IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 April 2023, accepted 9 May 2023, date of publication 18 May 2023, date of current version 7 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3277785

SURVEY

Survey on Unsupervised Domain Adaptation for Semantic Segmentation for Visual Perception in Automated Driving

MANUEL SCHWONBERG[1,*], JOSHUA NIEMEIJER[2,*], JAN-AIKE TERMÖHLEN[3,*], (Graduate Student Member, IEEE), JÖRG P. SCHÄFER[2], NICO M. SCHMIDT[1], HANNO GOTTSCHALK[4], AND TIM FINGSCHEIDT[3], (Senior Member, IEEE)

[1]CARIAD SE, 38440 Wolfsburg, Germany
[2]Deutsches Zentrum für Luft- und Raumfahrt (DLR) e.V., 38108 Braunschweig, Germany
[3]Institute for Communications Technology, Technische Universität Braunschweig, 38106 Braunschweig, Germany
[4]Institute of Mathematics, Technical University Berlin, 10623 Berlin, Germany

Corresponding authors: Joshua Niemeijer (joshua.niemeijer@dlr.de), Manuel Schwonberg (manuel.schwonberg@cariad.technology), and Jan-Aike Termöhlen (j.termoehlen@tu-bs.de)

This work was supported by the German Federal Ministry for Economic Affairs and Energy within the Project "KI Delta Learning" (Förderkennzeichen) under Grant 19A19013C and Grant 19A19013K. The authors would like to thank the consortium for the successful cooperation.

*Manuel Schwonberg, Joshua Niemeijer, and Jan-Aike Termöhlen contributed equally to this work.

ABSTRACT Deep neural networks (DNNs) have proven their capabilities in the past years and play a significant role in environment perception for the challenging application of automated driving. They are employed for tasks such as detection, semantic segmentation, and sensor fusion. Despite tremendous research efforts, several issues still need to be addressed that limit the applicability of DNNs in automated driving. The bad generalization of DNNs to unseen domains is a major problem on the way to a safe, large-scale application, because manual annotation of new domains is costly, particularly for semantic segmentation. For this reason, methods are required to adapt DNNs to new domains without labeling effort. This task is termed unsupervised domain adaptation (UDA). While several different domain shifts challenge DNNs, the shift between synthetic and real data is of particular importance for automated driving, as it allows the use of simulation environments for DNN training. We present an overview of the current state of the art in this research field. We categorize and explain the different approaches for UDA. The number of considered publications is larger than any other survey on this topic. We also go far beyond the description of the UDA state-of-the-art, as we present a quantitative comparison of approaches and point out the latest trends in this field. We conduct a critical analysis of the state-of-the-art and highlight promising future research directions. With this survey, we aim to facilitate UDA research further and encourage scientists to exploit novel research directions.

INDEX TERMS Computer vision, deep neural networks, unsupervised domain adaptation, semantic segmentation, automated driving.

I. INTRODUCTION

Perception of the environment using a variety of sensors is an essential component of modern autonomous systems, e.g., automated vehicles [1], [2]. Visual perception using camera sensors is of particular interest, as this type of sensor is inexpensive and provides diverse information not only on the geometry of the environment but also on the surfaces, e.g., colors, textures, and text on traffic signs. The perception that converts sensory to semantic information utilizes machine learning methods such as deep neural networks (DNNs) that are trained on a large amount of human-labeled training data. This principle led to remarkable results [3] and a newly emerging field of machine learning research. Nowadays,

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

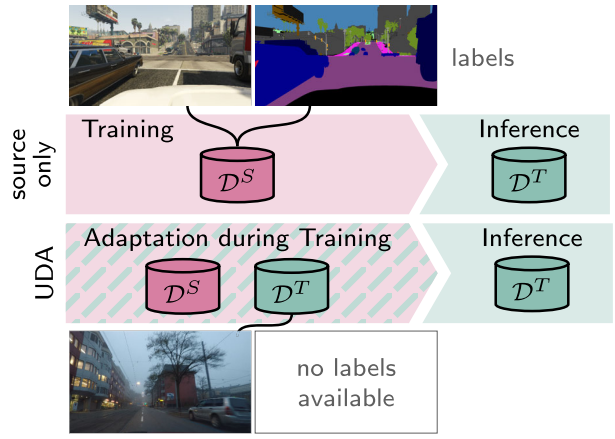


FIGURE 1. Unsupervised domain adaptation scheme illustrating the difference between source-only training and unsupervised domain adaptation (UDA). Red indicates (the use of) labeled data from the source domain and green indicates that unlabeled data from the target domain is utilized.

DNNs are employed for a large variety of tasks and areas, but we will focus on DNNs for semantic segmentation in this survey.

Despite the capabilities of DNNs to learn complex relations, there are still several challenges to solve. First, adversarial attacks [4] that are designed to cause wrong predictions are a severe threat to DNNs in safety-critical areas such as automated driving. Second, the large amounts of human-labeled data that are required for training are costly, and the training of large models on thousands or millions of images also. It is worth mentioning that self-supervised learning substantially reduced the need for labels [5]. One of the most severe issues is the need to generalize DNNs to samples outside their training distribution. Domain shifts, i.e., when training and inference distribution differ, occur often and can have multiple causes, such as changes in illumination, location, weather conditions, or sensor noise. Significant performance degradation of the model is usually the consequence. A domain shift also occurs when training on synthetic data and employing the model on real data. Utilizing synthetic data for training is of particular interest, because it offers labels without human annotation effort and allows critical situations to be simulated that would be too dangerous or too rare in real data recordings, e.g., accidents and children running onto the street. For semantic segmentation, manual labeling effort by a human annotator is very high and can take up to 90 minutes per image [6]. However, since synthetic data can substantially differ from real data, significant performance drops are caused when a model trained on synthetic data receives real data.

To counteract the problems introduced by domain shifts, unsupervised domain adaptation (UDA) methods have emerged that require only unlabeled samples of the target domain to adapt the network to it. Figure 1 illustrates the basic principle. Instead of training only on labeled images from the source domain $\mathcal{D}^S$, for UDA there are also images from the target domain $\mathcal{D}^T$ available for adaptation, but without any labels. In both cases, the inference is supposed to be performed on data from $\mathcal{D}^T$.

One of the most important and challenging applications is automated driving, where the autonomous system has to handle a broad range of driving scenarios. Domain shifts that cause a perception performance drop can seriously threaten human lives. For this reason, UDA methods are essential for automated driving. Consequently, this survey focuses on UDA methods for environment perception in autonomous driving, particularly on semantic segmentation of camera images. Semantic segmentation not only provides important information about objects but also about the environment surrounding the vehicle (the background classes), which can serve as the basis of a local grid map [7] or for map verification [8], [9], [10]. This makes segmentation a crucial part of the perception system. It is also one of the most commonly used applications in scientific publications on unsupervised domain adaptation methods for visual perceptions. UDA for object detection is also an active research area [11] but is out of the scope of this survey.

The UDA works included in this survey focus on the synthetic-to-real domain shift. As described, this shift is of special importance for automated driving, and the prioritization is valuable to assess how useful synthetic data can be for real applications. Because the synthetic-to-real domain shift benchmarking dominates UDA research, it is a reasonable choice for the focus of the survey to provide an extensive, valuable large-scale comparison of different approaches. However, since many more domain shifts are relevant for automated driving, we critically discuss this question in Section VI-B.

The evolvement of UDA publications for semantic segmentation for synthetic-to-real adaptation supports the importance of UDA for automated driving in research. This work focuses on UDA methods proposed for semantic segmentation since 2017. The research area of domain adaptation for machine learning already emerged early before the development of deep learning. However, only in 2017, the very first work [12] for UDA for semantic segmentation utilizing modern DNNs was proposed. For this reason, this year is taken as the initial year of research for UDA for semantic segmentation.

Several survey papers have attempted to provide a structured overview of this research field in recent years. Here we have to distinguish between surveys on domain adaptation in general [13], [14], [15] and surveys specifically for *unsupervised* domain adaptation for semantic segmentation [16], [17]. The general domain adaptation surveys do not focus on domain adaptation for semantic segmentation and therefore provide not the same depth and quantity as our survey. Toldo et al. [16] and Csurka et al. [17] published specialized surveys about domain adaptation for semantic segmentation and are closest to our survey. Both cover a significantly smaller amount of papers (factor of three). Toldo et al. [16] include a performance comparison clustered according to the

backbones, but with its publication in 2020, it is too old to cover recent trends. The clustering of UDA approaches is similar for both Toldo et al. [16] and Csurka et al. [17]. Our clustering also has commonalities with theirs. However, we extend the known taxonomy by the area of hybrid domain adaptation and provide more fine-grained sub-groupings for each area.

In summary, with this survey, our key contributions are:

- We propose a simple taxonomy that helps to group the different works. We also cover vision transformer networks for UDA as the very first survey.
- We provide the most complete literature study and comparison up to date by surveying three times more papers than previous studies.
- We provide a quantitative comparison of the different methods, showing methodical and performance trends of the last years.
- We take a critical look at the current approaches to overcome domain shifts and highlight common problems in the training process and evaluation of the adaptation approaches.
- We point out promising future research questions in this research area.

In addition, we have created an interactive project website that allows a more detailed comparison of all methods than in written form alone. The raw data of our quantitative comparison is also provided for further utilization by other researchers. The website can be accessed at: https://uda-survey.github.io/survey/.[1]

Our survey is expected to be valuable for three groups of readers: beginners, experts, and lecturers. This survey provides a structured introduction for readers without any prior knowledge of the topic (unsupervised) domain adaptation. We also discuss the most important benchmarking methods, and together with the quantitative comparison, our survey provides entrance for future UDA researchers. We also hope that expert-level readers find this survey helpful because of the more complete field coverage compared with prior works. In the dynamic research environment of UDA, expert-level readers will potentially find our overview of the most recent developments useful. For lecturers, the taxonomy and comparison of the individual methods provide interesting information. In addition, we compare the task of unsupervised domain adaptation with other related methods.

The survey is structured as follows. In Section II, we provide context for related research topics and introduce mathematical definitions and principles of domain adaptation. In Section III, we present our taxonomy and explain the common methods for each part of the taxonomy. Section V describes the employed metrics and provides a quantitative comparison of the approaches. Finally, we discuss the current research, presenting proposals for refinements and best practices as well as promising research directions in Section VI, before finishing with our conclusions in Section VII. Overall, we categorize, analyze and compare more than 140 approaches methodologically and quantitatively.

[1]We update the website with the latest approaches on a bi-monthly basis so that it can serve as a data hub for UDA researchers and keeping track of the current state-of-the-art.

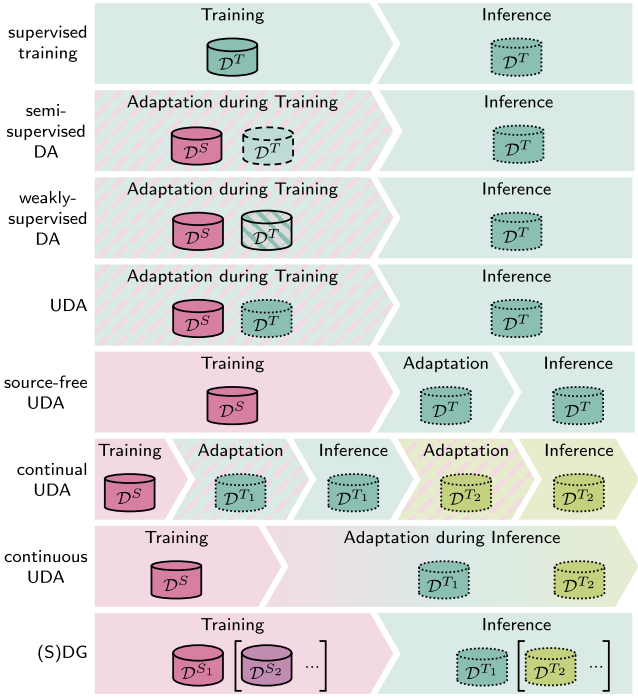


FIGURE 2. Overview of adaptation paradigms. Simplified schematic visualization. Red and green colors represent the source and target domains, respectively. Dotted lines indicate that no labels are available and dashed lines indicates a subset of labels. For weakly-supervised DA noisy labels are available.

II. RESEARCH CONTEXT AND DEFINITIONS

As mentioned before, UDA assumes that no labels are available for the target domain (cf. Fig. 1). Some tasks are closely related to domain adaptation. Figure 2 shows an overview of the most important adaptation paradigms. In the following section, we first give a brief overview of these related methods. Next, we introduce a mathematical notation to facilitate understanding of the various UDA methods. Then, we define domain shifts and describe the most commonly used benchmarks.

A. RELATED METHODS

In the following, we briefly explain other approaches for domain adaptation, which either use some form of labels or consider other constraints that apply to the availability of source domain data. Figure 2 provides an overview that categorizes the existing methods for domain adaptation found in the literature. We also differentiate the topic from domain generalization and explain the difference between closed- and open-set adaptation.

1) SUPERVISED TRAINING

As visible in Figure 2 (first row), the setting with full labels available in the target domain is called supervised training. In this case, no adaptation methods are required. However, this training paradigm is infeasible since manual annotation is costly and the number of domains is high. This way of adaptation is not scalable and will not be considered further.

2) SEMI-SUPERVISED DA

In semi-supervised domain adaptation (cf. Figure 2, second row), only a subset of the target domain data has labels. In contrast to self-training, these labels are not generated by the segmentation network itself but originate from human annotations. Only a few works study methods for semi-supervised domain adaptation [18], [19], [20], [21], but this research area is small compared to UDA and not the focus of this survey.

3) WEAKLY-SUPERVISED DA

For weakly-supervised domain adaptation (cf. Figure 2, third row), samples with noisy labels are available where, e.g., only bounding box labels are given as a weak label for the task of semantic or instance segmentation [22]. Another option is image-level labels, e.g., to predict the presence of classes, as done in WDA [23].

4) SOURCE-FREE UDA

This is the task of UDA when there is a given model but no access to the data of the source domain (cf. Figure 2, fifth row). For standard UDA, the adaptation process primarily utilizes source and target domains in parallel. In source-free UDA, in contrast, the adaptation process to the new domains must occur without forgetting essential information from the source domain. In this case, the only information that may be used from the source domain is the implicit information in the network weights from the pre-training on the source domain, which includes normalization parameters [24], [25].

5) CONTINUAL/CONTINUOUS UDA

The main idea behind continual and continuous domain adaptation is that a segmentation network is first trained on the labeled source domain and only afterward adapted to a target domain. During the adaption process, the source domain is unavailable. It is, therefore, a decoupling of training and adaptation, which happen simultaneously in standard UDA. Here, we want to propose a distinction between continual and continuous UDA (cf. Figure 2, sixth and seventh row). While continual means that something happens at (regular) intervals, continuous means that the adaptation happens without interruption, e.g., on a single-frame basis [25], [26]. Usually, continual and continuous UDA does not prohibit the usage of some form of source domain representation, e.g., a generator network that creates samples [27]. However, continual/continuous UDA methods can also be source-free when no source information is available during the adaptation process [25].

6) DOMAIN GENERALIZATION

This is the task of training networks to perform better in unseen domains without using any data from the target domain for adaptation (cf. Figure 2, last row). The model should generalize well, and the performance is usually evaluated on multiple unseen domains. The task is called single-domain generalization (SDG) if only one source domain is used. It is simply called domain generalization (DG) if multiple source domains or additional real auxiliary domains are employed. Compared to UDA, only a few works exploit the potential of domain generalization for semantic segmentation [28], [29], [30]. If it were possible to train networks that generalize perfectly, then adaptation to a target domain would no longer be necessary. On the other hand, it has been shown that even UDA approaches, e.g., DLOW [31], can also ensure that the trained network generalizes better. However, since this is not the focus of the UDA task, these results are often not reported in the papers.

7) CLOSED- AND OPEN-SET ADAPTATION

Concerning the labeled classes in the source and target domains, a distinction is made between closed-set and open-set adaptation. Closed-set adaptation refers to the more commonly employed method of domain adaptation, where the set of labeled classes is the same in the source and target domain. Thus it is assumed that only the visual domain changes, but the number of pre-defined semantic classes remains unchanged. In contrast, open-set adaptation assumes that the sets of labels do not have to be identical and that, e.g., new classes that have to be learned can occur in the new domain. Even though this field has not yet been subject to intensive research, there are first approaches that enable class- [32] and domain-incremental learning [33]. Although the open-set scenario is realistic, it is seldom considered in the academic context; see Tada et al. [34], however, for first steps. This survey only covers UDA methods that perform closed-set adaptation.

B. MATHEMATICAL NOTATION

So far, papers on domain adaptation do not use a harmonized mathematical notation. We first present a unified mathematical notation to improve understanding and allow a more straightforward comparison of different methods.

As input to the segmentation network, we define the image $\mathbf{x} \in \mathbb{G}^{H \times W \times C}$, where $\mathbb{G}$ denotes the set of integer color intensity values, H and W the image height and width in pixels, and $C = 3$ the number of color channels, respectively. A semantic segmentation network transforms an image into an output $\mathbf{y} = (y_{i,s}) \in \mathcal{B}^{H \times W \times S}$ with posterior probability (score) $y_{i,s} = P(s|i, \mathbf{x})$ for each class $s \in \mathcal{S}$ at pixel index $i \in \mathcal{I} = \{1, 2, \ldots, H \cdot W\}$. Here, $\mathcal{S} = \{1, 2, \ldots, S\}$ denotes the set of S classes and $\mathcal{B} = [0, 1]$. The final segmentation

map $\mathbf{m} = (m_i) \in \mathcal{S}^{H \times W}$ is obtained with arg max operating on each pixel i of the network output $\mathbf{y} = (\mathbf{y}_i)$ individually so that $m_i = \arg\max_{s \in \mathcal{S}} y_{i,s}$. Note that $\mathbf{y}_i = (y_{i,s})$ is the vector of class posteriors at a pixel with index i. Superscripts "S" and "T" on $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{m}$ denote the domain from which the variables stem, with, e.g., $\mathcal{D}^S$ being the source domain and $\mathcal{D}^T$ being the target domain.

C. DEFINITION OF DOMAIN SHIFTS

The domain adaptation problem can be viewed as overcoming the dataset shift between the source and target domain distributions: $p_S(a, b) \neq p_T(a, b)$. Where p_S and p_T represent the source and target distribution, and a and b are the feature and class variables, respectively, where both a and b are defined and used separately only for this explanation of domain shifts. We can distinguish between three distribution shifts to describe how the domains differ: the prior, covariate, and concept shift [35].

The prior shift occurs when $p_S(a|b) = p_T(a|b)$ but $p_S(b) \neq p_T(b)$. The prior shift describes a change in class distribution. An example of this shift can be found in the distribution of classes that may differ between domains. In a synthetic source domain an abundance of pedestrians might be rendered, while they are rare in the real-world target domain.

For the covariate shift, in contrast, $p_S(b|a) = p_T(b|a)$ but $p_S(a) \neq p_T(a)$ which means the input distribution changes. An example of the covariate domain shift is the difference in styles of the two domains, which can differ concerning, e.g., brightness, contrast, saturation, and hue. Similarly, distributions can differ because objects or textures look different.

The concept shift refers to the case when $p_S(a) = p_T(a)$ but $p_S(b|a) \neq p_T(b|a)$ so that the conditional distribution differs and, therefore, the relations between a and b are different. The same features in the source and target domain describe different classes. An example can be found in the synthetic-to-real domain shift case. If a car in the synthetic world has a similar shape or texture as a truck in the real world, a concept shift has occurred.

In many practically relevant domain adaptation settings, the overall domain shift is caused by a mixture of prior, covariate, and concept distribution shifts. There are several such domain shifts relevant to computer vision systems. Training models on synthetic data for the application to real-world images introduces the synthetic-to-real domain shift. Several real-to-real domain gaps exist, too. Different sensors, locations, weather, day and night times, etc., can cause them. Further domain gaps occur when a new generation of sensors is implemented in an autonomous vehicle, or the same sensor is mounted at different positions on a different car type. Slight differences in illumination, resolution, noise, etc., can also lead to significant domain shifts. Since each domain gap would require retraining of models with new data and thus collecting and labeling this data is required, this can become very costly for large-scale applications. For this reason, domain adaptation or domain generalization methods are desired to overcome this issue and provide autonomous driving functions without needing a large-scale data selection and the corresponding data labeling effort.

TABLE 1. List of datasets that are typically employed for UDA research. Shown are the total numbers of available labeled images, as well as the resolution of the images.

Syn/Real	Dataset Name	# Labeled Images	Resolution
syn	GTA5 [36]	24,966	1914×1052
syn	SYNTHIA [37]	9,400	1280×760
real	Cityscapes [6]	5,000	2048×1024
real	NTHU [38]	400	1280×647
real	ACDC [39]	4,006	1920×1080

D. BENCHMARKS

In this seection we will discuss commonly employed datasets, network architectures, and experimental setups in the field of UDA.

1) DATASETS

The selection of datasets used for UDA research is small and contains only two major synthetic and three real dataset, as shown in Table 1, where only the Cityscapes [6] dataset is commonly employed as a target domain. This simplifies the quantitative comparison. For the synthetic datasets, usually, GTA5 [36], extracted from the video game with the same name, and SYNTHIA [37], specifically rendered for autonomous driving research, are utilized. It is worth mentioning that while GTA5 provides synthetic images from an ego-vehicle view perspective, SYNTHIA contains perspectives from both street-level and bird-eye-level views. CARLA [40] is another popular driving simulator that can be used to generate synthetic datasets. However, since no established dataset from CARLA exists, it rarely appears in UDA research for semantic segmentation [19].

For most works, the target domain dataset is the established Cityscapes [6] dataset. We also include NTHU [38] and ACDC [39] in this list. NTHU is rarely applied as a real-to-real domain shift evaluation benchmark in UDA papers. This contrasts with the additional value from NTHU since it shows how the approaches perform on real city-to-city data. ACDC does not appear in UDA synthetic-to-real works but is an often used benchmark for real-to-real adaptation from Cityscapes to ACDC with direct scene correspondences under adverse weather conditions.

SYNTHIA, GTA5, and Cityscapes in Table 1 represent the de-facto standard in UDA research for semantic segmentation. See Csurka et al. [17] for a more extensive overview, including the number of classes, conditions, etc.

2) DNN ARCHITECTURES

The most used segmentation networks in UDA are the VGG16-FCN8 [41] and the DeepLabv2 with a ResNet-101 backbone [42]. The more modern ResNet-based architecture

dominates in more recent works. The variety of network architectures for UDA research is small, simplifying this survey's quantitative comparison.

Other architectures that were used in UDA works are the MobileNetv2 as a backbone [43], [44], DRN-26 [45], [46], [47], DRN-105 [48], and smaller versions of the ResNet like ResNet-18 [49], ResNet-38 [50], [51], [52] and ResNet-50 [53], [54]. However, all these architectures appear only rarely in UDA research.

The number of vision transformer architectures (see Section III-E4.b) used for UDA is small since the research on such architectures for UDA only started recently. DAFormer [55], which is based on SegFormer [56] was the first work; HRDA has DAFormer as the basis. TransDA [57] uses SwinFormer [58] as its transformer architecture.

3) STANDARD EXPERIMENTAL SETTINGS

There exists a consensus in the research community for benchmarking UDA approaches because both datasets and architectures are the same in many works or at least very similar. Therefore, they provide a basic experimental setting for UDA benchmarks. However, in many training details the approaches differ significantly (like resolution, hyperparameters, dataset splits, etc.), so there are no unified benchmarking settings. A detailed discussion of these aspects is part of our discussion in Section VI.

III. UNSUPERVISED DOMAIN ADAPTATION APPROACHES AND METHODS FOR SEMANTIC SEGMENTATION

We discuss the methods developed for unsupervised domain adaptation in this chapter. First, we explain our taxonomy. Afterward, we present the UDA methods of each part of our taxonomy in detail. Finally, we review the latest UDA approaches using vision transformer networks.

We must fix two terms that we clearly distinguish throughout the remaining survey. An *approach* or *method* is an entire paper that may include several different standalone *techniques*. For instance, the Fourier domain adaptation (FDA) approach contains the techniques of Fourier-based style transfer and self-training, so FDA is an approach with two different techniques.

A. ADAPTATION SPACES

In order to approach the problem in a structured way, we have categorized the approaches. In deep learning, there are three typical spaces, i.e., the *input space*, the *latent representations* (*features*) within the network, and the *output* of the network. These spaces make up our three main adaptation categories illustrated in our taxonomy in Figure 3. Approaches can combine methods in different spaces, which we view as approaches belonging to the category of *hybrid* approaches. Additional surrogate tasks can help with adaptation but cannot be seen as a standalone category. Toldo et al. [59] and Csurka et al. [17] propose a similar clustering by using the three spaces. Different from them and as one of our contributions, we introduce and analyze the category of hybrid domain adaptation, where we group approaches that combine two or more spaces.

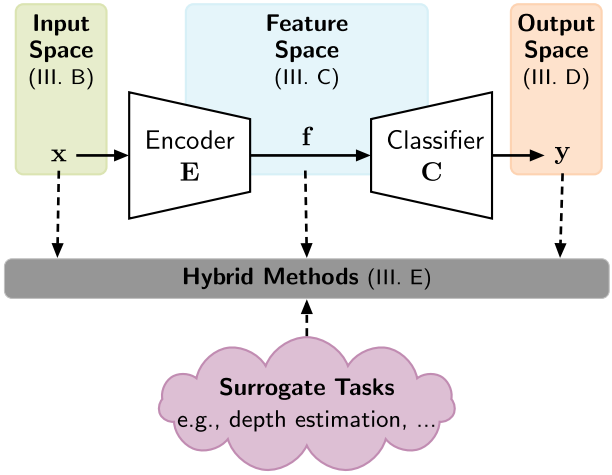


FIGURE 3. Overview of adaptation spaces that are covered in this survey. The subchapters dealing with the respective space are indicated. Hybrid methods perform adaptation in at least two spaces or utilize surrogate tasks.

