UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR MEDIZINISCHE INFORMATIK

Aus dem Institut für Medizinische Informatik
der Universität zu Lübeck
Direktor: Prof. Dr. rer. nat. habil. Heinz Handels

Optimizing Synthetic and Real Training Data Distributions for Deep Learning in Image Recognition

Inauguraldissertation
zur
Erlangung der Doktorwürde
der Universität zu Lübeck

Aus der Sektion Informatik/Technik

vorgelegt von
Joshua Niemeijer
aus Vechta

Lübeck, 2025

1. Berichterstatter: Prof. Dr. rer. nat. habil. Heinz Handels
2. Berichterstatter: Prof. Dr. rer. nat. Thomas Martinetz

Tag der mündlichen Prüfung: 19.11.2025

Zum Druck genehmigt. Lübeck, den 12. Dezember 2025

Abstract

The recent advances in deep learning have enabled a large variety of applications. Among these are, for example, the environment perception of robots, including self-driving cars, and medical image analysis, which helps identify medical conditions or planning treatment. To build deep learning systems that generalize well, large quantities of relevant human-labeled data must be available for training.

This requirement introduces several challenges. Annotations are costly due to the large amounts of data that need to be labeled and the complex nature of the annotation process. This is made more complex by the fact that relevant data needs to be recorded before data can be labeled. Depending on the field of application, this can be challenging. The challenge arises because relevant data is seldom available, which introduces the need to capture large quantities of data to find rare but critical cases.

The work investigates a more efficient use of manual annotation through intelligent data selection for labeling, utilizing active learning (AL). In this context, semi-supervised learning (SSL), which aims to replace manual annotation, is utilized. The thesis investigates the use of synthetic data to replace the acquisition of data itself. The work presents strategies to guide the generation process towards creating rare but critical data. Finally, it is shown how to utilize these insights to create models that generalize well toward unseen distributions with minimal human intervention.

For each of these methodologies, the thesis contributes novel approaches and analyses. It is shown that the choice of active learning approaches is highly dependent on the type of distribution the selection is performed on and the annotation budget. Next, the work shows how AL and semi-supervised learning are effectively integrated. This insight shows how to develop best practices for the application of AL and SSL. For the use of SSL in adapting networks to novel data domains, this work provides an extensive review of this dynamic field and derives novel low-complexity methods from it. These methods prove useful in their application to the environment perception of autonomous vehicles and the medical domain, as well as for adapting from synthetic to real data. The work provides novel methods for the targeted creation of synthetic data. Building on the creation of synthetic data and the research on SSL, the thesis presents an approach for generalizing to unseen domains.

Overall, this thesis provides solutions for minimizing the cost and human effort involved in annotating and acquiring relevant data. The solutions provide efficient adaptation and generalization to new domains and distributions.

Zusammenfassung

Die Fortschritte im Bereich Deep Learning bzw. der tiefen neuronalen Netzwerke (DNNs) haben eine Vielzahl an Anwendungen ermöglicht. Beispiele hierfür sind die Umfeldwahrnehmung im Bereich der Robotik bzw. der autonomen Fahrzeuge sowie die Analyse medizinischer Bilddaten. Um Deep Learning Systeme zu erschaffen, welche gut generalisieren, werden sehr große von Menschen annotierte Datenmengen benötigt.

