g_\theta^S(I'_{CDM}), S'_{CDM}), \tag{20}$$

where q'_{CDM} weighs the loss according to the certainty of the pseudo-label as described in Sect. 3.4.

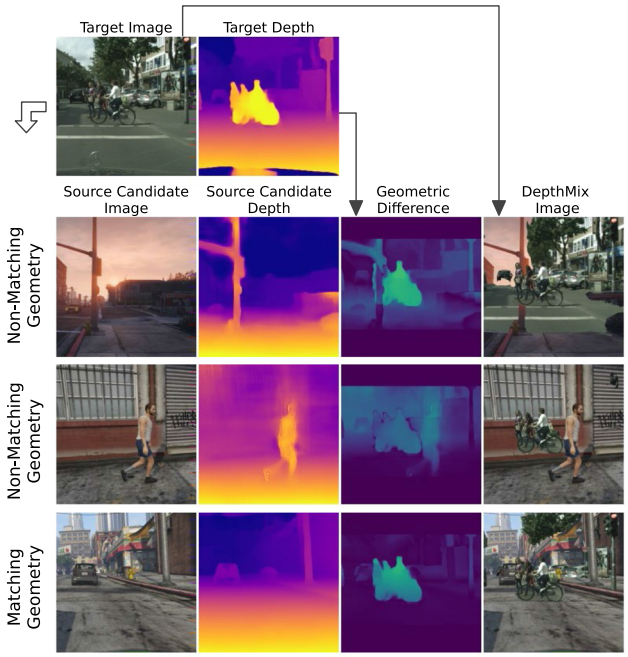


Fig. 6 Examples of the geometric domain gap and the Matching Geometry Sampling. Images and their SDE are shown for the target (first row) and the source domain (remaining rows). Some samples from the source domain (second and third row) have a different depth distribution compared to the target domain, which results in unrealistic DepthMix images (last column). Matching geometry sampling avoids sampling those domain pairs by selecting pairs with a small geometric difference (fourth row)

The final SSDA loss combines all four segmentation losses as well as the SDE loss on the target domain

$$L_{SSDA} = L_{ce}^{trg} + L_{ce}^{src} + L_{CDM} + L_{TDM} + L_{D}^{trg}, \tag{21}$$

where the loss components are weighted equally.

3.5.3 Matching Geometry Data Sampling

For samples from two different domains, the camera pose can differ between the domains as can be seen in the first three rows of Fig. 6. The geometric distribution difference between domains can impede the transfer of knowledge from the source to the target domain. For example, GTA contains samples from the view of a pedestrian while all Cityscapes samples are recorded from a front-facing camera of a car. This leads to different camera perspectives, which can result in unrealistic mixed samples such as a car "flying" in the sky (second row), or samples out of the target distribution such as images captured right in front of a building (third row).

We address this problem by sampling image pairs from the source and the target domain with a similar geometry with respect to the camera. The sampling is guided by the target geometry, which allows us to better match the geometric target distribution with mixed images. We define the geometric difference $G(i, j)$ of two samples i and j as the L1 distance of the log-scale disparity (inverse depth) estimates in camera space

$$G(i, j) = || \log\left(1 + \frac{1}{D_i}\right) - \log\left(1 + \frac{1}{D_j}\right) ||_1, \tag{22}$$

which corresponds to the metric used for the uncertainty sampling of our automatic data selection in Eq. (3). When calculating the geometric difference, we exclude the 80 pixels at the top of the image and the 100 pixels at the bottom from the geometric difference. This prevents SDE artifacts in the sky and the hood of the ego car from contaminating the geometric difference. The pixel-wise geometric difference is visualized in the third column of Fig. 6. It can be observed that it is generally higher for samples that do not have a matching geometry or camera perspective.

Based on a single target sample i^{trg} and a set of candidate source samples $\mathcal{C}^{src}$, which are both sampled randomly, the source sample with the smallest geometric difference is selected for training

$$j^{src} = \text{argmin}_{c^{src} \in \mathcal{C}^{src}} G(i^{trg}, c^{src}). \tag{23}$$

As the target sample is fixed during a matching step, it guides the selection towards the target distribution. The number of candidate samples $|C^{src}|$ balances between a good geometric match and a higher sampling diversity. A larger number of candidates results in a potentially better geometric match of the chosen sample, but it reduces the diversity of the chosen samples as it limits the sampling to the set of source samples that have a small geometric distance to the target domain in general.

This Matching Geometry Sampling allows our method to avoid the described issues of naive sampling and results in realistic DepthMix images, which are closer to the target distribution as can be seen in the last row of Fig. 6.

4 Experiment Setup

4.1 Datasets

Cityscapes We mainly evaluate our method on the Cityscapes dataset (Cordts et al., 2016), which consists of 2975 training and 500 validation images with semantic segmentation labels from European street scenes. We downsample the images to 1024×512 pixels. Besides, random cropping to a size of 512×512 and random horizontal flipping are used during the training. Importantly, Cityscapes provides 20 unlabeled frames before and 10 after the labeled image, which are used for our SDE training. During the semi-supervised segmen-

tation, only the 2975 images of the core dataset are used. If not stated otherwise, the same processing steps are applied to the following datasets as well.

CamVid The CamVid dataset (Brostow et al., 2009) contains 367 training, 101 validation, and 233 test images with dense semantic segmentation labels for 11 classes from street scenes in Cambridge. To ensure a similar feature resolution as for Cityscapes, we upsample the CamVid images from 480×360 to 672×512 pixels and randomly crop them to a size of 512×512 pixels.

GTA5 The GTA5 dataset (Richter et al., 2016) originates from a computer game, which enabled time-efficient semi-automatic semantic segmentation annotation. It contains about 25k training images labeled using the same 19 classes as Cityscapes. The SDE is trained on another part of the dataset (Richter et al., 2017), which provides image sequences.

Synthia The Synthia dataset (Ros et al., 2016) provides synthetic images with automatically generated annotations from a simulated urban environment. For semantic segmentation, we use the SYNTHIA-RAND-CITYSCAPES subset, which contains 9400 samples labeled with 16 semantic classes common with Cityscapes. Following the standard protocol for domain adaptation, we train our method for the 16 semantic classes that are common with Cityscapes and evaluate on 13 of them. The SDE is trained on the SYNTHIA-SEQS video sequence subset.

4.2 Network Architecture

Our network consists of a shared ResNet101 (He et al., 2016) encoder with output stride 16, a decoder for segmentation, and a decoder for SDE. The decoder consists of an ASPP (Chen et al., 2017) block with dilation rates of 6, 12, and 18 to aggregate features from multiple scales and another four upsampling blocks with skip connections (Ronneberger et al., 2015). For SDE, the upsampling blocks have a disparity side output at the respective scale. For effective multi-task learning, we additionally follow PAD-Net (Xu et al., 2018) and deploy an attention-guided distillation module after the third decoder block. It serves the purpose of exchanging useful features between segmentation and depth estimation. The design of the network architecture was chosen to facilitate both transfer and multi-task learning. To enable effective transfer learning, the task decoder branches have the same architecture and combine elements from typical semantic segmentation architectures such as the ASPP (Chen et al., 2017) as well as the commonly used U-Net decoder structure (Ronneberger et al., 2015) for depth estimation. This allows for pretraining the segmentation decoder branch with SDE and repurposing it for semantic segmentation afterward. For the pose estimation network, we use the same design as in (Godard et al., 2019). For the SDE network on the source domains, we use an output stride of 32 and a reduced number of decoder channels in order to improve convergence.

4.3 Training

For the SDE pretraining, the depth and pose network are trained using the Adam optimizer, a batch size of 4, and an initial learning rate of 1×10^{-4}, which is divided by 10 after 160k iterations. The SDE loss is calculated on four scales with three subsequent frames. During the first 300k iterations, only the depth decoder and the pose network are trained. Afterwards, the depth encoder is fine-tuned with an ImageNet feature distance $\lambda_F = 1 \times 10^{-2}$ for another 50k iterations. The encoder is initialized with ImageNet weights, either before depth pretraining or before semantic segmentation if depth pretraining is ablated. The *baseline* is trained with the same hyperparameters but only with a cross-entropy loss on the labeled samples. Its encoder is initialized with ImageNet pretrained weights.

For the semi-supervised multi-task learning, we train the network using SGD with a learning rate of 1×10^{-3} for the encoder and depth decoder, 1×10^{-2} for the segmentation decoder, and 1×10^{-6} for the pose network. The learning rate is reduced by 10 after 30k iterations and the network is trained for another 10k iterations. A momentum of 0.9, a weight decay of 5×10^{-4}, and a gradient norm clipping to 10 are used. The loss for segmentation and SDE are weighted equally. The mean teacher has $\alpha = 0.99$ and within an iteration, the network is trained on a clean labeled and an augmented mixed batch with size 2, respectively. The latter uses DepthMix with $\epsilon = 0.03$, color jitter, and Gaussian blur. If only pseudo-labeling but no mixing is used in an experiment, color jitter and Gaussian blur are still applied to the augmented batch.

For SSDA, the same hyperparameters as in the SSL setting are used. A batch consists of two source samples, two labeled target samples, and two (pseudo-)labeled target samples, which are used to compute L_{SSDA} (see Fig. 5). For the Matching Geometry Sampling, the number of random source candidate samples is set to 5: $|C^{src}| = 5$.