The description of hybrid adaptation approaches is two-fold. First, the different techniques employed in the hybrid approaches are described in detail according to the standalone category of input, feature, or output space adaptation. FDA as a hybrid approach, for instance, will be mentioned in the input and output space subsection since it contains techniques of both spaces. Secondly, how the techniques of the different spaces interact and a more approach-level description will be provided in the section describing the hybrid approaches, where FDA will also appear. The advantage of this two-fold description is that the individual techniques of the hybrid approaches are contextually embedded within their category. In the hybrid section, the focus is purely on the different interactions between the spaces.

Unlike the previous surveys, we provide a fine-grained sub-grouping for each of the four categories. We provide a table that shows the sub-categorization of the approaches. The idea behind these tables is that readers interested in a particular topic, e.g., output-level adversarial adaptation, can find a compact collection of all approaches employing this technique in the table. Approaches will appear multiple times if one approach consists of multiple techniques.

B. INPUT SPACE DOMAIN ADAPTATION

Unsupervised domain adaptation in the input space often refers to a change in the style of the images. The three main techniques employed in this adaptation space are depcited in Figure 4.

Typically, a distinction is made between the style and content of images. The semantic structure is often considered the same as the content. Usually, it is described by well-defined low-level properties of images, such as hue,

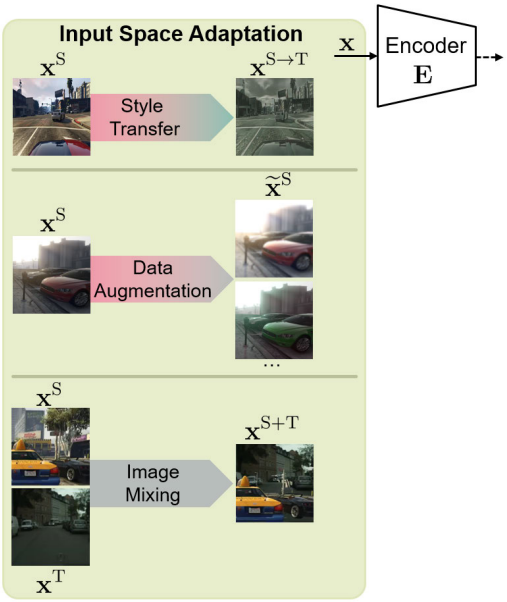


FIGURE 4. The three main input space adaptation methods are illustrated in this figure. Style transfer and data augmentation usually change the full appearance of the image, but style transfer does it in a more target domain directed manner. Image mixing generates images consisting of source and target domain pixels.

TABLE 2. Adaptation techniques in the input space. The papers are clustered and sub-clustered according to similar methodology.

Technique	Sub-Cluster	Approach
Style Transfer	Feature Transforms	[60], [61], [63]
	Normalization	[65], [83], [84], [85], [86], [87], [88], [89]
	GAN-Based	[31], [45], [46], [47], [59], [62], [83], [85], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99], [100], [101], [102], [103], [104], [105], [106], [107], [108], [109], [110], [111], [112], [113], [114]
	Frequency Domain	[69], [70]
	Histogram Matching	[66], [67], [68]
Data Augmentation		[79], [80], [81], [82], [115]
Image Mixing		[72], [73], [74], [75], [76], [77], [78]
Others		[23], [116]

saturation, contrast, brightness, image-noise, depth of field, etc. However, the appearance of cars, e.g., their shape or texture, can also be counted as a style, even if the previously mentioned properties cannot express this. The general idea with style transfer is to align the source and target domain distributions on a pixel-level in the input space. This can happen, in the simplest form, at low-level image properties, such as hue, saturation, brightness, etc., e.g., by histogram matching algorithms. More complex methods, e.g., GAN-based methods, can vary the style even more and can change textures, depth of field, etc. However, a limitation of many approaches is that the semantics of the source images must remain the same so that the labels can still be used. It is still hard to find a style transfer that maps, e.g., northern central-European vegetation, cars, traffic signs, etc., to, e.g., middle-eastern vegetation, different cars, and traffic signs. The adaptation techniques that include object shape must be modeled by feature space adaptation.

When not only the style of images, but also the early feature maps are modified by the method, we would refer to it as a feature-level adaptation approach (cf. Section III-C). During adaptation in the input space, the data samples $\mathbf{x}$ used as the input for the method are modified. Multiple methods can be used to alter the input images and improve the performance in the target domain. Usually, style transfer, content mixing, or data augmentation methods are employed. Style transfer methods try to match the target and source domains' style while not changing the samples' semantics. For unsupervised domain adaptation, the style of the source domain, which is usually synthetic in scientific benchmarking, is transferred to be similar to the anticipated target domain, which is usually a real-world domain.

Domain adaptation, in most cases, is performed by more than style transfer methods. These methods are often only part of the domain adaptation process and facilitate a more rapid convergence for subsequent methods. Nevertheless, some approaches solely rely on style transfer [60], [61], [62]. Some methods perform the style transfer the other way around so that the target domain images are more similar to the source domain. By doing this, the target domain does not have to be anticipated beforehand, but the style transfer method must be a part of the inference. These approaches can even be used for continuous domain adaptation under changing domains [26]. So far, no simple style transfer technique has been able to achieve a state of the art performance for UDA on its own. The style transfer methods in the input space are typically detached from the further training process and are often combined with other methods.

The style transfer is the most popular approach and is usually performed utilizing feature transforms [60], [61], [63], GAN-based networks, e.g., CycleGANs [64], normalization techniques, such as AdaIN [65], histogram matching [66], [67], [68], or image processing in the frequency domain [69], [70]. In recent years several image content mixing methods were proposed that mix the source and target directly in the image domain [71], [72], [73], [74], [75], [76], [77], [78]. Also, several data augmentation methods [79], [80], [81], [82] were proposed for input space UDA.

In the following, we will briefly discuss the methods. An overview of these references is given in Table 2. Since the methods listed under Others do not specify the style transfer type used, they are not discussed in detail in the remainder of this chapter.

1) STYLE TRANSFER

As already mentioned, style transfer is the primary input space adaptation technique. We will discuss style transfer using feature transforms, normalization techniques, image processing in the frequency domain, histogram matching, and GANs. Usually, style transfer is applied in one of two ways. First, the source images can be transferred to match the target domain during training (cf. Figure 4, upper part). In this case, during inference, no style transfer is needed. Second, the target domain images can be transferred to match the source domain. With this setting, style transfer is also needed during inference.

a: FEATURE TRANSFORMS

Style transfer using feature transforms must be distinguished from feature space domain adaptation. The feature transforms presented here are methods that convert the images of the source domain into the style of the target domain with the help of a style transfer network. The features of the original segmentation network are not adapted during this process. Instead, an additional network (usually an autoencoder) is trained on source and target images to transfer the style of the source images in their bottleneck features.

Early works that employed a style transfer for unsupervised domain adaptation used simple feature transforms such as FastPhotoStyle [117], which comprises a two-step stylization and smoothing process. At first, the style of a content image is stylized in the style of a style image from the target domain using an enhanced whitening and coloring transform (WCT) [118], which is called PhotoWCT. In PhotoWCT, the upsampling layers of the style transfer network are replaced by unpooling layers. Afterward, smoothing is performed to ensure that semantically similar regions are stylized consistently. The FastPhotoStyle method [117] was utilized by the domain stylization (DS) [60] and the mask-aware gated discriminator (MAGD) [63] methods, which both randomly match source and target domain samples. Restyling data (RD) [61] also employs FastPhotoStyle [117] as a style transfer method and improves the sample matching by computing so-called perceptual hashes in the frequency domain of the images. These hashes are then used to match samples for the style transfer, and it is based on the Hamming distance of the respective hashes.

b: NORMALIZATION METHODS

The efficacy of normalization methods for style transfer has been known for some time [88]. Adaptive instance normalization (AdaIN) [65] is particularly relevant in this context. The style transfer with AdaIN uses an encoder-decoder structure (usually based on a VGG-19 [119] architecture), where the AdaIN layer receives the features of a content image (in the case of domain adaptation, usually an image from $\mathcal{D}^S$) and a style image (from $\mathcal{D}^T$ respectively). AdaIN then performs the style transfer by transferring the channel-wise mean and variance statistics of the features. AdaIN allows as many different style transfers to be learned for a (content) image as there are style images.

Methods such as DCAN [89] employ AdaIn and assume that the mean and standard deviation of the feature maps in an image generator encodes an image's style information. They hence follow the idea to train an autoencoder in a way that it reconstructs images from the source domain $\mathcal{D}^S$. However, simultaneously, the mean and standard deviations are aligned between the source image that is to be reconstructed and a randomly selected image from the target domain $\mathcal{D}^T$. Given that the feature statistics are matched, the generator will produce the source image in the target domain style. As we shall see later in Section III-C, this idea is also significant for the distribution alignment in the feature space. The bidirectional style-induced domain adaptation (BiSIDA) [86] employs a source-to-target style transfer for supervised training and a target-to-source style transfer for the unsupervised learning branch of the framework. The style transfer is performed using the standard AdaIN method. Also, the CFContra method by Tang et al. [87] employs an encoder-decoder network with standard AdaIN layers for style transfer.

The adversarial style mining (ASM) method [84] uses a newly proposed random AdaIN (RAIN) module for style transfer. RAIN adds a style variational autoencoder (VAE) in the latent space to encode the features' channel-wise mean and variance statistics into a Gaussian distribution that can be sampled from the latter. During training, the RAIN module is trained to iteratively generate harder stylized images around the initial target sample according to the current learning state. That way, the segmentation model learns more potential styles in the target domain.

The target-guided and cycle-free data augmentation (TGCF-DA) method [85] employs a cycle-free generator network that is based on multimodal unsupervised image-to-image translation (MUNIT) [120]. The generator is extended by AdaIN layers, which enable several style transfers (as many as there are style images) to be learned. The network is trained by a discriminator (distinguishing whether the image stems from the source or the target domain) and a semantic loss, ensuring that the semantics between the original source image and the style-transferred source image remains unchanged.

c: FREQUENCY DOMAIN

Domain adaptation in the frequency domain is a relatively new field. Yang et al. [69] proposed a new form of style transfer by implanting low-frequency information from the target images into the source images. This Fourier domain adaptation (FDA) is performed in the frequency domain. Only parts of the amplitude spectrum are exchanged, as these are assumed to contain the general style of the images. Similar to FDA, the authors of SUDA [70] employ a style transfer in the frequency domain. They decompose the input image into multiple frequency components and train a transformer network to recompose a newly stylized image from

these frequency components. The transformer network learns to suppress domain-variant contents and enhance domain-invariant contents.

d: HISTOGRAM MATCHING

Histogram matching is a long-established method [121] to match the style of images. However, only recently has there been research for its use for domain adaptation. Huang et al. [66] tackle the task of panoptic segmentation, but the technique can also be employed for classical semantic segmentation. They propose an inter-style consistency, where the input images get stylized, and the segmentation masks between different styles, e.g., illumination or weather conditions, are learned to be equal. This is then combined with an inter-task consistency, which enforces consistent labels between a semantic segmentation and an instance segmentation network. They employ a histogram-matching algorithm [121] for the stylization. Ma et al. [67] propose a global photometric alignment for style transfer. They align the source and target images style by histogram matching in the three channels of the L*a*b* color space. The same global photometric alignment is also employed by BiSMAP [68].

e: GAN-BASED METHODS

Generative adversarial networks (GANs) currently dominate the field of input space adaptation methods. GANs modify an image by a generator network so that a subsequent discriminator network can no longer distinguish from which domain the image originates. By training a discriminator network, high-quality style transfers can be performed. In particular, CycleGAN [64] has proven to be a successful choice. It provides a photorealistic transformation between different image styles and mostly prevents semantic changes in the image due to the cycle consistency. The goal is to learn a mapping function $\mathbf{G}: \mathcal{D}^S \mapsto \mathcal{D}^T$ as well as an inverse mapping function $\mathbf{G}^{-1}: \mathcal{D}^T \mapsto \mathcal{D}^S$ and employ the cycle consistency to enforce that an image remains semantically the same after mapping and inverse mapping. However, most GAN-based methods are limited in terms of the variability of the stylized images.

Methods such as MUNIT [120] that combine GANs with, e.g., AdaIN, try to overcome this limitation. They use AdaIN in their generator network to generate more specific style transfers. The LSD method by Sankaranarayanan et al. [91] was about the first to employ a standard GAN-based style transfer for domain adaptation. Also, Chen et al. [90] employed a GAN for style transfer. The domain invariant structure extraction (DISE) method [92] tries to disentangle the images' structure and texture during style transfer. This way, the structure and the texture of different source or target images can be combined. The method employs a least squares GAN (LSGAN) [122] and can be used in both directions.

Li et al. [62] follow a slightly different strategy as they do not employ a style transfer directly on the image level. Instead, they propose a label-to-image domain adaptation (L2I-DA) transfer where they generate image-label pairs in the target domain style. They also employ a standard GAN for the image translation process.

DRANet [46] improves the style transfers from the generator network by searching the target features whose content component is most similar to the source features. The domain transfer is performed by incorporating style information from more suitable target features.

SPIGAN [103] simplifies the CycleGAN architecture by only using a single sim-to-real generator (no cycle consistency) and a downscaled generator network. The light-weight calibrator (LWC) method [45] employs the ResNet generator proposed by Johnson et al. [123] as a data calibrator. The calibrator can be seen as the generator. Two discriminators are employed, one on pixel level, and one on feature maps from the feature extractor. The translation process is based on an adversarial distribution alignment of the feature space and a pixel-wise calibration network in the input space. The pixel-wise calibration is based on an encoder-decoder architecture and is applied during inference, too.

Cai et al. [112] propose a condition-guided style transfer by employing a standard conditional GAN [124] that is trained with a semantic consistency loss. They also utilize concepts from StarGAN [125] and BicycleGAN [126]. This way, preferred styles like `foggy` or `cloudy` can be added to the images as needed.

SUIT [113] allows an improved style transfer by designing a novel semantic-content loss that focuses on label- and content-consistency between original and stylized images to guide the style transfer. The content-consistency is employed by comparing features of a pre-trained network for the stylized and the normal input images.

The stochastic image translation method by Chiou et al. [114] is based on MUNIT [120]. The authors propose not performing an image-based but stochastic-style translation. A source encoder encodes the content of the source image, and a target generator generates stylized versions of this image by sampling from a style distribution of the target domain.

The CycleGAN architecture, in particular, has been used and expanded by many papers as their style transfer network of choice. The CyCADA method [95] was among the first to perform a style transfer with a CycleGAN. It also explicitly encourages high semantic consistency before and after image translation for the source domain samples with a pre-trained source segmentation network. Also, CrDoCo [47], MSS [59], and CADA [96] employ a standard CycleGAN for their image translation. Zhou et al. [100] show that their ASANet+ is complementary to style transfer by combining their method with the image translation module from CyCADA [95].

The SE-GAN method [107] makes adversarial training more stable and employs a simple CycleGAN for style transfer. Yang et al. [93] utilize a CycleGAN that uses both a cycle consistency and a phase consistency loss. They show that the semantic information is mostly encoded in the phase from the complex spectrum of the image and enforce its

similarity for the transformation and inverse transformation of the CycleGAN.

In DLOW [31], the authors generate a sequence of intermediate domains between the source and target. They define a domainness factor z that affects the generator and the discriminator. They also employ a cycle consistency loss and build their method upon CyCADA [95].

Another idea is to use a content invariant representation (CIR) [108], which can be seen as an intermediate domain between the source and target with the same content as the source domain and the same style distribution as the target domain. They use a vanilla CycleGAN to generate this CIR.

A popular approach on which many works build is the bi-directional learning (BDL) method [98], which improves the image-to-image translation model by iteratively improving the translation model with feedback from the subsequent semantic segmentation model. This way, the image-to-image translation is not fixed but improves during training and adaptation. The authors also published their style-transferred images from the GTA5 [36] and SYNTHIA [37] datasets, which were used by many subsequent methods. For example, CDGA [99], SIM [101], MCSSF [105], and BDL+ESL [110] use this method or the already transferred images.

In contrast to previous works, the authors of LDR [102] train a style translation model that transfers the target domain images in order to make them look like source domain images. They employ the general translation model of BDL [98] but add a cycle-reconstruction loss to enforce semantic consistency between the image and the image reconstructed from the labels. The active pseudo-labeling (APL) method [94] first adapts the target domain images to the source domain using a style transfer. Afterward, the style-transferred images are used to create pseudo labels that are later used for self-supervised training in the target domain (cf. III-E). The style transfer is similar to that of LDR [102], but it replaces the transposed convolutions with bilinear upsampling and convolutions.

Ramirez et al. [97] employ a CycleGAN for style transfer from the source to the target domain in their image-level domain adaptation (ILDA) method. They enforce the similarity of segmentation masks based on style-transferred images and unaltered synthetic images using a discriminator in the generation process to avoid artifacts and guide the synthesis. The DISE-CT method [104] is based on DISE [92] but adds a cycle consistency to the generator training. It also adapts the zero loss [127] to a zero-style loss. A content transfer is employed for long-tail classes of the target domain to incorporate more of these classes into the training samples.

Dual path learning (DPL) [106] employs two pipelines, where images are transferred from the source to the target domain or from the target to the source domain. Both pipelines are trained interactively with a so-called dual path adaptive segmentation.

With KATPAN, Dong et al. [109] employ a modified CycleGAN for the image translation process. They extend the standard CycleGAN with a transferability-aware information bottleneck that guides the encoder to encode only discriminative features.

Musto et al. propose a new semantically adaptive image-to-image (SA-ITI) translation [83]. They utilize the segmentation maps from the source image provided by the segmentation network to guide the style transfer of the source domain images to the target domain. As their style transfer network, they design two coupled GANs similar to a CycleGAN and adaptively denormalize each pixel based on the semantic information. The translated image is then fed to the segmentation network again. Consistency is enforced between the two output posteriors using a new symmetric cross-entropy loss.

However, there are also further enhancements of the CycleGAN architecture, e.g., the symmetric adaptation consistency (SAC) method uses a StarGAN [125] for image-to-image translation.

2) DATA AUGMENTATION

An additional idea for domain adaptation in the input space is data augmentation. With data augmentation, the styles of the images are changed in a less targeted manner than with a style transfer (cf. Figure 4, middle part). Thus, no attempt is made to represent the target domain as precisely as possible. Instead, the images are changed as diversely as possible to train a network that is as robust as possible against various domain shifts. This is related to domain randomization, which is often used for domain generalization.

Zhou et al. [82] perform a *class out strategy* in the input space by employing a ClassDrop mask generation algorithm that provides class-wise perturbations. The learning texture invariant representation (LTIR) method [79] generates a stylized version of the commonly used GTA5 [36] and SYNTHIA [37] datasets to force the model to learn texture invariant representations, which are usually not learned from style-transferred images.

Huang et al. [80] train a more robust network against domain shifts by learning Fourier domain adversarial attacks and iteratively learning to defend against these attacks. These attacks are some form of style augmentation. Araslanov et al. [115] perform heavy data augmentation and then calculate output consistency using differently augmented images. The unsupervised contrastive domain adaptation (UCDA) method [81] also employs multiple augmentation techniques on source and target domain images.

3) IMAGE MIXING

Similar to data augmentation techniques, more and more methods have recently been developed that mix source domain images with portions of target domain images (cf. Figure 4, lower part). One popular method is domain adaptation via cross-domain mixed sampling (DACS) [71], on which many other methods have been built since. DACS mixes samples from the two domains along with the corresponding source labels and target pseudo-labels. The labeled source domain images and the mixed images are used for

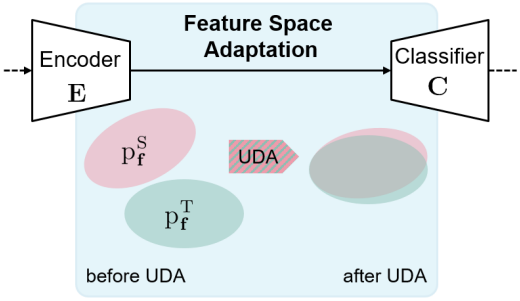


FIGURE 5. Feature space adaptation methods: The encoder of a CNN projects the input image to the feature space. Here before domain adaptation, the source and target distributions are not aligned. Hence the source domain-trained classifier does not generalize to the target domain. After feature space adaptation methods are used, the feature distributions are generally aligned much better, which improves performance on the target domain.

training. It also applies color augmentation and Gaussian blurring to the training samples.

The RCCR approach [75] employs ClassMix [128] and CutMix [129] as proposed by DACS [71]. Also BAPA-Net [74] solely employs CutMiX [129]. Likewise, the CorDA method by Wang et al. [77] is based on DACS [71] and utilizes all of its input space adaptations. Zhou et al. propose a new image mixing method termed CAMix [76], where they leverage contextual information on relationships to guide the image mixing. It can be seen as an improved version of DACS.

DBST [130] adds depth guidance to DACS and the authors explicitly propose their method as a module that can be combined with any other UDA method like, e.g., ProDA [131]. The dual soft-paste (DSP) method [72] improves on DACS [71] by pasting mainly long-tail classes from the source domain in source and target domain images. It creates two intermediate domains, which serve as a bridge between the domains. They preserve the original domain information by keeping objects, layout, and general structure the same.

PixMatch [73] employs a consistency training with two different perturbations added to the images in its best working model. The authors show that Fourier domain and CutMix [129] perturbations yield the best results.

C. FEATURE SPACE DOMAIN ADAPTATION

As identified in the previous sections, the distribution shift between the source and the target domain leads to decreased performance. Since the pre-logit feature space (the output of the last layer before the classifier) distributions of the source and target domain differ, a classifier trained on one cannot generalize well to the other (see Figure 5). Hence, we discuss approaches that try to adapt the model from the source to the target domain, trying to align the distributions in the feature space. In this case, the alignment of the distributions depends on the learned encoder-decoder function that maps the input to the (pre-logit) feature space. Therefore, distribution alignment between input data from the source and the target domain means learning the encoder-decoder function in a way that maps input from both domains that semantically represent the same things to a similar point in the feature space. In this case, the classification hyperplane learned on the source domain will generalize well to the target domain.

TABLE 3. Adaptation methods in the feature space. The papers clustered and sub-clustered according to similar methodology.

Technique		Approach
Distribution Divergence	Adversarial Training	[12], [23], [38], [45], [47], [53], [59], [95], [96], [106], [116], [132], [133], [134], [135], [136], [137], [138], [139], [140], [141], [142], [143], [144], [145], [146], [147], [148] , [149], [150]
	Instance Norm/ Gaussian	[143]
	L2 Distance	[145]
	Max Classifier Discrepancy	[48], [151], [152]
	Max Mean Discr. (MMD)	[72]
	Shared Style GAN	[91], [104], [153], [154], [155],
Instance Norm		[24], [51], [89], [156], [157], [158]
Self-Supervised	Contrastive	[75], [81], [159], [160], [161], [162], [163]
	Semantic Clustering	[44], [53], [74], [87], [99], [105], [164], [165], [166], [167]
	Depth & Ego Motion	[77], [149], [168]
	Augmentation	[169], [170]
	Weak Supervision	[23] [171]

One can identify different subclusters in the methods for distribution alignment in the feature space (see Table 3) that we will discuss in the following subsections.

1) DISTRIBUTION DIVERGENCE

Methods that fall in this cluster try to minimize a divergence measure describing the distance between the source and the target domain.

a: ADVERSARIAL ADAPTATION

Ganin et al. [172] introduced the first and most prominent among these methods. Although this paper deals with image classification, this work significantly impacted unsupervised domain adaptation for segmentation. The authors define the distance between the source and the target domain as the so-called H-divergence. The H-divergence is computed based on a classifier, classifying whether the feature representation of an image is from the source or the target domain. Given such an optimal domain classifier, the H-divergence is minimal if the error of the optimal classifier is maximized. Minimizing the H-divergence poses a min-max problem. The H-divergence is minimal if the encoder is learned, so the

domain classifier yields a maximum error rate. In contrast one has to minimize the error to obtain the optimal domain classifier. Ganin et al. [172], propose a gradient-reversal layer (GRL) between the domain classifier and the feature space to solve this problem. During the backward pass, the gradients of the domain classification loss are applied to the domain classification head but inverted for the encoder-decoder function and thereby minimizes the H-divergence.

There is a multitude of methods (see Table 3) that make use of this idea for the task of semantic segmentation. *FCNs in the Wild* [12] is the earliest example of adversarial learning in the feature space. The proposed method applies the adversarial loss function on the pre-logit feature map (the last representation before the classifier). Hoffman et al. [12], however, implement the adversarial principle in a different way. They define a domain classification loss that is used to optimize the domain classifier and the encoder-decoder weights of the segmentation network and use the inverse of this loss to update only the encoder-decoder weights. Both losses are applied in an alternating way. This way of implementing the adversarial principle of the H-divergence, without a gradient reversal layer, is applied by the majority of UDA approaches.

Building on *FCNs in the Wild* [12], many approaches use adversarial training as an additional tool in their domain adaptation strategy, e.g., WDC [138], RPT [148], DPL [106], CAA-Net [155], and SWLS [53]. There are, however, many works that introduce new strategies to improve adversarial learning.

b: SUB-DISTRIBUTION ALIGNMENT

The alignment of the global distributions of the source and target domain can lead to issues. A possible consequence of global distribution alignment is that sub-distributions of source and target domain that are closely aligned even before the adaptation are affected negatively by the global alignment (see Luo et al. [135]). Sub-distributions in our context denote parts of the source or target domain feature distributions that depend on, e.g., the classes or the spatial position. Wang et al. [140] argue that parts of the class-wise sub-distributions might get mixed up through the global distribution alignment. They further point out that the different frequencies of classes lead to the situation that the sub-distributions of frequent classes are aligned better than rare classes' sub-distributions. Finally, Chen et al. [147] speculated that another issue with global distribution alignment through GANs are non-contributing ambiguous features.