Durch diese Voraussetzung ergeben sich verschiedene Herausforderungen. Der Annotierungsprozess ist teuer, was sich durch die erforderliche große Menge und die zuweilen komplexe Natur der Annotation ergibt. Zudem lohnt es sich nur relevante Daten zu annotieren. Da allerdings die Relevanz eines Datenpunktes nicht zuletzt durch seine Seltenheit gegeben ist, ist die Aufnahme solcher Daten ein großes Problem. Dies führt häufig zu großen aufwendigen Messkampagnen. Diese Arbeit führt zunächst Methoden ein, welche die manuelle Annotation effizienter machen, indem die wertvollsten Daten ausgewählt werden. Solche Methoden werden unter dem Begriff Active Learning (AL) zusammengefasst. In diesem Zusammenhang wird auch das unüberwachte Lernen auf nicht annotierten Daten eingeführt, welches den menschlichen Aufwand weiter reduziert und als Semi-Supervised Learning bezeichnet (SSL) wird. Um das Problem anzugehen, dass relevante Daten nur schwer aufzunehmen sind, wird in dieser Arbeit zudem der Einsatz von synthetischen Daten und deren gezielte Generierung untersucht. Die gesammelten Erkenntnisse werden dazu eingesetzt, um Netzwerke zu trainieren, die gut auf neue Anwendungsdomänen und Datenverteilungen generalisieren. Für jede der genannten Felder führt diese Arbeit dabei neue Ansätze und Analysen ein. Es wird gezeigt, dass die Wahl des AL Ansatzes stark von den gegebenen Verteilungen und dem Annotierungsbudget abhängt. Weiter wird gezeigt, wie man SSL effektiv in das AL integriert um die menschliche Annotation weiter zu reduzieren. Das Feld des SSL wird dabei noch einmal spezieller im Bereich der Anpassung von Netzwerken auf neue Anwendungsdomänen betrachtet. Dieses Forschungsfeld wird in seinen Ansätzen analysiert und reflektiert. Diese Erkenntnisse werden dann verwendet, um neue Ansätze mit geringer Komplexität einzuführen. Weiter führt diese Arbeit Ansätze ein, welche die Generierung von seltenen aber kritischen Ereignissen ermöglichen und somit den Bedarf für die Datenaufnahme reduzieren. In den Kapiteln wird auf Anwendungen in der Medizin und der Umfeldwahrnehmung eingegangen.

Diese Arbeit trägt also zu neuen Lösungen bei, welche die effiziente Generierung von synthetischen und realen Datensätzen ermöglichen. Dies erlaubt, Netzwerke zu trainieren, die gut auf neue Anwendungsdomänen adaptieren und generalisieren.

Contents

Chapter 1

Introduction

1.1 Motivation

The recent advances in computer vision systems have allowed for significant improvements in a diverse variety of applications. Such applications e.g. include infrastructure monitoring [95, 119], big data analysis [118] or localization [8]. Especially notable among such applications are the environment perception of robotic systems or the medical image analysis [49, 202]. The advent of high-quality object detection and segmentation systems, for example, allowed for the development of self-driving cars capable of autonomy in certain operational design domains. For example, semantic segmentation, a pixel-wise classification of an image, provides localized information about the shape of the drivable area and objects like cars or pedestrians. In medical image processing, semantic segmentation or image classification gained traction for automatically identifying medical conditions or anatomies in medical image data. For example, the classification of histological images facilitates the identification of pathologies or the segmentation of retinal fluids in OCT images and allows for predicting illnesses like AMD.

These systems are usually based on so-called deep neural networks (DNNs) and provide high-quality predictions for tasks like image classification, object detection, or semantic (instance) segmentation. Such systems that discriminate between classes are called discriminative artificial intelligence (AI) systems. To be used in applications such as those mentioned above, these systems should:

- Provide high-quality predictions and
- Be robust towards changes in the input distributions.

In the case of autonomous vehicles, a model trained on images recorded on a sunny day should work well on this domain and distributions of images from novel domains, e.g., rainy or night images. A common challenge in analyzing medical images occurs if the training data is not recorded from the same image sensor that the model is applied to (e.g., different MRT scanners). Therefore, robustness towards these novel data distributions is crucial in this application, also.

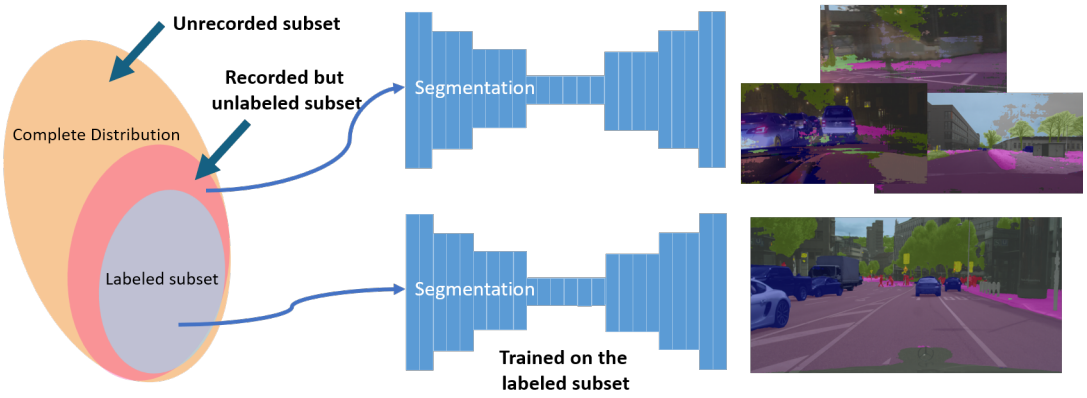


Figure 1.1: DNNs that were trained on a certain distribution of data achieve high quality on samples from the same distribution (c.f. bottom) but struggle on unknown parts of the relevant distribution (c.f. top).