4.4 Automatic Data Selection for Annotation

For the automatic data selection, we use a slimmed network architecture for f_{SIDE} with a ResNet50 backbone, reduced decoder channels, and BatchNorm (Ioffe & Szegedy, 2015) in the decoder for efficiency and faster convergence. The depth student network is trained with a berHu loss using Adam with a learning rate of 1×10^{-4} and polynomial decay with exponent 0.9. For calculating the depth feature diversity, we use the output of the second depth decoder block after SDE pretraining. It is downsampled by average pooling to a size of 8×4 pixels and the feature channels are

Table 1 Comparison of data selection methods (DS: diversity sampling based on depth features, US: uncertainty sampling based on depth student error)

# Labeled	1/30 (100)	1/8 (372)	1/4 (744)
Random	48.75 ± 1.61	59.14 ± 1.02	63.46 ± 0.38
Entropy	53.63 ± 0.77	63.51 ± 0.68	66.18 ± 0.50
Ours (US)	51.75 ± 1.12	62.77 ± 0.46	66.76 ± 0.45
Ours (DS)	53.00 ± 0.51	63.23 ± 0.69	66.37 ± 0.20
Ours (DS + US)	**54.37** ± 0.36	**64.25** ± 0.18	**66.94** ± 0.59

mIoU in %, std. dev. over 3 seeds

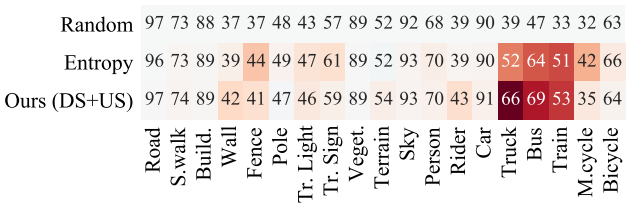
normalized to zero-mean and unit-variance over the dataset. The student depth error is weighted by $\lambda_E = 1000$. The number of the selected samples ($\sum_{t'=1}^{t} n_{t'}$) is incrementally increased to 25, 50, 100, 200, 372, and 744. For each subset, a student depth network is trained from scratch for 4 k, 8 k, 12 k, 16 k, and 20 k iterations, respectively, to calculate the student depth error and select the samples for the next subset. The quality of the selected subset with annotations $\mathcal{G}_A$ is evaluated for semantic segmentation using our default architecture and training hyperparameters. For the entropy baseline, a semantic segmentation network is trained on $\mathcal{G}_A$ and the samples with the highest mean pixel-wise Shannon entropy of the semantic segmentation prediction are greedily chosen from $\mathcal{G}_U$ to extend $\mathcal{G}_A$. Apart from that, the entropy baseline uses the same hyperparameters as our method.

5 Results

5.1 Automatic Data Selection for Annotation

First, we evaluate the proposed automatic data selection (see Sect. 3.2) on the Cityscapes (Cordts et al., 2016) dataset. Table 1 shows a comparison of our method with a baseline and a competing method for different numbers of selected labeled samples. The first baseline selects the labeled samples randomly, while the second, strong competitor uses active learning and iteratively chooses the samples with the highest segmentation entropy. In contrast to our method, this requires a human in the loop to create the semantic labels for iteratively selected images. Table 1 shows that our method with diversity sampling (DS) works better than with uncertainty sampling (US) for few labeled samples. We hypothesize that, for a small number of annotated samples, it is more important to better cover the underlying distribution with a diverse subset than just covering uncertain/difficult samples. For a larger subset, however, it makes sense to focus on the uncertain samples as the common cases are most likely already covered. Further, it can be seen that combining diversity sampling and uncertainty sampling (DS + US) performs better than using them individually showing that these criteria are complementary and cover two relevant aspects of selecting data for annotation. When comparing our method with both sampling criteria (DS + US) with the baselines "Random" and "Entropy", it can be seen that our method outperforms both comparison methods, demonstrating the effectiveness of ensuring diversity and exploiting difficult samples based on depth estimation. It also supports the assumption that depth estimation and semantic segmentation are correlated in terms of sample difficulty. With 1/4 of the labeled samples, our method achieves 98.8% of the fully-supervised performance and with only 1/8 samples it still reaches 94.8%. Furthermore, the standard deviation of the achieved segmentation performance with our data selection is noticeably lower than for the random baseline when using few labeled samples, resulting in better reproducibility.

Table 2 Comparison of the class-wise IoU in % of the data selection methods for 372 labeled samples



The color visualizes the IoU difference with respect to the baseline

To better understand the underlying reasons for the improved performance, we analyze the class-wise IoU for 372 labeled samples in Table 2. It shows that our automatic data selection significantly improves the performance of difficult classes with a low IoU of the random baseline such as wall, fence, truck, bus, and train. In comparison to the strong active learning entropy baseline, our method achieves even better performance for the classes wall, rider, truck, and bus.

In order to investigate possible reasons for the improved performance of the automatic data selection, we visualize the ratio of the automatically selected pixels and total dataset pixels grouped by the ground truth class for 372 selected samples in Fig. 7. As expected, the ratio is about 0.125 for most of the classes when selecting 1/8 of the samples randomly (Fig. 7 left). For the entropy baseline and our method, it can be seen that a higher ratio of difficult/rare classes (e.g. truck, bus, and train) are sampled from the underlying training set, while a smaller ratio of common classes such as road and building are sampled. When comparing the class-wise IoU (Table 2) and the ratio of selected pixels (Fig. 7), it can be seen that the improvement for difficult classes is correlated with them being selected more frequently by the automatic data selection. Intuitively, more samples of rare and easy to confuse classes such as car, truck, bus, and train as well as wall and fence will help the classifier to distinguish them.

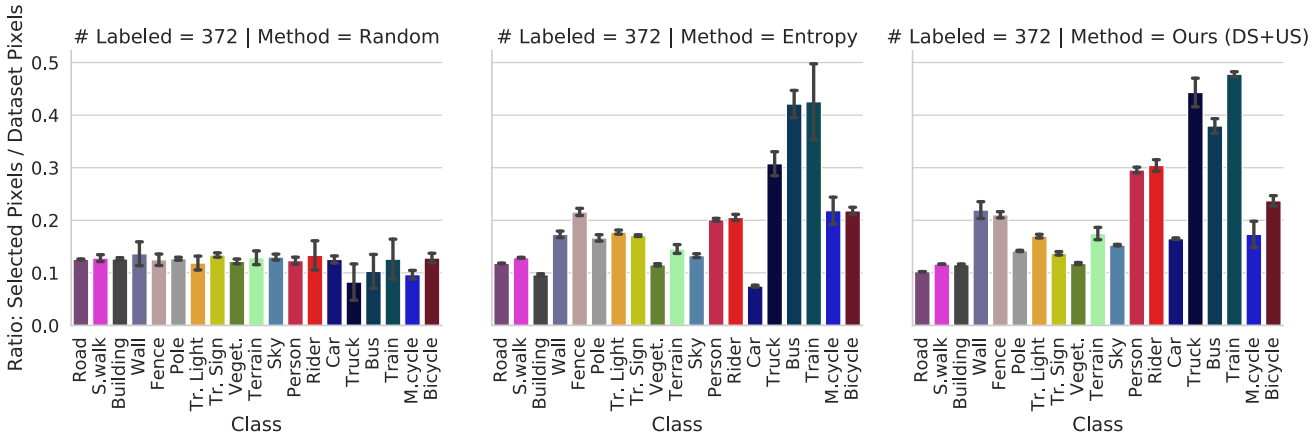


Fig. 7 Class frequency analysis of the data selection behavior. The ratio of selected pixels (372 samples) and dataset pixels (2975 samples) grouped by ground truth class for different data selection methods is shown. The values are averaged over 3 random seeds. The error bars indicate the standard deviation

When comparing the active learning entropy baseline to our method, Fig. 7 shows that our method selects a higher ratio of wall, person, rider, and truck, which directly connects to the higher class IoU for these classes as shown in Table 2. This is also reflected by a high Pearson correlation coefficient $\rho = 0.92$ of the increase in class selection ratio and the improvement of the class-wise IoU. Please note that the class-statistics of Fig. 7 are not available to our method during the entire selection process. This demonstrates that our method is able to correctly estimate the utility of samples for subsequent semantic segmentation without knowing the ground truth labels during the selection.

5.2 DepthMix Data Augmentation

Second, we study the proposed geometry-guided mixing strategy DepthMix (see Sect. 3.4). We evaluate the performance for the SSL setting with 372 of the labeled training samples (which corresponds to 1/8 of the labeled samples in Cityscapes) and the fully-supervised setting with 2975 samples. The subset of labeled samples is chosen randomly. Table 3 shows the mean and standard deviation of the mIoU in percent over three random seeds. Additionally, the improvement in percentage points of the analyzed components over the baseline, which only uses a cross-entropy loss on labeled samples, is shown. In accordance with the literature on semi-supervised mixing (French et al., 2020; Olsson et al., 2021; Sohn et al., 2020), we first add self-training with pseudo-labels from the mean teacher to the framework. As can be seen in Table 3, this already significantly improves the performance in the SSL setting by + 3.24 mIoU percentage points. Still, our proposed DepthMix module further increases the performance by another + 1.76 (+ 2.06) percentage points for 372 (2975) labeled samples. Note that the high variance for few labeled samples is mostly due to the high influence of the randomly selected labeled subset. The chosen subset affects all configurations equally and the reported improvements are consistent for each subset.

Table 3 Comparison of different mixing strategies

	372 labels		2975 labels	
Baseline	59.14 ± 1.02	↴	67.77 ± 0.13	↴
Pseudo-labels	62.39 ± 0.86	+ 3.24	–	
ClassMix	63.16 ± 0.89	+ 4.02	69.60 ± 0.32	+ 1.83
DepthMix	64.14 ± 1.34	+ 5.00	69.83 ± 0.36	+ 2.06

mIoU in %, standard deviation over 3 seeds

When comparing DepthMix directly to the competitor ClassMix (Olsson et al., 2021) in Table 3, the performance of DepthMix is still + 0.98 (+ 0.23) percentage point higher for 372 (2975) samples. This demonstrates the effectiveness of the geometry-aware mixing, which better handles occlusions as described in Sect. 3.4. The higher improvement of DepthMix for fewer labeled samples might be since the SDE for DepthMix can be trained on a large set of unlabeled samples, resulting in precise depth contours over the whole (un)labeled training set. ClassMix in contrast uses segmentation pseudo-labels for mixing, which were only supervised on the subset of labeled samples. Therefore, on the unlabeled samples, the mixing contours can be less accurate than for DepthMix.

Further, we analyze the class-wise IoU for 372 labeled samples as shown in Table 4. Pseudo-labels generally improve the IoU through self-training. However, for the rare class motorcycle, the IoU decreases compared to the baseline. The reason for that is probably a pseudo-label drift of motorcycle towards the similar class bicycle during the self-training. Both mixing strategies mitigate the drift by a better generalization from labeled to unlabeled data through providing different contexts and occlusions during the training. The

Table 4 Comparison of the class-wise IoU in % of the different mixing strategies for 372 labeled samples

	Road	S.walk	Build.	Wall	Fence	Pole	Tr. Light	Tr. Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.cycle	Bicycle
Baseline	97	73	88	37	37	48	43	57	89	52	92	68	39	90	39	47	33	32	63
P. Labels	97	77	89	43	41	49	47	59	90	55	93	69	41	91	50	61	42	29	64
ClassMix	97	78	89	46	43	50	49	62	90	54	93	70	43	91	45	61	43	32	65
DepthMix	97	76	89	49	44	50	49	62	90	52	93	71	44	91	53	63	46	34	65

The color visualizes the IoU difference with respect to the baseline

better generalization leads to less erroneous pseudo-labels and consequently to less drift. Additionally, this also results in a higher IoU for other difficult classes with a low baseline IoU such as sidewalk, wall, fence, traffic light, traffic sign, rider, truck, bus, and train. When comparing DepthMix and ClassMix, it can be seen that DepthMix improves over ClassMix for difficult classes with usually pronounced depth contours such as wall, traffic light, rider, bus, train, and motorcycle. However, there is a slight decrease in IoU for the classes sidewalk and terrain. These are classes, which can be easily confused with each other and with road. DepthMix might experience difficulties with these classes as there are usually no depth contours between them, which results in fewer mixing boundaries.