Chen et al. [38] and Du et al. [132] introduced early approaches to align the class-conditional sub-distributions. Their idea is to introduce a class-wise adversarial training. The discriminator classifies between the source and target domain only for feature representations of the same class. The approaches use self-inferred pseudo labels on the target domain to implement the class-dependent domain classifier. Additionally, the approach weighs the adversarial loss higher for classes with low average confidence. The work by Du et al. [132] improves the approach of Chen et al. [38] by addressing the inconsistent adaptation issue. CCDA [144] addresses the alignment of the class-conditional sub-distributions of classes with different frequencies. Wang et al. [144] employ two discriminator networks, one for coarse-level alignment and one for pixel-level alignment. For the coarse-level alignment, the discriminator network predicts the domain label of every coarse feature representation element and the classes present in the receptive field. The second discriminator computes the adversarial loss pixelwise. The influence of each class is normalized with its frequency, giving frequent and rare classes a similar weight. Additionally, they weigh spatial elements higher which possess a high classification uncertainty. FADA [140] and CCDA [144] follow a very similar idea.

CCD [147] tackles the problem that non-productive ambiguous features are learned during global distribution alignment through GANs. To prevent this, they also train a segmentation loss on the sub-network in addition to the discriminator. However, the segmentation on this network is not backpropagated to the shared backbone.

Finally, ROAD [145] assumes that similar classes occur at similar spatial positions in an image and uses the adversarial loss dependent on the spatial position. Their domain classification loss is computed for predefined regions (grid elements) in the image. Given that similar classes and objects appear at similar spatial positions, the source and target domain distributions of grid elements match well a priori. The adversarial distribution alignment hence matches sub-distributions that are similar.

c: ADVERSARIAL TRAINING WITH ATTENTION

The following approaches aim at guiding the adversarial adaptation process to the most relevant regions. For this purpose, the approaches utilize an implicit or an explicit way of guiding the attention of the adaptation process.

Li et al. [133] use spatial and channel attention to achieve this goal. They create a so-called highly embedded feature vector representing information about the feature space, the network prediction, and spatial and channel-wise attention maps. The adversarial training is done based on this feature vector so the goal goal is to align the distributions of the source and target domain of the vector.

DAST [139] uses discriminator confidences to measure the alignment of the source and target domain. After an initial adversarial alignment, the authors weight the feature map of the target domain with the domain classification output. A high confidence score of the discriminator indicates that a feature representation is easily identifiable as part of the target domain. Hence such a feature representation still has to be aligned to the source domain and is given a high weight.

Chen et al. [146] do not directly model the attention via a measure for the distribution alignment or a spatial attention module. They instead assume that the semantic edges or boundaries between classes are significant for predicting semantic segmentation. Thus the network comprises semantic and edge (class boundaries) segmentation branches.

In order to make the edge predictions domain-invariant, adversarial training in the edge branch feature space and feature fusion between the semantic and edge branch is applied. The edge feature distribution alignment guides the attention implicitly to the class boundaries.

d: ADVERSARIAL TRAINING ON STYLE

As described in Section III-B, style transfer enables supervised training on source domain images with the style of target domain images. Apart from that, several approaches utilize style transfer for adversarial adaptation in the feature space.

CyCaDa presented by Hoffman et al. [95] and as described in Section III-B trains a CycleGAN network to transform source domain images into the target domain and vice versa. Apart from this main contribution, the approach applies adversarial learning between the stylized source domain images and the not transformed target domain images. The discriminator distinguishes between the respective feature representation of stylized source domain and the target domain images.

CrDoCo [47] trains two segmentation networks on the source domain labels, a source, and a target domain network. The target domain network is trained with the source domain labels but with style-transferred images. Two separate discriminators enable the adversarial training in the feature space of the two networks. The discriminator for the source domain network takes feature representations of source domain images and target domain images that were transferred to the source domain. Vice versa, the target domain network is trained. A consistency loss between the outputs of the two networks for the target domain images and the transferred target domain images is applied. The fact that the source and target domain only differ in style but not in content facilitates the distribution alignment. The authors of MSS [59] follow a similar approach as in CrDoCo [47]. The main difference to CrDoCo [47] is that the encoder computing the feature representations is shared between the domains.

LWC [45] is different from the previous works because it tries to align the distributions of the source and target domain not by altering the encoder but by transforming the input image. The authors of LWC [45] present a calibrator strategy for domain adaptation. Given a model trained on the source domain, the aim is to train a calibrator network that transforms the input image in such a way that the distributions of the source and target domain feature representations are aligned in the feature space of the source-trained segmentation model.

e: SHARED STYLE GAN

Most approaches that use a shared style GAN rely on the original GAN principle presented by Goodfellow et al. [173]. These approaches usually have four elements: A shared encoder **E** that generates a shared feature space; a segmentation classifier **C** that computes the semantic segmentation from the feature map produced by **E**; a decoder **G** that reconstructs the input image (mostly trained by $L1$ loss); And finally a discriminator **D** that tries to classify the image output of G into either being ''fake'', or ''real''.

The approach presented in LSD [91] has all the architectural elements described. The discriminator **D** distinguishes between fake and real source domain images and fake and real target domain images. The decoder **G** generates fake target and source domain images by adding dropout noise to the feature embeddings generated by **E**. An adversarial loss is computed between real and fake images inside each domain and cross-domain. This way, the encoder **E** is trained to output similar feature space embeddings for the source and target domain. **C** takes this domain-aligned feature map to compute the segmentation. CAA-Net presented by Ruan et al. [155] follows a similar direction as LSD [91].

The architecture in PTP [154] consists of the elements and again follows a similar principle as CAA-Net [155] and LSD [91]. Similar to LSD [91], the final objective is to achieve similar feature embeddings for both domains by applying an adversarial loss. The biggest difference is that, e.g., on the source domain ''real'' would be the reconstruction of the source domain image and ''fake'' would be the reconstruction of the same image in a target domain style.

CLADA [153] computes a transformation that is added to the pre-logit feature space of a segmentation network to transform the source domain features to the target domain. The classifier is trained on a target domain feature distribution given such a transformed source domain feature space. The conditional generator G takes in a noise channel and a low-level source domain feature map of encoder E. The two inputs are concatenated and passed through a ResNet architecture, computing the transformation. The discriminator distinguishes between transformed and non-transformed source domain feature maps.

The network architecture of Lee et al. [104] has three encoders that share the first convolution layers. One encoder extracts the content, and two other encoders extract the source and target domain style information, respectively. The decoder computes the segmentation, and the other two decoders compute the image reconstruction in the source or target domain style. The authors use a zero loss function, which minimizes the $L1$ norm so that the two encoders capture unique information that only exist in the source domain and target domain. According to the authors, the source encoder only learns the style-independent content features.

f: MAXIMUM CLASSIFIER DISCREPANCY (MCD)

Next to the H-divergence, other distribution discrepancy metrics are used for distribution alignment in the feature space. Saito et al. [48] introduce an approach based on the maximum classifier discrepancy (MCD). The MCD is computed by first training two classifiers for the source domain, mapping the same feature map into a segmentation. In the second step, the discrepancy of the probability output of these two classifiers is maximized for the target domain. The discrepancy between

the class probabilities is computed using the $L1$ norm. The segmentation loss is trained on the source domain in parallel to keep the source domain performance from degrading. The result is two classifiers that agree with each other for samples with support from the source domain distribution and disagree with each other for samples that are not represented well in the source domain. The latter case characterizes most of the target domain samples. In the third step, the feature extractor is then optimized to minimize the MCD for the target domain. This causes those samples from the target domain far away from the source domain distribution to move closer to the source domain distribution (here, the support of the source domain is given, and the two classifiers agree). The three steps are iterated, which results in an adversarial optimization.

Lee et al. [152] advance this work by introducing an improved way of computing the discrepancy between the classifier probability outputs. The authors propose the sliced Wasserstein discrepancy, which considers the properties of the underlying geometry of probability space and thus improves upon the $L1$ norm used in [48]. Further follow-up work is presented in Li et al. [151] where two classifiers are trained on the source domain while also updating the feature generator. Then they maximize the classifier discrepancy on the target domain while ensuring the source domain classification stays the same. In the final stage, they train the feature generator to minimize the classifier discrepancy, pushing the target domain data to the statistical support of the source domain.

g: OTHER METHODS FOR DISTRIBUTION DIVERGENCE MINIMIZATION

MMD [72] uses the soft paste algorithm, combining two images by a weighted overlay (see Section III-B). A reference source domain image is pasted into a target and source domain image. This is done based on a mask containing relevant classes in the reference image. The authors try to align the feature space representation of the source and target image in the region of the mask. The minimization of the squared difference of the kernel-mean-embedding of the feature representations in the mask regions in the reproducing kernel Hilbert space introduces the alignment. In addition to the alignment in the mask region, the authors apply a global alignment using the same method (MMD) without filtering with the reference image mask. The general activation matching (GAM) [143] approach trains two networks, one for the target and one for the source domain. The authors apply an $L2$ minimization of the difference of the weights between the source and target domain networks and Jensen-Shannon divergence matching between the output of source and target domain. The latter is optimized in an adversarial manner. Additionally, the feature maps of the target domain are scaled to the source domain mean and variance. These adaptation methods are applied in each layer of the network. PFR [174] approach utilizes the $L2$ distance of the feature representations of source and target domain images. The style features and content features are computed using the method presented by Gatys et al. [175]. The $L2$ distance is minimized at different feature levels.

2) SELF-SUPERVISED LEARNING

Self-supervised learning is based on so-called pretext tasks that can be annotated automatically without human effort. The assumption is that the training on pretext tasks results in an encoder that produces features that are relevant to the actual task that should be solved, i.e., semantic segmentation. Since self-supervised learning can be used for unsupervised feature learning, it is often applied for pre-training. More importantly, in our case, it is an essential method for unsupervised training on the unlabeled target domain, too.

a: AUGMENTATION AND DEPTH

SSL-UDA [176] and SSDA [170] introduce a process to make use of self-supervised learning for an implicit alignment of the source and target domain. In addition to the main task of semantic segmentation, they employ the pretext tasks image rotation, image flipping, and location prediction of image crops. These tasks are trained on the source and the target domain jointly. The idea is that by training the encoder to produce relevant features on the source and target domain, the distributions of the source and target domain will also align in the feature space. The authors of SSL-UDA [176] show that the centroids of both distributions get closer over the training epochs. However, the quality of such approaches is dependent on the pretext task. The approaches presented in GUDA [168], CTRL [149], and CorDA [77], show that depth prediction and ego-motion estimation are meaningful pretext tasks. GUDA [168] makes use of recent advances in the domain of unsupervised depth estimation. In addition to the semantic segmentation on the source, domain the authors train an unsupervised depth estimation on the target domain that implicitly predicts the ego-motion. Since the prediction of pixel-wise depth maps requires similar features as semantic segmentation, utilizing depth for the pretext task trains the encoder to extract relevant features even on the target domain. CorDA [77] and CTRL [149], in contrast to GUDA [168], do not train the unsupervised monocular depth estimation as a pretext task. Instead, the authors assume fixed depth labels, which unsupervised monocular depth estimation approaches can compute, too. In Wang et al. [77], another difference can be found in how depth estimation is incorporated via spatial attention into semantic segmentation. For further details of how approaches incorporate depth and ego-motion pre-text tasks for self-supervised feature learning, refer to Section III-D.

b: CONTRASTIVE LEARNING

Unlike the implicit alignment of the feature distributions, self-supervised approaches are also used for direct alignment. DACL [161], SPCL [160], UCDA [81], PWCL [162],

RCCR [75], and CLST [159] make use of contrastive self-supervised learning.

Contrastive self-supervised methods are based on so-called positive and negative pairs. In general, such methods aim to make the feature representation of positive pairs more similar and those of negative pairs more different. It depends on the task and method how positive and negative pairs are constructed. A positive pair are, e.g., two instances of the same class or two augmented versions of the same object. In the case of domain adaptation, the construction of positive and negative pairs is not trivial because no labels are available in the target domain.

The approach presented by Shim et al. [161] uses a CycleGAN-generated style transfer from the source to the target domain. The resulting pseudo-target domain images with ground truth labels yield pixelwise class information. The contrastive loss can be computed based on the so-constructed positive and negative pairs.

The authors of CLST [159] follow a different approach. The idea is to construct high-quality pseudo labels for the target domain. Given such pseudo labels, one can construct positive and negative pairs across the source and target domain. The positive and negative pairs are constructed between the source domain class centroids and the target domain class centroids for each target domain image. The feature representations of the target domain are clustered towards their respective source domain centroids and moved away from the wrong source domain class centroids. The approach of SPCL [160] is similar to CLST [159]. The authors compute the average feature representations of each class on the source domain and update them in a moving average way. The contrastive loss is computed from the feature representation of each pixel to the centroids. In the target domain, the assignment is done by the pseudo-labels of the network.

RCCR [75] combines contrastive learning with knowledge distillation and introduces both a teacher and a student for the projection head. Differently from other works, the projection head consists of convolution layers. The positive and negative pairs are constructed by the student and teacher network based on a source-target mixed image and a regular target image. As the only UDA approach, RCCR utilizes a memory bank to include negative samples from previous batches to increase the variety of the negative pairs and thereby improve the discriminability of the learned representations. UCDA [81] follows SimCLR [177]. It adds two MLP layers to transform the feature representations into a 128-dimensional vector representation. Class prototypes are computed per batch. Each feature vector contributes to each class prototype according to the softmax probability of the teacher network. Then anchor features are chosen within the same domain, and the contrastive loss is computed. Additionally, they choose anchor features in the source domain and assign them to the corresponding target domain centroid.

PWCL [162] determines positive and negative pairs between source and target domain image patches. The use of image patches is a distinct feature and is done through multi-level spatial pyramid matching. Their contrastive approach is close to the idea of MoCo [177] and utilizes the cosine similarity.

SCDA [167] differs from the previously described contrastive approaches because it does not create positive and negative pairs based on concrete feature representation, but rather operates on class distributions. The authors estimate the distributions of each class in the feature space based on source domain statistics using the mean and covariance. It is computationally infeasible to compute the contrastive loss for multiple positive and negative pairs. To resolve this limitation Li et al. derive a loss that directly utilizes the gaussian distributions of the positive and negative classes.

c: SEMANTIC CLUSTERING

Apart from the implicit adaptation through self-supervised learning and the construction of semantic pairs in the source and target domain, one can identify a third class of self-supervised domain adaptation approaches. Semantic self-supervised approaches as presented in DANCE [178] CAM [166], CFContra [87], SCDA [167], BAPA-Net [74], SWLS [53], and SSS+ST [165] which all aim to cluster the pre-logit feature space towards so-called class prototypes directly. These class prototypes are vectors that represent the pre-logit feature representations of their respective class.

By advancing the method proposed by DANCE [178] Niemeijer et al. with SSS+ST [165] present an approach for semantic self-supervised learning for semantic segmentation. The class prototypes are computed as the moving average of source domain feature representations of the respective class during the training.

The authors of CFContra [87] compute the average feature representations on the source and on the target domain. The target domain centroid is computed by assigning pseudo labels based on the distance of a feature representation to the source domain class centroids. Based on that, the two closest centroids are computed. The authors compute the contrastive loss between each combination of source domain features, target domain features, source domain centroids, and target domain centroids.

The authors of CAM [166] apply prototype clustering on both source and target domains. For each class, a single target domain feature representation is selected to serve as the prototype for the class. This prototype feature is computed by determining the feature representation that has the maximum cosine similarity to all the other feature representations of the same class. The similarity matrix and the entropy minimization are computed similarly to SSS+ST [165]. Distinct from this paper, the authors propose a contrastive clustering loss. This loss takes normalized first-order statistics (mean representation) of each class cluster from the source and target domain and uses the euclidean distance as a distance metric for the clustering of the mean representations.

The authors of OCE [44] apply feature clustering in the source and target domain, aiming to group feature vectors of

the same class together and those of different classes away from each other. Notably, OCE differs in the computation of the cluster centroids and the distance metric compared to the previous approaches. The class centroids are computed based on the current batch, both on the source and target domain. The distance metric that is used to define the similarity is the $L1$ norm. During optimization, the $L1$ norm between the current feature representation is minimized to centroids of the predicted class and maximized to centroids of the other classes. Additionally, they introduce an orthogonality requirement meaning that feature vectors of different classes are forced to be orthogonal in the feature space. The orthogonality requirement is based on the cosine similarity between the current feature representation and the class centroids.

The method introduced in MCSSF [105] is also based on clustering. The authors introduce a dictionary containing the correctly classified feature representations in the source domain is defined. The target domain feature representations of the current batch are stored in a dictionary, also. A cosine similarity matrix is computed between the target and source domain features of the same class. Elements of this matrix representing low similarities are eliminated by thresholding and the cosine similarity of the remaining elements is maximized. Therefore, this approach does not optimize the feature representations of different classes to be dissimilar. This is also the case for LSR [179] and BAPA-Net [74].

Similarly, the authors of LSR [179] apply non-contrastive clustering to prototypes. The prototypes are computed for source and target domain and updated via a moving average. The authors minimize the $L2$ norm of each feature representation to its corresponding class prototype. The correspondence to a class is determined based on the prediction probability. Additionally, they enforce perpendicularity between prototypes of different classes (as in OCE [44]) and the norm of the target and source domain features to be the same. They assume, according to recent research by Xu et al. [180], that target domain feature vectors have a smaller norm. Enforcing the norm to be the same in the target and source domains introduces domain alignment.

SIM [101] is also based on non-contrastive clustering but distinct from the previous approaches the clustering is done differently for stuff classes like road or sky and thing or instance classes like car or pedestrian. For the stuff classes, the authors compute multiple average feature representations per class by averaging the feature representations. For a given target domain centroid, the $L1$ norm is minimized towards the closest source domain centroid of the predicted class. For a given target domain instance centroid, the $L1$ norm is minimized towards the closest source domain instance centroid of the predicted class.

Li et al. [74] (BAPA-Net) assume that near-boundary pixels are hard to classify and propose a special handling of the boundary regions, different from the previous approaches. The authors employ the CutMix [129] operator to paste source pixels and labels to the target domain, artificially creating more boundary pixels that are assigned a higher weight. They employ a prototype clustering algorithm between the source and mixed target domain images. The prototypes of the mixed target domain in the current batch are computed by assigning the feature vectors to the predicted class and filtering out those feature vectors for the centroid computation that are too close to a boundary. The class-wise centroids of the mixed images are optimized to minimize the $L1$ norm to the closest source centroid of the same class.

As we have seen, the above clustering approaches often use the classification of feature representations to determine to which centroid the current (target domain) feature representation should be clustered. Hence a good classification is necessary. Based on this, CaCo [164] shows that existing domain adaptation methods can profit from an additional feature space clustering, given that they provide a good classifier to determine the clustering target centroid.

D. OUTPUT SPACE DOMAIN ADAPTATION

Output space adaptation methods can be formally defined by the distinctive property that the pixel-wise logits or softmax probability outputs $y_{i,s} = P(s|i, \mathbf{x})$ of the network are utilized for the adaptation.

Output space adaptation methods can be subdivided into different subcategories, which are shown in Table 4. The two most popular and commonly employed output space adaptation methods are self-training and adversarial learning, while several other methods have also been utilized for adaptation, such as entropy-based methods and consistency or contrastive learning.

1) SELF-TRAINING

The general idea is to retrain the network on labels that are generated by itself (see Figure 6). In the unsupervised domain adaptation setting, the network $\mathbf{F}(\cdot)$ is trained in the source domain $\mathcal{D}^S$ in a supervised manner as the first step. In the second step, the trained network generates the raw predictions by running inference in the target domain $\mathcal{D}^T$ delivering $\mathbf{y} = \mathbf{F}(x^{\mathrm{T}}; \boldsymbol{\theta})$. Because of the domain shift, $\mathbf{y}$ is noisy and contains wrong labels, so a direct utilization as pseudo-ground truth is not optimal. Instead, methods are required to discriminate between reliable and non-reliable predictions.

For this reason, the distinctive operation of self-training methods is mostly the filter operation $\overline{\mathbf{y}}^{\mathrm{T}_{\mathrm{pseudo}}} = \mathbf{U}(\mathbf{y})$, which removes predictions with low confidence. A small subtaxonomy of self-training methods is given in the second column of Table 4. These methods are particularly often used in hybrid approaches and only rarely as stand-alone adaptation methods (cf. Section III-E).

Some methodological characteristics are shared among self-training methods. At the beginning of UDA research, one of these was a so-called warm-up step [159], [185], [214], where a different adaptation method is employed to obtain an initial adaptation of the network and a better start performance for the pseudo-labels. However, with the rise of hybrid methods, dedicated warm-up steps became obsolete. Also,

TABLE 4. Adaptation methods in the output space. The papers are clustered and sub-clustered according to similar methodology.

Technique		Approach
Self-Training	Global Thresholding	[71], [79], [83], [96], [98], [100], [104], [106], [107], [140], [149], [162], [181], [182], [183]
	Adaptive Training	[50], [52], [53], [54], [66], [67], [94], [101], [104], [105], [109], [115], [131], [139], [142], [160], [166], [167], [184], [185], [186], [187], [188]
	Image-Level Self-Training	[23], [50], [51], [144], [171], [189], [190], [191], [192]
	Entropy-Based	[66], [76], [87], [116], [146], [165], [184], [186], [193], [194], [195], [196], [197], [198]
	Ensemble Learning	[50], [69], [111], [151], [179], [182], [199], [200]
	Discriminator Confidence	[138], [201], [202], [203]
	Others	[68], [77], [116]
Adversarial	Basic	[54], [79], [83], [92], [94], [98], [99], [101], [102], [104], [106], [107], [108], [111], [139], [142], [185], [192], [195], [198], [199], [200], [201], [203], [204], [205], [206], [207], [208]
	Depth	[53], [90], [138], [149], [174], [207]
	Entropy-Based	[114], [146], [194], [195], [197], [209], [210], [211]
	Modified Input	[100], [137]
	Bi-Classifier	[48], [135], [151], [152]
	Intra-Domain	[194], [197], [208]
	Multi-Level	[49], [116]
	Multi Discriminator	[63], [96], [202], [145]
	Class-Wise	[132], [140], [141], [144], [155], [191]
Consistency	Augmentations	[67], [70], [71], [73], [192], [193]
	Style Transfer	[47], [59], [108], [111], [113], [198]
	Knowledge Distillation	[72], [76], [85], [107], [115], [131], [199], [212], [213]
	Others	[99], [100], [148], [171], [188]
Depth-Based		[90], [103], [130], [149], [168], [207]
Contrastive Learning		[75], [81], [159], [162], [163], [167]
Others		[74], [99], [146]

often multiple iterations or stages of self-training are performed to iteratively increase the performance of the pseudo-labels [52], [131], [184], [185].

a: GLOBAL THRESHOLDING

Global maximum softmax thresholding is the simplest method employed by several approaches (see Table 4). These approaches take the softmax probability distribution $P(s|i, \mathbf{x})$ as a pixel-wise confidence estimation of the network and filter out every pixel whose maximum softmax probability is below a certain class-independent threshold, often 0.9 or 0.95.

b: ADAPTIVE TRAINING

Several other researchers propose extended softmax thresholding mechanisms belonging to the adaptive softmax thresholding category. One important motivation for this group of methods is not to treat all classes in the same way but employ different adaptive thresholds to the classes since not all classes have similar output probability distributions due to the domain shift. Class-balanced self-training (CBST) [52] introduced these class-wise thresholds as one of the first works. It combines output normalization and class-specific quantile-guided thresholding. The best p_i percent of the pixels per class i are chosen as a pseudo-label, and p_i is increased over the self-training iterations. Several self-training methods are directly based on CBST, e.g., CVRN [66], MRNet [187], MLSL [50], and CSCL [142]. Other approaches only use the class-wise quantile-guided thresholding as a self-training method, where the top $p\%$ pixels of each class is selected, e.g., CCM [186], CSCL [142], APL [94], PA+CCR [67], and SCDA [167]. Additionally, regularization methods for CBST were proposed. The cross-view regularized network (CVRN) [66] extends the CBST method by an inter-task and inter-style regularized multi-task self-training, which enforces consistency between instance and semantic segmentation and two different styles. Confidence-regularized self-training (CRST) [184] introduces label- and model-regularized self-training. In addition to CBST three different regularization techniques are utilized to penalize over-confident labels to output a more uniform probability distribution. All three regularization methods $L2$, entropy regularization, and KL divergence regularization, achieve similar performance.

Stuff and instance matching (SIM) [101] utilizes a partially adaptive class-wise thresholding by computing the class-wise median of the maximum softmax probabilities across the entire target dataset. The median is used as a threshold if being smaller than a fixed threshold of 0.9.

Contrary to previous works, ProDA [131] proposes not iteratively to change the hard pseudo-labels but to keep them fixed and weigh them instead. The authors show that both stage-wise self-training and the parallel update of network and pseudo-labels (trivial solution) lead to sub-optimal results. This is notably different from popular self-training strategies such as CBST, where a self-paced scheme with an increasing amount of labels is used. The ProDA weights are computed based on prototypical features (see Section III-C), and ProDA outperformed previous self-training approaches by a large margin. Note that also additional methods such as knowledge distillation (see Section III-D4) contribute to the performance increase.

