Empirical Generalization Study: Unsupervised Domain Adaptation vs. Domain Generalization Methods for Semantic Segmentation in the Wild

Fabrizio J. Piva Daan de Geus Gijs Dubbelman
Eindhoven University of Technology
{f.j.piva, d.c.d.geus, g.dubbelman}@tue.nl

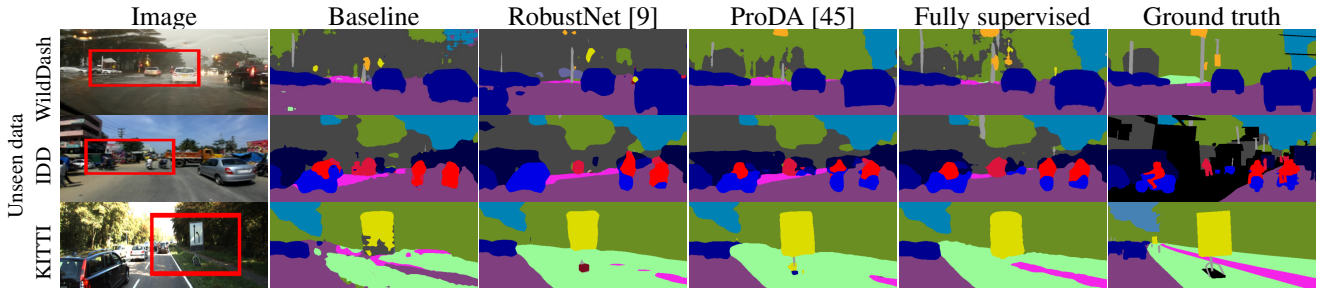

Figure 1. **Qualitative results of state-of-the-art domain generalization (RobustNet [9]), unsupervised domain adaptation (ProDA [45]) and fully-supervised [6] methods on unseen datasets WildDash [44], IDD [37], and KITTI [1].** The baseline model and RobustNet are trained only on Cityscapes [10], ProDA is trained on the labeled data of Cityscapes and the unlabeled data of BDD-100K [42] and Mapillary Vistas [25], and the fully-supervised model is trained on the respective 'unseen' dataset. The ProDA method, which leverages unlabeled data, consistently performs best, showing that generalization can be improved by using non-annotated examples.

Abstract

For autonomous vehicles and mobile robots to safely operate in the real world, i.e., the wild, scene understanding models should perform well in the many different scenarios that can be encountered. In reality, these scenarios are not all represented in the model's training data, leading to poor performance. To tackle this, current training strategies attempt to either exploit additional unlabeled data with unsupervised domain adaptation (UDA), or to reduce overfitting using the limited available labeled data with domain generalization (DG). However, it is not clear from current literature which of these methods allows for better generalization to unseen data from the wild. Therefore, in this work, we present an evaluation framework in which the generalization capabilities of state-of-the-art UDA and DG methods can be compared fairly. From this evaluation, we find that UDA methods, which leverage unlabeled data, outperform DG methods in terms of generalization, and can deliver similar performance on unseen data as fully-supervised training methods that require all data to be labeled. We show that semantic segmentation performance can be increased up to 30% for a priori unknown data without using any extra labeled data.

1. Introduction

Semantic segmentation, where each pixel in an image needs to be classified, is a useful computer vision task for applications like autonomous driving and mobile robotics. When applied to images from cameras mounted on such mobile agents, it can provide them with actionable information. However, state-of-the-art semantic segmentation methods are all based on Deep Neural Networks (DNNs), for which it is typically challenging to perform well in all real-world conditions, i.e., *the wild*. To combat this, several methods have been introduced that aim to improve the *generalization* capabilities of such DNNs, so that they perform well under all imaging conditions. However, all these methods have been researched in different settings, which makes it difficult to make a fair comparison and identify the best solution for a given application. To solve this, in this work, we conduct an empirical study on the generalization capabilities of different methods under fair conditions, and provide key insights and take-away messages.

Semantic segmentation networks typically perform quite well on images that are similar to the data that they are trained on [6, 7, 46]. However, when they encounter images captured under different conditions, e.g., in different weather, with other lighting, or with changed camera properties, their performance degrades [16, 40]. This is prob-

task	method	semantic segmentation architecture		training domains		validation domains
		encoder	decoder	labeled	unlabeled	
UDA	ProDA [45]	ResNet-101 [14]	DeepLab v2 [6]	GTA V [29]	Cityscapes [10]	Cityscapes [10]
	DSP [12]	ResNet-101 [14]	DeepLab v2 [6]			
	SAC [2]	ResNet-101 [14]	DeepLab v2 [6]	Synthia [31]	Cityscapes [10]	Cityscapes [10]
		VGG-16 [35]	DeepLab v2 [6]			
DG	WildNet [18]	ResNet-50 [14]	DeepLab v3+ [7]	GTA V [29]	n.a	Cityscapes [10], BDD-100K [42], Mapillary [25], Synthia [31]
		ResNet-101 [14]	DeepLab v3+ [7]			
		VGG-16 [35]	DeepLab v3+ [7]			
	RobustNet [9]	ResNet-50 [14]	DeepLab v3+ [7]	Cityscapes [10]	n.a	BDD-100K [42], Mapillary [25], GTA V [29], Synthia [31]
		ShuffleNet v2 [21]	DeepLab v3+ [7]			
		MobileNet v2 [32]	DeepLab v3+ [7]			

Table 1. **Semantic segmentation architectures and training settings commonly used by top performing UDA and DG methods.** This shows that UDA and DG methods typically use very different network architectures, training data and evaluation data, making direct comparisons between methods of different tasks very difficult.

lematic when deploying semantic segmentation networks in the wild, because it is *a priori* unknown if the data that a vehicle or robot will capture is similar to the training data. One obvious solution is to gather more, possibly heterogeneous [23], training data, captured under as many different conditions as possible. However, especially for semantic segmentation, obtaining per-pixel labels is expensive and time-consuming [10]. Moreover, there is no guarantee that a gathered dataset contains images with all the conditions that can be encountered during deployment.