The effective occlusion handling of DepthMix can be seen in Fig. 8a–c for samples from Cityscapes. It shows input images in orange and blue as well as their SDE used for mixing. The column “DepthMix Select." visualizes from which input image the regions, chosen by DepthMix, originate. As can be seen in Fig. 8a, DepthMix is able to handle occlusions at multiple levels. The biker from the blue image occludes buildings from the orange image, but the blue biker is itself also partly occluded by the closer biker from the orange image. Similar cases can be seen for trees, traffic signs, and cars in Fig. 8b, c. The column “Mixed Image I'" shows the resulting image without the selection overlay. It can be seen that due to the spatially accurate depth contours, the mixed images contain only minor mixing border artifacts and have a realistic appearance. The same is the case for the mixed segmentation as can be seen in the column “Mixed Segm. S'”.

However, there are also some cases in which DepthMix fails to correctly mix images according to their geometry. Examples of typical failure cases are shown in Fig. 8d, e. First, the SDE can be inaccurate for dynamic objects due to the violation of the static world assumption, which can cause an inaccurate structure within the mixed image. This is particularly the case if a car is driving in front of the ego car (Fig. 8d). However, this type of failure case is common in ClassMix and its frequency is greatly reduced with DepthMix. A remedy might be SDE extensions that incorporate the motion of dynamic objects (Casser et al., 2019; Dai et al., 2020; Klingner et al., 2020b). Second, in some cases, the SDE can be imprecise and the depth discontinuities do not appear at the same location as the class border. This can cause artifacts in the mixed image as well as in the mixed segmentation as can be seen for the sky within the building in Fig. 8e. Note that the same can happen for ClassMix when the pseudo-labels, used for the mixing, do not have accurate segmentation borders.

5.3 Transfer and Multi-task Learning

Third, we study the proposed transfer and multi-task learning of semantic segmentation and the auxiliary task self-supervised depth estimation in Table 5. For 372 (2975) samples, SDE transfer learning of the encoder and decoder (with previous ImageNet pretraining of the encoder) improves performance by + 1.31 (+ 1.23) percentage points mIoU over the baseline with only ImageNet pretraining of the encoder. This demonstrates the usefulness of the features learned by SDE for semantic segmentation, both in the semi- and fully-supervised case. Additional regularization of the encoder with an ImageNet feature distance loss during SDE pretraining improves the performance by another + 0.35 (+ 0.48) percentage points. Furthermore, multi-task learning in addition to transfer learning results in a performance increase of + 0.45 (+ 0.29) percentage points.

The class-wise analysis for 372 labeled samples (see Table 6) shows that SDE transfer learning without ImageNet Feature distance loss significantly improves the performance of classes, where segmentation border coincides with depth discontinuities such as fence, pole, traffic light, and traffic sign. This is possibly due to their characteristic depth profile learned during SDE. For example, a good depth estimation performance requires correctly segmenting poles or traffic signs as missing them can cause large depth errors. However, there is a performance drop for classes that have slight semantic differences such as truck, bus, train, and motorcycle. We hypothesize that the SDE pretraining causes forgetting important semantic features from the ImageNet pretraining that are relevant for semantic segmentation but not for SDE. For example, for SDE it is not relevant if an object is a bus or a train but for semantic segmentation it is. Adding the ImageNet feature distance loss to the SDE pretraining in order to avoid forgetting these semantic features, prevents the performance drop for truck, bus, and train. The additional multi-task learning further improves the performance for the small difficult classes rider and motorcycle.

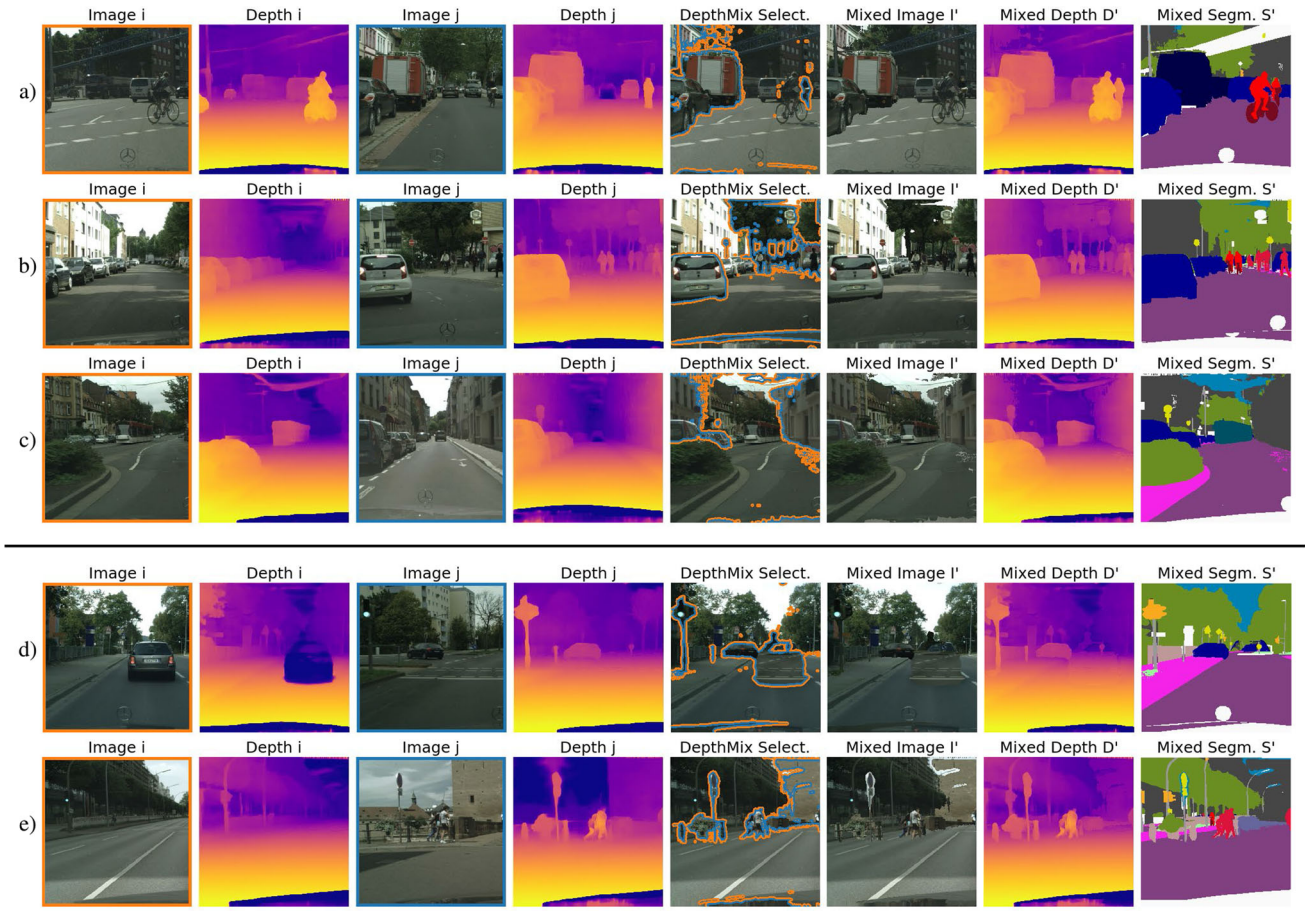


Fig. 8 Examples of DepthMix applied to Cityscapes crops. From left to right, the source images with their SDE estimate, the mixed image I' overlaid with the border of the mix mask M in blue/orange depending on the adjacent source image (i—orange, j—blue), the mixed image without visual guidance I', the mixed depth D', and the mixed segmentation S' are shown. For simplicity, the source segmentations for the mixed segmentation S' originate from the ground truth labels. Rows **a**–**c** demonstrate the strength of DepthMix to handle occlusions, while rows **d**, **e** show typical failure cases

5.4 Combined Framework for SSL

Next, we combine the three contributions multi-task learning, DepthMix, and automatic data selection for annotation into a unified semi-supervised semantic segmentation framework. The first part of Table 7 summarizes the performance of these components from the previous sections for a better comparison. The component with the most improvement is the automatic data selection for annotation with diversity and uncertainty sampling with + 5.11 mIoU percentage points for 372 labeled samples. However, it is not applicable to the full dataset as there is no need for sample selection—all samples are used. The second-most effective component is DepthMix with pseudo-labeling, which also has a pronounced mIoU improvement of + 5.00 (+ 2.06) for 372 (2975) samples. The smallest but still significant improvement comes from multi-task learning with + 2.00 (+ 1.99) percentage points. The direct comparison of the class-wise IoU for 372 labeled samples in Table 8 shows that data selection mostly improves the performance of difficult classes with a low baseline IoU (e.g. wall, fence, truck, bus, and train), SDE multi-task learning of classes with surrounding depth discontinuities (e.g. fence, pole, traffic light, traffic sign, and rider), and DepthMix of both.