Figure 1.1, however, shows the detrimental effect that data that was not part of the original training distribution can have on the performance of a DNN for semantic segmentation. Apart from the network architecture, the most decisive factor in achieving a model that generalizes well is the availability of large quantities of high-quality annotated data for training. Creating relevant datasets is a challenging task.

1. **Annotations are costly:** This problem arises due to the large amounts of data that need to be labeled and the complex nature of the annotation.

This problem is especially prominent in the field of medical image processing. The annotation of medical data requires specialists like medical doctors, which are costly and seldom available in large quantities. In autonomous driving, most people can do data annotation. Even though the qualification of annotators can be lower, the effort to label an image is often high and, in the case of semantic segmentation, can, e.g., take up to 90 minutes for complex scenes c.f. Cordts et al. [36].

2. **Data acquisition can be challenging and expensive:** Depending on the field of application, the acquisition process of relevant data is costly, or data privacy regulations impede it. In most applications, the acquisition processes yield data that is redundant. Adding value to the training dataset requires annotating data that is novel w.r.t. the existing distribution.

Given the use of measurement campaigns to record new data to create datasets for autonomous vehicles, much of the recorded data will likely be redundant. This redundancy has two primary reasons. On the one hand, the scene composition in driving environments is restricted. On the other hand, it is unlikely that completely novel scenarios occur since if they were likely to occur, they would probably already be contained in the preexisting training data. Designing measurement campaigns to record

rare data, e.g., driving in under-explored scenarios, helps somewhat. However, the problem is that identifying the challenging cases a priori is impossible. All of these issues are also common in medical image processing. Additionally, data privacy regulations exist in this context, meaning that not all recorded data can be used for annotation. The image data acquisition process is more expensive and less planable than the use case of autonomous driving. The lack of control over image recording results from the cost of medical imaging devices, such as MRTs or CTs, during use. The high costs make deliberate measurement campaigns difficult, so data is mainly produced as a byproduct of hospital and clinical practice operations.

Therefore, this work serves the purpose of taking the first steps to solve the two problems (challenges) identified. To tackle problem 1. it contributes to the active learning (AL) and semi-supervised learning (SSL) literature. That means this thesis provides solutions that help to intelligently select the most meaningful data from a given pool (active learning) of non-annotated data for human labeling and even provides solutions for utilizing parts of the data that were not annotated before (semi-supervised learning). Additionally, the work contributes to solving problem 2. by utilizing synthetic data to substitute real-world data. This work presents novel methods for creating synthetic data representing relevant missing parts of the training distribution. This thesis, therefore, creates robustness in the trained DNN, even for rare but critical scenarios that are hard to record during real-world measurement campaigns.

The solutions of this work could be applied to the scenario of a large car company trying to create a robust perception system for a self-driving car. Given large unlabeled pools of data that the company recorded from measurement campaigns or the live operation of their car fleet, the best practices in active learning from Chapter 2 provide a way of selecting an optimal subset of images for labeling. Semi-supervised learning methods described and introduced in Chapter 2 and 3 allow the utilization of the remaining unlabeled part through unsupervised training and adapting to novel domains or environments that were not part of the original training distribution. Failure cases remain even after utilizing the available data through optimal labeling and unsupervised learning. The solutions provided in Chapter 4 allow for creating labeled data specifically for such rare but critical scenarios in the synthetic world. That means that the car manufacturer can spend less money on measurement campaigns to record or stage such data, a task that would otherwise be difficult since the perception system's failure cases seldom follow human categories and are usually unknown. Finally, Chapter 5 shows how to utilize such synthetic data in combination with data-driven generative models to create a training distribution that allows for training image recognition networks that generalize well to unseen data distribution.