As an alternative to increasing the training dataset, significant research is focused on finding methods that allow deep learning models to generalize better to environments that are not part of the training data. On a high level, we can identify two tasks that focus on improving generalization of deep learning models in different ways: a) *domain generalization* (DG), and b) *unsupervised domain adaption* (UDA). DG methods take one or multiple labeled datasets and apply techniques to generate a model that performs well on multiple datasets that were not seen during training [9, 18]. UDA methods assume that they have access to unlabeled images from the so-called *target* environment where the model is to be deployed. Therefore, they train a model jointly on a single labeled dataset and the unlabeled images from the target environment, with the goal of achieving a good performance on other images from this target environment [2, 45]. Some UDA methods have also shown that they can also boost performance on unseen environments, but this is not explored extensively thusfar [28, 30].

Although both DG and UDA aim to explicitly or implicitly improve the generalization capabilities, it is unclear which of the two actually leads to better generalization to unseen data, and under what conditions. To find this out, a quantitative comparison should be made under circumstances that are as equal as possible. Such a fair comparison is currently very difficult based on literature, due to multiple factors, illustrated in Tab. 1 with different colors. Specifically, it can be noticed that a) methods are not trained using the same semantic segmentation network architecture, while architectural differences can greatly impact performance (blue); b) each task uses different training settings, and UDA methods predominantly focus on adapting from synthetic to real data, even though abundant real-world data is available to perform real-to-real adaptations [25, 42, 44] (pink); and c) UDA methods are not evaluated for the task of generalization, i.e., they only measure performance on the *target* dataset, and do typically not report scores on unseen data (green). To address this, in this work, we propose an evaluation framework where methods are trained using a normalized architecture, on real-world labeled and unlabeled data, and are evaluated specifically on unseen datasets, to properly assess generalization capabilities.

With our proposed evaluation framework, we conduct a thorough quantitative comparison between DG and UDA methods, and assess their generalization capabilities under various conditions. Most importantly, we find that leveraging unlabeled data like in UDA, greatly boosts the generalization performance beyond the target domain. To provide additional insights, we also assess the impact of choosing a particular training dataset, and the proportion of labeled and unlabeled data that is used for training.

To summarize, the contributions of this work are:

- We provide a new evaluation framework where semantic segmentation models can be tested for generalization to unseen data in the wild.
- Using this framework, to the best of our knowledge, we are the first to provide a quantitative comparison between DG and UDA methods for semantic segmentation, to properly assess their relative performances and provide recommendations on their usage.
- From this comparison, we find that unlabeled data is an important resource to achieve generalization to unseen data, achieving performances on par with fully supervised models.

The code of this work is made publicly available[1].

[1] https://fabriziojpiva.github.io/empirical-generalization-study/

2. Related Work

Unsupervised Domain Adaptation (UDA) refers to the process of training a model able to transfer the learned knowledge from a domain where labels are accessible, to a domain where annotations are not available. In the past years, UDA methods for semantic segmentation have shown impressive results [2, 19, 45], particularly by leveraging a combination of multiple strategies, that can involve data augmentations [2, 8, 20, 22, 24, 41], feature alignments [3, 17, 19, 20, 28, 33, 36], and/or self-supervised learning [2, 12, 19, 20, 28, 39, 45]. Regardless of the training strategy, UDA methods have strongly focused on adaptation settings where a synthetically generated dataset such as GTA V [29] or SYNTHIA [31] represents the labeled domain, and needs to be adapted to a real-world dataset such as Cityscapes [10] as the unlabeled target domain. While we consider these benchmarks challenging, it is worth noting that these settings 1) assume that the unlabeled domain is the one and only domain on which the model will be deployed, abandoning the possibility that the model could encounter other domains that are *a priori* unknown, and 2) focus strongly on adapting synthetic to real data, without considering other scenarios like real-to-real adaptations, despite the availability of many datasets with annotated real images. To properly assess the ability to generalize to unseen images and leverage the availability of real datasets, our proposed evaluation framework evaluates on multiple datasets that were not seen during training, and focuses only on real-world data.

Domain Generalization (DG) methods for semantic segmentation have surged significantly recently, where the goal is to train a model on data from one or multiple labeled datasets, and let it perform well on various datasets that were not seen during training. These methods predominantly operate at feature-level, in combination with data augmentations [9, 18, 26, 43], by creating augmented versions of the input images to either suppress style-related features [9, 26], or to overexpose the network to multiple styles [18, 43], encouraging the network to learn domain-invariant features. Regardless of the methodology involved, these methods only use data from labeled domains, and therefore they are unable to exploit rich information that can be extracted from data from unlabeled domains. We consider that this is one of the main drawbacks of DG methods, since unlabeled data is significantly easier and cheaper to collect than labeled data. For this reason, we allow the usage of unlabeled examples in our evaluation framework, and we assess the benefit of having access to unlabeled data on the generalization capabilities of a network, i.e., with UDA.

Comparisons between UDA and DG methods. Previous studies comparing UDA and DG methods for computer vision tasks are mostly surveys [34, 38, 47], where different theoretical aspects such as problem definitions, training strategies and related research areas are described. One of the main drawbacks of these surveys is the lack of a practical comparison, where UDA and DG models are evaluated using a common framework to assess their applicability to real-world scenarios. In line with this, a recent study has proposed a setting to compare UDA and DG methods in a practical fashion [13], but it is focused on clinical medicine and addresses the effect of a temporal dataset shift, i.e., when the distribution of the data changes gradually over time.

In this work, rather, we compare UDA and DG methods for the computer vision task of semantic segmentation, using a common practical framework, especially designed to assess the generalization capacity of these models on domains that were not seen during training, independent of distribution shifts that can occur over time. In particular, we evaluate the effect of the aforementioned differences that exist between DG and UDA approaches: 1) the effect of leveraging unlabeled data in the network, which happens for UDA but not for DG, 2) the effect of evaluating UDA approaches on unseen domains, which is typically not done for UDA methods, but is the main objective for DG.

3. Problem definition

In this work, we address the problem of applying semantic segmentation in challenging real-world conditions, i.e., in *the wild*. Assuming that there is no available labeled dataset large and varied enough to yield good performance in all real-world conditions, we focus on methods for training semantic segmentation using limited labeled data, along with unlabeled data, with the aim of generalizing to varied unseen data. To formally define this problem, we introduce the following notations.