The predictions from past iterations can be complementary utilized to refine the pseudo-labels. IAST [185] extends the

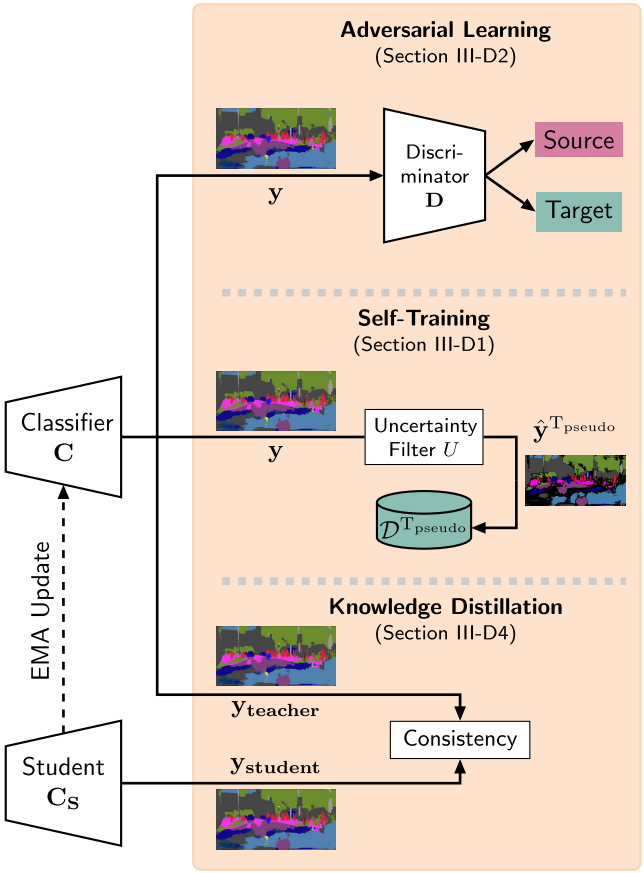


FIGURE 6. **Overview of output space adaptation methods: Simplified and exemplary schematic visualization. Deviations of certain approaches from this basic scheme are possible.**

quantile-guided class-wise thresholding to instance-adaptive thresholding, where the applied threshold is an exponential moving average of the quantile-based image-specific threshold and the history of thresholds from previous samples. In this way, information from both other instances of the dataset and the current instance are considered for thresholding. Similar to that, SSAC [115] utilizes an exponential moving average of class-specific prior probabilities (certain pixel belonging to a certain class) and introduces two hyperparameters so that the threshold for often-occurring classes, such as road, etc. remains unaffected but decreases for rare classes such as traffic light, etc. CLST [159] uses a temporal ensemble by storing predictions from past iterations in a weighted manner. The pseudo-labels are obtained by a majority vote of the stored predictions, and a class weighting based on the class frequencies is also employed.

Next to the pixel-wise self-training methods, a smaller cluster of methods emerged, which can be seen as image-level self-training. It can be distinguished by its characteristic to employ self-training for patch- or image-level predictions. Often approaches of this cluster make use of multi-scale domain alignment to obtain both local and global alignment. A popular representative of this cluster is PyCDA [215].

c: IMAGE-LEVEL SELF-TRAINING

Next to the standard pixel-wise self-training with softmax thresholding, self-training on larger patches with patch-level pseudo-labels is conducted in a curriculum manner. The patch-level labels are obtained by average pooling on the pixel-level labels and thresholding. This method's highest level of abstraction is the prediction of the global label distribution of the entire image. CDA [190] shares significant similarities with PyCDA since it also utilizes the prediction of global image class distribution and superpixel class distribution. In contrast to PyCDA, logistic regression, and a support vector machine are used for the corresponding tasks and no pixel-level pseudo-labels are utilized. The authors argue that estimating global class distributions is easier than pixel-wise pseudo-label prediction. Similarly, pivot interaction transfer (PIT) [171] utilizes multi-nomial logistic regression. It is trained on the source domain with the image-level class distribution to train multiple region expansion units. These units consist of a convolution, and an up-sampling operation and with different smoothing parameters in the aggregation layer afterward, the different units focus on different object sizes. This is combined with a knowledge transfer to the pixel-level source segmentation branch.

WDA [23] simplifies the training task further by training a network with a class-wise binary cross-entropy loss on the image-level to predict whether a class exists in a training sample. This is done by training class-wise classification networks with class-wise features, so each class has its classifier and feature representation. CCDA [144] developed a different method for the same aim. Instead of class-specific classifiers like WDA, this approach utilizes two discriminator branches to predict which classes are present in a particular patch using binary classification loss and an adversarial loss. For the target domain, the existing classes of a patch are predicted using the pixel-wise thresholded pseudo-labels. This coarse patch-level is accompanied by a fine-grained pixel-level discriminator, reaching a multi-scale domain alignment.

d: ENTROPY-BASED

The output probability distribution's entropy is a popular uncertainty estimation tool. It is utilized for several UDA approaches since it provides more information about the network output than the maximum softmax prediction. Just like maximum softmax-based self-training, entropy-based self-training can be easily applied complementary to other UDA methods [110], [194] and often used for adversarial learning (see Section III-D2). One of the first works using entropy information was ADVENT [211]. This approach employs entropy minimization, which can be seen as entropy-based self-training without a thresholding mechanism and, thus, without one-hot encoded labels. FDA [69] followed a similar idea and included a Charbonnier penalty weighting [216] for the entropy minimization, which assigns a higher loss to high entropy regions. Next, the entropy can replace the maximum softmax value as the uncertainty measure and be utilized

analogously with thresholding. Several works are employing this method with different thresholding techniques: Niemeijer et al. [165] use a global quantile-based threshold, ESL [110] and LSE+FL [193] both apply a class-based quantile-like threshold for the entropy taking the most confident $p\%$ pixels as pseudo-labels. CRA [194] employs a manually chosen threshold which is halved for rare classes.

In contrast, other works avoid the usage of thresholds. SEDA [146] does not utilize thresholding but takes the inverse of the entropy values as weights for the unfiltered target pseudo-labels. UncerDA [195] fits a Gaussian mixture model to obtain a class-wise positive and negative distribution which is employed to assess if predictions are used as pseudo-labels.

Two approaches aim to utilize the entropy by developing new loss functions next to the entropy minimization described previously. MSL [182] argues that the entropy minimization is not optimal for rare classes due to a higher gradient for higher probabilities. Consequently, the maximum squares loss is introduced, where the loss is the square of the predicted probabilities, which leads to a linearly increasing gradient and makes both easy and hard classes better transferable since the gradient grows not exponentially for easy classes. PTP [154] also modifies the standard cross-entropy loss so that the loss becomes symmetric, and very high predicted probabilities are penalized for discouraging the network from overfitting. BiMaL [196] introduces a new loss function that is a generalized form of the entropy minimization of ADVENT [211]. The bijective maximum likelihood loss uses a sequence of bijective mapping functions to map the segmentation output to the latent space, where the log-likelihood loss is computed. This approach should better capture image-level characteristics than entropy minimization since not every pixel is treated individually.

e: ENSEMBLE LEARNING

Ensemble learning is a commonly applied method for various applications, e.g., uncertainty quantification [217]. Ensemble learning refers to a group of methods where two or more predictions of different DNNs or different (segmentation) heads are included to obtain the final prediction. SAC [111] proposes a classic ensemble learning method by training two segmentation networks with correspondingly translated source and target images. The predictions on the target image are averaged and filtered by a softmax threshold to obtain the pseudo-labels. The MRNet [199] method is different by only using a second classifier head which takes features from a different layer for weighted prediction summation. EPS-UDA [200] employs three semantic segmentation heads with a shared encoder but differs from other ensemble learning methods in two ways. First, each head is trained with the outputs from the two other branches. The valid pseudo-labels are only assigned when both outputs agree, making this the only method where this hard constraint is employed instead of, e.g., averaging. Second, the agreement between the heads is measured using KL divergence and then used as a weighting factor.

For UDA approaches, ensemble learning is often closely connected to the group of multi-inference self-training methods, where different versions of one image are fed into the network, and multiple predictions are combined to get a more robust final prediction. These methods can be seen as an extension of the standard ensemble learning methods with Monte Carlo uncertainty estimation. In FDA [69], three independently trained networks are utilized, and each network receives differently style-transferred input images. The predictions of all three networks are averaged, and an argmax operation obtains the prediction. SIT [114]] follows a similar scheme. However, instead, it uses a stochastic style transfer (see Section III-B1.d) to vary the style of the translated images and trains a triplet of networks with a large style variety of the translated input images. One of these networks is trained with ten different style transfers per image, which can be seen as a Monte-Carlo-like uncertainty reduction by averaging over the predictions. The outputs are averaged across the three networks, and class-balanced self-training is applied to further filter the pseudo-labels. DPL [106] employs a similar method having a target and a source network. With both target images and source style translated target images, for the source network, two different predictions are obtained, averaged like in FDA, and the known maximum softmax thresholding is applied.

A crucial design choice for ensemble learning architectures is the number of different networks or network heads. STAR [218] introduced an alternative to a fixed number of classifiers. Instead, the authors propose to model a distribution of classifiers as a multivariate Gaussian and randomly sample the model weights from this learned distribution. During training, the weights themselves are not optimized, but the distribution parameters from which the weights are then sampled. In practice, the method employs two different classifiers, which should lead to similar results as training with an infinite number of classifiers. It can therefore be seen as a stochastic variant of CLAN [135].

Most methods do not explicitly estimate or utilize the source and target domain shift. However, only a few works aim to utilize the domain shift estimation for self-training. The authors of CorDA [77] argue that depth prediction can be used as a proxy for the actual domain shift estimation. The target images are processed through a source and a target depth prediction network, and the difference between these predictions is computed. If the depth prediction difference is high, the according pixel gets a low weight in the self-training loss assigned, and vice versa.

f: DISCRIMINATOR CONFIDENCE

Using discriminator confidence is a less popular self-training method and is only possible adjacent to an adversarial learning framework. The underlying assumption of these works is that those pixels where the discriminator has high confidence are also good pseudo-labels. AL+ST [202] exploits

a pixel-wise domain discriminator to use these outputs as a confidence estimate for the target predictions. The thresholding of the discriminator confidence values is the same as the class-wise quantile-based softmax thresholding. MADA [203] applies a very similar principle for its two pixel-wise discriminator confidence maps but combines it with the softmax probabilities, so only pixels where both confidence estimates are above the threshold get qualified for the pseudo-label.

DMLC [219] is significantly distinct from the other self-training methods since it utilizes meta-learning for UDA. It aims to correct wrong pseudo-labels with a noise transition matrix (NTM). This matrix contains class transition probabilities and is jointly optimized with the segmentation network in a three-step scheme that includes a meta-optimization step. In this step, the source data with a domain predictor is used to serve as a so-called metadata set, estimate the network's generalization, and update the NTM accordingly.

2) ADVERSARIAL OUTPUT ADAPTATION

Adversarial output adaptation is one of the most often applied methods in UDA research.

a: BASIC

It introduces the basic idea of attaching a discriminator after the segmentation probability output and trains it to predict if a certain prediction sample originates from a source or target input on the image-level. A simple visualization of this idea is shown in Figure 6 (top). By backpropagating this adversarial loss, the segmentation networks should output similar segmentation distributions for both the source and target domain since the segmentation network will try to fool the discriminator with similar outputs. The target predictions will become more similar to the source, and the network gets adapted to the target domain. This principle also works for two different discriminators as proposed by AdaptSegNet. One receives the standard high-level softmax output, and the second discriminator ensures a low-level adaptation by getting segmentation predictions only based on lower-level features. Several other works employ adversarial output adaptation as proposed by AdaptSegNet in addition to other methods [63], [79], [92], [101], [102], [206]. Some approaches employ a reduced adversarial learning method by leaving out the utilization of low-level features [98], [142]. A minor change to the original adversarial learning method is to replace the source input with a style-transferred source image whose style is more similar to the target domain [83], [94], [107]. Similarly, DPL [106] uses two adversarial losses. One loss for translated source and real target images and the second for translated target and real source images. Another minor change to the original scheme is proposed in UncerDA [195]. A sophisticated sampling strategy for the source images in the adversarial adaptation process is proposed to show rare or hard classes according to their entropy uncertainty. Classes with high uncertainty are shown more often and vice versa.

However, there are a lot of extensions and improvements of this original adversarial method which can be clustered as shown in Table 4. A straightforward extension is multi-level adversarial learning, which distinctive characteristic of AdaptSegNet is the utilization of features in multiple different network layers for the adversarial adaptation. Therefore, these works closely correlate to approaches operating in the feature space. SASP [49] applies two types of adversarial adaptation. Next to the known output adversarial learning, it concatenates multiple latent layers before sending the concatenated result to a classification layer and applying the adversarial loss. The authors reason that also the earlier layers receive a strong learning signal from this multi-layer fusion. MLAN [116] also proposes multi-level adversarial adaptation but has a different argumentation and approach. In global alignment, no local distributions can be adapted, so MLAN introduces region-level adversarial learning where relations between small patches are utilized to reach fine-grained region-level adaptation. In addition to the connection between local and global alignment, consistency maps on multiple levels are calculated.

b: MULTI-DISCRIMINATOR

AdaptSegNet employs two discriminators for different levels but with the same objective. A straightforward extension is two discriminators with different objectives, as proposed by MDD [202]. Here a second discriminator is trained to distinguish between the source predictions and the source ground truth. CADA [96] trains three different discriminators. They all align the source and target domain. However, since two discriminators receive their prediction based on the output of a feature attention mechanism, CADA can also be seen in the group of multi-discriminator approaches. Next to the standard discriminator, MAGD [63] proposes a gated discriminator next to the standard discriminator that additionally takes foreground-background segmentation masks as the input since both areas have different adaptation difficulties. The gated convolutions make it possible to utilize the masks without hard thresholding, and the additional input for the discriminator makes distribution alignment easier for fore- and background.

The basic adversarial output adaptation does not distinguish between classes, so a group of works addresses class-dependent adversarial learning. CCDA [144] introduces two extensions to the adversarial adaptation from AdaptSegNet [204]. First, it has one coarse- and one fine-grained branch enforcing adaptation at different levels of granularity.

c: CLASS-WISE ADVERSARIAL LEARNING

Second, a special focus lies on class-dependent adaptation, and class-conditional loss functions are used. The adversarial part of the coarse branch is trained to distinguish the image-level class predictions of the two domains. The pixel-wise predictions of the fine-grained branch to discriminate between source and target domain are computed by a class-conditional binary cross-entropy loss.

SSF-DAN [132] computes class-wise semantic features by separate convolutional layers for each class and calculates the sum across all classes before computing the loss. SSF-DAN introduces a pixel-wise weighting based on the softmax prediction confidences to let the adversarial loss focus more on difficult classes. CGDA [141] utilizes a so-called cross-domain grouping network that performs clustering of the segmentation output to better align the particular classes. The discriminator receives an input conditioned on the learned sub-clusters of the grouping network and, therefore, can better learn to distinguish classes. FADA [140] proposes a significantly different approach. Instead of only binary domain labels, the discriminator is trained to output the entire softmax probability prediction from the segmentation network. This additional knowledge is expected to enable a better class-adaptive alignment since the same network predicts class distribution and domain labels. CLS [191] employs a similar method and proposes a shared classifier and discriminator for better conditional alignment. The shared decoder has one more output class in addition to the predefined semantic classes to support the discriminator. CAA-Net [155] follows a similar idea but applies class-wise masks to the prediction maps so that the discriminator predicts a domain label for each separate class map.

d: ENTROPY-BASED

Several approaches use entropy to replace the softmax predictions for adversarial training. ADVENT [211] was one of the early works training the discriminator to distinguish between domains based on the entropy maps of source and target output probability maps. SCDS [209] slightly modifies this method by using random patches for adversarial training since they argue that the distribution of an entire image differs too much. Some works simply integrate ADVENT into their framework as an additional method [114], [210]. IntraDA [197] employs the same method and uses entropy to split the target domain into hard and easy samples for an adversarial intra-domain adaptation. SEDA [146] extends this approach and introduces a weighting factor for the adversarial loss based on the entropy of the images so that hard samples with a high entropy can have a higher impact on the loss. CRA [194] generates binary labels for confident and non-confident regions based on entropy thresholding and trains a discriminator to distinguish between these regions within the target domain. In UncerDA [195], the entropy is used to resample the input of the adversarial learning according to the class-wise uncertainties on the target domain.

Next to the entropy, the basic softmax probability input for the discriminators can also be replaced with other inputs aiming for more meaningful alignment. ASANet [100] introduces affinity maps for adversarial adaptation. These maps are computed by also taking the predictions of adjacent pixels into account and therefore enforcing the discriminator to focus on the structural properties of the domains. As the only work for semantic segmentation, APODA [198] uses an adversarial attack to compute adversarial features and to attack both the discriminator and the classifier. It trains the attacked discriminator to output the same domain label for both the clean and the perturbed prediction. DRP [137] combines adversarial learning and patch-wise clustering. It trains an additional network on the source domain to cluster output patches and uses adversarial learning for alignment with the source domain. MLAN [116] is mostly similar to that but uses DBSCAN clustering [220] instead of training a separate network for clustering.

Adversarial learning is usually employed between the source and target domain for aligning the two different domain distributions. However, a group of works aims to obtain intra-domain alignment with adversarial learning, which means alignment between different distributions within one of the domains. While this is technically no domain adaptation, this group of methods helps to learn target representations. IntraDA [197] combines both inter- and intra-adaptation. First, the known entropy-based adversarial alignment is applied, and second, the image-level entropy is used to divide the target domain into easy and difficult samples for adversarial training. PixIntraDA [208] extends IntraDA to the pixel-level since they argue that the image-level does not capture the local differences of the prediction. However, no entropy is involved there, but simple softmax thresholding is used to distinguish between easy and hard samples. CRA [194] can be seen as a combination of IntraDA and PixIntraDA since it combines entropy with pixel-level adversarial learning. It also classifies so-called trusted and untrusted regions of the target domain by their corresponding entropy and trains a discriminator in an adversarial manner to align both regions. This method should help to transfer the knowledge from the well-segmented parts of the target domain to the less-confident areas.

CSCL [142] focuses on separating transferable knowledge as one of only a few works since the authors argue that global adversarial adaptation may cause misleading knowledge transfer. For that purpose, a transferability quantizer and critic are introduced to distinguish between these different types of knowledge. The critic is necessary to inform the quantizer where the transferability estimates are not accurate. The critic is trained using a reward based on how good the segmentation predictions are.

Depth information as an additional modality is popular in UDA research and adversarial output adaptation several approaches make use of it. GGIO [90] is one of the simplest combinations of depth and segmentation information. It concatenates the two predictions and provides this as the input to the discriminator. DADA [207] includes the depth modality in a more sophisticated way. It fuses the segmentation entropy with the depth prediction as the element-wise product and forwards the fused results to the discriminator. The discriminator is trained to distinguish between the source and the target domain. The authors argue that depth and geometric information are similar between the domains and, therefore, beneficial for better domain alignment. CTRL [149] follows a similar idea but on task-level. Depth prediction is

incorporated here to improve the adaptation. A dedicated cross-task relation layer is employed, where the entropy of the semantic prediction, the semantics refinement head prediction and the depth prediction are concatenated and forwarded to the domain discriminator. The authors reason that an adversarial adaptation on both semantic and depth entropy maps makes the adaptation easier to difficult to transfer classes.

e: BI-CLASSIFIER

Maximum classifier discrepancy (MCD) [48], SWD [152], and BCDM [151] belong to a group of adversarial methods that are use two different classifiers instead of or in addition to a discriminator. These methods obtain an adversarial alignment in the feature space, so a detailed description is provided in Section III-C. However, there is a mutual connection to the output space alignment because the output discrepancy between their two classifiers is iteratively maximized to construct the adversarial learning process. The bi-classifier method proposed in CLAN [135] differs from the previous settings. First, an additional discriminator is employed after the two classifiers, receiving the summed output, and the cosine similarity of the classifiers weights the adversarial loss. The cosine similarity is used as the discrepancy loss between the two classifiers. No alternating optimization with the same loss function as in the three other works is applied [48], [151], [152].

The growing research area of knowledge distillation/transfer is combined with adversarial domain adaptation in the SE-GAN approach [107]. A student-teacher network architecture with an exponential moving average update for the student replaces the standard generator to stabilize the adversarial training.

3) CONTRASTIVE OUTPUT ADAPTATION

Contrastive learning is mainly applied directly in the feature space as described in Section III-C, but some approaches exploit it for the output space. The basic principle here is the same: the network is trained to output similar representations for similar inputs or classes and vice versa.

The approaches PWCL [162], CLST [159], SDCA [167], RCCR [75], and UCDA [81] all have in common that the contrastive adaptation operates in the feature space, and a detailed description is provided in III-C2.b. However, to compute positive and negative pairs, all access the output space to obtain the pseudo-labels, which directly correlates to output space alignment since reliable pseudo-labels are important for the adaptation process.

This also applies to PLCA [163]but it is the only work that conducts multi-level contrastive learning on both the feature maps and the semantic predictions. For the latter one in the output space, the authors choose a different metric to compute the similarity and their positive pairs between source and target prediction, namely the Kullback-Leibler divergence.

4) CONSISTENCY OUTPUT ADAPTATION

The idea of consistency output adaptation is to enforce two or more different network outputs to be similar using a dedicated loss function. In UDA research, several approaches employ consistency learning in the output space.

RPT [148] proposes an entire consistency framework combining three different levels of consistency. For patch-wise consistency, superpixels are computed, and all pixels within these superpixels are enforced to have the same predicted class. A similar strategy is conducted for cluster-wise consistency, where the superpixels are grouped into clusters and enforced to have the same predicted majority-voted class. On top, an LSTM is used to enforce source and target to have a similar spatial structure.

a: AUGMENTATIONS

It is a widely adopted method in consistency learning to use two or more different style versions of the same image and enforce the network to predict the same outputs since the semantic content, i.e., the classes, are the same. Generally, one can distinguish two ways to generate different versions of the same image: rule-based like image augmentations and learnable such as GANs and CycleGANs. SUDA [70] creates two different spectral views of the same target images and applies an $L1$ consistency loss to obtain similar predictions. SVmin [192] utilizes the same loss for scale-invariance. The target images are downscaled and enforced to be similar to the original resolution's prediction. LSE+FL [193] applies the same but patch-wise with a cross-entropy loss for consistency. A popular and simple way are image augmentations which can severely change image characteristics such as sharpness, contrast, hue, etc. Similarly, PA+CCR [67] augments the target images with color jitter and enforces the prediction to be similar to the clean prediction. The standard cross-entropy loss can be used since the clean prediction is treated as a one-hot encoded pseudo-label that the augmented prediction has to match. PixMatch [73] applies more sophisticated and multiple different augmentations, including the discrete Fourier transformation. The consistency loss (cf. PA+CCR) is applied, making consistency learning and self-training very similar in this setting. DACS [71] introduces cross-domain image-level mixing and blurs the distinctive boundaries between consistency learning and self-training. The training can be understood as both self-training on mixed labels and consistency learning to predict the same classes independently from added source content in the target image.

b: STYLE TRANSFER

Another line of work uses a GAN or CycleGAN to obtain different image styles. SUIT [113] employs a GAN to transfer source images to the target style and then enforces consistency between style-transferred and real source images by cross-entropy loss. SAC [111] follows a similar idea but trains two distinct networks and enforces consistency using an $L2$ loss. CrDoCo [47] is similar to this approach but uses a

CycleGAN and two domain-specific networks to enforce the output prediction consistency with a bi-directional KL divergence. MSS [59] follows a similar approach but applies the consistency loss for both the source and target domain and utilizes the cross-entropy as the consistency loss. APODA [198] employs a more sophisticated technique since the features are perturbed with an adversarial attack, and an $L2$ loss enforces the prediction of both the clean and the perturbed maps to be the same.

c: KNOWLEDGE DISTILLATION

A popular and straightforward application of consistency learning is knowledge distillation, where the knowledge should be transferred from a teacher network to a student network. SEAN [213] proposes a typical UDA knowledge transfer framework. After being augmented, the target images are processed by both a student and a teacher network, and an $L2$ consistency loss enforces the two different target predictions to be similar. SE-GAN [107], TGCF-DA [85], and BiSMAP [68] (with KL divergence consistency loss) follow very similar methods. Augmentations and teacher-student learning are expected to make consistency enforcement more effective in this setting. UACR [82] extends this basic idea with an uncertainty module and a second consistency loss. Two uncertainty-weighted mean squared error losses (MSE) are applied as the consistency loss to enforce student and teacher to generate similar predictions. At the same time, a class-wise mask is used to enforce consistency between perturbed and non-perturbed images. Notably, these losses are the only adaptation losses applied in this approach. MRNet [199] is distinct from the other works since the authors argue that a second additional classifier with a shared encoder can also act as a teacher and regularize the main model; the KL divergence loss is used to obtain output consistency.

CAMix [76] uses the method from DACS and extends it by knowledge transfer and a so-called significance mask. This is computed based on the entropy of the target prediction and a contextual mask using spatial similarities between the source and target domain. The original domain mixture idea from DACS is further extended by DSP [72]. It pastes domain-specific content in both directions, so source and target domain images are modified with content from the other domain. The cross-entropy loss is a combination to enforce both the source and the target content-based predictions to be consistent with the corresponding unmixed labels. The clean predictions are obtained from the teacher, and the mixed prediction from the student model, so it also enforces consistency between these two models. SAC [115] relies on strong image augmentations for the inputs for a momentum network that is updated as a moving average of the student network. In contrast to UACR, a single focal loss enforces the consistency between the momentum and segmentation network. Similar to UACR, predictions from multiple crops are averaged to obtain more confident pseudo-labels. BiSIDA [86] combines knowledge distillation and style transfer. It processes the original target image and several different style transferred images of that original image through the network. The style transfer predictions are averaged and utilized as the pseudo-label in the consistency loss.

ProDA [131] introduces two novel extensions worth mentioning. First, it initializes the student network with weights from a self-supervised pre-training on ImageNet, which provides a strong bias towards diverse real-data representations. Second, it performs multiple iterations of distillation, providing further performance improvement.

MFA [212] proposes the probably most complex UDA consistency framework by combining two knowledge distillation units each consisting of a teacher and a student, resulting in four networks overall. However, the consistency mechanism, embedded into the larger hybrid framework, is similar to other works enforcing the student and teacher networks prediction to be similar by optimizing for an $L1$ loss.