Table 5 Comparison of SDE feature transfer methods (F: ImageNet feature distance loss)

Aux. SDE	F	372 labels		2975 labels	
		59.14 ± 1.02	↴	67.77 ± 0.13	↴
Transfer		60.46 ± 0.64	+ 1.31	69.00 ± 0.70	+ 1.23
Transfer	✓	60.80 ± 0.69	+ 1.66	69.47 ± 0.38	+ 1.71
Multi-task	✓	61.25 ± 0.55	+ 2.10	69.76 ± 0.39	+ 1.99

mIoU in %, standard deviation over 3 seeds

Considering that the three contributions follow different approaches and improve the performance of a different subset

Table 6 Comparison of the class-wise IoU in % of SDE feature transfer methods for 372 labeled samples (F: ImageNet feature distance loss)

	Road	S.walk	Build.	Wall	Fence	Pole	Tr. Light	Tr. Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.cycle	Bicycle
Baseline	97	73	88	37	37	48	43	57	89	52	92	68	39	90	39	47	33	32	63
Transfer	97	75	89	38	40	52	50	63	90	54	93	70	42	90	37	43	29	30	65
Transfer+F	97	74	89	41	40	51	47	62	90	52	93	70	41	90	43	48	33	30	65
Multi-Task+F	97	73	89	40	40	50	47	61	90	53	93	70	44	90	44	48	34	37	64

The color visualizes the IoU difference with respect to the baseline

Table 7 Comparison of the combinations of the proposed framework components (S: data selection, DM: DepthMix, MTL: SDE multi-task learning)

S	DM	MTL	372 labels		2975 labels	
			59.14 ± 1.02	↴	67.77 ± 0.13	↴
		✓	61.25 ± 0.55	$+ 2.10$	69.76 ± 0.39	$+ 1.99$
	✓		64.14 ± 1.34	$+ 5.00$	69.83 ± 0.36	$+ 2.06$
✓			64.25 ± 0.18	$+ 5.11$	–	
✓		✓	65.35 ± 0.10	$+ 6.21$	–	
✓	✓		66.48 ± 0.27	$+ 7.34$	–	
	✓	✓	66.66 ± 1.05	$+ 7.52$	71.16 ± 0.16	$+ 3.40$
✓	✓	✓	68.01 ± 0.83	$+ 8.87$	–	

mIoU in %, standard deviation over 3 seeds

Table 8 Comparison of the class-wise IoU in % of the combinations of the proposed framework components for 372 labeled samples (see Table 7 for abbreviations)

	Road	S.walk	Build.	Wall	Fence	Pole	Tr. Light	Tr. Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.cycle	Bicycle
Baseline	97	73	88	37	37	48	43	57	89	52	92	68	39	90	39	47	33	32	63
MTL	97	73	89	40	40	50	47	61	90	53	93	70	44	90	44	48	34	37	64
DM	97	76	89	49	44	50	49	62	90	52	93	71	44	91	53	63	46	34	65
S	97	74	89	42	41	47	46	59	89	54	93	70	43	91	66	69	53	35	64
S+MTL	96	73	89	43	43	49	46	62	90	55	93	70	44	91	71	73	58	32	64
S+DM	97	77	89	47	45	50	49	63	90	54	93	72	45	92	69	77	55	34	65
DM+MTL	97	77	90	49	46	53	52	65	90	53	94	72	48	92	60	69	54	38	66
S+DM+MTL	97	77	90	47	47	52	51	65	90	55	94	73	51	92	66	79	65	35	67

The color visualizes the IoU difference with respect to the baseline

of classes, we further study the combination of our contributions as shown in the second part of Tables 7 and 8. The improvement over the baseline performance is + 6.21 when combining multi-task learning with data selection, + 7.34 when combining DepthMix and data selection, and + 7.52 (+ 3.40) when combining multi-task learning and DepthMix for 372 (2975) samples. In all cases, the combination is better than every single component. The class-wise analysis for 372 labeled samples in Table 8 reveals that the class performance of the combination usually is the highest class performance of the components. As the components perform well on different classes, this already attributes to the improved performance of the combinations. Moreover, there are some classes such as fence, traffic sign, rider, truck, bus, and train, where the performance of the combination is even higher than its best component. This might be due to self-reinforcing effects. For example, the improved segmentation detail at depth contours from multi-task learning is propagated into DepthMix and results in even better pseudo-label supervision for mixed samples. The last row of Table 7 shows the combination of all three contributions. With an improvement of + 8.87 percentage points for 372 labeled samples, it achieves the best results so far. It combines the strength of our three contributions and significantly improves the performance for classes with depth discontinuities and for difficult classes. The most improvement is achieved for truck, bus, and train, where the mIoU is more than 50% better than the baseline.

5.5 Influence of Depth Estimation on SSL Performance

To better understand the influence of the depth estimation performance on semi-supervised semantic segmentation, we study the performance of our framework with different depth estimation networks. In particular, we compare the used SDE method with fully-supervised depth estimation, where the supervision comes from the official Cityscapes stereo depth maps. In contrast to SDE, which only provides relative depth maps with an unknown scale factor, the fully-supervised depth estimates have a metric scale and correctly predict the depth of dynamic objects. Further, we compare the default SDE network, which was trained with all 83,300 Cityscapes frames, with SDE networks, which were trained only with a randomly selected subset of the frames. The performance of the depth network variants is evaluated on the Cityscapes validation set with the depth RMSE and RMSE log metrics with a 50 m depth cap. For the evaluation of SDE, the common practice of per-image median ground truth scaling is used (Zhou et al., 2017; Godard et al., 2019).

The results are shown in Table 9. It can be seen that the supervised depth estimation achieves the best RMSE and RMSE log (lower values are better). SDE performs best when all training frames are used. When reducing the training frames of SDE, the depth performance gradually drops.

The semi-supervised semantic segmentation performance is correlated with the performance of the depth estimation network. A smaller depth estimation error usually results in a higher segmentation mIoU. However, the differences in mIoU are relatively small when compared with the standard

Table 9 Influence of the depth estimation method on the semi-supervised semantic segmentation performance for 100 labels

Depth estimation	RMSE (↓)	RMSE log (↓)	mIoU (↑)
Supervised depth	4.252	0.159	62.69 ± 0.80
SDE 100% frames	5.693	0.243	62.09 ± 0.39
SDE 10% frames	5.873	0.249	62.29 ± 1.21
SDE 1% frames	6.003	0.256	61.40 ± 1.10
SDE 0.1% frames	7.063	0.303	59.65 ± 1.03

mIoU in %, standard deviation over 3 seeds

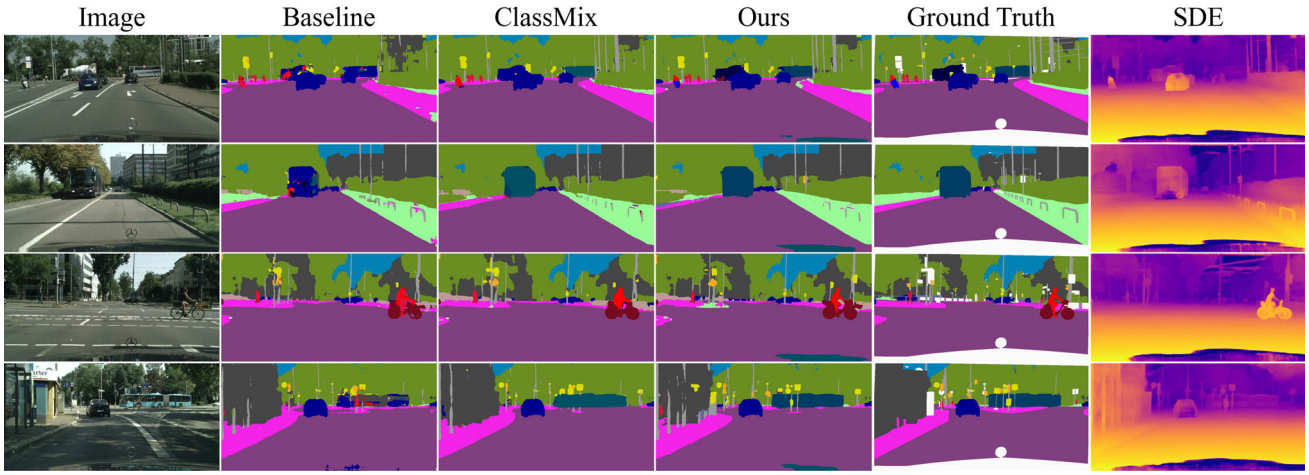


Fig. 9 Example semantic segmentations and self-supervised depth estimates of our method for 100 labeled samples in comparison with ClassMix (Olsson et al., 2021) and the baseline

deviations. Only for the SDE with 0.1% frames, which has the highest depth error, there is a significant mIoU drop. This shows that the proposed semi-supervised learning framework is relatively robust with respect to the precision of the depth estimates (within certain bounds) and that already a rough understanding of the scene geometry is sufficient to learn useful representations for semantic segmentation. Intuitively, depth estimation requires distinguishing objects from their background, which facilitates semantic grouping. The precise distance of an object to the camera plays probably only a subordinate role for semantic segmentation as long as the depth estimates are sufficiently correct to facilitate visual grouping.

Further, this experiment shows that the relative depth maps, which have an unknown scaling factor, are sufficient for the proposed semi-supervised semantic segmentation framework. The metric depth maps from supervised depth estimation only slightly improve the segmentation performance by a margin that is still within the standard deviation. When considering SDE as a self-supervised representation learning method for semantic segmentation, the unknown scaling factor of SDE has probably only a minor influence because it does not affect foreground/background relations, which are important for learning a semantic grouping. For example, SDE maps a traffic sign to the same depth value while its surrounding would most likely be mapped to a larger depth. The required SDE features can be re-utilized for semantic segmentation independent of the exact depth values. When considering DepthMix, an unknown scaling factor is also not problematic as long as it is consistent across the mixed images (i.e. both images have a similar scale factor). As qualitatively observed in Figs. 8 and 9, the self-supervised depth estimates are sufficiently consistent for DepthMix.

5.6 Comparison with State-of-the-Art SSL Methods

Next, we compare our approach with several state-of-the-art SSL approaches. The results are summarized in Table 10. The performance (mIoU in %) of the SSL methods and their baselines (which use the same backbone network but are only trained on the labeled dataset) are shown over a different number of labeled samples. As the performance of the baselines differs, there are columns showing the absolute improvement for better comparability. As our baseline utilizes a more capable network architecture due to the U-Net decoder with ASPP as opposed to a DeepLabv2 decoder used by most previous works, we also reimplemented the state-of-the-art method, ClassMix (Olsson et al., 2021) with our network architecture and training parameters to ensure a direct comparison.