Notation. Let $\mathcal{X}$ be the input images, and let $\mathcal{Y}$ be their corresponding pixel-wise ground truth for semantic segmentation. A labeled domain $\mathcal{D}_l$ is defined as the joint distribution $P(\mathcal{X}, \mathcal{Y})$ on $\mathcal{X} \times \mathcal{Y}$, and it typically consist of multiple sub-domains $\mathcal{D}_l^1, ..., \mathcal{D}_l^n$. A dataset represents a random subset of samples from one or multiple (sub)domains. Respectively, an unlabeled domain is a domain where pixel-wise ground truths are not available for training, denoted as $\mathcal{D}_{nl}$, and an unseen domain $\mathcal{D}_u$ is a domain where both input images and ground truths are not available during training.

Goal of generalization. When a neural network is deployed in a mobile agent, it is a priori unknown whether the images that are captured by the sensing devices fall into domains $\mathcal{D}_l$, $\mathcal{D}_{nl}$, or $\mathcal{D}_u$. Therefore, to have a system that performs well in all circumstances, it is important that the semantic segmentation network performs well on all domains, i.e., not only on the typically evaluated training domains $\mathcal{D}_l$ and $\mathcal{D}_{nl}$, but especially on unseen domains $\mathcal{D}_u$, as illustrated in Fig. 2.

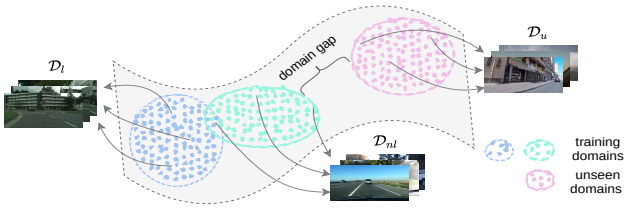

Figure 2. **Conceptual illustration of domains in the representation space.** Each domain can be seen as taking up a specific subspace of the representation space. The distance between two domains is often referred to as their domain gap and datasets can be seen as random samples, visualized by the colored dots, coming from one or multiple domains. The goal of generalization is to train a model on a labeled domain $\mathcal{D}_l$ (blue) and possibly an unlabeled domain $\mathcal{D}_{nl}$ (green) such that the model also performs well on an unseen domain $\mathcal{D}_u$ (pink). Conceptually, the envelope of the model (grey), should ideally contain all domains.

experiment	domains	datasets		# images		metrics	
		training	testing	training	testing	calculation	name
1A, 2	$\mathcal{D}_l$	CS train. split	CS val. split	2975	500	avg. over mIoU of CS, BDD, MAP val. splits	seen mIoU avg.
	$\mathcal{D}_{nl}$	BDD train. split MAP train. split	BDD val. split MAP val. split	7000 18000	1000 2000		
1B	$\mathcal{D}_l$	MAP train. split	MAP val. split	18000	2000	avg. over mIoU of CS, BDD, MAP val. splits	seen mIoU avg.
	$\mathcal{D}_{nl}$	BDD train. split CS train. split	BDD val. split CS val. split	7000 2975	1000 500		
1, 2	$\mathcal{D}_u$	- - -	WILD val. split IDD val. split KITTI val. split	3404 6993 160	852 973 40	avg. over mIoU of WILD, IDD, KITTI val. splits	unseen mIoU avg.

Acronyms for datasets: Cityscapes (CS) [10], BDD-100K (BDD) [42], Mapillary Vistas (MAP) [25], WildDash (WILD) [44], IDD (IDD) [37], KITTI (KITTI) [1].

Table 2. **Datasets and metrics for each experiment.** The proposed performance metrics, the seen and unseen mIoU avg., are computed over the validation splits of the datasets for the training domains, and the datasets for the unseen domains, respectively.

DG and UDA. Given the aforementioned notation and the description of these tasks in Sec. 2, we note that DG methods only use the labeled domain $\mathcal{D}_l$ during training, and evaluate on $\mathcal{D}_u$, which is unseen for them. In contrast, UDA methods use the labeled domain $\mathcal{D}_l$ and the unlabeled domains $\mathcal{D}_{nl}$ during training. Whereas UDA methods are normally evaluated on an unseen split of $\mathcal{D}_{nl}$, we now evaluate on other, unseen domains $\mathcal{D}_u$, to assess their generalization capabilities.

4. Experiments

4.1. Overview of experiments

The main goal of this work is to thoroughly assess and compare the generalization capabilities of DG and UDA methods for semantic segmentation. Moreover, we are interested in the effect that the used data has on the final performance. To this end, we conduct the following experiments:

1. Quantitative comparison UDA vs. DG. To make a proper comparison, we propose a training setting where UDA and DG methods are trained on the same labeled dataset $\mathcal{D}_l$. Additionally, we pick multiple datasets to represent the unlabeled domains $\mathcal{D}_{nl}$. By definition, this unlabeled data is only leveraged by UDA methods. To assess the generalization capabilities, we select multiple, other datasets as unseen domains $\mathcal{D}_u$, on which the evaluation is performed. The better the performance on these unseen domains $\mathcal{D}_u$, the better the generalization capabilities of the network.

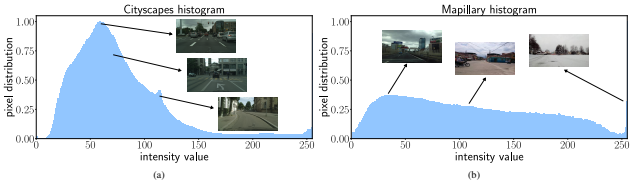

Figure 3. **Intensity histograms for Cityscapes and Mapillary.** Cityscapes shows a clear peak in the histogram, indicating that many images have a similar style. Mapillary has a much more uniform distribution, indicating there is no dominant style.

In practice, the availability of labeled data for the labeled domain $\mathcal{D}_l$ may vary. Because the level of variation in terms of image conditions within a dataset generally influences the generalization capabilities of a network trained on that dataset, we also expect the availability of heterogeneous labeled training data to have an impact on the performance of UDA and DG methods. Therefore, we conduct this experiment in two different configurations:

A) **Homogeneous dataset as labeled domain.** This experiment considers a situation where there is only a simple, homogeneous labeled dataset available, as is often the case in practical environments. This dataset consists of images captured at similar locations, with similar conditions and lighting properties.

B) **Heterogeneous dataset as labeled domain.** In this experiment, there is a labeled dataset that consists of images captured under many different conditions, from many distinctive subdomains.