A company providing DNN-based software for classifying or segmenting pathologies could utilize the solutions in a similar fashion but with a different emphasis on the challenges. The challenges would be similar, but the importance of the solutions would be different. Since data, generally speaking, is less available, synthetic data, i.e.,

Chapter 4, becomes more important, and since data annotation is a lot harder due to the demand for medical professionals, the selection of the correct labels is even more important, i.e., Chapter 2.

Therefore, in the following chapters, this thesis will describe methods of optimizing synthetic and real training data distributions for deep learning in image recognition. This work aims to achieve progress in constructing the methodology for the adaptation and generalization of DNNs to new domains and distributions.

1.2 Scope and Objectives of the Work

This work aims to achieve progress in constructing the methodology for the adaptation and generalization of DNNs to new domains and distributions. As Figure 1.1 on page 2 underlines, the given training distribution for DNNs in most cases only represents a small subset of the complete distribution relevant for training. Figure 1.1 illustrates the consequence, i.e., a decline in performance on data that was not part of the original distribution. The following two challenges introduce the limitation.

1. **Annotations are costly**

2. **Data acquisition can be challenging and expensive**

This work presents two interesting applications: medical image processing and the environment perception of autonomous cars. Challenges 1 and 2 are especially severe in medical image processing. Annotation can only be done by costly medical professionals, and data acquisition itself is difficult due to privacy regulations and often costly scanning processes (e.g., MRT). Even though the annotation of street scenes is time-consuming (90 minutes per semantic segmentation [36]), for the environment perception in autonomous driving, the largest problem is the acquisition of relevant data (problem 2.). Recording relevant data is a difficult problem because the temporal data of driving scenes is inherently redundant. This means that huge amounts of data need to be acquired to find novel and relevant data.

This work aims to make progress in all of the challenges. From a methodological point of view, the chapters provide novel approaches and analyses for the following fields,

- Active Learning
- Semi-Supervised / Unsupervised Learning
- Synthetic Data

Active Learning (AL) → challenge 1: Active learning refers to a branch of methods that aims to select the optimal subset of images from a given unlabeled data pool.

Optimal means that the selected images, when labeled, improve the performance of the network that is trained on them maximally. Selecting an optimal subset achieves a better performance of the trained DNN with the same labeling effort, thereby saving costs over a less ideal image selection. The method for selecting the images for labeling is called the acquisition function. This work aims to analyze the connection between the distribution of the given unlabeled data, the annotation budget, and the properties of the acquisition function. The aim is to define best practices for active learning.

Semi-Supervised Learning (SSL) → challenge 1: Semi-supervised learning can alleviate human labeling. SSL consists of a parallel supervised training on a given labeled set of data and an unsupervised training on unlabeled data. Often, self-inferred labels are used for the unsupervised training part. This way, training and adapting to new distributions without human labeling is possible. This work investigates two applications of SSL. On the one hand, it analyzes how different active learning methods interact with SSL for the optimal use of semi-supervised learning. On the other hand, it investigates the use of SSL in overcoming structural changes between a labeled set of data and the unlabeled distribution. I.e., the use of SSL for unsupervised domain adaptation is investigated.

Synthetic Data → challenge 2: Synthetic data can be created by simulation engines or data-driven generative models. The created data is already annotated without human intervention, and it is possible to control what kind of data is simulated. These are attractive properties for solving challenge 2 since real-world data acquisition is largely defined by chance. Since the scenarios and properties of the generated data can be controlled, the generation process can also be directed toward producing "relevant" data. This control allows for replacing the untargeted recording of unlabeled data. Additionally, the image recording process is almost free, a property that is especially attractive for medical image processing. The generation of synthetic data, however, comes with two challenges that this work addresses: There is a large appearance gap between simulated and real-world data, and it is unclear how to find the parametrization to create meaningful data.

Combined, the approaches in this work provide a framework for minimizing human intervention in creating training datasets to train models that generalize well to domains and distributions. Active learning yields an effective selection of training data, semi-supervised learning allows us to utilize recorded unlabeled data for training, and synthetic data can be used to create targeted data for the missing parts of the relevant training distribution. Either image classification or semantic segmentation is utilized in the experiments to demonstrate the effectiveness of the provided solutions. The created systems are applied to the environment perception of autonomous vehicles and medical image processing.