PIT [171] introduces a relaxed consistency loss between the fine- and coarse-grained network branches. The known $L2$ loss enforces the class activation maps of both branches to be similar. However, learnable weights for the coarse branch allow some adaptation between the branches and, therefore, relaxation of the consistency condition. CDGA [99] shows that consistent adaptation can also be conducted on the class distributions predicted from an additional network, which are enforced to be similar according to an $L2$ loss. In contrast to the other works, ASAnet [100] enforces local region-wise affinity consistency within the same image for both the source and target domain. The goal is to obtain the same predicted class for all pixels in a certain region except at the semantic boundaries.

SAM [188] is the only work combining consistency and a self-attention learning mechanism. A self-attention module receives the segmentation output, and an $L1$ loss then enforces the predicted output to be similar to two self-attention maps. This should improve adaptation since the attention maps enforce a focus on inter-pixel correlations.

5) DEPTH-BASED OUTPUT ADAPTATION

It is a straightforward idea to enrich the domain adaptation process with additional or surrogate information to simplify the adaptation process. A dominant modality is depth information because of its close relation to the actual semantic segmentation map and because it is possible to obtain ground truth without human labeling effort. Since the output of the depth estimation is mostly utilized, depth-enriched adaptation can be seen as another category of output space adaptation. Depth-based adversarial output adaptation methods were already described in Section III-D2.

SPIGAN [103] proposes a framework that may utilize multiple kinds of additional information from the source domain but evaluates on depth data. It trains a second decoder (with a shared encoder) network for depth estimation in the source domain with an $L1$ loss. GUDA [168] builds upon a similar architecture as SPIGAN, but extends it with new components. Next to the depth estimation, the additional prediction of

depth surface normals serves as a regularization for the depth prediction task. More importantly, domain adaptation may benefit in two ways from the depth information. First, via the shared encoder, which additionally learns depth prediction for the source domain. Second, via an image synthesis task, where both the target depth prediction and the previous frames are required to predict the target image.

DBST [130] is the only approach that incorporates self-supervised depth estimation on the target domain to obtain depth labels for this domain, which is different from CorDA, where the depth is not used as a label in the target domain. DBST contains two separable units that rely on depth information. The first unit trains one network on the depth labels in both domains and a second network on the segmentation labels of the source domain. A transfer network then predicts the semantic output from the depth network so that the depth knowledge of the target is utilized for the segmentation task. The second unit can be seen as a depth-guided version of DACS [71]. The depth information is leveraged to mix source and target content in a more meaningful way and to generate a more diverse dataset for self-training.

6) OTHER METHODS

Next to the already described large methodological cluster with many different proposed methods, there are a few works in the line of UDA output space adaptation that cannot be assigned to any previous clusters.

Clustering-based adaptation methods are often applied in the feature space, as described in Section III-C. However, CDGA [99] attaches an additional clustering network of two convolutional layers directly after the semantic prediction output. The clustering network is trained with two different losses. One loss enforces class distribution clustering consistency between source, and target and the second loss minimizes a cosine similarity across the predicted clustered class distributions. The original class distribution will be clustered into a fixed number of sub-clusters strengthening the inter-class adaptation.

The accurate segmentation along the boundaries between objects is still a challenge for segmentation in general. Most UDA works ignore the particular adaptation of object boundaries, but two approaches specifically aim to utilize the boundaries for adaptation purposes.

BAPA-Net [74] builds upon DACS [71] and uses semantic boundaries in two ways. First, it weighs the standard cross-entropy loss by the distance of each pixel to the mixed boundary of the source and target mixed image. This weighting should enforce the network to focus on the domain mixed boundaries. Second, the opposite strategy is applied for prototype alignment in the feature space, and the mixed boundary pixels are excluded to not confuse the prototype alignment. In contrast, SEDA [146] proposes an entire semantic boundary prediction framework. A second network branch is trained to predict the semantic boundaries in the source domain, and a feature-level adversarial loss helps to obtain accurate semantic boundary predictions in the target domain.

TABLE 5. Hybrid adaptation approaches employing techniques in multiple adaptation spaces. The papers are clustered and sub-clustered according to the employed adapation spaces.

Adaptation Space			Approach
Input	Feature	Output	
✓	✓		[31], [45], [84], [89], [95], [161]
✓		✓	[63], [66], [69], [70], [71], [76], [79], [82], [83], [85], [86], [90], [92], [93], [94], [98], [99], [100], [102], [103], [106], [107], [110], [111], [114], [116], [137], [154], [155], [168], [188] [208]
	✓	✓	[43], [53], [54] [81], [116], [131], [139], [141], [142], [146], [148], [159], [160], [163], [164], [165], [166], [167], [170], [174], [183], [189], [191], [194], [198], [199], [200], [203], [204], [206], [207], [210], [213], [214], [221]
✓	✓	✓	[23], [47], [59], [67], [68], [72], [74], [75], [77], [78], [96], [99], [101], [104], [105], [108], [109], [112], [115], [130], [162], [212]

An $L1$ consistency loss between the predicted boundaries of the second network and the actual boundaries of the predicted semantic segmentation map makes it possible to transfer the knowledge of the boundary branch to the actual segmentation branch.

E. HYBRID DOMAIN ADAPTATION

It became evident early in the research that methods of the different adaptation spaces can be combined to increase performance. A large group of research works has emerged from this idea, and we refer to these approaches as *hybrid* domain adaptation approaches. The complexity of different approaches and ways to combine techniques is large. Therefore, we provide a two-leveled grouping to ease the overview. The first-level grouping is done according to the variations of how the different spaces can be combined so that we obtain four different fields, as shown in Table 5.

For the second level grouping, we introduce the terms mutually independent and mutually dependent approaches. Mutually independent describes approaches where the different methods are combined independently so that the approach would still work without one of the spaces. That, in turn, means that the methods from the different spaces do not directly rely on each other w.r.t. the information flow. A simple example would be, e.g., a style transfer method with multiple loss functions for input space alignment. To increase output alignment, softmax-based self-training can be ''attached'' so that both techniques build a framework but are still independent. Mutually dependent approaches combine techniques that closely interact with each other and are directly dependent on the other space, e.g., style transfer provides the input for output consistency learning.

The advantage of hybrid methods is that the performance increase in the target domain is, in most cases, significant. A detailed analysis of the performance capabilities of the hybrid approaches will follow in Section V. A critical analysis of the limitations and disadvantages of hybrid approaches follows in Section VI as part of our meta-analysis.

1) INPUT AND FEATURE SPACE ADAPTATION

In this section we will discuss approaches that combine input and feature space adaptation techniques. We will start with with mutually independent followed by mutually dependent approaches.

a: MUTUALLY INDEPENDENT APPROACHES

CyCADA [95] is one of the most popular approaches that combines input and feature-level techniques in a mutually independent manner. Next to a style transfer with a CycleGAN, adversarial learning on the feature-level is applied. The approach DLOW [31] works in the same way and only extends the style transfer by a domainness factor for higher style diversity. Closely related to that, GAM [143] utilizes CycleGAN-transferred images for pre-training and independently applies deep activation matching afterward. Likewise, the idea of DACL [161] is similar but applies contrastive learning in the feature space.

b: MUTUALLY DEPENDENT APPROACHES

LWC [45] combines input-level and feature-level adversarial learning within one framework. However, both techniques interact, and the feature-level adversarial learning is enabled by the input style transfer forming a mutually dependent approach.

ASM [84] is different because it utilizes an autoencoder-based style transfer to generate mini-batches with different stylized versions of the same image. This is necessary to enforce feature consistency across the mini-batch.

2) INPUT AND OUTPUT SPACE ADAPTATION

In this section we will discuss approaches that combine input and output space adaptation techniques. Again, we will start with with mutually independent followed by mutually dependent approaches.

a: MUTUALLY INDEPENDENT APPROACHES

APL [94] and DISE [92] are exemplary approaches for this sub-cluster with a focus on input space adaptation. APL consists of an input-level image reconstruction adaptation along with self-training. DISE employs a complex input adaptation module in combination with output space adversarial learning. Similarly, LTIR [79] first aims to learn texture-agnostic representations by both domain-randomized and translated images, followed by the second stage with adversarial learning and self-training. Unlike these approaches, PCEDA [93] focuses on input and output adaptation by Fourier phase consistent style transfer and an additional network to encode the source segmentation priors in the output space.

A large group of approaches focuses on output space adaptation and where the input space adaptation is added as an independent sub-component. The three methods PixIntraDA [208], MAGD [63], and MLAN [116] have in common that they focus on output-level adversarial learning but additionally utilize a Cycle-GAN-based style transfer to increase the performance further. ASANet+ [100] focuses on output space structure learning but includes a style transfer to show the orthogonality of their method. In contrast to the other approaches, SPIGAN [103] and GUDA [168] include depth information in their adaptation methods, and both conduct image-level alignment. Additionally, SPIGAN attaches an adversarial-based technique to their multi-task depth and segmentation network. Different from that, GUDA combines depth prediction with a view synthesis module.

PTP [154] and CAA-Net [155] are distinct from the other approaches by combining image reconstruction techniques with output space methods. PTP is special since it utilizes the so-called conservative loss in the output space.

b: MUTUALLY DEPENDENT APPROACHES

The mutually dependent combination of style transfer and self-training closely relates to ensemble-like learning. FDA [69], SAC [111], and SIT [114] all share the same hybrid idea of generating multiple versions of the same image using style transfer and training multiple networks to obtain the pseudo labels. DPL [106] employs two networks to process images in both translation directions. All three methods, style transfer, adversarial learning, and self-training, are applied for both translation streams. This group of approaches obtains a better-aligned input space and directly utilizes that to increase the confidence of the pseudo labels for output space alignment. In contrast to these approaches, the hybrid idea of DACS [71] is more straightforward because it computes pseudo labels only based on mixed images from both domains. CVRN [66] and SUDA [70] both differ from the other approaches since they focus on consistency between different styles. CVRN combines inter-style and inter-task regularization loss, and SUDA combines input adversarial learning with a consistency loss for the different stylized image versions.

Several other methods integrate style transfer, self-training (or a different output space adaptation method), and adversarial output learning into adaptation frameworks. SA-ITI [83] combines these three methods, while BDL [98] has to be highlighted because they propose a framework that utilizes more interaction between the two spaces. The learned segmentation model is utilized for the perceptual loss of the translated images. The framework uses an iterative interaction between input and output space next to self-training and adversarial learning.

Another combination of input and output adaptation as a mutually dependent framework is knowledge distillation with a teacher and student network. TGCF-DA [85] proposes an exemplary framework where the source images are translated to a target-like style and used as input for the student network.

UACR [82] and CAMix [76] follow a similar scheme, but CAMix inputs domain-mixed images to the student network instead of a style transfer. In contrast, BiSIDA [86] employs style transfer in both directions; therefore, student and teacher networks receive images from both domains with a shared style.

3) FEATURE AND OUTPUT SPACE ADAPTATION

In this section we will discuss approaches that combine feature and output space adaptation techniques. Again, we will start with with mutually independent followed by mutually dependent approaches.

a: MUTUALLY INDEPENDENT APPROACHES

AdaptSegNet [204], SEDA [146], MLAN [116], CGDA [141], and CrCDA [189] all utilize the adversarial learning for distribution alignment in the output and the feature space. Similarly, CLS [191] and DAST [139] combine the adversarial alignment of distributions in the feature space with a self-training method in the output space. This cluster contains self-supervised learning techniques introduced in Section III-C and self-training approaches described in Section III-D. The approaches SSS+ST [165] and SePiCo [183] apply contrastive clustering as described in III-C and self-training in the output space. SWLS [53] falls in a similar category, but they utilize an adversarial loss for output space alignment. A common strategy for mutually independent feature and output space alignment is the utilization of feature-level adversarial learning [204] in addition to another output technique. Several approaches follow this idea. RPT [148] proposes output patch consistency. SSDA [170] combines adversarial learning with self-supervised pretext tasks. JAL [206] adds a weight transfer, while CRA [194] is proposed as an additional technique to any UDA method and can be combined with adversarial learning. VAE-UDA [54] applies an autoencoder-based output space alignment and adversarial alignment. PFR [174] and SRDC [210] are slightly distinct from these works because they utilize output adversarial learning in combination with style minimization and feature clustering, respectively. The authors of SEAN [213] instead combine a self-attention mechanism in the feature space with an output consistency loss.

The approaches DADA [207] and CTRL [149] apply an implicit distribution alignment in the feature space by training depth regression on the source and target domain and an adversarial alignment of the distributions in the output space. Similarly, GUDA [168] and CorDA [77] utilize depth regression as self-supervised training but use self-training in the output space.

b: MUTUALLY DEPENDENT APPROACHES

The approaches MCD [48], SWD [152], and BCDM [151], which utilize the maximum classifier discrepancy, fall into this category and have a very close and crucial interaction between feature and output space. Their three-step iterative adversarial learning scheme (see Section III-C) works because the feature extractor and two classifier heads are updated alternatingly, so that feature- and output-alignment directly support each other.

A crucial challenge for contrastive learning is the definition of semantically meaningful positive and negative pairs. Often class information is accessed to guide the selection of positive and negative pairs, which gives a close mutual dependence between feature and output space. Different approaches such as ProDA [131], CLST [159], SPCL [160], and EPS-UDA [200] follow this principle. The actual adaptation happens in the feature space, but reliable pseudo labels in the target domain are required, so both spaces are strongly dependent. The additional application of self-training is widespread.

Similar to this principle, another group of mutually dependent approaches directly utilizes the feature prototypes or anchors for assigning pseudo labels. This provides a strong dependency of both spaces since the quality of pseudo labels directly relies on the extracted prototypes. SCDA [167] is an exemplary work for this, and also UCDA [81], CAM [166], and CAG [214] utilize this idea.

The approach presented in MADA [203] presents an example of mutual dependency between feature and output space. The authors apply adversarial training at low-level and high-level feature maps combined with self-training based on the classifier and discriminator confidences. CSCL [142] utilizes a more complex mutual interaction. Next to self-training and adversarial learning, a critic function aims to distinguish between domain-specific and domain-invariant knowledge and closely interacts in the feature- and output space.

4) INPUT, FEATURE AND OUTPUT SPACE ADAPTATION

In this section we will discuss approaches that combine techniques from all three adapation spaces (input, feature, and output space). Again, we will start with with mutually independent followed by mutually dependent approaches.

a: MUTUALLY DEPENDENT ADAPTATION

A notable pattern of mutually dependent adaptation is the utilization of input space based domain mixing, i.e. content from source images is pasted to target images and/or vice versa. All three approaches RCCR [75], DAP [78] and BAPA-Net [74] are building upon this mechanism that was initially proposed by DACS; DISE-CT [104] also employs source and target domain mixture. RCCR closely connects the three spaces by processing the mixed images with a student-teacher framework and a consistency loss in the output space. The latent features of the student and teacher network are used for contrastive learning. BAPA-Net [74] instead uses the domain mixture to enforce the boundary consistency on feature and pseudo label-level. DAP [78] has to be highlighted in this context since it introduces a novel extension at the intersection of feature and output space also including input space alignment. As the only currently known UDA approach for semantic segmentation it introduces another modality by

using word2vec embeddings [222] as domain invariant priors and projects them together with the mixed semantic output to enforce similarity between the priors and actual network features.

Another group of mutually dependent approaches is formed by CrDoCo [47] and MSS [59]. They both utilize feature-level adversarial learning and then connect input and output space by applying style transfer to compute consistency loss between the predictions of the different stylized images. DSP [72] and SSAC [115] are very similar because they connect input and output space utilizing a teacher-student framework. For DSP, the student network receives domain mixed images, and a weighted CE-loss is computed for both source and target mixed images in the output space. Independently from this, a local and global MMD loss is applied in the feature space. Similarly, SSAC applies augmentations to obtain different versions of the same image. Additionally, target BatchNorm adaptation is conducted independently in the feature space.

Unlike from previously described works, another familiar pattern of mutually independent approaches is the close interaction between feature and output space. A popular representative of this idea is SIM [101]. Feature and output space are closely connected to compute class-wise feature representations in the latent space and minimize the distance between source and target features. Independently from that input space adaptation following BDL and adversarial feature adaptation is applied. DCAA [112] also has an independent input adaptation module. However, the attention-based feature adaptation and self-training output adaptation interact by using the attention weights for the pseudo labels and an attention discriminator. BiSMAP [68] instead introduces a novel utilization of the three adaptation spaces. First, a gaussian mixture model in the feature space is used to assign the pseudo labels, which are used to train a student-teacher framework along with a consistency loss.

Distinct from the previous works, KATPAN [109] employs three mostly independent domain adaptation modules in input, feature, and output space. The feature adaptation module has a connection to the style transfer module making, KATPAN a mutually dependent approach. The feature transferability information is used to weigh the style transfer bottleneck and improve the input-level transfer of well-transferable regions.

b: MUTUALLY INDEPENDENT ADAPTATION

Some works mainly vary in feature space adaptation methods among these approaches. An exemplary approach for this is CIR [108], where the style transfer with a CycleGAN and adversarial discriminator acts independently from the attention mechanism in the feature space and output-level self-training. CADA [96] is very similar to that in the input and output space; only the feature-level channel and spatial-wise attention mechanism are different. The same applies to MCSSF [105], which employs standard input and output space methods but uses a cosine similarity-based feature centroid alignment. WDA [23] shares the same underlying idea with slightly different modules. It combines an attention mechanism in the feature space with class-wise discriminators and image-level class existence prediction on the output level.

There are two contrastive learning frameworks among the mutually independent approaches. CFContra [87] and PWCL [162] share the idea of embedding feature-based contrastive learning methods into a larger framework with style transfer. Both apply entropy minimization in the output space. In PWCL, a patch-matching module is required to compute positive and negative pairs, and self-training is conducted. CorDA [77] and DBST [130] utilize domain mixture techniques along with depth information in slightly different ways. Building upon DACS, CorDA uses depth information and a feature-level attention mechanism to enable knowledge exchange between the depth and semantic stream, followed by a depth disparity-based output alignment. DBST connects the three spaces in a different mutually independent manner because it first applies a feature-level transfer between two networks before a depth-extended DACS version is used.

In contrast to the other works, only the input space alignment of PA+CCR [67] enforces inter-domain alignment by style transfer. The feature space centroid alignment and output space consistency alignment work independently and only within the source and target domain, respectively.

IV. VISION TRANSFORMER NETWORKS FOR UDA

This section will describe the novel and recently emerging field of vision transformers. We aim to give the reader a better intuition of why vision transformers show promising results on domain adaptation benchmarks and why they could be an exciting research direction for domain adaptation. For this reason, we will first briefly overview this research field in general and focus on the novel properties of these networks. The following will review the existing UDA works utilizing vision transformer networks.

Transformer networks with attention mechanisms were initially developed for language processing [223]. Starting with the foundational work from Dosovitskiy et al. with ViT [224], the transformer networks recently gained much attention in computer vision and showed promising results on several benchmarks and applications [225]. For semantic segmentation, several new architectures were developed, e.g., Swin transformer [58], pyramid transformer [226], SegFormer [56], and recently HRViT [227].

The self-attention mechanism is the major change compared to standard convolutional architectures such as the ResNet versions. The self-attention mechanism, originating from language processing, learns the relations between the elements of a sequence of inputs and aims to capture how the sequence element influence each other. In computer vision, the input often is not a sequence but a single image. For this reason, ViT [224] divides one image into image patches of

16×16 pixels replacing the sequence known from language processing. The self-attention mechanism learns the relations between the image patches. SegFormer [56] uses smaller 4×4 patches for semantic segmentation.

Several works indicate that vision transformer networks have higher robustness against perturbations and better generalization capabilities than CNNs [228], [229], [230]. In contrast, Bai et al. [231] and Wang et al. [232] show with their works that CNNs can achieve similar adversarial robustness when transferring certain elements except for the self-attention mechanism of vision transformer training to CNNs. Also, the recently proposed ConvNeXt [233] outperforms vision transformer networks with modifications for a CNN-based architecture, thereby questioning the superiority of the vision transformers. That indicates that research for vision transformers is still at its beginning, and more new research findings can be expected. Next to these partially contradicting results, there are some first findings on how the learned representations from vision transformer networks differ from CNNs and why that might impact their robustness and generalization capabilities. The first difference is related to the self-attention mechanism, which is supposed to learn the relations between different patches. This causes a larger receptive field [56] and enables the transformer networks to incorporate more global contextual information at the early layers [234], which might be one reason why occlusions cause a smaller performance drop than for CNNs [229]. Second CNNs are considered sensitive to texture [235], a crucial problem in domain adaptation. In contrast to that and following the current research, vision transformers focus more on the shape of objects, making them more robust to texture shifts [229]. Third, Raghu et al. [234] observed that vision transformer networks could better propagate location information through the network than CNNs, which is a beneficial property for localization tasks such as detection and segmentation.

DAFormer [55] can be seen as the foundational work with vision transformers for unsupervised domain adaptation. It proposes novel contributions to both the method- and architecture-level. This combination caused a major step in the SOTA performance outperforming the previous best approach ProDA [131], by more than 10 % mIoU. On the architecture-level DAFormer builds upon SegFormer [56], which is used as the encoder architecture. Two well-known methods from the segmentation DNNs are utilized. First DAFormer introduces skip-connections between the encoder and decoder to transfer low-level knowledge better. It then uses an ASPP-like [236] fusion where the stacked encoder outputs from different levels are processed with different dilation rates, which should further increase the receptive field. On the method-level DAFormer partially adapts known UDA methods for CNNs. Self-training with a teacher-student framework, strong augmentations, and softmax-based confidence weighting is employed. In addition, rare class sampling on the source domain and a feature distance loss to the pre-trained ImageNet features are part of the DAFormer approach. An interesting side observation is that learning rate warm-up methods can be beneficial for UDA.

The second important vision transformer contribution and current SOTA work HRDA [237] work directly builds upon DAFormer. Its major contribution is a scale attention mechanism. The network receives two inputs; one high and one low resolution input. The scale attention then learns to assign attention scores that decide whether low- or high-resolution input should get higher weighted. The idea behind that method is that different classes and objects are easier to learn on specific scales, and, e.g., contextual information can be better extracted from smaller crops. Self-training is applied using a sliding window to generate pseudo labels. This overall further improves the DAFormer performance but still leaves a performance gap.

TransDA [57] observes that the vision transformer can have a so-called high-frequency problem. Using the Swin Transformer [58] architecture, Liu et al. show that it generates target pseudo labels and features that change more significantly and with a higher frequency over the iterations than for a ResNet-101. Therefore they argue that the high-frequency problem only affects vision transformer networks. TransDA proposes feature and pseudo label smoothing using a momentum network to reduce the high-frequency flickering along with self-training and weighted adversarial output adaptation. This is similar to the teacher-student adaptation approaches known for CNNs, as described in Section III-D4.

Next to these three approaches with methods tailored explicitly for vision transformer networks, a rising number of works evaluates on them. ProCST [238] follows the idea of hybrid adaptation and applies style transfer in the input space in addition to DAFormer [55] and HRDA [237]. Several other works, which are already described in the previous sections, combine their methods with the DAFormer framework and further improve the performance by small margins [76], [183], [239], [240] but without beating HRDA. However, CLUDA [239] also builds upon HRDA and further improves this performance.

V. QUANTITATIVE COMPARISON OF UDA APPROACHES

After the extensive review of UDA techniques in the previous section, we will conduct a large-scale performance analysis, including all UDA approaches. Therefore, first, an overview of the most common UDA performance metrics and tools is given. Afterward, the comparison method is described, and finally, new insights about the performance capabilities and development are revealed.

A. PERFORMANCE METRICS AND TOOLS

The dominant quantitative evaluation metric is the mean intersection over union $\text{mIoU} = \frac{TP}{TP+FP+FN}$, with TP being the true positive predicted pixels, FP the false positive, and FN the false negative ones. It is a well-established metric for semantic segmentation to quantify the segmentation quality and is used by all included papers of our quantitative comparison. Depending on the complexity of the specific approach,

the mIoU is utilized to assess and compare the performance of several different configurations. In addition, usually, the class-wise IoUs are reported. That is important since not all classes may benefit similarly from the adaptation, and the class-wise IoUs allow a more fine-grained assessment.

Some works utilize a t-SNE [241] analysis, to visualize domain alignment: [44], [154], [189], [208], [219] Our analysis shows that it is the second most used method to assess domain alignment, even though it is a qualitative method and not a quantitative metric. T-SNE can visualize high-dimensional feature distributions in the 2-dimensional space by applying nonlinear dimensionality reduction [241]. With this technique, t-SNE is appropriate to visualize and assess the aligned feature distribution of UDA approaches. A typical change of the t-SNE plots after adaptation is a better alignment of the target feature centroids with the source centroids [189]. Next to this, most of the papers present segmentation maps as a qualitative verification of the performance of their approach. However, this can only serve to highlight specific achievements exemplarily.

Next to these often applied methods, some metrics only appear in single works like the t-test for comparison with other approaches [109] or similarity and sparsity scores [44].

Overall it becomes clear that the set and variety of different metrics are limited; namely, only mIoU for quantitative and t-SNE for qualitative comparison are established, while only the first one allows large-scale comparisons. That can hurt the kind of findings researchers can draw from their evaluation.

B. QUANTITATIVE PERFORMANCE COMPARISON

1) METHOD

For our large-scale performance comparison across the UDA segmentation approaches, we utilize the mIoU to measure the segmentation performance. These values are reported by all papers of the comparison and can therefore serve as a comparison metric. Cityscapes is the de-facto standard benchmark for the target domain. Other datasets like NTHU [38], A2D2 [242], or BDD [243] would be valuable additional evaluations for the real-to-real domain shift. However, their appearance in the evaluations is too rare for a valid large-scale comparison. The mIoU values which we are reporting in this survey are, in all cases, taken directly from the original papers without any modifications, so no individual experiments were conducted That means that differences in the evaluation protocols of the different papers like resolution etc. (as discussed in Section VI-A) can also have an impact on the reported values. The best performance is reported in the case of different reported performance values for different configurations of the same approach.