2. Impact of using unlabeled data. To investigate the effect of leveraging unlabeled data to achieve generalizable semantic segmentation, we conduct an experiment where we vary the quantity of available unlabeled data for the best-performing UDA method. To set a baseline, we train a fully-supervised network with the same increasing portions of images, but then with labeled data. Specifically:

- Train the best performing UDA method from experiment 1.A on $\mathcal{D}_l$ with increasing unlabeled portions $\{0, 25, 50, 75, 100\}\%$ of $\mathcal{D}_{nl}$.
- Train a fully-supervised segmentation model on $\mathcal{D}_l$ together with increasing portions $\{0, 25, 50, 75, 100\}\%$ of the datasets for $\mathcal{D}_{nl}$ but now using their labels.

This experiment is designed to provide insights in the amount of unlabeled data that is needed to achieve generalization to unseen domains, and how this compares with using the same number of images for supervised training.

4.2. Evaluation protocol

In order to compare different generalization and adaptation methods quantitatively, we need to select datasets for the labeled and unlabeled training domains as well as the unseen domains, as defined in Sec 3. With these datasets, which are described below and summarized in Tab. 2, models are trained and evaluated using the performance metrics reported in Tab. 2.

4.2.1 Datasets

For our evaluation protocol, we need several datasets that all consist of similar categories. In the context of autonomous driving, multiple datasets are available:

Cityscapes [10] is an urban street scene dataset collected in several cities in an around Germany. **BDD-100K** [42] also contains images of urban scenes, but then captured at different locations in the United States. Likewise, **IDD** [37] gathered numerous urban scenes from several cities in India. On the other hand, **Mapillary Vistas** [25] is a very diverse dataset, and contains street scenes from all over the world, under different conditions. Similarly, **WildDash** [44] consists of a vast number of images, including high-hazard scenarios, from places all over the world. It is designed to benchmark the robustness of semantic segmentation models. Finally, **KITTI** [1] is a relatively small dataset with very similar images, captured around the same city in Germany.

4.2.2 Training domains

Given the six datasets described in Sec. 4.2.1, we need to select the datasets that will be used as labeled domains $\mathcal{D}_l$ and unlabeled domains $\mathcal{D}_{nl}$ during training.

Labeled domains. As explained in Sec. 4.1, we aim to conduct experiments in two high-level settings, where A) the labeled domain consists of a homogeneous dataset and B) the labeled domain is a heterogeneous dataset. Given the properties of the datasets, we select Cityscapes [10] to be the homogeneous dataset for $\mathcal{D}_l$ in experiment 1.A, because its images are captured under very similar conditions. For experiment 1.B, we pick Mapillary Vistas [25] as the heterogeneous dataset for $\mathcal{D}_l$ in experiment 1.B, because it contains images captured under many different conditions, in many different locations.

To support our selection, we conduct a histogram analysis on the pixel intensities of the images in each dataset. We expect that a homogeneous dataset, captured with the same camera setup and under similar lighting conditions, will show a clear peak at certain pixel intensity values. On the other hand, a heterogeneous dataset with images captured under many different conditions, should have a more uniform distribution. Therefore, this histogram should give a rough indication of the homogeneity of a dataset. We acknowledge that there exist more advanced techniques for this purpose, but we consider this to be out of scope, since we only use these histograms as an auxiliary tool.

From the histograms depicted in Fig. 3, we find that the histogram for the Cityscapes shows a clear peak around certain intensity values, whereas Mapillary Vistas images have a more uniform pixel intensity distribution. This supports our decision for picking Cityscapes as a homogeneous dataset and Mapillary Vistas as a heterogeneous dataset. We refer to the supplementary material for more information about these histograms, and the histograms for other datasets.

Unlabeled domains. In experiment 1.A, where Cityscapes is labeled domain $\mathcal{D}_l$ we pick the training splits of BDD-100K and Mapillary Vistas to be the unlabeled training domains $\mathcal{D}_{nl}$. We choose BDD-100K and Mapillary because together they consist of more varied images than Cityscapes, which can now be leveraged by UDA methods to boost generalization. In experiment 1.B, $\mathcal{D}_{nl}$ is composed of the training splits of BDD-100K and Cityscapes, two datasets which are both homogeneous compared to the labeled dataset $\mathcal{D}_l$, Mapillary Vistas.

4.2.3 Unseen domains

For all experiments, unseen domains $\mathcal{D}_u$ are composed of a combination of the validation splits of WildDash, IDD, and KITTI. Note that these are only used during testing and never during training. We choose these datasets because they together represent the realistic and challenging streams of images that a driving system encounters during deployment: they consist images from multiple environments captured under different conditions (WildDash), and images captured at very specific locations under very specific conditions (IDD and KITTI). In the end, a good generalized model should perform well on both types of datasets.

4.3. Method selection

In our evaluation, we aim to include the best-performing state-of-the-art methods for domain generalization and unsupervised domain adaptation. To select only the best-performing and properly reproducible methods, we set the following requirements: 1) the implementation of the method should be publicly available, providing training and evaluation scripts, 2) the method should not require training external networks not available in the published code, 3) the methods should obtain state-of-the-art results in the standard benchmarks for their research domains, and 4) reproducing the reported results should not lead to a performance drop of more than 5%. Methods fitting these criteria are adapted using their original code so that they all use a normalized architecture for the semantic segmentation

model's encoder and decoder (i.e., ResNet-101 [14] and DeepLabv2 [6] from [45]), to allow for a fair comparison. The final selected models are ProDA [45] and SAC [2] for UDA, and WildNet [18] and RobustNet [9] for DG. Please note that our comparison is not meant to disqualify methods in any way. Our aim is merely to find out what type of methods should be used in what situation. More information on the selection process and the adaptation of these methods is provided in the supplementary material.

4.4. Baselines

To further complement the quantitative comparison, and provide informative baselines, we also train the following fully-supervised segmentation models:

Single-dataset training. To provide additional insights in all the datasets used in our evaluation protocol, and to evaluate how well models generalize by training on just a single labeled dataset, we train fully-supervised models separately on each individual dataset using a standard cross-entropy loss, and evaluate them on all the datasets. By doing so, we can identify what type of dataset can be used to achieve good results on unseen data, without using any adaptation/generalization technique. We briefly discuss takeaways on these models in Sec. 5.1.

Labeled-domain-only training. This is a fully-supervised model trained only on the dataset that is the labeled domain $\mathcal{D}_l$. We consider DG and UDA methods effective only if they outperform this baseline.