1.3 Contributions

This work is divided into five chapters. The next four chapters, "Optimizing Real Datasets with Active and Semi-Supervised Learning," "Reducing Manual Annotation Effort with Unsupervised Domain Adaptation," "Optimizing Synthetic Datasets by Targeted Image Generation," and "Knowledge-Based Optimization of Synthetic Data for Real-World Domain Generalization" provide novel ideas and analysis to overcome the three challenges in adapting and generalizing to new domains and distributions. The following points describe the content of these chapters. Additionally, the section introduces the contributions of the papers on which these chapters are based. For the papers, I highlight the work done by my co-authors. The content of the papers that is not explicitly mentioned as provided by others is my contribution.

- **Chapter 2** analyzes the current state of the art of active learning (AL) w.r.t. to dimensions that were under-explored: Data distribution (diverse vs. redundant), size of acquired batch size (low high), and application to different data domains (medical, car). The chapter shows that different combinations of these dimensions require different kinds of acquisition functions (batch-based vs. single sample selection). Given that the current state of the art highly optimizes its methods on the high diversity distribution and high budget use case, the gained insights help steer the field towards more realistic benchmarks (of which this work proposes one) and enable the development of more suitable approaches. Since the purpose of active learning is to reduce manual labeling efforts, the integration of semi-supervised learning is an interesting extension. This research investigates the integration of SSL into different types of AL functions under the given redundancy and batch size variations. It also identifies the best combinations and provides theoretical background for the synergistic effects observed in the integration of SSL with certain types of AL functions. The general results of this work are novel best practices for the field of active learning and semi-supervised learning. This chapter is derived from my previously published works, specifically "Best Practices in Active Learning for Semantic Segmentation"[158], which won the best paper award at GCPR 2023, and "Realistic Evaluation of Deep Active Learning for Image Classification and Semantic Segmentation"[159]. This work was created in cooperation with Sudhanshu Mittal, who is an equal contribution co-author in these papers. We co-designed the scenarios and experiments and co-wrote the paper. My contributions had a stronger emphasis on identifying a more realistic evaluation of AL in real-world scenarios containing different levels of redundancies, and his contributions came from the perspective of integrating SSL in high and low-budget scenarios.

- **Chapter 3** provides the most extensive overview of the rapidly growing field of unsupervised domain adaptation (UDA), and it reflects the current direction of the field. This part of the thesis is based on the paper "Survey on Unsupervised Do-

main Adaptation for Semantic Segmentation for Visual Perception in Automated Driving"[201] that is the result of a collaboration with my equal contribution co-authors Manuel Schwonberg and Jan-Aike Termöhlen. My focus was on the feature space adaptation methods, Manuel focused on the output space adaptation, and Jan-Aike focused on the output space adaptation. Building on the overview of the UDA research field coming from the survey, this thesis provides several novel approaches with applications in medical image analysis and autonomous driving. The work presented in this context is based on the papers "Combining Semantic Self-Supervision and Self-Training for Domain Adaptation in Semantic Segmentation" [165] and "Overcoming the Sensor Delta for Semantic Segmentation in OCT Images"[168]. This work designs and analyzes a feature space adaptation method that is based on clustering the target domain feature space representations towards their corresponding class centroids in the source domain. Contributions were made by Jan Ehrhardt and Jörg P. Schäfer, who helped edit the final papers. Jan Ehrhardt helped in the design of the medical experiments by, among other things, providing and selecting the necessary medical data. This thesis builds on this approach, focusing on overcoming the appearance gap between synthetic and real data. That is because synthetic data does not require human labeling, and the generation process can be controlled to yield relevant scenarios. Therefore, this work provides a low-complexity method for this scenario and analyzes the effects of this method on the generalization to real-world data. The content presented in this thesis is based on the paper "Domain Adaptation and Generalization: A Low-Complexity Approach" [166]. Contributions were made by Jörg P. Schäfer, who helped in editing the final paper.

- **Chapter 4** explores the creation of synthetic data itself. Synthetic data can be created utilizing different strategies: 1. Simulation engines 2. Generative models. The chapter provides strategies for utilizing these options optimally. For simulation engines, the thesis showcases how to construct acquisition functions that score the value of generated synthetic data for training a model for a specific target domain. This content is derived from the previously published work "Synthetic Dataset Acquisition for a Specific Target Domain"[169]. Sudhanshu Mittal is a co-author, who helped with the implementation and execution of the experiments. For generative models, this thesis contributes a novel method to guide the generation process directly to generate images that represent missing parts of the original training distribution. The content presented in this thesis is based on the paper "TSynD: Targeted Synthetic Data Generation for Enhanced Medical Image Classification" [170, 172], which won the best paper award at the "Simulation and Synthesis in Medical Imaging" workshop at MICCAI 2024 and the best Poster Award at BVM 2025. Co-authors include Jan Ehrhardt and Hristina Uzunova. They helped select and train the auto-encoder model, execute the experiments, and write the paper.