For the comparison, both the performance and the improvement over the source-only baseline are of interest, and we provide both values for each paper if possible. In several cases, the source-only performance of the approaches is not reported. Therefore, it is impossible to provide values for the improvement of these methods. For GTA5 and SYNTHIA source-only training often, 36.6% and 38.6% mIoU, respectively, are reported. However, we cannot assume these values for all papers due to several possible modifications in the source-only training. It has to be mentioned that this improvement has to be carefully considered since weaker baselines can cause a larger improvement while the performance is still low. Another highly interesting performance assessment for UDA approaches would be the performance gap to the oracle performance, which means supervised training with labels for the target domain. The number of papers reporting this performance is also limited, so no valid comparison is possible.

We present the performance comparison as a plot over time with an assignment to input, feature, output, and hybrid space adaptation. This makes the quantitative progress in UDA research over the years directly observable and reveals new insights about the capabilities of specific methods. For better interaction and more detailed information, we also provide the plots as interactive graphics on our project website with links to the papers and more information. To obtain a fair comparison w.r.t the time of publication, we took the earliest publication date we found for each paper. Figure 7 shows the performance, and Figure 8 shows the performance improvement in % absolute for both GTA5 and SYNTHIA for the VGG-16 and ResNet-101 backbone. As seen by the decreasing number of data points for the VGG-16 backbone, more recent works mainly utilize a ResNet-101 as the backbone. However, we include the VGG-16 performance to verify that the observations are valid for both backbones. Vision transformer networks are not included in these plots because they use a different architecture and would not contribute to a fair comparison. Additionally, the number of approaches using these architectures is still small and does not allow a meaningful comparison yet.

2) META ANALYSIS

The first interesting observation we can draw from the performance comparison is the performance limitation of approaches that only adapt in the feature space. None crosses the 50% mIoU line for GTA5 as the source and with a ResNet-101 backbone. All other feature space approaches reach a performance between 40-50 % mIoU. The observation for SYNTHIA is similar, where only one feature-based approach exceeds the 50% mIoU target performance. The plot showing the absolute improvement confirms this observation. Among the only feature-based approaches CaCo [164] reports the best improvement of 12.6 % absolute for GTA5 as the source and 15.0 % absolute for SYNTHIA as the source. For GTA5 many approaches do not reach an improvement of more than 20% absolute. These observations also hold for the VGG-16 backbone, where only one of the feature-based approaches reaches an improvement of more than 20% absolute. The observation is, therefore, significant across both datasets and architectures. This empirical observation could indicate that aligning the synthetic and real distributions in

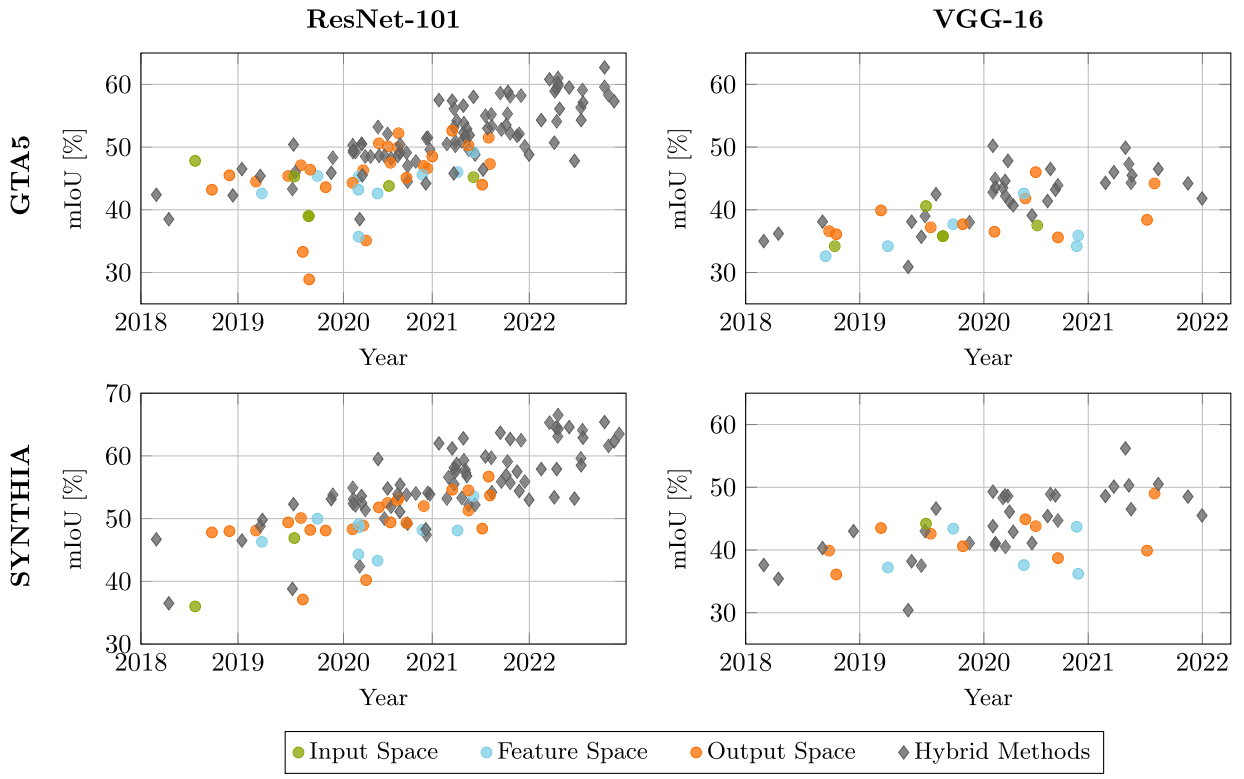


FIGURE 7. Performance (mIoU (%)) on the Cityscapes validation set after training on the source domains **GTA5** (top row) or **SYNTHIA** (bottom row) with simultaneous adaptation to Cityscapes. The results are shown for models based on a ResNet-101 feature extractor (left column) or a VGG-16 feature extractor (right column). The reported values are taken from the respective papers.

the feature space has a limit and cannot provide full alignment standalone.

The observation flips for output and hybrid space adaptation approaches. Both significantly outperform the feature space adaptation, particularly the hybrid approaches, which are one of the most important reasons for the performance increase and new SOTA performances in the past two years. Remarkably, in 2022 all proposed approaches were hybrid adaptation approaches, indicating their major importance for UDA. In strong contrast, the pure input space adaptation approaches, both quantitatively and qualitatively, play a minor role, indicating that a standalone pixel-level adaptation is insufficient to bridge the complete domain shift and provide a large performance improvement. The best-performing input space adaptation-only approach is DS [60], with 47.2% mIoU with GTA5 as the source dataset even close to the feature-space performance. The observed limitations intuitively make sense since input space adaptation does not take aspects like different label distributions, different semantic content, or different geometry into account.

Pure output space adaptation approaches are significantly more numerous than input and feature-space. However, similar to the other spaces, we can observe a performance boundary, also. For GTA5 and the ResNet-101 backbone the highest mIoU is reached by UncerDA [195] with 52.6% mIoU and the highest improvement with 16.7% absolute by IAST [185]. However, in contrast to the other spaces, several other approaches [140], [171], [186], [192] provide improvements in a similar range like the two highlighted. For both SYNTHIA and the VGG-16 backbone, a similar pattern can be observed. Notably, for VGG-16, the improvements are slightly higher with up to $\approx$ 20% absolute, but the number of data points is limited.

We can observe that the hybrid approaches made it possible to cross the line of $\approx$ 15% absolute improvement in mIoU for both GTA5 and SYNTHIA. In particular, hybrid approaches clearly carried out the latest performance raise within the last two years that exceeded the 60% mIoU performance with GTA5. Before this development, between the middle of 2019 and the end of 2020, for both SYNTHIA and GTA5, we can see saturation in the performance where most of the approaches remained close to the 50 % boundary (slightly higher for SYNTHIA). At the beginning of 2021, several better-performing hybrid works were published, ending this saturation trend. No standalone method worth highlighting, but the combination of known methods from other deep learning areas, like knowledge distillation, contrastive learning, self-supervised learning, and self-training, led to this increase in the SOTA performance. ProDA [131] used and combined all these techniques, which obtained a new SOTA performance of 57.5% mIoU and an improvement of 20.9% absolute, outperforming previous approaches by a

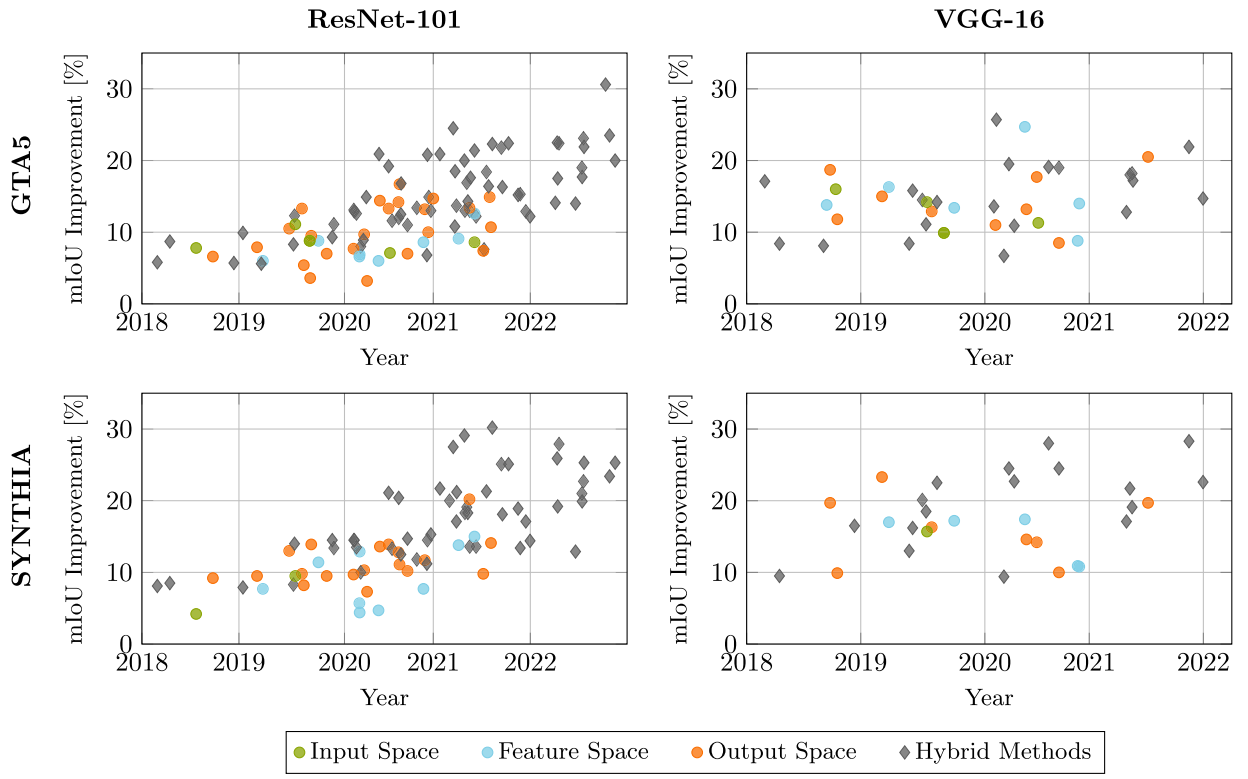


FIGURE 8. Performance improvement (mIoU (% absolute)) on the Cityscapes validation set after training on the source domains **GTA5** (top row) or **SYNTHIA** (bottom row) with simultaneous adaptation to Cityscapes. The results are shown for models based on a ResNet-101 feature extractor (left column) or a VGG-16 feature extractor (right column). The reported values are taken from the respective papers. Note that not all papers provide a baseline performance without adaptation.

significant margin. Several other works [130], [188], [194] attached additional methods to ProDA, leading to a relatively high density of approaches close to 58% mIoU but without providing significant methodical progress. The same phenomenon can be observed for vision transformer approaches where several works build upon HRDA [237] or DAFormer [55]. Those approaches [239], [240], [244], [245], [246] provide an additional improvement of 1-2% absolute and reach up to 75.9% mIoU [247] with the current SOTA approach masked image consistency (MIC). That marks a $\approx$ 30% mIoU increase over the respective source-only baseline. In comparison, the strongest reported improvement of a hybrid approach using a ResNet-101 backbone is 30.6% absolute from DDB [248], but a significantly weaker source-only baseline caused this large improvement. More realistically, most of the recent approaches with a similar or slightly weaker performance, compared to DDB, report smaller improvements of 22.4% absolute [183], 22.5% absolute [249], or 21.8% absolute [194]. This indicates that the maximum improvement by UDA techniques for vision transformer networks is significantly larger than for ResNet-101-based architectures.

These observations do not directly apply to the performance with a VGG-16 backbone, where the performance is clearly saturating since 2020. Notably, also newer hybrid approaches do not reach new SOTA performance for this setting. This does not contradict the ResNet-101 trend because it is mainly caused by the fact that VGG-16 was replaced as the standard architecture by the ResNet. Therefore newer works either entirely leave the VGG-16 out or may not perform such an extensive hyperparameter optimization as for the ResNet-101. For 2022, none of the works in the database reports their performance for the VGG-16 backbone.

Closely related to this is the observation that the performance variance of the feature space approaches is smaller than that of the hybrid space. Visually described, the performance band of the feature space is narrower. In contrast, the hybrid space shows a larger scattering of the performances where we still observe works with around 45% mIoU and high performers with over 60%. That is reasonable since hybrid approaches can strongly vary w.r.t. performance depending on the selection of components That shows that strong performance is not directly guaranteed but requires carefully selecting the hybrid components.

VI. DISCUSSION

In this section, both the benchmarking settings in UDA and the adaptation approaches themselves will be critically discussed. Based on this, this section will conclude with promising future research directions.

A. BENCHMARK PROBLEMS

The task of unsupervised domain adaptation is defined clearly. The entire training and validation process, on the other hand, has been inconsistent so far. The following issues are observed when comparing different methods for unsupervised domain adaptation.

1) DATASET SPLITS

The SYNTHIA dataset does not provide an official split into a training, validation, and test set. Therefore, the whole dataset is usually used for training. Validation on the source domain is not considered. For GTA5, there is an official split of the dataset, but it is not used universally. Some publications train solely on the training split, but others train on the entire GTA5 dataset, i.e., on training, validation, and test set combined. However, in both approaches, there is no validation of the method on the source domain, contributing to the next issue.

2) CHECKPOINT SELECTION

The training and adaptation process on the source and target domains typically takes several epochs, respectively up, to roughly 250,000 iterations. During this lengthy training and adaption process, almost all methods periodically save checkpoints of the neural network on which an evaluation is performed. The results using the checkpoint with the best performance are then reported in the publication. This leads to the following issues: First, the interval, in which checkpointing is performed, is not standardized. In some publications, a checkpoint is saved every 2000 iterations [211], and in others, every 2500 iterations [69]. Additionally, the total number of iterations can differ from method to method. Secondly, the validation for the checkpointing is often performed on the validation set of the target domain (Cityscapes), which is also used to report on in the publication. This means that the validation, which is also used for hyperparameter optimization and checkpointing, is used as a test set, which goes against the basic guidelines of machine learning. There are already some publications using a subset of the training set for validation in order to use an unseen validation set as a provisional test set [51], [115] or criticizing this and suggesting to use the validation set for validation and the original test set with the benchmark server for testing [115]. Furthermore, the use of labeled target data for validation is, in our opinion, essentially misleading, as it undermines the concept of domain adaptation and misrepresents the actual performance of the methods in real world applications. Suppose there were labeled samples of the target domain. In that case, these should be used for training since significantly better results can be achieved with supervised multi-domain training than with an adaptation. Furthermore, checkpointing on the target domain favors selecting models that perform significantly better on this domain. This can also lead to overfitting the hyperparameters, as already described in Section VI-B above.

3) TRAINING HYPERPARAMETERS

Another issue concerns the hyperparameters of training and evaluation, which can significantly impact performance but whose influence is often not adequately reported. One of the factors is the resolution of the images from the source and target domains that is adopted. For example, images from GTA5 (source domain) are usually downsampled to 1280 px× 720 px, and Cityscapes images (target domain) are usually downsampled to 1024 px×512 px during training. From here, there are a wide variety of strategies that some papers have followed but have not explicitly analyzed. Some papers use the full images as their input [204], [211] and some papers use image crops for parts of their training [115]. During the evaluation process on the target domain, most papers use the same downsampled resolution as in training, but some methods use less downsampled images, e.g., 1344 px × 576 px [69]. Many papers do not address the role of resolution and cropping further, although reduced resolution, in particular, can affect performance, especially for small structures. The different choice of these hyperparameters also contributes to the next point.

4) BASELINE PERFORMANCE

Another common issue is the lack of comparison with an own source-only baseline. Many methods only compare their adaptation performance with that of other methods. It is ignored that the source-only performance can already provide substantially different performances, e.g., by a different choice of hyperparameters and augmentations.

5) NON-DETERMINISM IN DL FRAMEWORKS

Most methods that provide code cannot be re-simulated perfectly due to non-determinism in deep learning frameworks, e.g., PyTorch [250]. This is likely because either deterministic convolutions algorithms are unavailable or take significantly longer than non-deterministic ones. Therefore, for many methods, the results may differ from the reported results. If no code is published, this makes it even more complicated since, for example, the choice of the random seed can also significantly influence the final results. There are now publications that address this problem by repeating their training several times and reporting a mean value with the standard deviation as their result [55].

B. METHODICAL REFLECTION

When analyzing the current state-of-the-art methods, one can observe that the top-performing approaches tend to be complex hybrid models. A few examples of such methods are given with ProDA [131], DPL [106], and MFA [212]. ProDA [131] applies self supervised pre-training on image net and is comprised of four training stages in which knowledge distillation, symmetric cross-entropy, contrastive, and adversarial loss functions are applied. DPL [106] combines a warmup strategy with a cycle GAN for style transfer, and it applies four segmentation losses and two adversarial losses

for the final training. Finally, MFA [212] consists of two training stages, a warmup phase and a domain adaptation phase in which co-learning is applied. In the latter, six self-training losses are applied.

Recently Sakaridis et al. [39] showed that many approaches that show good performance on the synthetic-to-real domain shift struggle on real-to-real domain changes. The synthetic to real domain change is exemplified through the adaptation from either the SYNTHIA [37] or GTA5 [36] dataset to the Cityscapes dataset [6]. The real-to-real domain change tested in Sakaridis et al. [39] is the change from Cityscapes to the ACDC dataset that contains diverse environment conditions. There might be various reasons for the different performance of the same domain adaptation approach on a different domain adaptation benchmark. First, the approaches are optimized for the benchmarked synthetic-to-real domain shift, as a result of this introducing a bias in the selection and development of new approaches toward this domain change. Second, many approaches that are comprised of many elements and hence bare a considerable amount of complexity can be finetuned toward the given benchmarks more easily by finding the optimal hyperparameters. Such finetuning is done based on the labeled target domain validation set, hence introducing a dependency on target domain supervision.

Connected to the issue of approach complexity is the topic of training stability. Given that, in practice, often limited or no validation data is given for the target domain, a training process for domain adaptation should be robust against hyperparameter setting since no finetuning might be possible. This touches on the issue of complexity but also the choice of technology. Adversarial training, e.g., is known to be very sensitive to hyperparameter settings. Approaches like self-training where a certain closeness of the source and target distribution is assumed, might suffer when applied to strong domain changes. A question especially relevant for the synthetic-to-real-world domain change is whether domain adaptation also introduces domain generalization to unknown real world domains. Since the real world target domain is comprised of a nearly infinite amount of subdomains, domain adaptation to each of them is infeasible. Hence domain adaptation approaches must introduce domain generalization to many real world sub-domains. This topic is seldoml addressed in the current unsupervised domain adaptation research.

Connected to this question is what information or knowledge can be transferred by unsupervised domain adaptation methods from the source to the target domain. Given, e.g., synthetic data of street scenes under rainy weather conditions, are unsupervised domain adaptation approaches capable of transferring the explicit knowledge about these scenes to the real world? This question is seldom analyzed in current research works. Domain adaptation approaches and papers are optimized to increase the general mIoU metric on generic domain changes such as from GTA5 to Cityscapes. We hence suggest two improvements. On the one hand, analyzing what kind of knowledge can be transferred requires datasets that offer meta tags about subdomains. On the other hand more sophisticated metrics than the mIoU are required.

Closely related is the question of how realistic the GTA5- or SYNTHIA-to-Cityscapes domain change still is. Considering that current simulation engines can generate more realistic data than the synthetic datasets GTA5 and SYNTHIA, the task of unsupervised domain adaptation from synthetic to real has changed. This probably influences the methods that could be used, and we, therefore advocate the creation of new benchmarks.

All UDA papers discussed in this survey deal with closed-set adaptation (see Section II-A) meaning both source and target domain have the same classes. However, for real large-scale application, it is likely that the classes between source and target are distinct, e.g., the target domain contains the class ''E-Scooter'' while the source does not. The performance of the researched closed-set UDA methods for this setting is unclear. A stronger focus on open-set adaptation can further simplify the application for real settings.

Training time and the number of iterations are rarely addressed in most approaches. Since training is relatively cheap for scenarios where the source and target domain are comprised of small datasets this issue is of a lesser importance. Such scenarios are, e.g., given in the current scientific benchmarks. In contrast, real autonomous driving datasets are of a large scale and their size might increase even further in the future. Hence this aspect is essential and should be considered when judging a method.

So far, the presented benchmarks deal with adapting large-scale models only applicable in real-time, given powerful hardware. UDA approaches should consider presenting a third knowledge distillation step showing how the learned target domain model could be used to provide a real-time capable model.

Finally, one of the most critical questions for unsupervised domain adaptation is its relationship with semi-supervised domain adaptation. Semi-supervised domain adaptation might be an efficient trade-off between the cost of labeling and the performance on the target domain. When looking at the highly relevant synthetic-to-real domain change in autonomous driving, we already showed that no UDA method achieves equal or better performance than supervised training on the target domain. Given that we can even achieve equal or better performance by labeling only a few images of the target domain, the question arises of where and when to apply unsupervised domain adaptation compared to semi-supervised methods.

C. FURTHER RESEARCH DIRECTIONS

In contrast to the recent performance progress, particularly the step obtained by the vision transformer architectures, the domain performance gap is still significant. Because oracle performance is mostly not reported, it is impossible to accurately quantify that remaining gap. However, an estimate between $\approx$ 5-10 % mIoU for both CNN and vision transformer architectures is realistic. This purely

performance-based assessment and the previously discussed issues show that the demand for research in domain-robust deep neural networks is still great. The aspects which we will discuss in the following as future research go beyond the perspective of simply closing the performance gap for the synthetic-to-real domain gap.

1) STANDARDIZATION OF EVALUATION

As described in Section VI-A, the current evaluation settings are based on similar architectures and the same datasets. However, they differ in several other aspects with an unclear impact on the reported performance. That is not a clean scientific standard and particularly questionable for comparisons where less than one percent can be crucial for a new SOTA performance. As a first step and necessary basis work for further research, we highly suggest standardizing the evaluation protocols.

Crucial points like the resolution, augmentations, architecture, dataset split, reporting standard, and statistically valid evaluation must be the same for all works and are easy to standardize. Other points like checkpoint selection, how to treat the labeled validation set, and a standard source training setting instead might need to be the objective of a scientific discussion to find a common standard.

2) METRICS AND DOMAIN SHIFT ANALYSIS

As shown in Section V-A, the variety of employed metrics or assessment tools is limited, directly impacting the acquired knowledge of the evaluations. More in-depth insights and a better understanding of the actual underlying mechanism within the network would further push the development of methods. These may better address the occurring domain shifts the more knowledge exists. One possible direction could be the broader utilization of feature visualization tools, e.g., t-SNE [241], to evaluate the alignment of the two domains. Another option is to develop new metrics to quantify the alignment in feature space. With this survey, we would like to encourage researchers of future UDA works to use a more diverse set of evaluation tools without diminishing the importance of performance-based metrics like mIoU.

The major focus of UDA research in the past years was reducing the performance gap. To the best of our knowledge, no works so far focus on analyzing the network behavior under domain shift. There are many important questions only answered at maximum implicitly by existing works: Which classes mostly affect the domain shift? Which factors (style, semantic content, class distribution) are most difficult for domain alignment? How do different network architectures respond under domain shift? Can we find a general metric for network generalization capabilities? Focusing more on understanding and analysis by answering these questions could be a valuable contribution to future UDA works.

3) NEW NETWORK ARCHITECTURES

As described, both VGG-16 and ResNet-101 were the defacto standard backbones used in UDA research for several years. Vision transformer architectures recently gained attention in UDA research and reached new SOTA performances (see Section III-E4.b). The full potential of these architectures now needs to be exploited after first works with promising results. Therefore utilizing and researching vision transformers as a new architecture type is one of the promising future research trends. For researchers, we recommend including vision transformer networks like DAFormer [55] in future publications.

However, it remains unclear at the current point if vision transformer networks will close the performance domain gap and how strong they perform for other settings such as domain generalization. For this reason another interesting research direction can be the development of a real next generation deep learning method with stronger and more human-like generalization capabilities than the vision transformer and more fundamental architectural changes as it was suggested by Marcus et al. [251]. For this reason, another interesting research direction can be developing a real next-generation deep learning method with stronger and more human-like generalization capabilities than the vision transformer and more fundamental architectural changes, as it was suggested by Marcus et al. [251]. The complex hybrid methods show that CNNs, by default, do not have strong generalization capabilities and therefore require much effort and additional methods (see Section VI-B) to obtain a better generalization or adaptation. This might be a strong motivation to push research from the methodic to the architecture level to provide much better generalization capabilities with a new generation of deep networks. This could align with the research of other areas such as robustness and adversarial stability that also may benefit from increased generalization.