Multi-dataset training. We train fully-supervised models on multiple datasets. Specifically:

1. We train a model on the datasets used as labeled domain $\mathcal{D}_l$ and unlabeled domains $\mathcal{D}_{nl}$, with all labels, to compare with UDA trained using the same datasets $\mathcal{D}_l$ and $\mathcal{D}_{nl}$. The goal of the DG and UDA methods is to reach comparable performance, but using significantly less labeled training data.
2. We train a model jointly on all the datasets used as $\mathcal{D}_u$, to see what the 'oracle' performance is, i.e., the performance when a fully-supervised model is trained on the datasets that are used for evaluation.
3. We train a model on all the datasets proposed for $\mathcal{D}_l$, $\mathcal{D}_{nl}$ and $\mathcal{D}_u$, to serve as an 'upper bound'.

4.5. Implementation details

As explained before, currently reported results in literature for DG and UDA methods are not comparable, because these methods use different semantic segmentation architectures and are trained and tested on different datasets (Fig. 1). To solve this, our experiments aim for a normalized comparison, where each method is trained and tested on the same datasets and all methods use the same semantic segmentation architecture. For all experiments, all models are trained and evaluated on the same 19 classes, i.e., the 19 classes typically evaluated on the Cityscapes dataset. For Mapillary Vistas, we map the class labels to the Cityscapes definition as in [15].

Hardware and network architecture. In our experiments, we have implemented all methods using PyTorch [27], training them on two NVIDIA A6000 GPUs with 48GB memory each. Due to the heterogeneity in architectures, we adapt the code of all methods, so that all networks use the same version of DeepLabv2 [6, 45] with a ResNet-101 backbone [14] pretrained on ImageNet [11]. More details are provided in the supplementary material.

Hyperparameters. We run all selected candidates using the set of hyperparameters that led to achieve their best mIoU performance. The fully-supervised models, which are DeepLabv2 models as described previously, are trained using an SGD [5] optimizer with momentum of 0.9, initial learning rate of 2.5×10^{-4}, polynomial schedule with decay of 0.9, and a standard cross-entropy loss [4]. In addition, all baseline models are trained on random crops of 896×512 (W×H) pixels, for $N_d \times 180k$ iterations, where N_d is the amount of datasets used for training, and using early-stopping, i.e., stopping training if there are no performance improvements during 20 consecutive epochs.

5. Results

In this section, we provide the results for the experiments as listed in Sec. 4.1. But first, to a) provide insights in the different datasets that are used in our evaluation framework, and b) show that DG or UDA methods are necessary for better generalization, we briefly discuss the results of the single-dataset training baselines described in Sec. 4.4.

5.1. Single-dataset training

In Tab. 4, we report the performance of the fully-supervised models trained individually on each of the datasets used in our evaluation framework. When looking at this table, it is immediately clear that Mapillary Vistas and WildDash lead to the best overall performance across all datasets, as reflected in the average mean IoU. This is expected, because we found that those datasets contain the most varied images, captured under many different conditions (see also Fig. 3), making it more likely that the images used for evaluation are similar to images in the training set.

In this table, the results for training and evaluating on the same dataset – but not the same split, as we train on the training set and evaluate on the validation set – are highlighted in blue across the diagonal. These numbers represent the mIoU that can be achieved if the network has access to labeled data from the domain that is used for evaluation. It can be seen that, in the vast majority of cases, the best results for each dataset are the results highlighted in blue.

task	method	training domains $\mathcal{D}_l$	training domains $\mathcal{D}_{nl}$	seen domains CS	seen domains BDD	seen domains MAP	seen mIoU avg.	unseen domains WILD	unseen domains IDD	unseen domains KITTI	unseen mIoU avg.
	Labeled-domain-only (baseline)	CS	n.a.	69.8	43.2	39.8	50.9	32.9	43.5	49.6	42.0
UDA	ProDA [45]	CS	BDD, MAP	74.4$^{\uparrow+4.6}$	53.8$^{\uparrow+10.6}$	51.9$^{\uparrow+12.1}$	60.0$^{\uparrow+9.1}$	50.9$^{\uparrow+18.1}$	55.7$^{\uparrow+12.2}$	61.6$^{\uparrow+12.0}$	56.1$^{\uparrow+14.1}$
UDA	SAC [2]	CS	BDD, MAP	68.4$^{\downarrow-1.4}$	51.4$^{\uparrow+8.2}$	45.3$^{\uparrow+5.5}$	55.0$^{\uparrow+4.1}$	44.2$^{\uparrow+11.3}$	51.0$^{\uparrow+7.5}$	52.9$^{\uparrow+3.3}$	49.4$^{\uparrow+7.3}$
DG	WildNet [18] w/ class balancing	CS	n.a.	70.6$^{\uparrow+0.8}$	46.1$^{\uparrow+2.9}$	56.3$^{\uparrow+16.6}$	57.7$^{\uparrow+6.8}$	44.1$^{\uparrow+11.2}$	50.4$^{\uparrow+6.9}$	47.3$^{\downarrow-2.3}$	47.3$^{\uparrow+5.3}$
DG	WildNet [18] w/o class balancing	CS	n.a.	69.3$^{\downarrow-0.4}$	45.4$^{\uparrow+2.1}$	53.9$^{\uparrow+14.2}$	56.2$^{\uparrow+5.3}$	42.3$^{\uparrow+9.4}$	49.4$^{\uparrow+5.8}$	51.0$^{\uparrow+1.5}$	47.6$^{\uparrow+5.6}$
DG	RobustNet [9] w/ class balancing	CS	n.a.	**74.6**$^{\uparrow+4.8}$	47.9$^{\uparrow+4.7}$	55.0$^{\uparrow+15.3}$	59.1$^{\uparrow+8.2}$	38.1$^{\uparrow+5.2}$	50.8$^{\uparrow+7.3}$	52.8$^{\uparrow+3.3}$	47.2$^{\uparrow+5.2}$
DG	RobustNet [9] w/o class balancing	CS	n.a.	70.4$^{\uparrow+0.6}$	44.4$^{\uparrow+1.2}$	51.4$^{\uparrow+11.7}$	55.4$^{\uparrow+4.5}$	37.1$^{\uparrow+4.2}$	48.7$^{\uparrow+5.2}$	49.6$^{\uparrow+0.0}$	45.1$^{\uparrow+3.1}$
	Fully sup. training domains	CS, BDD, MAP	n.a.	73.1$^{\uparrow+3.3}$	**60.3**$^{\uparrow+17.1}$	57.8$^{\uparrow+18.0}$	**63.7**$^{\uparrow+12.8}$	53.2$^{\uparrow+20.4}$	56.7$^{\uparrow+13.2}$	56.5$^{\uparrow+6.9}$	55.5$^{\uparrow+13.5}$
	Fully sup. unseen domains	WILD, IDD, KITTI	n.a.	45.2$^{\downarrow-24.6}$	43.7$^{\uparrow+0.5}$	43.8$^{\uparrow+4.0}$	44.2$^{\downarrow-6.7}$	47.7$^{\uparrow+14.8}$	57.1$^{\uparrow+13.6}$	50.1$^{\uparrow+0.6}$	51.6$^{\uparrow+9.6}$
	Fully sup. all domains	CS, BDD, MAP, WILD, IDD, KITTI	n.a.	70.8$^{\uparrow+1.1}$	58.3$^{\uparrow+15.1}$	**58.8**$^{\uparrow+19.0}$	62.6$^{\uparrow+11.7}$	**59.1**$^{\uparrow+26.2}$	**66.5**$^{\uparrow+23.0}$	**63.3**$^{\uparrow+13.8}$	**63.0**$^{\uparrow+21.0}$