- **In Chapter 5**, this thesis leverages generative models and semi-supervised learning to achieve a trained model that generalizes well to unseen domains. The chapter contributes a novel approach that shows how to utilize synthetic data created by simulation engines and generative models to transform such data into a more realistic and diverse distribution. The work utilizes methods of unsupervised domain adaptation to leverage the created synthetic data to train models that set a new benchmark in the task of domain generalization. This chapter is derived from my previously published works, specifically "Generalization by Adaptation: Diffusion-Based Domain Extension for Domain-Generalized Semantic Segmentation" [171]. Co-authors include Jan-Aike Termöhlen and Manuel Schwonberg, who helped run the experiments and write and edit the paper.
- **Chapter 6** provides a summary and reflection of the achieved contributions for solving the problems of adapting and generalizing to new domains and distributions by optimizing synthetic and real training data distributions for deep learning in image recognition. Finally, given the new state, this work proposes future work.

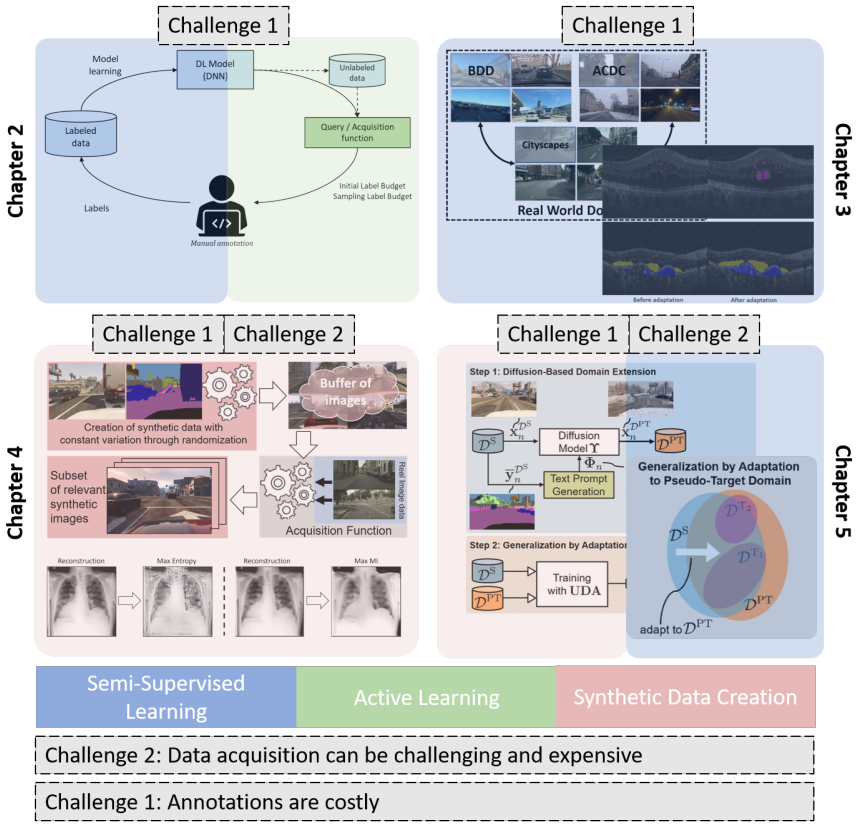


Figure 1.2: Chapters addressing challenges 1 and 2 with SSL, AL and synthetic data.

Chapter 2

Optimizing Real Datasets with Active and Semi-Supervised Learning

This chapter is partially derived from my previously published works, specifically "Best Practices in Active Learning for Semantic Segmentation" [158] and "Realistic Evaluation of Deep Active Learning for Image Classification and Semantic Segmentation"[159]. Some text, figures, and findings have been re-utilized or adapted from these publications. Co-Authors of these Publications are Sudhanshu Mittal, Jan Ehrhardt, Özgün Çiçek, Maxim Tatarchenko, Jörg P. Schäfer, Heinz Handels and Thomas Brox.