4) LARGE-SCALE DOMAIN ADAPTATION

There are no works that research large-scale unsupervised domain adaptation, as would be the case for industrial applications. Instead, for example, a target dataset is utilized, which is relatively old and contains a small number of images, significantly smaller than more recent datasets like A2D2 [242], which would be an exciting addition. The focus of the works in this survey is clearly a scientific benchmarking setting which is unquestionably necessary. However, more research is necessary to use these algorithms for real applications because the field of applicability of UDA algorithms still needs to be completed. One step in this direction can be open set domain adaptation as described in Section II-A, where the target classes are distinct from the source classes and which might be a common problem for industrial applications across different countries. In Section VI-B, we have shown that already small settings changes can lead

to severe performance decreases, so research needs to exploit how these algorithms can be transferred into real industrial settings.

There is currently a strong focus in research on camera-based domain adaptation, while some works tackle domain adaptation for lidar sensors [252]. However, from a recent point of knowledge, an entirely autonomous driving stack will come with some kind of sensor fusion that combines the camera information with the inputs from other sensors like LiDAR and RaDAR. To our knowledge, there are currently no works covering the impact of the domain shift on the system level, which means for the entire perception system of an autonomous vehicle. Researching this question and proposing new methods to tackle the domain shift on a perception system level would be a promising and valuable future research direction.

5) DOMAIN GENERALIZATION

Our survey focuses on unsupervised domain adaptation where unlabeled data of the target domain is available to perform adaptation. However, we must question how feasible this setting is outside a fixed research context in a large-scale industrial setting. The data collection might not be the largest problem since an intelligent data collection mechanism may help, depending on the application. Data utilization for large-scale applications such as the automotive industry can lead to severe problems. The pure number of domains a network needs to be adapted to will cost a lot of effort in terms of computational and human resources. Also the formal safety verification that the networks fulfill the safety standards in all domains can be time-intensive. It is unlikely that the current generation of DNNs, including vision transformers, can perform well on a very large range of domains in parallel, even with good adaptation.

For these reasons, domain generalization can be even more promising as a future research direction because the adaptation step can be left out, making large-scale adaptation easier under the assumption that strong generalization methods are available. This would have the major advantage that the network does not need to be adapted to every single domain, but we obtain a model that generalizes well across all or most unknown domains. However, these methods are still missing and could be a promising research direction for their development. Since it is unclear if current models are powerful enough for such learning strategies, this could be connected to the research direction of novel architectures.

VII. CONCLUSION

In this work, we have given the most thorough review of the highly relevant and fastly evolving field of unsupervised domain adaptation for semantic segmentation. We have categorized and explained the ideas and methods of the vast majority of approaches that were published in this field. We have created a unique knowledge base that provides the reader with a comprehensive overview of the field and an extensive quantitative comparison of the approaches. There was strong methodical progress over the past years in UDA research, successively decreasing the domain gap. Hybrid adaptation methods, a combination of multiple standalone methods, can be highlighted as the currently most sophisticated way for UDA and the carrier of the latest SOTA performances. Recently also, vision transformer architectures raised the SOTA performances into new dimensions.

However, the domain gap has yet to be closed. For this reason, the scope and claim of this survey were not only the categorization and the description of recent approaches but also to analyze the current status of the research critically. Following this idea, we pointed out several issues in UDA research, like very complex approaches, bad generalization to other settings, missing strict standards for benchmarking, limited evaluation metrics, and more points.

Based on this critical analysis, we eventually recommend promising future directions. Here we include aspects like new common standards, new architectures, or an application for real large-scale industrial settings. Going beyond the scope of classic UDA research, we discuss new directions like system-level adaptation, symbolic deep learning, and domain generalization. By doing so, we gave new impulses for domain adaptation and believe to have facilitated research further.

REFERENCES

[1] T. Fingscheidt, H. Gottschalk, and S. Houben, Eds., *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety*. Cham, Switzerland: Springer, 2022. [Online]. Available: https://library.oapen.org/handle/20.500.12657/57375

[2] S. Houben et al., "Inspect, understand, overcome: A survey of practical methods for AI safety," in *Deep Neural Networks and Data for Automated Driving*. Cham, Switzerland: Springer, 2022, pp. 3–78.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[4] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in adversarial attacks and defenses in computer vision: A survey," *IEEE Access*, vol. 9, pp. 155161–155196, 2021.

[5] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, pp. 1–22, 2020.

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 3213–3223.

[7] A. Meyer, N. O. Salscheider, P. F. Orzechowski, and C. Stiller, "Deep semantic lane segmentation for mapless driving," in *Proc. IROS*, Madrid, Spain, Oct. 2018, pp. 869–875.

[8] C. Plachetka, N. Maier, J. Fricke, J. Termöhlen, and T. Fingscheidt, "Terminology and analysis of map deviations in urban domains: Towards dependability for HD maps in automated vehicles," in *Proc. IV*, Las Vegas, NV, USA, Oct. 2020, pp. 63–70.

[9] C. Plachetka, J. Fricke, M. Klingner, and T. Fingscheidt, "DNN-based recognition of pole-like objects in LiDAR point clouds," in *Proc. ITSC*, Montreal, QC, Canada, Sep. 2021, pp. 2889–2896.

[10] C. Plachetka, B. Sertolli, J. Fricke, M. Klingner, and T. Fingscheidt, "3DHD CityScenes: high-definition maps in high-density point clouds," in *Proc. ITSC*, Macau, China, Oct. 2022, pp. 627–634.

[11] P. Oza, V. A. Sindagi, V. V. Sharmini, and V. M. Patel, "Unsupervised domain adaptation of object detectors: A survey," 2021, *arXiv:2105.13502*.

[12] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "FCNs in the wild: Pixel-level adversarial and constraint-based adaptation," 2016, *arXiv:1612.02649*.

[13] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, pp. 1–46, Oct. 2020.

[14] Y. Zhang, "A survey of unsupervised domain adaptation for visual recognition," 2021, *arXiv:2112.06745*.

[15] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[16] M. Toldo, A. Maracani, U. Michieli, and P. Zanuttigh, "Unsupervised domain adaptation in semantic segmentation: A review," *Technologies*, vol. 8, no. 2, pp. 1–35, 2020.

[17] G. Csurka, R. Volpi, and B. Chidlovskii, "Unsupervised domain adaptation for semantic image segmentation: A comprehensive survey," 2021, *arXiv:2112.03241*.

[18] Z. Wang, Y. Wei, R. Feris, J. Xiong, W.-M. Hwu, T. S. Huang, and H. Shi, "Alleviating semantic-level shift: A semi-supervised domain adaptation method for semantic segmentation," in *Proc. CVPR Workshops*, Seattle, WA, USA, Jun. 2020, pp. 936–937.

[19] A. Mütze, M. Rottmann, and H. Gottschalk, "Semi-supervised domain adaptation with CycleGAN guided by a downstream task loss," 2022, *arXiv:2208.08815*.

[20] Y. Chen, X. Ouyang, K. Zhu, and G. Agam, "Semi-supervised domain adaptation for semantic segmentation," 2021, *arXiv:2110.10639*.

[21] S. Chen, X. Jia, J. He, Y. Shi, and J. Liu, "Semi-supervised domain adaptation based on dual-level domain mixing for semantic segmentation," in *Proc. CVPR*, Jun. 2021, pp. 11018–11027.

[22] N. Hanselmann, N. Schneider, B. Ortelt, and A. Geiger, "Learning cascaded detection tasks with weakly-supervised domain adaptation," in *Proc. IEEE IV*, Jul. 2021, pp. 532–539.

[23] S. Paul, Y.-H. Tsai, S. Schulter, A. K. Roy-Chowdhury, and M. Chandraker, "Domain adaptive semantic segmentation using weak labels," 2020, *arXiv:2007.15176*.

[24] M. Klingner, J. Termöhlen, J. Ritterbach, and T. Fingscheidt, "Unsupervised BatchNorm adaptation (UBNA): A domain adaptation method for semantic segmentation without using source domain representations," in *Proc. WACV Workshops*, Waikoloa, HI, USA, Jan. 2022, pp. 210–220.

[25] M. Klingner, M. Ayache, and T. Fingscheidt, "Continual BatchNorm adaptation (CBNA) for semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20899–20911, Nov. 2022.

[26] J.-A. Termöhlen, M. Klingner, L. J. Brettin, N. M. Schmidt, and T. Fingscheidt, "Continual unsupervised domain adaptation for semantic segmentation by online frequency domain style transfer," in *Proc. ITSC*, Sep. 2021, pp. 2881–2888.

[27] M. Wulfmeier, A. Bewley, and I. Posner, "Incremental adversarial domain adaptation for continually changing environments," in *Proc. ICRA*, Brisbane, QLD, Australia, May 2018, pp. 4489–4495.

[28] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in *Proc. ICCV*, Seoul, South Korea, Oct. 2019, pp. 2100–2110.

[29] S. Lee, H. Seong, S. Lee, and E. Kim, "WildNet: Learning domain generalized semantic segmentation from the wild," in *Proc. CVPR*, New Orleans, LA, USA, Jun. 2022, pp. 9936–9946.

[30] S. Choi, S. Jung, H. Yun, J. T. Kim, S. Kim, and J. Choo, "RobustNet: Improving domain generalization in urban-scene segmentation via instance selective whitening," in *Proc. CVPR*, Jun. 2021, pp. 11580–11590.

[31] R. Gong, W. Li, Y. Chen, and L. Van Gool, "DLOW: Domain flow for adaptation and generalization," in *Proc. CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 2477–2486.

[32] S. Uhlemeyer, M. Rottmann, and H. Gottschalk, "Towards unsupervised open world semantic segmentation," 2022, *arXiv:2201.01073*.

[33] T. Kalb, M. Roschani, M. Ruf, and J. Beyerer, "Continual learning for class- and domain-incremental semantic segmentation," in *Proc. IV*, Jul. 2021, pp. 1345–1351.

[34] R. Gong, M. Danelljan, D. Dai, D. P. Paudel, A. Chhatkuli, F. Yu, and L. Van Gool, "TACS: Taxonomy adaptive cross-domain semantic segmentation," 2021, *arXiv:2109.04813*.

[35] W. M. Kouw and M. Loog, "An introduction to domain adaptation and transfer learning," 2018, *arXiv:1812.11806*.

[36] S. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. ECCV*, Amsterdam, The Netherlands, Oct. 2016, pp. 102–118.

[37] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 3234–3243.

[38] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C.-F. Wang, and M. Sun, "No more discrimination: Cross city adaptation of road scene segmenters," in *Proc. ICCV*, Venice, Italy, Oct. 2017, pp. 1992–2001.

[39] C. Sakaridis, D. Dai, and L. Van Gool, "ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding," in *Proc. ICCV*, Oct. 2021, pp. 10765–10775.

[40] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. CoRL*, Mountain View, CA, USA, Nov. 2017, pp. 1–16.

[41] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[42] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[43] C.-H. Chao, B.-W. Cheng, and C.-Y. Lee, "Rethinking ensemble-distillation for semantic segmentation based unsupervised domain adaptation," 2021, *arXiv:2104.14203*.

[44] M. Toldo, U. Michieli, and P. Zanuttigh, "Unsupervised domain adaptation in semantic segmentation via orthogonal and clustered embeddings," in *Proc. WACV*, Jan. 2021, pp. 1358–1368.

[45] S. Ye, K. Wu, M. Zhou, Y. Yang, S. H. Tan, K. Xu, J. Song, C. Bao, and K. Ma, "Light-weight calibrator: A separable component for unsupervised domain adaptation," in *Proc. CVPR*, Jun. 2020, pp. 13736–13745.

[46] S. Lee, S. Cho, and S. Im, "DRANet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation," in *Proc. CVPR*, Jun. 2021, pp. 15252–15261.

[47] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "CrDoCo: Pixel-level domain transfer with cross-domain consistency," in *Proc. CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 1791–1800.

[48] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 3723–3732.

[49] Y. Shan, C. M. Chew, and W. F. Lu, "Semantic-aware short path adversarial training for cross-domain semantic segmentation," *Neurocomputing*, vol. 380, pp. 125–132, Mar. 2020.

[50] J. Iqbal and M. Ali, "MLSL: Multi-level self-supervised learning for domain adaptation with spatially independent and semantically consistent labeling," in *Proc. WACV*, Aspen, CO, USA, Mar. 2020, pp. 1864–1873.

[51] Q. Lian, L. Duan, F. Lv, and B. Gong, "Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach," in *Proc. ICCV*, Seoul, South Korea, Oct. 2019, pp. 6758–6767.

[52] Y. Zou, Z. Yu, B. V. K. V. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 289–305.

[53] J. Dong, Y. Cong, G. Sun, and D. Hou, "Semantic-transferable weakly-supervised endoscopic lesions segmentation," in *Proc. ICCV*, Seoul, South Korea, Oct. 2019, pp. 10712–10721.

[54] Z. Li, R. Togo, T. Ogawa, and M. Haseyama, "Variational autoencoder based unsupervised domain adaptation for semantic segmentation," in *Proc. ICIP*, Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 2426–2430.

[55] L. Hoyer, D. Dai, and L. Van Gool, "DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation," in *Proc. CVPR*, New Orleans, LA, USA, Jun. 2022, pp. 9924–9935.

[56] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. NeurIPS*, Dec. 2021, pp. 12077–12090.

[57] R. Chen, Y. Rong, S. Guo, J. Han, F. Sun, T. Xu, and W. Huang, "Smoothing matters: Momentum transformer for domain adaptive semantic segmentation," 2022, *arXiv:2203.07988*.

[58] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. ICCV*, Oct. 2021, pp. 10012–10022.

[59] M. Toldo, U. Michieli, G. Agresti, and P. Zanuttigh, "Unsupervised domain adaptation for mobile semantic segmentation based on cycle consistency and feature alignment," *Image Vis. Comput.*, vol. 95, pp. 103889–103899, Mar. 2020.

[60] A. Dundar, M.-Y. Liu, T.-C. Wang, J. Zedlewski, and J. Kautz, "Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation," Jul. 2018, *arXiv:1807.09384*.

[61] V. Gkitsas, A. Karakottas, N. Zioulis, D. Zarpalas, and P. Daras, "Restyling data: Application to unsupervised domain adaptation," 2019, *arXiv:1909.10900*.
[62] R. Li, W. Cao, S. Wu, and H. Wong, "Generating target image-label pairs for unsupervised domain adaptation," *IEEE Trans. Image Process.*, vol. 29, pp. 7997–8011, 2020.
[63] Y. Lin, D. S. Tan, W. Cheng, and K. Hua, "Adapting semantic segmentation of urban scenes via mask-aware gated discriminator," in *Proc. ICME*, Shanghai, China, Jul. 2019, pp. 218–223.
[64] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. ICCV*, Venice, Italy, Oct. 2017, pp. 2223–2232.
[65] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. ICCV*, Venice, Italy, Oct. 2017, pp. 1501–1510.
[66] J. Huang, D. Guan, A. Xiao, and S. Lu, "Cross-view regularization for domain adaptive panoptic segmentation," 2021, *arXiv:2103.02584*.
[67] H. Ma, X. Lin, Z. Wu, and Y. Yu, "Coarse-to-fine domain adaptive semantic segmentation with photometric alignment and category-center regularization," in *Proc. CVPR* Jun. 2021, pp. 4051–4060.
[68] Y. Lu, Y. Luo, L. Zhang, Z. Li, Y. Yang, and J. Xiao, "Bidirectional self-training with multiple anisotropic prototypes for domain adaptive semantic segmentation," Apr. 2022, *arXiv:2204.07730*.
[69] Y. Yang and S. Soatto, "FDA: Fourier domain adaptation for semantic segmentation," in *Proc. CVPR*, Seattle, WA, USA, Jun. 2020, pp. 4085–4095.
[70] J. Zhang, J. Huang, Z. Tian, and S. Lu, "Spectral unsupervised domain adaptation for visual recognition," 2021, *arXiv:2106.06112*.
[71] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson, "DACS: Domain adaptation via cross-domain mixed sampling," in *Proc. WACV*, Waikoloa, HI, USA, Jan. 2021, pp. 1379–1389.
[72] L. Gao, J. Zhang, L. Zhang, and D. Tao, "DSP: Dual soft-paste for unsupervised domain adaptive semantic segmentation," in *Proc. 29th ACM Int. Conf. Multimedia*, Chengdu, China, Oct. 2021, pp. 2825–2833.
[73] L. Melas-Kyriazi and A. K. Manrai, "PixMatch: Unsupervised domain adaptation via pixelwise consistency training," in *Proc. CVPR*, Jun. 2021, pp. 12435–12445.
[74] Y. Liu, J. Deng, X. Gao, W. Li, and L. Duan, "BAPA-Net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation," in *Proc. ICCV*, Oct. 2021, pp. 8801–8811.
[75] Q. Zhou, C. Zhuang, R. Yi, X. Lu, and L. Ma, "Domain adaptive semantic segmentation via regional contrastive consistency regularization," Oct. 2021, *arXiv:2110.05170*.
[76] Q. Zhou, Z. Feng, Q. Gu, J. Pang, G. Cheng, X. Lu, J. Shi, and L. Ma, "Context-aware mixup for domain adaptive semantic segmentation," 2021, *arXiv:2108.03557*.
[77] Q. Wang, D. Dai, L. Hoyer, L. Van Gool, and O. Fink, "Domain adaptive semantic segmentation with self-supervised depth estimation," in *Proc. ICCV*, Oct. 2021, pp. 8515–8525.
[78] X. Huo, L. Xie, H. Hu, W. Zhou, H. Li, and Q. Tian, "Domain-agnostic prior for transfer semantic segmentation," 2022, *arXiv:2204.02684*.
[79] M. Kim and H. Byun, "Learning texture invariant representation for domain adaptation of semantic segmentation," in *Proc. CVPR*, Seattle, WA, USA, Jun. 2020, pp. 12975–12984.
[80] J. Huang, D. Guan, A. Xiao, and S. Lu, "RDA: Robust domain adaptation via Fourier adversarial attacking," in *Proc. ICCV*, Oct. 2021, pp. 8988–8999.
[81] F. Zhang, V. Koltun, P. Torr, R. Ranftl, and S. R. Richter, "Unsupervised contrastive domain adaptation for semantic segmentation," 2022, *arXiv:2204.08399*.
[82] Q. Zhou, Z. Feng, Q. Gu, G. Cheng, X. Lu, J. Shi, and L. Ma, "Uncertainty-aware consistency regularization for cross-domain semantic segmentation," 2020, *arXiv:2004.08878*.
[83] L. Musto and A. Zinelli, "Semantically adaptive image-to-image translation for domain adaptation of semantic segmentation," 2020, *arXiv:2009.01166*.
[84] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Adversarial style mining for one-shot unsupervised domain adaptation," in *Proc. NeurIPS*, Dec. 2020, pp. 20612–20623.
[85] J. Choi, T. Kim, and C. Kim, "Self-ensembling with GAN-based data augmentation for domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 6830–6840. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/papers/Choi_Self-Ensembling_With_GAN-Based_Data_Augmentation_for_Domain_Adaptation_in_Semantic_ICCV_2019_paper.pdf
[86] K. Wang, C. Yang, and M. Betke, "Consistency regularization with high-dimensional non-adversarial source-guided perturbation for unsupervised domain adaptation in segmentation," 2020, *arXiv:2009.08610*.
[87] S. Tang, P. Tang, Y. Gong, Z. Ma, and M. Xie, "Unsupervised domain adaptation via coarse-to-fine feature alignment method using contrastive learning," 2021, *arXiv:2103.12371*.
[88] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*.
[89] Z. Wu, X. Han, Y.-L. Lin, M. G. Uzunbas, T. Goldstein, S. N. Lim, and L. S. Davis, "DCAN: Dual channel-wise alignment networks for unsupervised scene adaptation," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 535–552.
[90] Y. Chen, W. Li, X. Chen, and L. Van Gool, "Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach," in *Proc. CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 1841–1850.
[91] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 3752–3761.
[92] W.-L. Chang, H.-P. Wang, W.-H. Peng, and W.-C. Chiu, "All about structure: Adapting structural information across domains for boosting semantic segmentation," in *Proc. CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 1900–1909.
[93] Y. Yang, D. Lao, G. Sundaramoorthi, and S. Soatto, "Phase consistent ecological domain adaptation," in *Proc. CVPR*, Seattle, WA, USA, Jun. 2020, pp. 9011–9020.
[94] L. Song, Y. Xu, L. Zhang, B. Du, Q. Zhang, and X. Wang, "Learning from synthetic images via active pseudo-labeling," *IEEE Trans. Image Process.*, vol. 29, pp. 6452–6465, 2020.
[95] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proc. ICML*, Stockholm, Sweden, Jul. 2018, pp. 1989–1998.
[96] J. Yang, W. An, C. Yan, P. Zhao, and J. Huang, "Context-aware domain adaptation in semantic segmentation," in *Proc. WACV*, Jan. 2021, pp. 514–524.
[97] P. Z. Ramirez, A. Tonioni, and L. Di Stefano, "Exploiting semantics in adversarial training for image-level domain adaptation," in *Proc. IPAS*, Sophia Antipolis, France, Dec. 2018, pp. 49–54.
[98] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *Proc. CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 6936–6945.
[99] M. Kim, S. Joung, S. Kim, J. Park, I.-J. Kim, and K. Sohn, "Cross-domain grouping and alignment for domain adaptive semantic segmentation," 2020, *arXiv:2012.08226*.
[100] W. Zhou, Y. Wang, J. Chu, J. Yang, X. Bai, and Y. Xu, "Affinity space adaptation for semantic segmentation across domains," *IEEE Trans. Image Process.*, vol. 30, pp. 2549–2561, 2021.
[101] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W.-M. Hwu, T. S. Huang, and H. Shi, "Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation," in *Proc. CVPR*, Seattle, WA, USA, Jun. 2020, pp. 12635–12644.
[102] J. Yang, W. An, S. Wang, X. Zhu, C. Yan, and J. Huang, "Label-driven reconstruction for domain adaptation in semantic segmentation," in *Proc. ECCV*, Glasgow, U.K., Aug. 2020, pp. 480–498.
[103] K.-H. Lee, G. Ros, J. Li, and A. Gaidon, "SPIGAN: Privileged adversarial learning from simulation," in *Proc. ICLR*, New Orleans, LA, USA, Apr. 2019, pp. 1–14.
[104] S. Lee, J. Hyun, H. Seong, and E. Kim, "Unsupervised domain adaptation for semantic segmentation by content transfer," 2020, *arXiv:2012.12545*.
[105] I. Chung, D. Kim, and N. Kwak, "Maximizing cosine similarity between spatial features for unsupervised domain adaptation in semantic segmentation," 2021, *arXiv:2102.13002*.
[106] Y. Cheng, F. Wei, J. Bao, D. Chen, F. Wen, and W. Zhang, "Dual path learning for domain adaptation of semantic segmentation," in *Proc. ICCV*, Oct. 2021, pp. 9082–9091.
[107] Y. Xu, F. He, B. Du, D. Tao, and L. Zhang, "Self-ensembling GAN for cross-domain semantic segmentation," 2021, *arXiv:2112.07999*.
[108] L. Gao, L. Zhang, and Q. Zhang, "Addressing domain gap via content invariant representation for semantic segmentation," in *Proc. AAAI*, 2021, vol. 35, no. 9, pp. 7528–7536.
[109] J. Dong, Y. Cong, G. Sun, Z. Fang, and Z. Ding, "Where and how to transfer: Knowledge aggregation-induced transferability perception for unsupervised domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Nov. 16, 2021, doi: 10.1109/TPAMI.2021.3128560.

[110] A. Saporta, T.-H. Vu, M. Cord, and P. Pérez, "ESL: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation," 2020, *arXiv:2006.08658*.

[111] Z. Li, R. Togo, T. Ogawa, and M. Haseyama, "Unsupervised domain adaptation for semantic segmentation with symmetric adaptation consistency," in *Proc. ICASSP*, Barcelona, Spain, May 2020, pp. 2263–2267.

[112] B. Cai, H. Fu, R. Jia, B. Zhao, H. Li, and Y. Xu, "Exploiting diverse characteristics and adversarial ambivalence for domain adaptive segmentation," 2020, *arXiv:2012.05608*.

[113] R. Li, W. Cao, Q. Jiao, S. Wu, and H.-S. Wong, "Simplified unsupervised image translation for semantic segmentation adaptation," *Pattern Recognit.*, vol. 105, Sep. 2020, Art. no. 107343.

[114] E. Chiou, E. Panagiotaki, and I. Kokkinos, "Beyond deterministic translation for unsupervised domain adaptation," 2022, *arXiv:2202.07778*.

[115] N. Araslanov and S. Roth, "Self-supervised augmentation consistency for adapting semantic segmentation," in *Proc. CVPR*, Jun. 2021, pp. 15384–15394.

[116] J. Huang, D. Guan, S. Lu, and A. Xiao, "MLAN: Multi-level adversarial network for domain adaptive semantic segmentation," 2021, *arXiv:2103.12991*.

[117] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, "A closed-form solution to photorealistic image stylization," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 453–468.

[118] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Proc. NeurIPS*, Long Beach, CA, USA, Dec. 2017, pp. 386–396.

[119] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, San Diego, CA, USA, May 2015, pp. 1–27.

[120] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 1–18.

[121] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. T. H. Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Comput. Vis., Graph., Image Process.*, vol. 39, no. 3, pp. 355–368, Sep. 1987.

[122] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. ICCV*, Venice, Italy, Oct. 2017, pp. 2794–2802.

[123] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. ECCV*, Amsterdam, The Netherlands, Oct. 2016, pp. 694–711.

[124] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[125] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 8789–8797.

[126] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Proc. NeurIPS*, Long Beach, CA, USA, Dec. 2017, pp. 465–476.

[127] S. Benaim, M. Khaitov, T. Galanti, and L. Wolf, "Domain intersection and domain difference," in *Proc. ICCV*, Seoul, South Korea, Oct. 2019, pp. 3445–3453.

[128] V. Olsson, W. Tranheden, J. Pinto, and L. Svensson, "ClassMix: Segmentation-based data augmentation for semi-supervised learning," in *Proc. WACV*, Waikoloa, HI, USA, Jan. 2021, pp. 1369–1378.

[129] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. ICCV*, Seoul, South Korea, Oct. 2019, pp. 6023–6032.