Table 3. **Quantitative comparison when labeled domain $\mathcal{D}_l$ is *homogeneous***. The reported deltas (in green and red) are with respect to the baseline (first row). The highest mIoU values are highlighted as follows: bold, considering all methods; and underlined, considering only UDA and DG methods.

training \ validation	CS	BDD	MAP	WILD	IDD	KITTI	mIoU avg.	non-diagonal mIoU avg.
CS	69.8	43.2	39.8	32.9	43.5	49.6	46.5	41.8
BDD	51.1	54.4	41.0	43.3	51.8	47.2	48.1	46.9
MAP	60.1	54.6	55.0	48.2	55.0	52.8	54.3	54.1
WILD	56.4	51.4	49.3	56.5	56.0	46.9	52.8	52.0
IDD	42.8	42.1	38.7	37.2	68.9	41.8	45.2	40.5
KITTI	29.9	27.2	24.1	23.1	27.2	48.9	30.1	26.3

Table 4. **Results for fully-supervised networks trained individually on a single dataset.**

This means that, when the evaluation images are dissimilar from the training images, the performance drops. This is exactly the lack of generalization that we described in the introduction, and that the DG and UDA methods aim to solve. Interestingly, we can also see some cases where the blue highlighted number is not the best result. Specifically, 1) training on Mapillary Vistas yields slightly better results on BDD-100K and KITTI than training on those datasets, and 2) training on Cityscapes also leads to better performance on KITTI. We hypothesize that for case 1, the Mapillary Vistas training simply yields very good results because the dataset is large and varied. For case 2, we expect that Cityscapes images look very similar to KITTI, meaning that the training and evaluation conditions are similar. Moreover, Cityscapes consists of much more training images than KITTI, further boosting the performance (see Tab. 2). Overall, though, there is a clear lack of generalization that needs to be addressed, to allow for a successful application in *the wild*, e.g., with UDA or DG methods.

5.2. Quantitative comparison UDA vs. DG

Overall findings. We apply the selected state-of-the-art UDA and DG methods to our evaluation framework, as described in Sec. 4, and report the results in Tab. 3 and Tab. 5. Note that Tab. 3 reports the results for the setting with a *homogeneous* labeled domain $\mathcal{D}_l$, and Tab. 5 has a *heterogeneous* labeled domain. Because we are interested in the generalization capacity of models to unseen domains, the unseen mIoU avg. is the most relevant and important metric. At first glance, both Tab. 3 and Tab. 5 show that ProDA [45], a UDA method, significantly outperforms all other methods on the unseen mIoU avg., improving the baseline with an average mIoU of +14.1 and +4.7 points, respectively. Surprisingly, this performance is also on par with – or even better than – the fully-supervised baseline trained on all training domains, achieving a generalization equivalent to training a segmentation network with all the training datasets fully labeled.

Furthermore, it is also remarkable that both UDA methods, ProDA and SAC, generalize to unseen domains more effectively than domain generalization methods. This is an interesting finding, because UDA methods are not designed to perform well on unseen domains $\mathcal{D}_u$, but rather to perform well on seen, unlabeled domains $\mathcal{D}_{nl}$. We expect that the good performance by UDA methods is caused by the fact that they have access to unlabeled data, which allows them to use techniques like feature alignment and self-training, as mentioned in Sec. 2. In Sec. 5.3, we further analyze the impact of leveraging portions of unlabeled data.

In Fig. 1, we show some qualitative results for the best performing methods of the setting with the heterogeneous training domain, to demonstrate what an increase in mIoU means in terms of actual segmentation quality. In this figure, we observe that ProDA works consistently well, regardless of the unseen domain, as also supported by quantitative results.

Homogeneous vs. heterogeneous labeled domain. Although most results are the same for the settings with a) a homogeneous labeled domain and b) a heterogeneous labeled domain, there are also notable differences. Specifically, we note that DG methods suffer a significant drop in generalization performance when the labeled domain is heterogeneous (Tab. 5), compared to the baseline trained on the labeled domain only, but perform quite well when the labeled domain is homogeneous (Tab. 3). We expect that this is caused by the techniques employed by the DG methods. Specifically, they try focus on the style component of a specific dataset, and try to enhance or suppress it. However, when applied to a dataset with a high statistical variability (see Fig. 3b), the style component becomes hard to estimate, harming the learning process of these methods. This is not the case for ProDA and SAC, as their underly-