2.1 Introduction and Motivation

Deep Neural Networks (DNNs) require annotated datasets for training. In the first step, the annotation process requires recording sensor data that can be labeled. Therefore, car manufacturers build large unlabeled pools of data from measurement campaigns or the live operation of their car fleet. Similarly, image data from hospitals or medical practices can be recorded and saved into such data pools to build a foundation for creating real-world datasets for training DNNs. In both cases, and especially in the case of medical images, annotation is costly. In the case of medical images, it can only be done by professionals (c.f. problem 1 on Chapter 1). Therefore, only a fraction of the recorded data can and should be annotated. This restriction of annotating only a subset from the data pool makes the selection process that determines this subset crucial. This chapter deals with active learning (AL), which allows the selection of the most relevant data points to reduce the manual annotation effort. Additionally, this chapter explores how the non-selected part of the given data pools can be utilized through unsupervised training. Therefore, this chapter additionally explores semi-supervised learning (SSL).

Active learning (AL): Figure 2.1 on page 10, which depicts the active learning circle, illustrates the general principle of active learning. A subset selection is performed based on the pool of unlabeled data. The subset is then annotated, and the model can be trained. Active learning (AL) is the selection of relevant data for annotation. This general process leaves the question of selecting the relevant data from the set of

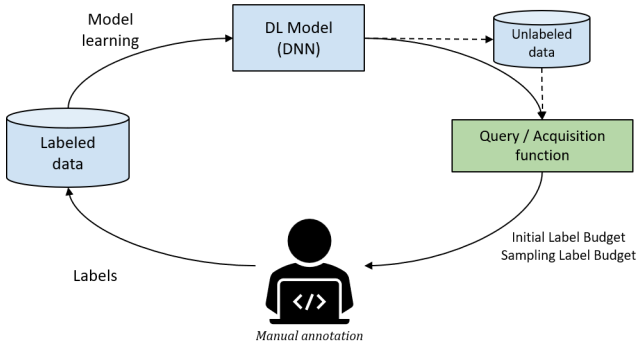


Figure 2.1: The active learning circle: The DNN is trained on the existing labeled data set. The unlabeled data is analyzed and scored by the acquisition function. An image's score reflects the value of training the DNN on it. The acquisition function selects a batch of images, which are then manually labeled and added to the training set. Given the updated training set, the circle starts again.

unlabeled data for annotation. The active learning literature describes approaches to create an intelligent selection of data [159]. The aim is to define a so-called acquisition function that selects the images/data points from the unlabeled data pool that are the most relevant w.r.t. the current state of the DNN. Most relevant, in most cases, means that the subset of data yields the most significant improvement in performance for the DNN when it is labeled and added to the existing labeled training set. The acquisition function selects a batch of b data points (images) in each iteration of the circle. The AL circle is performed until the global labeling budget of B is exhausted.

Semi-supervised learning (SSL) allows the utilization of unlabeled data for training neural networks. During SSL, the subset chosen by utilizing AL is used for supervised training, and the remaining unlabeled subset is used for unsupervised training. Such unsupervised training can consist of, e.g., the utilization of self-inferred labels.

Hence, both AL and SSL reduce the manual annotation effort. Therefore, they provide a solution for the first challenge identified in Chapter 1. This chapter first introduces the fundamentals of active learning and semi-supervised learning. The second part of the chapter provides an analysis of the integration of these tools. It analyzes how AL and SSL interact with each other while using different types of data distributions and annotation budgets. This chapter investigates the application of medical image processing and the environment perception of autonomous vehicles. Small annotation budgets are an especially interesting use case for medical image processing, as costly medical professionals must be employed for annotation. When annotating datasets for the environment perception of self-driving cars, data distributions are usually redundant, but the annotation budget is larger. This chapter results in new best practices for integrating AL and SSL in a variety of relevant real-world scenarios.