Table 10 Comparison with state-of-the-art SSL semantic segmentation methods on the Cityscapes validation set (mIoU in %, standard deviation over 3 random seeds)

Labeled samples	1/30 (100)		1/8 (372)		1/4 (744)		Full (2975)	
Baseline (Hung et al., 2018)	–		55.50	↴	59.90	↴	66.40	↴
Adversarial (Hung et al., 2018)	–		58.80	+ 3.30	62.30	+ 2.40	–	
Baseline (Mittal et al., 2019)	–		56.20	↴	60.20	↴	66.00	
s4GAN (Mittal et al., 2019)	–		59.30	+ 3.10	61.90	+ 1.70	65.80	− 0.20
Baseline (Feng et al., 2020a)	45.50	↴	56.70	↴	61.10	↴	66.90	
DST–CBC (Feng et al., 2020a)	48.70	+ 3.20	60.50	+ 3.80	64.40	+ 3.30	–	
Baseline (Feng et al., 2020b)	49.54	↴	59.65	↴	–		68.16	
DMT (Feng et al., 2020b)	54.80	+ 5.26	63.03	+ 3.38	–		–	
Baseline (French et al., 2020)	44.41 ±1.11	↴	55.25 ±0.66	↴	60.57 ±1.13	↴	67.53 ±0.35	↴
CutMix (French et al., 2020)	51.20 ±2.29	+ 6.79	60.34 ±1.24	+ 5.09	63.87 ±0.71	+ 3.30	67.68 ±0.37	+ 0.15
Baseline (Mendel et al., 2020)	–		55.96 ±0.86	↴	60.54 ±0.85	↴	–	
ECS (Mendel et al., 2020)	–		60.26 ±0.84	+ 4.30	63.77 ±0.65	+ 3.23	–	
Baseline (Olsson et al., 2021)	43.84 ±0.71	↴	54.84 ±1.14	↴	60.08 ±0.62	↴	66.19 ±0.11	
ClassMix (Olsson et al., 2021)	54.07 ±1.61	+10.23	61.35 ±0.62	+ 6.51	63.63 ±0.33	+ 3.55	–	
ATSO (Huo et al., 2021)[a]	53.1		61.8		63.2		–	
Baseline	48.75 ±1.61	↴	59.14 ±1.02	↴	63.46 ± 0.38	↴	67.77 ±0.13	↴
ClassMix (Olsson et al., 2021)[b]	56.82 ± 1.65	+ 8.07	63.86 ± 0.41	+ 4.72	65.57 ± 0.71	+ 2.11	–	
ClassMix (Olsson et al., 2021) (+ Video)	56.79 ±1.98	+ 8.04	63.22±0.84	+ 4.08	65.72±0.18	+ 2.26	68.23 ±0.70	+ 0.46
Ours w/o Data Selection	58.40 ± 1.36	+ 9.65	66.66 ± 1.05	+7.52	68.43 ± 0.06	+4.98	**71.16** ± 0.16	**+ 3.40**
Ours	**62.09** ± 0.39	**+ 13.34**	**68.01** ± 0.83	**+ 8.87**	**69.38** ± 0.33	**+ 5.92**	–	

The best results are shown in bold font and the second-best results are underlined

[a] ATSO does not provide baseline results

[b] Results of the reimplementation in our experiment setting

Table 11 Semantic segmentation performance on the CamVid test set with SDE trained on Cityscapes sequences (mIoU in %, standard deviation over 3 random seeds)

# Labeled	50		100		367 (Full)	
Baseline	59.2± 1.8	↴	63.1 ± 0.6	↴	68.2 ± 0.1	↴
ClassMix	65.9± 0.3	+ 6.7	67.5 ± 1.0	+ 4.4	–	
Ours w/o S	66.8 ± 1.2	+ 7.6	68.9 ± 0.6	+ 5.8	**71.5** ± 0.2	**+3.3**
Ours	**68.2** ± 0.4	**+9.0**	**69.6** ± 0.6	**+6.5**	–	

As shown in Table 10, our method (without data selection) outperforms all other approaches on each labeled subset size for both the absolute performance as well as the improvement to the baseline. The only exception is the absolute improvement of the original results of ClassMix for 100 labeled samples. However, if we consider ClassMix trained in our setting, our method outperforms it also in this case. This can be explained by the considerably higher baseline performance in our setting, which increases the difficulty to achieve a high improvement. Adding data selection even further increases the performance by a significant margin, so that our method, trained with only 1/8 of the labels, even slightly outperforms the fully-supervised baseline.

To identify whether the improvement originates from access to more unlabeled data or from the effectiveness of our approach, we compare it to another baseline "ClassMix (+Video)". More specifically, we also provide all unlabeled image sequences to ClassMix and see how much it can benefit from this additional amount of unlabeled data. Experimental results show no significant difference. This is probably due to the high correlation between the Cityscapes image dataset and the video dataset (the images are the 20th frames of the video clips).

The adequacy of our approach is also reflected in the example predictions in Fig. 9. We can observe that the contours of classes are more precise. This is particularly the case for classes, which are surrounded by depth discontinuities such as poles, traffic signs, rider, or person. Moreover, difficult objects such as bus, train, rider, or truck can be better distinguished. As discussed in Sect. 5.4, this observation is also quantitatively confirmed by the class-wise IoU improvement shown in Table 8. On the downside, SDE sometimes fails for cars driving directly in front of the camera (see 7th row in Fig. 9) and violating the reconstruction assumptions. Those cars are observed at the same location across the image sequence and can not be correctly reconstructed during SDE training, even with correct depth and pose estimates. However, the network-internal differentiation between moving and non-moving cars does not hinder the transfer of SDE-learned features to semantic segmentation but can cause problems with DepthMix (see Sect. 5.2).

5.7 Learning SDE and Semantic Segmentation on Different Datasets

In this section, we show that the unlabeled image sequences and the labeled segmentations can also originate from different datasets within similar visual domains. For that purpose, we train the SDE on Cityscapes sequences and learn the semi-supervised semantic segmentation on the CamVid dataset (Brostow et al., 2009). As we assume in this scenario that there are no image sequences available for SDE training on CamVid, we only apply transfer learning but no multi-task learning.

Table 11 shows that the results on CamVid are similar to our main results on Cityscapes. For 50/100/367 labeled training samples, our method improves the mIoU by + 9.0/+ 6.5/+ 3.3 percentage points. In the end, our proposed method significantly outperforms ClassMix (Olsson et al., 2021) by + 2.3 percentage points for 50 labeled samples and + 2.1 percentage points for 100 labeled samples.

5.8 Component Study for SSDA

We study the components of the SSDA framework described in Sect. 3.5 on the commonly used benchmark GTA5 → Cityscapes, where the synthetic source training samples originate from the GTA5 dataset (Richter et al., 2016) and the real target training samples are obtained from Cityscapes (Cordts et al., 2016). After the training, the network is evaluated on the target validation samples from the Cityscapes validation set. First, we analyze our contributions from SSL in an SSDA setting by naively adding the additional source samples to the training according to Eq. (18). The remaining framework is the same as in the previous experiments. To indicate that DepthMix is applied specifically to the target domain in this experiment (as opposed to both domains), we denote it as Target-Domain DepthMix (TDM). TDM is equivalent to the single-domain DepthMix of the previous sections as both operate on Cityscapes.

The first part of Table 12 shows the results using the SSL framework without source domain supervision, while the second part shows the results for the framework with additional semantic segmentation supervision from the source domain according to Eq. (18).

Table 12 Comparison of the previous framework components in a SSDA setting (SD: additional source domain data, S: data selection, TDM: Target-Domain DepthMix, MTL: SDE multi-task learning)

SD	S	TDM	MTL	100 Trg. Labels	500 Trg. Labels
				48.75 ± 1.62	61.66 ± 0.90
	✓	✓	✓	62.09 ± 0.39	67.75 ± 0.10
✓				53.83 ± 1.09	60.99 ± 1.04
✓			✓	56.20 ± 0.92	62.46 ± 1.04
✓		✓		60.05 ± 1.91	66.19 ± 0.80
✓	✓			54.92 ± 0.68	61.97 ± 0.74
✓	✓	✓	✓	64.54 ± 0.12	68.63 ± 0.34

Table 13 Comparison of the class-wise IoU in % of the previous framework components in a SSDA setting for 100 labeled target samples (see Table 12 for abbreviations)

	Road	S.walk	Building	Wall	Fence	Pole	Tr. Light	Tr. Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.cycle	Bicycle
Baseline	96	66	86	22	22	43	35	45	88	47	90	62	22	87	20	16	16	7	56
SD	94	58	86	36	28	44	42	43	88	44	89	63	28	88	50	43	17	32	49
SD+MTL	94	63	87	38	34	46	43	48	88	45	89	66	36	89	52	41	22	32	57
SD+TDM	97	74	89	46	34	47	47	55	89	52	93	68	38	91	63	35	32	31	60
SD+S	93	57	86	38	30	42	41	47	88	46	89	64	30	87	52	53	15	32	51
SD+S+TDM+MTL	96	74	89	48	43	49	50	61	90	54	92	71	45	90	56	71	54	29	64

The color visualizes the IoU difference with respect to SD

For 100 labeled samples from the target domain, Table 12 shows that additional source domain supervision improves the performance of the baseline by + 5.08 percentage points. As can be seen in Table 13, this is mainly due to improvements for classes with a low baseline performance such as wall, fence, traffic light, rider, truck, bus, and motorcycle. However, additional source domain supervision deteriorates the performance for the classes sidewalk, terrain, and bicycle, which are easy to confuse and have a considerable domain gap. When applying our proposed methods from SSL, they also lead to an improved performance in the SSDA setting as shown in the second part of Table 12. For multi-task learning, the gain is + 2.37 percentage points with the same performance pattern of the class-wise IoU. For Target-Domain DepthMix, the improvement is + 6.22, while it also effectively counters the performance drop (from Baseline to SD) for the classes road, sidewalk, terrain, and bicycle (see Table 13). For automatic data selection, the improvement by additional source data is + 1.09. When combining the three contributions, the performance gain over the baseline with source supervision is + 10.71. This is + 2.45 percentage points better than our method for SSL.

For 500 labeled samples from the target domain, additional source domain supervision decreases the performance for the baseline by − 0.67 percentage points (see Table 12). This shows that additional source supervision is not helpful in this case, probably, because there is already decent supervision on the target domain and naively adding the source domain loss cannot close the domain gap. But also in this setting, multi-task learning/Target-Domain DepthMix/data selection can still improve the performance by + 1.47/+ 5.2/+ 0.98 over the baseline with source supervision. When being combined, their performance gain is + 7.64. This is + 0.88 percentage point better than our method for SSL.

Next, we analyze our contributions tailored to overcome the domain gap of SSDA: Cross-Domain DepthMix (see Sect. 3.5.2) and Matching Geometry Sampling (see Sect. 3.5.3). Table 14 shows that both Cross-Domain DepthMix (CDM) and Target-Domain DepthMix (TDM) significantly outperform the baseline. As shown in Table 15, this is due to an improved performance for difficult classes such as sidewalk, wall, traffic sign, terrain, rider, truck, train, and motorcycle. Through DepthMix presenting these objects with different backgrounds and occlusions, the network learns to generalize better within the target domain (for TDM) or across domains (CDM). When comparing the performance of CDM and TDM (see Table 14), it can be seen that CDM works better for 100 labeled target samples and TDM works better for 500. On the one side, CDM can exploit the labeled source data to propagate its knowledge to the target data through mixing. This is especially useful if there are only a few labeled target samples available and most supervision comes from the source domain. On the other side, TDM can use the already labeled target samples to propagate their knowledge to the unlabeled target through mixing, without being impeded by a domain gap. This is most effective when there are sufficient labels from the target domain available.