task	method	training domains $\mathcal{D}_l$	training domains $\mathcal{D}_{nl}$	seen domains CS	seen domains BDD	seen domains MAP	seen mIoU avg.	unseen domains WILD	unseen domains IDD	unseen domains KITTI	unseen mIoU avg.
	Labeled-domain-only (baseline)	MAP	n.a.	60.1	54.6	55.0	56.6	48.2	55.0	52.8	52.0
UDA	ProDA [45]	MAP	BDD, CS	67.7$^{\uparrow+7.6}$	58.0$^{\uparrow+3.4}$	55.8$^{\uparrow+0.7}$	60.5$^{\uparrow+3.9}$	54.8$^{\uparrow+6.6}$	57.5$^{\uparrow+2.5}$	58.0$^{\uparrow+5.2}$	56.8$^{\uparrow+4.7}$
UDA	SAC [2]	MAP	BDD, CS	61.4$^{\uparrow+1.4}$	57.3$^{\uparrow+2.7}$	57.9$^{\uparrow+2.9}$	58.9$^{\uparrow+2.3}$	48.4$^{\uparrow+0.1}$	54.2$^{\downarrow-0.8}$	55.0$^{\uparrow+2.2}$	52.5$^{\uparrow+0.5}$
DG	WildNet [18] w/ class balancing	MAP	n.a.	56.3$^{\downarrow-3.8}$	47.1$^{\downarrow-7.5}$	60.5$^{\uparrow+5.4}$	54.6$^{\downarrow-1.9}$	48.0$^{\downarrow-0.2}$	50.5$^{\downarrow-4.5}$	41.9$^{\downarrow-10.9}$	46.8$^{\downarrow-5.2}$
DG	WildNet [18] w/o class balancing	MAP	n.a.	53.2$^{\downarrow-6.9}$	45.8$^{\downarrow-8.8}$	56.0$^{\uparrow+1.0}$	51.6$^{\downarrow-4.9}$	45.5$^{\downarrow-2.7}$	50.8$^{\downarrow-4.2}$	42.8$^{\downarrow-10.0}$	46.4$^{\downarrow-5.6}$
DG	RobustNet [9] w/ class balancing	MAP	n.a.	58.1$^{\downarrow-1.9}$	50.5$^{\downarrow-4.1}$	**61.5**$^{\uparrow+6.5}$	56.7$^{\uparrow+0.2}$	47.6$^{\downarrow-0.7}$	52.1$^{\downarrow-3.0}$	45.9$^{\downarrow-7.0}$	48.5$^{\downarrow-3.5}$
DG	RobustNet [9] w/o class balancing	MAP	n.a.	53.5$^{\downarrow-6.5}$	46.8$^{\downarrow-7.8}$	56.0$^{\uparrow+0.9}$	52.1$^{\downarrow-4.5}$	42.6$^{\downarrow-5.6}$	49.8$^{\downarrow-5.2}$	44.4$^{\downarrow-8.4}$	45.6$^{\downarrow-6.4}$
	Fully sup. training domains	CS, BDD, MAP	n.a.	**73.1**$^{\uparrow+13.0}$	**60.3**$^{\uparrow+5.7}$	57.8$^{\uparrow+2.7}$	**63.7**$^{\uparrow+7.1}$	53.2$^{\uparrow+5.0}$	56.7$^{\uparrow+1.7}$	56.5$^{\uparrow+3.7}$	55.5$^{\uparrow+3.5}$
	Fully sup. unseen domains	WILD, IDD, KITTI	n.a.	45.2$^{\downarrow-14.9}$	43.7$^{\downarrow-10.9}$	43.8$^{\downarrow-11.2}$	44.2$^{\downarrow-12.3}$	47.7$^{\downarrow-0.6}$	57.1$^{\uparrow+2.1}$	50.1$^{\downarrow-2.7}$	51.6$^{\downarrow-0.4}$
	Fully sup. all domains	CS, BDD, MAP, WILD, IDD, KITTI	n.a.	70.8$^{\uparrow+10.8}$	58.3$^{\uparrow+3.7}$	58.8$^{\uparrow+3.7}$	62.6$^{\uparrow+6.0}$	**59.1**$^{\uparrow+10.9}$	**66.5**$^{\uparrow+11.5}$	**63.3**$^{\uparrow+10.5}$	**63.0**$^{\uparrow+11.0}$

Table 5. **Quantitative comparison when labeled domain $\mathcal{D}_l$ is *heterogeneous*.** The reported deltas (in green and red) are with respect to the baseline (first row). The highest mIoU values are highlighted as follows: bold, considering all methods; and underlined, considering only UDA and DG methods.

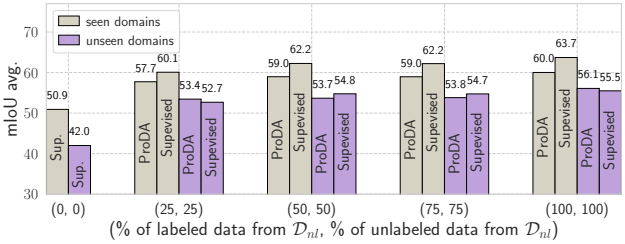

Figure 4. **Impact of amount of labeled and unlabeled data**. Models are trained on the labeled domain $\mathcal{D}_l$ along with portions of the unlabeled domain $\mathcal{D}_{nl}$, either labeled or unlabeled.

ing learning mechanisms do not involve the estimation of any style-related components. Therefore, when the labeled domain $\mathcal{D}_l$ is heterogeneous, we recommend using UDA or standard supervised learning over DG methods. Moreover, we find that the best average results by ProDA in both Tab. 5 and Tab. 3 are quite similar, even though the labeled domain $\mathcal{D}_l$ is very different. This implies that it does not significantly matter what training dataset is labeled; it seems to be more important that the network has access to heterogeneous data, whether it is labeled or not.

5.3. Impact of using unlabeled data

In the previous experiments, we found that having access to unlabeled data allows UDA methods to apply techniques that boost generalization to unseen domains, and that they can even perform on par with fully-supervised methods trained on the same data. This shows that there is a great benefit in just collecting and using data, without having to annotate it. To further investigate the benefits, we study the effect of the quantity of labeled and unlabeled data that is used by the network. The results of this analysis can be seen in Fig. 4. First of all, although this is not the focus of this work, when we consider the mIoU avg. on the *seen* domains (beige bars), the figure shows that ProDA with 100% of unlabeled data achieves roughly the same as the fully-supervised model using 25% of labeled data. This indicates that there is still room for improvement for methods using unlabeled data, as there is a considerable gap. However, in terms of generalization to unseen domains, there is no real gap between the supervised method and ProDA. It can be observed that most of the generalization capacity to unseen domains (purple bars) of the model is already achieved with 25% of the total amount of unlabeled data. With 25% percentage of unlabeled data, ProDA already performs very similarly to the model trained in fully-supervised fashion on the same images, i.e., using the labels. We even observe that using 100% of the unlabeled data is better than using 100% of labeled data. As recording unlabeled data is inexpensive compared to labeling, this shows that much labeling effort can be avoided by simply training with a UDA strategy, when the purpose is generalization.