[130] A. Cardace, L. De Luigi, P. Z. Ramirez, S. Salti, and L. Di Stefano, "Plugging self-supervised monocular depth into unsupervised domain adaptation for semantic segmentation," in *Proc. WACV*, Waikoloa, HI, USA, Jan. 2022, pp. 1129–1139.

[131] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," 2021, *arXiv:2101.10979*.

[132] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang, "SSF-DAN: Separated semantic feature based domain adaptation network for semantic segmentation," in *Proc. ICCV*, Seoul, South Korea, Oct. 2019, pp. 982–991.

[133] C. Li, D. Du, L. Zhang, L. Wen, T. Luo, Y. Wu, and P. Zhu, "Spatial attention pyramid network for unsupervised domain adaptation," in *Proc. ECCV*, Aug. 2020, pp. 481–497.

[134] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Significance-aware information bottleneck for domain adaptive semantic segmentation," in *Proc. ICCV*, Seoul, South Korea, Oct. 2019, pp. 6778–6787.

[135] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proc. CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 2507–2516.

[136] D. Hu, J. Liang, Q. Hou, H. Yan, Y. Chen, S. Yan, and J. Feng, "Semantic domain adversarial networks for unsupervised domain adaptation," 2020, *arXiv:2003.13274*.

[137] Y. Tsai, K. Sohn, S. Schulter, and M. Chandraker, "Domain adaptation for structured output via discriminative patch representations," in *Proc. ICCV*, Seoul, South Korea, Oct. 2019, pp. 1456–1465.

[138] Q. Zheng, J. Chen, Z. Wang, J. Jiang, and C. Liang, "Deep segmentation domain adaptation network with weighted boundary constraint," *IEEE Access*, vol. 7, pp. 93909–93918, 2019.

[139] F. Yu, M. Zhang, H. Dong, S. Hu, B. Dong, and L. Zhang, "DAST: Unsupervised domain adaptation in semantic segmentation based on discriminator attention and self-training," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10754–10762.

[140] H. Wang, T. Shen, W. Zhang, L.-Y. Duan, and T. Mei, "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation," in *Proc. ECCV*, Aug. 2020, pp. 642–659.

[141] Y. Luo, Z. Wang, D. Huang, N. Ge, and J. Lu, "Get away from style: category-guided domain adaptation for semantic segmentation," 2021, *arXiv:2103.15467*.

[142] J. Dong, Y. Cong, G. Sun, Y. Liu, and X. Xu, "CSCL: Critical semantic-consistent learning for unsupervised domain adaptation," in *Proc. ECCV*, Aug. 2020, pp. 745–762.

[143] H. Huang, Q. Huang, and P. Krahenbuhl, "Domain transfer through deep activation matching," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 590–605.

[144] Y. Wang, Y. Li, J. H. Elder, R. Wu, and H. Lu, "Class-conditional domain adaptation on semantic segmentation," 2019, *arXiv:1911.11981*.

[145] Y. Chen, W. Li, and L. Van Gool, "ROAD: Reality oriented adaptation for semantic segmentation of urban scenes," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 7892–7901.

[146] H. Chen, C. Wu, Y. Xu, and B. Du, "Unsupervised domain adaptation for semantic segmentation via low-level edge information transfer," 2021, *arXiv:2109.08912*.

[147] T. Chen, J. Zhang, G. Xie, Y. Yao, X. Huang, and Z. Tang, "Classification constrained discriminator for domain adaptive semantic segmentation," in *Proc. ICME*, Jul. 2020, pp. 1–6.

[148] Y. Zhang, Z. Qiu, T. Yao, C.-W. Ngo, D. Liu, and T. Mei, "Transferring and regularizing prediction for semantic segmentation," in *Proc. CVPR*, Jun. 2020, pp. 9621–9630.

[149] S. Saha, A. Obukhov, D. P. Paudel, M. Kanakis, Y. Chen, S. Georgoulis, and L. Van Gool, "Learning to relate depth and semantics for unsupervised domain adaptation," in *Proc. CVPR*, Jun. 2021, pp. 8197–8207.

[150] J.-A. Bolte, M. Kamp, A. Breuer, S. Homoceanu, P. Schlicht, F. Hüger, D. Lipinski, and T. Fingscheidt, "Unsupervised domain adaptation to improve image segmentation quality both in the source and target domain," in *Proc. CVPR Workshops*, Long Beach, CA, USA, Jun. 2019, pp. 1404–1413.

[151] S. Li, F. Lv, B. Xie, C. H. Liu, J. Liang, and C. Qin, "Bi-classifier determinacy maximization for unsupervised domain adaptation," 2020, *arXiv:2012.06995*.

[152] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced Wasserstein discrepancy for unsupervised domain adaptation," in *Proc. CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 10285–10295.

[153] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 1335–1344.

[154] Y. Zhu, H. Zhou, C. Yang, J. Shi, and D. Lin, "Penalizing top performers: Conservative loss for semantic segmentation adaptation," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 568–583.

[155] C. Ruan, W. Wang, H. Hu, and D. Chen, "Category-level adversaries for semantic domain adaptation," *IEEE Access*, vol. 7, pp. 83198–83208, 2019. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8731847

[156] R. Romijnders, P. Meletis, and G. Dubbelman, "A domain agnostic normalization layer for unsupervised adversarial domain adaptation," in *Proc. WACV*, Waikoloa, HI, USA, Jan. 2019, pp. 1866–1875.

[157] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Proc. NIPS*, Long Beach, CA, USA, Dec. 2017, pp. 1945–1953.

[158] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognit.*, vol. 80, pp. 109–117, Aug. 2018.
[159] R. A. Marsden, A. Bartler, M. Döbler, and B. Yang, "Contrastive learning and self-training for unsupervised domain adaptation in semantic segmentation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–8.
[160] B. Xie, M. Li, and S. Li, "SPCL: A new framework for domain adaptive semantic segmentation via semantic prototype-based contrastive learning," 2021, *arXiv:2111.12358*.
[161] D. Shim and H. J. Kim, "Learning a domain-agnostic visual representation for autonomous driving via contrastive loss," 2021, *arXiv:2103.05902*.
[162] W. Liu, D. Ferstl, S. Schulter, L. Zebedin, P. Fua, and C. Leistner, "Domain adaptation for semantic segmentation via patch-wise contrastive learning," 2021, *arXiv:2104.11056*.
[163] G. Kang, Y. Wei, Y. Yang, Y. Zhuang, and A. G. Hauptmann, "Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation," 2020, *arXiv:2011.00147*.
[164] J. Huang, D. Guan, A. Xiao, S. Lu, and L. Shao, "Category contrast for unsupervised domain adaptation in visual tasks," 2021, *arXiv:2106.02885*.
[165] J. Niemeijer and J. P. Schäfer, "Combining semantic self-supervision and self-training for domain adaptation in semantic segmentation," in *Proc. IV Workshops*, Jul. 2021, pp. 364–371.
[166] S. Wang, D. Zhao, Y. Li, C. Zhang, Y. Guo, Q. Zang, B. Hou, and L. Jiao, "More separable and easier to segment: A cluster alignment method for cross-domain semantic segmentation," 2021, *arXiv:2105.03151*.
[167] S. Li, B. Xie, B. Zang, C. H. Liu, X. Cheng, R. Yang, and G. Wang, "Semantic distribution-aware contrastive adaptation for semantic segmentation," 2021, *arXiv:2105.05013*.
[168] V. Guizilini, J. Li, R. Ambrus, and A. Gaidon, "Geometric unsupervised domain adaptation for semantic segmentation," 2021, *arXiv:2103.16694*.
[169] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," 2019, *arXiv:1909.11825*.
[170] J. Xu, L. Xiao, and A. M. López, "Self-supervised domain adaptation for computer vision tasks," *IEEE Access*, vol. 7, pp. 156694–156706, 2019.
[171] F. Lv, T. Liang, X. Chen, and G. Lin, "Cross-domain semantic segmentation via domain-invariant interactive relation transfer," in *Proc. CVPR*, Seattle, WA, USA, Jun. 2020, pp. 4334–4343.
[172] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, Lille, France, Jul. 2015, pp. 1180–1189.
[173] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, Montreal, QC, Canada, Dec. 2014, pp. 2672–2680.
[174] B. Zhang, S. Zhao, and R. Zhang, "Towards adaptive semantic segmentation by progressive feature refinement," in *Proc. ICIP*, Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 2221–2225.
[175] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 2414–2423.
[176] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," *CoRR*, vol. abs/1909.11825, pp. 1–15, Sep. 2019.
[177] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. ICML*, Jul. 2020, pp. 1597–1607.
[178] K. Saito, D. Kim, S. Sclaroff, and K. Saenko, "Universal domain adaptation through self supervision," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 16282–16292.
[179] F. Barbato, M. Toldo, U. Michieli, and P. Zanuttigh, "Latent space regularization for unsupervised domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2835–2845.
[180] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1426–1435.
[181] S. Stan and M. Rostami, "Unsupervised model adaptation for continual semantic segmentation," 2020, *arXiv:2009.12518*.
[182] M. Chen, H. Xue, and D. Cai, "Domain adaptation for semantic segmentation with maximum squares loss," in *Proc. ICCV*, Seoul, South Korea, Oct. 2019, pp. 2090–2099.
[183] B. Xie, S. Li, M. Li, C. H. Liu, G. Huang, and G. Wang, "SePiCo: Semantic-guided pixel contrast for domain adaptive semantic segmentation," 2022, *arXiv:2204.08808*.
[184] Y. Zou, Z. Yu, X. Liu, B. V. K. V. Kumar, and J. Wang, "Confidence regularized self-training," in *Proc. ICCV*, Seoul, South Korea, Oct. 2019, pp. 5982–5991.
[185] K. Mei, C. Zhu, J. Zou, and S. Zhang, "Instance adaptive self-training for unsupervised domain adaptation," in *Proc. ECCV*, Aug. 2020, pp. 415–430.
[186] G. Li, G. Kang, W. Liu, Y. Wei, and Y. Yang, "Content-consistent matching for domain adaptive semantic segmentation," in *Proc. ECCV*, Aug. 2020, pp. 440–456.
[187] Z. Zheng and Y. Yang, "Unsupervised scene adaptation with memory regularization in vivo," 2019, *arXiv:1912.11164*.
[188] I. Chung, J. Yoo, and N. Kwak, "Exploiting inter-pixel correlations in unsupervised domain adaptation for semantic segmentation," 2021, *arXiv:2110.10916*.
[189] J. Huang, S. Lu, D. Guan, and X. Zhang, "Contextual-relation consistent domain adaptation for semantic segmentation," in *Proc. ECCV*, Glasgow, U.K., Aug. 2020, pp. 705–722.
[190] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 3929–3938.
[191] X. Liu, Z. Guo, S. Li, F. Xing, J. You, C.-C.-J. Kuo, G. El Fakhri, and J. Woo, "Adversarial unsupervised domain adaptation with conditional and label shift: Infer, align and iterate," in *Proc. ICCV*, Oct. 2021, pp. 10367–10376.
[192] D. Guan, J. Huang, S. Lu, and A. Xiao, "Scale variance minimization for unsupervised domain adaptation in image segmentation," *Pattern Recognit.*, vol. 112, Apr. 2021, Art. no. 107764.
[193] M. N. Subhani and M. Ali, "Learning from scale-invariant examples for domain adaptation in semantic segmentation," in *Proc. ECCV*, Glasgow, U.K., Aug. 2020, pp. 290–306.
[194] Z. Wang, X. Liu, M. Suganuma, and T. Okatani, "Cross-region domain adaptation for class-level alignment," 2021, *arXiv:2109.06422*.
[195] Y. Wang, J. Peng, and Z. Zhang, "Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation," in *Proc. ICCV*, Oct. 2021, pp. 9092–9101.
[196] T.-D. Truong, C. N. Duong, N. Le, S. L. Phung, C. Rainwater, and K. Luu, "BiMaL: Bijective maximum likelihood approach to domain adaptation in semantic scene segmentation," in *Proc. ICCV*, Oct. 2021, pp. 8548–8557.
[197] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon, "Unsupervised intra-domain adaptation for semantic segmentation through self-supervision," in *Proc. CVPR*, Seattle, WA, USA, Jun. 2020, pp. 3764–3773.
[198] J. Yang, R. Xu, R. Li, X. Qi, X. Shen, G. Li, and L. Lin, "An adversarial perturbation oriented domain adaptation approach for semantic segmentation," in *Proc. AAAI*, 2020, vol. 34, no. 7, pp. 12613–12620.
[199] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, pp. 1106–1120, Jan. 2021.
[200] W. Xu, Z. Wang, and W. Bian, "Unsupervised domain adaptation with implicit pseudo supervision for semantic segmentation," 2022, *arXiv:2204.06747*.
[201] U. Michieli, M. Biasetton, G. Agresti, and P. Zanuttigh, "Adversarial learning and self-teaching techniques for domain adaptation in semantic segmentation," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 3, pp. 508–518, Sep. 2020.
[202] T. Spadotto, M. Toldo, U. Michieli, and P. Zanuttigh, "Unsupervised domain adaptation with multiple domain discriminators and adaptive self-training," 2020, *arXiv:2004.12724*.
[203] T. Shen, D. Gong, W. Zhang, C. Shen, and T. Mei, "Regularizing proxies with multi-adversarial training for unsupervised domain-adaptive semantic segmentation," 2019, *arXiv:1907.12282*.
[204] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 7472–7481.
[205] Y. Liu, W. Zhang, and J. Wang, "Source-free domain adaptation for semantic segmentation," 2021, *arXiv:2103.16372*.
[206] Y. Zhang and Z. Wang, "Joint adversarial learning for domain adaptation in semantic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 6877–6884. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6169/6025
[207] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. P. Pérez, "DADA: Depth-aware domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 7364–7373. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/papers/Vu_DADA_Depth-Aware_Domain_Adaptation_in_Semantic_Segmentation_ICCV_2019_paper.pdf

[208] Z. Yan, X. Yu, Y. Qin, Y. Wu, X. Han, and S. Cui, "Pixel-level intra-domain adaptation for semantic segmentation," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 404–413.

[209] S. Cicek, N. Xu, Z. Wang, H. Jin, and S. Soatto, "Spatial class distribution shift in unsupervised domain adaptation: Local alignment comes to rescue," in *Proc. ACCV*, Kyoto, Japan, Dec. 2020, pp. 1–16.

[210] H. Tang, X. Zhu, K. Chen, K. Jia, and C. L. P. Chen, "Towards uncovering the intrinsic data structures for unsupervised domain adaptation using structurally regularized deep clustering," 2020, *arXiv:2012.04280*.

[211] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proc. CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 2517–2526.

[212] K. Zhang, Y. Sun, R. Wang, H. Li, and X. Hu, "Multiple fusion adaptation: A strong framework for unsupervised semantic segmentation adaptation," 2021, *arXiv:2112.00295*.

[213] Y. Xu, B. Du, L. Zhang, Q. Zhang, G. Wang, and L. Zhang, "Self-ensembling attention networks: Addressing domain shift for semantic segmentation," in *Proc. AAAI*, Honolulu, HI, USA, Jan. 2019, pp. 5581–5588.

[214] Q. Zhang, J. Zhang, W. Liu, and D. Tao, "Category anchor-guided unsupervised domain adaptation for semantic segmentation," 2019, *arXiv:1910.13049*.

[215] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," 2018, *arXiv:1805.10180*.

[216] A. Bruhn and J. Weickert, "Towards ultimate motion estimation: Combining highest accuracy with real-time performance," in *Proc. ICCV*, vol. 1, 2005, pp. 749–755.

[217] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Inf. Fusion*, vol. 76, pp. 243–297, Dec. 2021.

[218] Z. Lu, Y. Yang, X. Zhu, C. Liu, Y.-Z. Song, and T. Xiang, "Stochastic classifiers for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9111–9120.

[219] X. Guo, C. Yang, B. Li, and Y. Yuan, "MetaCorrection: Domain-aware meta loss correction for unsupervised domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3927–3936.

[220] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, Portland, OR, USA, Aug. 1996, pp. 226–231.

[221] J. Niemeijer and J. P. Schäfer, "Domain adaptation and generalization: A low-complexity approach," in *Proc. 6th Conf. Robot Learn.*, vol. 205, K. Liu, D. Kulic, and J. Ichnowski, Eds., Dec. 2022, pp. 1081–1091. [Online]. Available: https://proceedings.mlr.press/v205/niemeijer23a.html

[222] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–9.

[223] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[224] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[225] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, no. 10, pp. 1–41, Jan. 2022.

[226] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. ICCV*, Oct. 2021, pp. 568–578.

[227] J. Gu, H. Kwon, D. Wang, W. Ye, M. Li, Y.-H. Chen, L. Lai, V. Chandra, and D. Z. Pan, "Multi-scale high-resolution vision transformer for semantic segmentation," in *Proc. CVPR*, Jun. 2022, pp. 12094–12103.

[228] R. Shao, Z. Shi, J. Yi, P.-Y. Chen, and C.-J. Hsieh, "On the adversarial robustness of vision transformers," 2021, *arXiv:2103.15670*.

[229] M. M. Naseer, K. Ranasinghe, S. H. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Intriguing properties of vision transformers," in *Proc. NeurIPS*, Dec. 2021, pp. 23296–23308.

[230] X. Mao, G. Qi, Y. Chen, X. Li, R. Duan, S. Ye, Y. He, and H. Xue, "Towards robust vision transformer," in *Proc. CVPR*, New Orleans, LA, USA, Jun. 2022, pp. 12042–12051.

[231] Y. Bai, J. Mei, A. L. Yuille, and C. Xie, "Are transformers more robust than CNNs?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 26831–26843.

[232] Z. Wang, Y. Bai, Y. Zhou, and C. Xie, "Can CNNs be more robust than transformers?" 2022, *arXiv:2206.03452*.

[233] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. CVPR*, New Orleans, LA, USA, Jun. 2022, pp. 11976–11986.

[234] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" in *Proc. NeurIPS*, Dec. 2021, pp. 12116–12128.

[235] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," 2018, *arXiv:1811.12231*.

[236] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 801–818.

[237] L. Hoyer, D. Dai, and L. Van Gool, "HRDA: Context-aware high-resolution domain-adaptive semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 372–391.

[238] S. Ettedgui, S. Abu-Hussein, and R. Giryes, "ProCST: Boosting semantic segmentation using progressive cyclic style-transfer," 2022, *arXiv:2204.11891*.

[239] M. Vayyat, J. Kasi, A. Bhattacharya, S. Ahmed, and R. Tallamraju, "CLUDA: Contrastive learning in unsupervised domain adaptation for semantic segmentation," 2022, *arXiv:2208.14227*.

[240] Y. Du, Y. Shen, H. Wang, J. Fei, W. Li, L. Wu, R. Zhao, Z. Fu, and Q. Liu, "Learning from future: A novel self-training framework for semantic segmentation," 2022, *arXiv:2209.06993*.

[241] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[242] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth. (2020). *A2D2: Audi Autonomous Driving Dataset*. [Online]. Available: https://www.a2d2.audi

[243] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. CVPR*, Jun. 2020, pp. 1–14.

[244] K. B. Koh and B. Fernando, "Consistency regularization for domain adaptation," in *Computer Vision—ECCV 2022 Workshops*. Tel Aviv, Israel: Springer, 2023, pp. 347–359.

[245] M. Chen, Z. Zheng, Y. Yang, and T.-S. Chua, "PiPa: Pixel- and patch-wise self-supervised learning for domain adaptative semantic segmentation," 2022, *arXiv:2211.07609*.

[246] K. Wang, D. Kim, R. Feris, K. Saenko, and M. Betke, "Exploring consistency in cross-domain transformer for domain adaptive semantic segmentation," 2022, *arXiv:2211.14703*.

[247] L. Hoyer, D. Dai, H. Wang, and L. Van Gool, "MIC: Masked image consistency for context-enhanced domain adaptation," 2022, *arXiv:2212.01322*.

[248] L. Chen, Z. Wei, X. Jin, H. Chen, M. Zheng, K. Chen, and Y. Jin, "Deliberated domain bridging for domain adaptive semantic segmentation," 2022, *arXiv:2209.07695*.

[249] Y. Liu, J. Deng, J. Tao, T. Chu, L. Duan, and W. Li, "Undoing the damage of label shift for cross-domain semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7042–7052.

[250] PyTorch Contributors. (2022). *PyTorch Documentation: Reproducibility*. Accessed: Sep. 19, 2022. [Online]. Available: https://pytorch.org/docs/stable/notes/randomness.html

[251] G. Marcus, "Deep learning: A critical appraisal," 2018, *arXiv:1801.00631*.

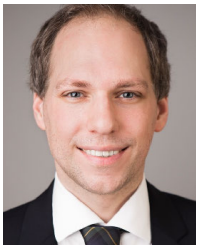
[252] L. T. Triess, M. Dreissig, C. B. Rist, and J. M. Zöllner, "A survey on deep domain adaptation for LiDAR perception," in *Proc. IV Workshops*, Jul. 2021, pp. 350–357.

MANUEL SCHWONBERG received the B.Eng. degree in automotive engineering from the Ostfalia University for Applied Sciences, in 2017, and the M.Sc. degree in robotics from the Technical University of Munich, in 2021. He is currently pursuing the Ph.D. degree with CARIAD SE, Munich. The dissertation is supervised by Prof. Hanno Gottschalk from the Chair of Mathematical Modeling of Industrial Life Cycles, TU Berlin. His research interests include unsupervised domain adaptation, domain generalization, and self-supervised learning methods for deep neural networks.

JOSHUA NIEMEIJER received the B.Sc. degree in medical informatics and the M.Sc. degree from the University of Lübeck, Germany, in 2015 and 2017, respectively. He is currently a Research Associate with DLR, Institute of Transportation Systems, Braunschweig, Germany. His research interests include domain adaptation/generalization and active learning approaches for neural networks. He applies this research to the real-world in the environmental perception of autonomous vehicles with DLR and in the analysis of medical image data. The latter research is done in context of a dissertation that is supervised by Dr. Heinz Handels with the Institute of Medical Informatics, University of Lübeck.

JAN-AIKE TERMÖHLEN (Graduate Student Member, IEEE) received the B.Sc. degree in industrial and electrical engineering and the M.Sc. degree in electrical engineering from Technische Universität Braunschweig, Germany, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the Faculty of Electrical Engineering, Information Technology, Physics. His research interests include domain generalization, unsupervised domain adaptation, and continuous adaptation approaches for neural networks with a focus on computer vision tasks. He was given the CVPR Workshop Best Paper Award, in 2019.

JÖRG P. SCHÄFER received the Dipl. degree in computer science and the Ph.D. degree from the Humboldt-University of Berlin, Germany, in 2011 and 2022, respectively. In 2011 and 2012, he gained valuable experience in research and development on distributed systems with Zuse-Institut, Berlin. He taught students with the Humboldt-University of Berlin and Technical University Braunschweig, from 2012 to 2023, in various fields, including computer science theory, database systems, compiler construction, and transportation systems. From 2015 to 2019, he worked on his Ph.D. thesis on theory and algorithms for high-dimensional time series retrieval. After that, he continued and extended his research activities with the Institute of Transportation Systems, DLR. He is currently leading research and development efforts in the area of perception for autonomous driving cars. His research interests include distributed systems, computer vision, and domain adaptation for neural networks.



NICO M. SCHMIDT received the B.Sc. degree in bioinformatics from Freie Universität Berlin, Germany, in 2008, the M.Sc. degree in computational neuroscience from Technische Universität Berlin, Germany, in 2011, and the Ph.D. degree in computer science from the University of Zurich, Switzerland, in 2015. He was worked on environment perception for automated driving with Carmeq Gmbh, Berlin, Germany, from 2015 to 2018. With Volkswagen AG, Wolfsburg, Germany, he was conducting research on AI technologies for automated driving (in the areas robustness, compression, domain adaptation, and scalability of deep neural networks), from 2018 to 2020. Since 2021, he has been with CARIAD SE on intelligent data collection and domain adaptation for data-driven development of autonomous driving functions.

HANNO GOTTSCHALK received the Diploma degrees in theoretical physics and in mathematics and the Ph.D. degree in mathematics from Ruhr University Bochum, in 1995, 1997, and 1999, respectively. From 2000 to 2001, he was a DAAD Fellow with University La Sapienza of Rome, before he became a Research Assistant and a Lecturer with the University of Bonn, in 2001 and 2005, respectively. After working with Siemens Energy, from 2007 to 2011, as a Core Competency Owner for Probabilistic Design, he became a Professor in stochastics with the University of Wuppertal, in 2011. In 2018, he co-founded the Interdisciplinary Center for Machine Learning and Data Analytics, University of Wuppertal. Since 2023, he has been the Chair for Mathematical Modeling of Industrial Life Cycles, Technical University Berlin (TU Berlin).

TIM FINGSCHEIDT (Senior Member, IEEE) received the Dipl.-Ing. degree in electrical engineering and the Ph.D. degree from RWTH Aachen University, Germany, in 1993 and 1998, respectively. He joined the AT&T Laboratories, Florham Park, NJ, USA, in 1998, and Siemens AG (Mobile Devices), Munich, Germany, in 1999. With Siemens Corporate Technology, Munich, he was leads the speech technology development activities, from 2005 to 2006. Since 2006, he has been a Full Professor with the Institute for Communications Technology, Technische Universität Braunschweig, Germany. His research interests include speech technology and vision for autonomous driving. He was a member of the IEEE Speech and Language Processing Technical Committee, from 2011 to 2018. He was a recipient of several awards, including the Vodafone Mobile Communications Foundation Prize, in 1999, and the 2002 ITG Prize of the Association of German Electrical Engineers (VDE ITG). In 2017 and 2020, he coauthored the ITG Award-Winning Publication. He was given the Best Paper Award of a CVPR Workshop, from 2019 to 2021. He has been the Speaker of the Speech Acoustics Committee ITG AT3, since 2015. He was an Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, from 2008 to 2010.