Based on this observation, we conclude that it might be useful to combine CDM and TDM to align labeled source and target samples as well as labeled target and unlabeled target samples. Table 14 shows that CDM + TDM indeed improves the performance over only CDM and only TDM by + 0.70 (+ 0.79) for 100 (500) labeled target samples due to an improved performance for the classes sidewalk, wall, fence, traffic sign, terrain, and train.

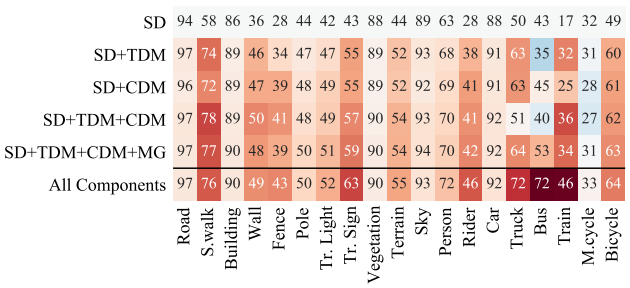
To further improve the Cross-Domain DepthMix, we apply the proposed Matching Geometry Sampling to overcome the geometric domain gap of source and target domain and to better align the geometric distribution of the mixed samples to the geometric target distribution as discussed in Sect. 3.5.3. Table 14 shows that it improves the mIoU by + 1.65 (+ 0.16) percentage points for 100 (500) labeled target samples. The geometry and view alignment is probably more important for fewer labeled target samples because it is more difficult to bridge the geometric domain gap. For 100 labeled

Table 14 Comparison of domain-adaptive mixing strategies (SD: additional source domain data, S: data selection, TDM: Target-Domain DepthMix, CDM: Cross-Domain DepthMix, MG: Matching Geometry Sampling, MTL: SDE multi-task learning)

SD	S	TDM	CDM	MG	MTL	100 Trg. labels	500 Trg. labels
✓						53.83 ± 1.09	60.99 ± 1.04
✓		✓				60.05 ± 1.91	66.19 ± 0.80
✓			✓			60.65 ± 1.88	65.34 ± 0.08
✓		✓	✓			61.35 ± 1.39	66.98 ± 0.88
✓		✓	✓	✓		63.00 ± 2.09	67.14 ± 0.42
✓	✓	✓	✓	✓	✓	66.01 ± 0.32	69.88 ± 0.39

mIoU in %, standard deviation over 3 seeds

Table 15 Comparison of the class-wise IoU in % of domain-adaptive mixing strategies for 100 labeled target samples (see Table 14 for abbreviations)



The color visualizes the IoU difference with respect to SD

samples, the improvement mainly originates from difficult vehicles such as truck, bus, and motorcycle (see Table 15).

When combining the domain adaptive strategies (combined CDM + TDM and Matching Geometry Sampling) with the previous contributions from SSL, the SSDA performance can be further improved by + 3.01 (+ 2.74) percentage points for 100 (500) labeled target samples (see Table 14). Overall, our contributions sum up to + 17.26 (+ 8.22) percentage points improvement over the baseline using only target supervision and + 12.18 (+ 8.89) percentage points improvement over the baseline with target and source supervision. Especially, the performance of truck, bus, and train is increased by more than 50% as shown in Table 15.

5.9 Comparison with State-of-the-Art SSDA Methods

Finally, we compare our framework with other state-of-the-art SSDA methods on the benchmarks Synthia → Cityscapes (Table 16) and GTA → Cityscapes (Table 17). For each method, its baseline performance is provided because the methods differ in their architecture and labeled subset. For better comparability between the architectures, we show the relative performance in % with respect to the fully-supervised baseline. As the previous SSDA methods did not publish their implementation, labeled subset, or variance over the subset selection, we adapted the UDA state-of-the-art methods DACS (Tranheden et al., 2021) to our framework for a fair comparison with a competitive method.

Considering the mIoU and the relative performance with respect to the fully-supervised baseline, our method notice-

Table 16 Comparison with other SSDA methods for GTA → Cityscapes

# Labeled (Target)	100		200		500		2975	
	mIoU	Rel	mIoU	Rel	mIoU	Rel	mIoU	Rel
Baseline (Wang et al., 2020)	43.6		47.1		53.6		65.9	Ref
ASS (Wang et al., 2020)	54.2	82.3	56.0	85.0	60.2	91.4	69.1	104.9
Baseline (Alonso et al., 2021)	–		–		–		66.4	Ref
Alonso et al. (2021)	59.9	90.2	62.0	93.4	64.2	96.7	–	
Baseline (Chen et al., 2021a)	41.9		47.7		55.5		65.3	Ref
Chen et al. (2021a)	61.2	93.7	60.5	92.6	64.3	98.5	69.8	**106.9**
Baseline	48.75 ± 1.52		54.04 ± 0.64		61.66 ± 0.90		67.77 ± 0.13	Ref
DACS (Tranheden et al., 2021)[a]	61.04 ± 0.64	90.1	63.14 ± 1.00	93.2	64.89 ± 0.45	95.8	66.51 ± 0.18	98.1
Ours w/o Data Selection	64.14 ± 1.96	94.6	66.13 ± 0.20	97.6	68.16 ± 0.40	100.6	**71.71** ± 0.44	105.8
Ours	**66.01** ± 0.32	**97.4**	**67.73** ± 0.43	**99.9**	**69.88** ± 0.39	**103.1**	–	

The mIoU in % on the Cityscapes validation set is shown for a different number of labeled target samples. Mean and standard deviation are aggregated over 3 random seeds. Additionally, the relative performance (Rel.) in % with respect to the fully-supervised baseline is shown. The best results are shown in bold font and the second-best results are underlined

[a]Results of the reimplementation in our experiment setting extending DACS from UDA to SSDA

Table 17 Comparison with other SSDA methods for Synthia → Cityscapes

# Labeled (Target)	100		200		500		2975	
	mIoU	Rel	mIoU	Rel	mIoU	Rel	mIoU	Rel
Baseline (Wang et al., 2020)	57.6		60.8		66.5		73.8	Ref
ASS (Wang et al., 2020)	62.1	84.1	64.8	87.8	69.8	94.6	77.1	104
Baseline (Chen et al., 2021a)	53		58.9		61		72.2	Ref
Chen et al. (2021a)	68.4	94.7	69.8	96.7	71.7	99.3	**77.2**	**106.9**
Baseline	58.00 ± 1.96		63.26 ± 0.91		67.74 ± 0.48		73.34 ± 0.21	Ref
DACS (Tranheden et al., 2021)[a]	64.88 ± 0.30	88.5	67.72 ± 1.19	92.3	71.32 ± 0.38	97.2	74.43 ± 0.41	101.5
Ours w/o Data Selection	68.89 ± 1.94	93.9	71.95 ± 0.49	98.1	74.06 ± 0.30	101.0	77.04 ± 0.31	105.0
Ours	**72.35** ± 0.23	**98.7**	**73.54** ± 0.67	**100.3**	**75.36** ± 0.26	**102.8**	–	

The mIoU in % of 13 classes on the Cityscapes validation set is shown for a different number of labeled target samples. Mean and standard deviation are aggregated over 3 random seeds. Additionally, the relative performance (Rel.) in % with respect to the fully-supervised baseline is shown. The best results are shown in bold font and the second-best results are underlined

[a]Results of the reimplementation in our experiment setting extending DACS from UDA to SSDA

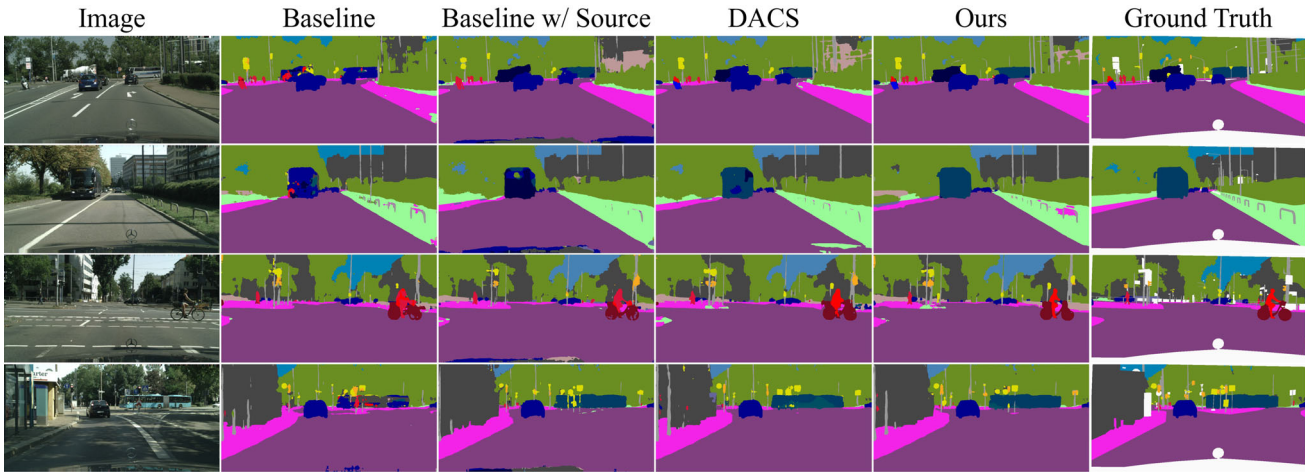


Fig. 10 Example semantic segmentations from GTA5 → Cityscapes of our method for 100 labeled target samples in comparison with DACS (Tranheden et al., 2021) adapted to SSDA and the baseline with/without source supervision

ably outperforms the competitors for 100, 200, and 500 labeled target samples on both benchmarks. Only in the fully-supervised case, Chen et al. (2021a) achieves slightly better results. Moreover, it can be seen that even if we remove the data selection for annotation from our method, the previous statements still hold.

We would like to highlight that our method achieves 97.4% (GTA → Cityscapes) and 98.7% (Synthia → Cityscapes) of the fully-supervised baseline performance with only about 1/30 (100) of the target labels. With about 1/15 of the target labels, it even reaches the fully-supervised baseline performance. The improved performance for 100 labeled target samples can also be observed in Fig. 10, where our method better distinguishes difficult classes such as truck, bus, and train and produces more detailed segmentation contours for classes such as pole, traffic sign, and rider.