6. Conclusions

In this work, we evaluated several state-of-the-art semantic segmentation training strategies in terms of their ability to generalize to data unseen during training. Whereas a fair comparison was not possible based on literature alone, we proposed a fair evaluation setting where normalized implementations of existing domain generalization (DG) and unsupervised domain adaptation (UDA) methods could be assessed. From the experiments conducted with this evaluation protocol, we found that UDA methods yield the best generalization performance, and we showed that the ability to use unlabeled data plays a key role in achieving this. Moreover, we showed that unlabeled data can be just as powerful as labeled data when the purpose is to generalize to unseen data. From this, we can conclude that it is highly advisable to train semantic segmentation models that need to work reliably and robustly in *the wild* with both labeled and much unlabeled data using a UDA strategy, especially when considering that unlabeled data is significantly easier and cheaper to collect than labeled data.

Acknowledgements This work is supported and funded by the Netherlands Organization for Scientific Research (NWO) in the context of the Efficient Deep Learning (EDL) programme.

References

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

[2] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *CVPR*, pages 15384–15394, 2021.

[3] Matteo Biasetton, Umberto Michieli, Gianluca Agresti, and Pietro Zanuttigh. Unsupervised domain adaptation for semantic segmentation of urban scenes. In *CVPR*, pages 1211–1220, 2019.

[4] David E. Booth. The cross-entropy method. *Technometrics*, 50(1):92, 2008.

[5] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, pages 177–186. Physica-Verlag, 2010.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2018.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, 2018.

[8] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *ICCV*, pages 6829–6839, 2019.

[9] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne Taery Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *CVPR*, pages 11580–11590, 2021.

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, pages 248–255, 2009.

[12] Li Gao, Jing Zhang, Lefei Zhang, and Dacheng Tao. DSP: dual soft-paste for unsupervised domain adaptive semantic segmentation. In *ACM Multimedia*, pages 2825–2833, 2021.

[13] Lin Lawrence Guo, Stephen R Pfohl, Jason Fries, Alistair Johnson, Jose Posada, Catherine Aftandilian, Nigam Shah, and Lillian Sung. Evaluation of domain generalization and adaptation on improving model robustness to temporal dataset shift in clinical medicine. *Scientific Reports*, 2021.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Yang He, Shadi Rahimian, Bernt Schiele, and Mario Fritz. Segmentations-leak: Membership inference attacks and defenses in semantic image segmentation. In *ECCV*, pages 519–535, 2020.

[16] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[17] Takashi Isobe, Xu Jia, Shuaijun Chen, Jianzhong He, Yongjie Shi, Jianzhuang Liu, Huchuan Lu, and Shengjin Wang. Multi-target domain adaptation with collaborative consistency learning. In *CVPR*, pages 8187–8196, 2021.

[18] Suhyeon Lee, Hongje Seong, Seongwon Lee, and Euntai Kim. Wildnet: Learning domain generalized semantic segmentation from the wild. In *CVPR*, pages 9926–9936, 2022.

[19] Y. Li, L. Yuan, and N. Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. *CVPR*, pages 6929–6938, 2019.

[20] Haoyu Ma, Xiangru Lin, Zifeng Wu, and Yizhou Yu. Coarse-to-fine domain adaptive semantic segmentation with photometric alignment and category-center regularization. In *CVPR*, 2021.

[21] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: practical guidelines for efficient CNN architecture design. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, 2018.

[22] Luke Melas-Kyriazi and Arjun K. Manrai. Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In *CVPR*, pages 12435–12445, 2021.

[23] Panagiotis Meletis and Gijs Dubbelman. Training of convolutional networks on multiple heterogeneous datasets for street scene semantic segmentation. In *IV*, pages 1045–1050, 2018.

[24] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim. Image to image translation for domain adaptation. In *CVPR*, pages 4500–4509, 2018.

[25] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, pages 5000–5009, 2017.

[26] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, pages 484–500, 2018.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, pages 8024–8035, 2019.

[28] Fabrizio J. Piva and Gijs Dubbelman. Exploiting image translations via ensemble self-supervised learning for unsupervised domain adaptation. *CoRR*, abs/2107.06235, 2021.

[29] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, 2016.

[30] Rob Romijnders, Panagiotis Meletis, and Gijs Dubbelman. A domain agnostic normalization layer for unsupervised adversarial domain adaptation. In *WACV*, pages 1866–1875, 2019.

[31] Germán Ros, Laura Sellart, Joanna Materzynska, David Vázquez, and Antonio M. López. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[32] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[33] Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Multi-target adversarial frameworks for domain adaptation in semantic segmentation. *CoRR*, abs/2108.06962, 2021.

[34] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *ArXiv*, abs/2108.13624, 2021.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[36] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, pages 7472–7481, 2018.

[37] Girish Varma, Anbumani Subramanian, Anoop M. Namboodiri, Manmohan Chandraker, and C. V. Jawahar. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*, pages 1743–1751, 2019.

[38] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. In *IJCAI*, 2021.

[39] Jinyu Yang, Weizhi An, Sheng Wang, Xinliang Zhu, Chaochao Yan, and Junzhou Huang. Label-driven reconstruction for domain adaptation in semantic segmentation. In *ECCV*, volume 12372, pages 480–498, 2020.

[40] Jinyu Yang, Chunyuan Li, Weizhi An, Hehuan Ma, Yuzhi Guo, Yu Rong, Peilin Zhao, and Junzhou Huang. Exploring robustness of unsupervised domain adaptation in semantic segmentation. In *ICCV*. IEEE, 2021.

[41] Yanchao Yang and Stefano Soatto. FDA: fourier domain adaptation for semantic segmentation. In *CVPR*, pages 4084–4094, 2020.

[42] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, pages 2633–2642, 2020.

[43] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto L. Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *ICCV*, pages 2100–2110. IEEE, 2019.

[44] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernández Domínguez. Wilddash - creating hazard-aware benchmarks. In *ECCV*, pages 407–421, 2018.

[45] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *CVPR*, pages 12414–12424, 2021.

[46] Xiong Zhang, Hongmin Xu, Hong Mo, Jianchao Tan, Cheng Yang, Lei Wang, and Wenqi Ren. DCNAS: densely connected neural architecture search for semantic image segmentation. In *CVPR*, 2021.

[47] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *CoRR*, abs/2103.02503, 2021.