6 Conclusions

In this work, we have studied how self-supervised depth estimation (SDE) can be utilized to improve semantic segmentation in the single-domain semi-supervised and the domain-adaptive semi-supervised setting.

We introduce four effective strategies capable of leveraging the knowledge learned from SDE. First, we present an automatic data selection for annotation algorithm based on SDE, which does not require human-in-the-loop annotations and, therefore, increases flexibility, efficiency, and scalability. By combining diversity sampling based on features from self-supervised depth estimation and uncertainty sampling based on the depth student error, our method significantly outperforms random data selection and even entropy-based active learning, which requires a human in the loop. We show that without knowledge of the class labels, our data selection for annotation prefers samples, which contain difficult/rare

classes (e.g. rider, truck, bus, and train). This results in a significantly higher semantic segmentation performance of these classes.

Second, we demonstrate that the proposed DepthMix strategy outperforms related mixing strategies by avoiding an inconsistent geometry of the generated images. We show that DepthMix effectively improves the performance for classes with a low baseline performance such as wall, fence, traffic light, rider, truck, bus, and train. We assume that DepthMix improves generalization by presenting labeled and pseudo-labeled instances with different backgrounds and occlusions.

Third, we show that the feature representation from self-supervised depth estimation can be transferred to semantic segmentation, by means of SDE pretraining and multi-task learning of semantic segmentation and SDE. This is particularly effective for difficult classes surrounded by depth discontinuities such as wall, fence, pole, traffic, light, traffic sign, rider, truck, and motorcycle. By using an ImageNet feature distance loss during the SDE pretraining, we mitigate forgetting useful semantic features from ImageNet pretraining and avoid the resulting performance drop for semantically similar classes such as truck, bus, train, and motorcycle.

And fourth, we show the effectiveness of combined Cross-Domain and Target-Domain DepthMix as well as Matching Geometry Sampling in a semi-supervised domain adaptation setting. The former effectively aligns source and target data as well as labeled target and unlabeled data to generate high-quality pseudo-labels for unlabeled target data. The latter samples source images with a similar scene geometry and camera pose with respect to target images to produce more realistic Cross-Domain DepthMix images.

A combination of the first three contributions in a single-domain semi-supervised framework can achieve even higher performance gains than the single components as the approaches address different aspects of the learning process. By using these SDE-based contributions, our approach results in state-of-the-art performance for semi-supervised semantic segmentation. Our method achieves 92% of the fully-supervised baseline performance with only 1/30 of the available labels and even slightly outperforms it with only 1/8 of the labels.

A combination of all four contributions in a semi-supervised domain adaptation framework improves the performance even further and outperforms previous state-of-the-art semi-supervised domain adaptation methods. On GTA $\rightarrow$ Cityscapes, our method achieves even 97% of the fully-supervised baseline performance with only 1/30 of the target labels. This roughly corresponds to only 150 working hours for data annotation for the target domain instead of 4460 working hours.

All in all, our findings suggest that SDE can be a valuable source of self-supervision for semantic segmentation, improving the semantic segmentation performance and reducing the number of necessary annotations.

Funding Open access funding provided by Swiss Federal Institute of Technology Zurich

Data availibility For this paper only publicly available datasets were used.

Code Availability The source code of this paper is available at https://github.com/lhoyer/improving_segmentation_with_selfsupervised_depth.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

Alonso, I., Sabater, A., Ferstl, D., Montesano, L., & Murillo, A. C. (2021). Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. arXiv:2104.13415.

Araslanov, N., & Roth, S. (2021). Self-supervised augmentation consistency for adapting semantic segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 15384–15394).

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: a holistic approach to semi-supervised learning. In *Adv. neural inform. process. syst.* (pp. 5049–5059).

Brostow, G. J., Fauqueur, J., & Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters, 30*(2), 88–97.

Casser, V., Pirk, S., Mahjourian, R., & Angelova, A. (2019). Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI conf. artif. intell.* (pp. 8001–8008).

Chapelle, O., Scholkopf, B., & Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks, 20*(3), 542–542.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(4), 834–848.

Chen, P. Y., Liu, A. H., Liu, Y. C., & Wang, Y. C. F. (2019a). Towards scene understanding: Unsupervised monocular depth estimation

with semantic-aware representation. In *IEEE conf. comput. vis. pattern recog.* (pp. 2624–2632).
Chen, S., Jia, X., He, J., Shi, Y., & Liu, J. (2021a). Semi-supervised domain adaptation based on dual-level domain mixing for semantic segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 11018–11027).
Chen, X., Yuan, Y., Zeng, G., & Wang, J. (2021b). Semi-supervised semantic segmentation with cross pseudo supervision. In *IEEE conf. comput. vis. pattern recog.* (pp. 2613–2622).
Chen, Y., Li, W., Chen, X., Gool, L. V. (2019b). Learning semantic segmentation from synthetic data: A geometrically guided input–output adaptation approach. In *IEEE conf. comput. vis. pattern recog.* (pp. 1841–1850).
Chen, Y., Schmid, C., & Sminchisescu, C. (2019c). Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *Int. conf. comput. vis.* (pp. 7063–7072).
Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *IEEE conf. comput. vis. pattern recog.* (pp. 3213–3223).
Dai, D., & Van Gool, L. (2018) Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *IEEE int. conf. on intell. transport. syst.* (pp. 3819–3824).
Dai, Q., Patil, V., Hecker, S., Dai, D., Van Gool, L., & Schindler, K. (2020). Self-supervised object motion and depth estimation from video. In *IEEE conf. comput. vis. pattern recog. workshops* (pp. 1004–1005).
Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Int. conf. comput. vis.* (pp. 1422–1430).
Dvornik, N., Mairal, J., & Schmid, C. (2019). On the importance of visual context for data augmentation in scene understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*, 2014–2028.
Feng, Z., Zhou, Q., Cheng, G., Tan, X., Shi, J., & Ma, L. (2020a) Semi-supervised semantic segmentation via dynamic self-training and class-balanced curriculum. arXiv:2004.08514.
Feng, Z., Zhou, Q., Gu, Q., Tan, X., Cheng, G., Lu, X., Shi, J., & Ma, L. (2020b). Dmt: Dynamic mutual training for semi-supervised learning. arXiv:2004.08514.
French, G., Laine, S., Aila, T., Mackiewicz, M., & Finlayson, G. (2020) Semi-supervised semantic segmentation needs strong, varied perturbations. In *Brit. mach. vis. conf.*
Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Int. conf. mach. learning* (pp. 1050–1059).
Garg, R., BG, V. K., Carneiro, G., Reid, I. (2016) Unsupervised CNN for single view depth estimation: geometry to the rescue. In *Eur. conf. comput. vis.* (pp. 740–756).
Gidaris, S., Singh, P., & Komodakis, N. (2018) Unsupervised representation learning by predicting image rotations. In *Int. conf. learn. represent*.
Godard, C., Mac Aodha, O., & Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *IEEE conf. comput. vis. pattern recog.* (pp. 270–279).
Godard, C., Mac Aodha, O., Firman, M., Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In *Int. conf. comput. vis.* (pp. 3828–3838).
Gonzalez Bello, J. L., & Kim, M. (2020) Forget about the lidar: Self-supervised depth estimators with med probability volumes. In *Adv. neural inform. process. syst*.
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Adv. neural inform. process. syst.* (pp. 2672–2680).
Gordon, A., Li, H., Jonschkowski, R., & Angelova, A. (2019). Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Int. conf. comput. vis.* (pp. 8977–8986).
Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., & Gaidon, A. (2020a). 3d packing for self-supervised monocular depth estimation. In *IEEE conf. comput. vis. pattern recog.* (pp. 2485–2494).
Guizilini, V., Hou, R., Li, J., Ambrus, R., & Gaidon, A. (2020b). Semantically-guided representation learning for self-supervised monocular depth. In *Int. conf. learn. represent*.
Guizilini, V., Li, J., Ambrus, R., & Gaidon, A. (2021) Geometric unsupervised domain adaptation for semantic segmentation. arXiv:2103.16694
Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *IEEE conf. comput. vis. pattern recog.* (pp. 1735–1742).
He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *IEEE conf. comput. vis. pattern recog.* (pp 9729–9738).
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conf. comput. vis. pattern recog.* (pp. 770–778).
Hoffman, J., Tzeng, E., Park, T., Zhu, J. Y., Isola, P., Saenko, K., Efros, A., & Darrell, T. (2018).Cycada: Cycle-consistent adversarial domain adaptation. In *Int. conf. mach. learning* (pp. 1989–1998).
Hoffman, J., Wang, D., Yu, F., Darrell, T. (2016). Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv:1612.02649.
Hoyer, L., Dai, D., Chen, Y., Köring, A., Saha, S., & Van Gool, L. (2021). Three ways to improve semantic segmentation with self-supervised depth estimation. In *IEEE conf. comput. vis. pattern recog*.
Hung, W. C., Tsai, Y. H., Liou, Y. T., Lin, Y. Y., & Yang, M. H. (2018). Adversarial learning for semi-supervised semantic segmentation. In *Brit. mach. vis. conf.*
Huo, X., Xie, L., He, J., Yang, Z., Zhou, W., Li, H., & Tian, Q. (2021). Atso: Asynchronous teacher–student optimization for semi-supervised image segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 1235–1244).
Hwa, R. (2004). Sample selection for statistical parsing. *Computational Linguistics, 30*(3), 253–276.
Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.
Jiang, H., Larsson, G., Maire Greg Shakhnarovich, M., & Learned-Miller, E. (2018) Self-supervised relative depth learning for urban scene understanding. In *Eur. conf. comput. vis.* (pp. 19–35).
Jiang, H., Sun, D., Jampani, V., Lv, Z., Learned-Miller, E., & Kautz, J. (2019). Sense: A shared encoder network for scene-flow estimation. In *Int. conf. comput. vis.* (pp. 3195–3204).
Jiao, J., Cao, Y., Song, Y., & Lau, R. (2018). Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Eur. conf. comput. vis.* (pp. 53–69).
Kalluri, T., Varma, G., Chandraker, M., & Jawahar, C. (2019). Universal semi-supervised semantic segmentation. In *Int. conf. comput. vis.* (pp. 5259–5270).
Kasarla, T., Nagendar, G., Hegde, G. M., Balasubramanian, V., & Jawahar, C. (2019). Region-based active learning for efficient labeling in semantic segmentation. In *IEEE winter conf. appl. of comput. vis.* (pp. 1109–1117).
Kim, M., & Byun, H. (2020). Learning texture invariant representation for domain adaptation of semantic segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 12975–12984).
Klingner, M., Bar, A., & Fingscheidt, T. (2020a). Improved noise and attack robustness for semantic segmentation by using multi-task training with self-supervised depth estimation. In *IEEE conf. comput. vis. pattern recog. workshops* (pp. 320–321).

Klingner, M., Termöhlen, J. A., Mikolajczyk, J., & Fingscheidt, T. (2020b). Self-supervised monocular depth estimation: solving the dynamic object problem by semantic guidance. In *Eur. conf. comput. vis.* (pp. 582–600).

Lai, X., Tian, Z., Jiang, L., Liu, S., Zhao, H., Wang, L., & Jia, J. (2021). Semi-supervised semantic segmentation with directional context-aware consistency. In *CVPR* (pp. 1205–1214).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

Lee, D. H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Int. conf. mach. learning*.

Lee, K. H., Ros, G., Li, J., & Gaidon, A. (2018) Spigan: Privileged adversarial learning from simulation. In *Int. conf. learn. represent*.

Li, C., Ma, H., Kang, Z., Yuan, Y., Zhang, X. Y., & Wang, G. (2020a). On deep unsupervised active learning. *Int Joint Conf Artif Intell*.

Li, C., Wang, X., Dong, W., Yan, J., Liu, Q., & Zha, H. (2018). Joint active learning with feature selection via cur matrix decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 41*(6), 1382–1396.

Li, G., Kang, G., Liu, W., Wei, Y., & Yang, Y. (2020b). Content-consistent matching for domain adaptive semantic segmentation. In *Eur. conf. comput. vis.* (pp. 440–456).

Lian, Q., Lv, F., Duan, L., & Gong, B. (2019). Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *Int. conf. comput. vis.* (pp. 6758–6767).

Liu, S., Johns, E., & Davison, A. J. (2019). End-to-end multi-task learning with attention. In *IEEE conf. comput. vis. pattern recog.* (pp. 1871–1880).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 3431–3440).

Mackowiak, R., Lenz, P., Ghori, O., Diego, F., Lange, O., & Rother, C. (2018). Cereals-cost-effective region-based active learning for semantic segmentation. In *Brit. mach. vis. conf.*

McCallumzy, A. K., & Nigamy, K. (1998). Employing em and pool-based active learning for text classification. In *Int. conf. mach. learning* (pp. 359–367).

Mendel, R., De Souza, L. A., Rauber, D., Papa, J. P., & Palm, C. (2020). Semi-supervised segmentation based on error-correcting supervision. In *Eur. conf. comput. vis.* (pp. 141–157).

Mittal, S., Tatarchenko, M., & Brox, T. (2019). Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*, 1369–1379.

Nie, F., Wang, H., Huang, H., & Ding, C. (2013). Early active learning via robust representation and structured sparsity. In *Int. joint conf. artif. intell.*

Novosel, J., Viswanath, P., & Arsenali, B. (2019). Boosting semantic segmentation with multi-task self-supervised learning for autonomous driving applications. In *Int. conf. comput. vis. workshops*.

Olsson, V., Tranheden, W., Pinto, J., & Svensson, L. (2021). Classmix: Segmentation-based data augmentation for semi-supervised learning. In *IEEE winter conf. on applications of comput. vis.* (pp. 1369–1378).

Ouali, Y., Hudelot, C., & Tami, M. (2020). Semi-supervised semantic segmentation with cross-consistency training. In *IEEE conf. comput. vis. pattern recog.* (pp. 12674–12684).

Pilzer, A., Lathuiliere, S., Sebe, N., & Ricci, E. (2019). Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. In *IEEE conf. comput. vis. pattern recog.* (pp. 9768–9777).

Pilzer, A., Xu, D., Puscas, M., Ricci, E., & Sebe, N. (2018). Unsupervised adversarial depth estimation using cycled generative networks. In *Int. conf. on 3D vision* (pp. 587–595).

Ramirez, P. Z., Poggi, M., Tosi, F., Mattoccia, S., & Di Stefano, L. (2018). Geometry meets semantics for semi-supervised monocular depth estimation. In *Asian conf. comput. vis.* (pp. 298–313).

Ramirez, P. Z., Tonioni, A., Salti, S., & Stefano, L. D. (2019). Learning across tasks and domains. In *Int. conf. comput. vis.* (pp. 8110–8119).

Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., & Black, M. J. (2019). Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 12240–12249).

Richter, S. R., Hayder, Z., & Koltun, V. (2017). Playing for benchmarks. In *Int. conf. comput. vis.* (pp. 2213–2222).

Richter, S. R., Vineet, V., Roth, S., & Koltun, V. (2016). Playing for data: Ground truth from computer games. In *Eur. conf. comput. vis.* (pp. 102–118).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Int. conf. medical image computing and computer-assisted intervention* (pp. 234–241).

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE conf. comput. vis. pattern recog.* (pp. 3234–3243).

Sakaridis, C., Dai, D., & Van Gool, L. (2018). Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision, 126*(9), 973–992.

Sakaridis, C., Dai, D., & Van Gool, L. (2021) ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Int. conf. comput. vis.*

Sener, O., & Savarese, S. (2018). Active learning for convolutional neural networks: A core-set approach. In *Int. conf. learn. represent*.

Settles, B. (2009). *Active learning literature survey*. Tech. rep.: University of Wisconsin-Madison Department of Computer Sciences.

Settles, B., & Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Conf. empirical methods natural language processing* (pp. 1070–1079).

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. In *Annual workshop computational learning theory* (pp. 287–294).

Shu, C., Yu, K., Duan, Z., & Yang, K. (2020) Feature-metric loss for self-supervised learning of depth and egomotion. In *Eur. conf. comput. vis.* (pp. 572–588).

Siddiqui, Y., Valentin, J., & Nießner, M. (2020) .Viewal: Active learning with viewpoint entropy for semantic segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 9433–9443).

Sinha, S., Ebrahimi, S., & Darrell, T. (2019). Variational adversarial active learning. In *Int. conf. comput. vis.* (pp. 5972–5981).

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., & Li, C. L. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Adv. neural inform. process. syst*.

Souly, N., Spampinato, C., & Shah, M. (2017). Semi supervised semantic segmentation using generative adversarial network. In *Int. conf. comput. vis.* (pp. 5688–5696).

Tarvainen, A., & Valpola, H. (2017) Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Adv. neural inform. process. syst.* (pp. 1195–1204).

Tranheden, W., Olsson, V., Pinto, J., & Svensson, L. (2021) Dacs: Domain adaptation via cross-domain mixed sampling. In *IEEE winter conf. on applications of comput. vis.* (pp. 1379–1389).

Tsai, Y. H., Hung, W. C., Schulter, S., Sohn, K., Yang, M. H., & Chandraker, M. (2018). Learning to adapt structured output space for semantic segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 7472–7481).

Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., & Van Gool, L. (2021). Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*, 3614–3633.

Verma, V., Lamb, A., Kannala, J., Bengio, Y., & Lopez-Paz, D. (2019). Interpolation consistency training for semi-supervised learning. In *Int. joint conf. artif. intell.* (pp. 3635–3641).

Vu, T. H., Jain, H., Bucher, M., Cord, M., & Pérez, P. (2019a). Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 2517–2526).

Vu, T. H., Jain, H., Bucher, M., Cord, M., & Pérez, P. (2019b). Dada: Depth-aware domain adaptation in semantic segmentation. In *Int. conf. comput. vis.* (pp. 7364–7373).

Wang, Q., Dai, D., Hoyer, L., Fink, O., & Van Gool, L. (2021). Domain adaptive semantic segmentation with self-supervised depth estimation. In*Int. conf. comput. vis*.

Wang, R., Pizer, S. M., & Frahm, J. M. (2019). Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *IEEE conf. comput. vis. pattern recog.* (pp. 5555–5564).

Wang, Z., Wei, Y., Feris, R., Xiong, J., Hwu, W. M., Huang, T. S., & Shi, H. (2020). Alleviating semantic-level shift: A semi-supervised domain adaptation method for semantic segmentation. In *IEEE conf. comput. vis. pattern recog. workshops* (pp. 936–937).

Wei, Y., Xiao, H., Shi, H., Jie, Z., Feng, J., & Huang, T. S. (2018). Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 7268–7277).

Xie, S., Feng, Z., Chen, Y., Sun, S., Ma, C., & Song, M. (2020). Deal: Difficulty-aware active learning for semantic segmentation. In *Asian conf. comput. vis*.

Xu, D., Ouyang, W., Wang, X., & Sebe, N. (2018). Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *IEEE conf. comput. vis. pattern recog.* (pp. 675–684).

Yang, G., Zhao, H., Shi, J., Deng, Z., & Jia, J. (2018). Segstereo: Exploiting semantic information for disparity estimation. In *Eur. conf. comput. vis.* (pp. 636–651).

Yang, L., Zhang, Y., Chen, J., Zhang, S., & Chen, D. Z. (2017). Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *Int. conf. medical image computing and computer-assisted intervention* (pp. 399–407).

Yang, Y., & Soatto, S. (2020) Fda: Fourier domain adaptation for semantic segmentation. In *IEEE conf. comput. vis. pattern recog.* (pp. 4085–4095).

Yin, Z., & Shi, J. (2018). Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *IEEE conf. comput. vis. pattern recog.* (pp. 1983–1992).

Yu, K., Bi, J., & Tresp, V. (2006). Active learning via transductive experimental design. In *Int. conf. mach. learning* (pp. 1081–1088).

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Int. conf. comput. vis.* (pp. 6023–6032).

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

Zhang, Y., David, P., Foroosh, H., & Gong, B. (2019). A curriculum domain adaptation approach to the semantic segmentation of urban scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(8), 1823–1841.

Zheng, H., Yang, L., Chen, J., Han, J., Zhang, Y., Liang, P., Zhao, Z., Wang, C., & Chen, D. Z. (2019). Biomedical image segmentation via representative annotation. In *AAAI conf. artif. intell.* (pp. 5901–5908).

Zheng, Z., & Yang, Y. (2021). Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision, 129*(4), 1106–1120.

Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. In *IEEE conf. comput. vis. pattern recog.* (pp. 1851–1858).

Zou, Y., Yu, Z., Kumar, B., & Wang, J. (2018). Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Eur. conf. comput. vis.* (pp. 289–305).