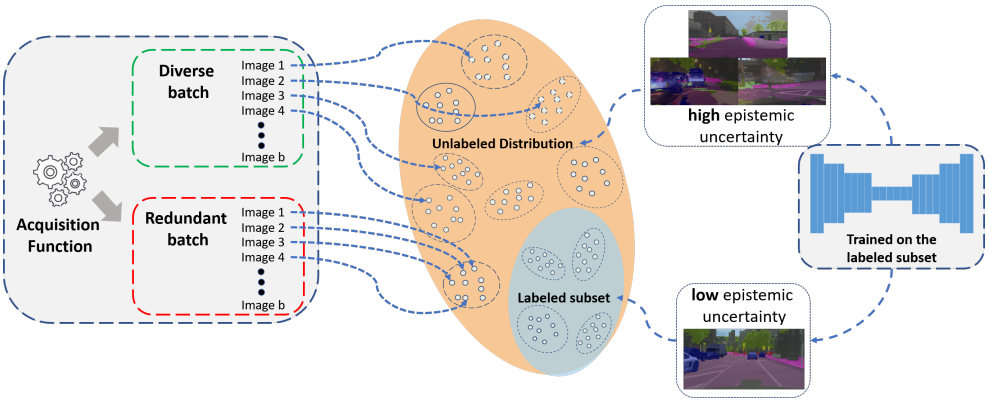


Figure 2.2: Illustration of the two major objectives for the acquisition function to select images from the unlabeled distribution. 1. The selected images should represent a missing part of the training distribution. The epistemic uncertainty is high if the network is applied to samples that were not part of the training distribution. 2. Each image in the selected batch should represent new information w.r.t. the other selected images, resulting in diverse batch.

2.2 Active Learning: Acquisition Functions

There are two main objectives that are relevant in scoring the value a data point contributes to the training distribution when labeled and added:

1. It should represent a missing part of the training distribution

2. It should be unique w.r.t. to the other samples selected in the current batch

Single sample acquisition functions: Select the top b samples (images) individually. The relevance score of the given data point is computed independently of the other already selected samples. Such a score is usually an estimate of the epistemic uncertainty [100]. The **epistemic uncertainty** measures how well a given data point is represented in the training distribution. As Figure 2.2 indicates, a trained DNN, therefore, yields a low epistemic uncertainty for images represented in the given training distribution and a high epistemic uncertainty for novel data that was not represented in the training distribution. The epistemic uncertainty of a model can be reduced by adding the data that was not yet represented in the training distribution. This distinguishes it from aleatoric uncertainty (data uncertainty), which measures the given data's ambiguity and can not be reduced by adding more data. Epistemic uncertainty, hence, is a measure of the first point of the desired features of an acquisition function.

The simplest single sample selection methods utilize the entropy [205] over the output probabilities to score the epistemic uncertainty. Even though it does not provide a good measure of epistemic uncertainty from a theoretical point of view, it often provides

a strong baseline as a score for the acquisition function. Further methods that estimate the epistemic uncertainty based on the output probabilities include EquAL [64], Ensemble+AT [112], CEAL [240]. Methods like BALD [76] and DBAL [54] use Monte Carlo Dropout [53] (MCD) to estimate the epistemic uncertainty. The MCD method is motivated by Bayesian deep learning and represents a better approximation of the epistemic uncertainty. Methods like DAAL [243], VAAL [212], and WAAL [209] select the most representative samples, i.e., samples that are not covered by the training distribution. To score this representativeness, they, e.g., utilize an auxiliary network trained in an adversarial manner. This work utilizes the approaches Entropy, EquAL, and BALD to represent single-sample acquisition methods. Section 2.4.4 further describes their methodology and implementation.

A problem that arises from selecting samples independent of each other is that samples with redundant information might be selected. Given the current state of the DNN trained on the labeled set, the epistemic uncertainty will be similar for similar images. Given two similar image, if the score is high for one, it is high for the other, as well, and therefore, both are selected, which leads to the selection of redundant images. As Figure 2.2 on page 11 illustrates, single-sample acquisition functions would hence select clusters of images. A data pool with many redundant samples to select from results in the selection of redundant samples. This problem of redundant selection is also known as the **mode collapse problem**. In theory, it could be solved by updating the DNN's uncertainties through retraining after each new image is selected. Due to the large datasets required and hardware constraints, this solution is not feasible. Batch-based acquisition functions aim to mitigate the mode collapse problem.

Batch-based acquisition functions: compute a cumulative information gain score of the whole batch of b selected samples. Since redundant samples do not increase the information gain, unique samples are selected. This way, batch-based acquisition functions address the second objective of acquisition functions (c.f. Figure 2.2).

Sener et al. introduced the CoreSet [203] approach for selecting diverse batches. The CoreSet algorithm selects new samples with respect to the already selected set by minimizing the distance to the farthest neighbor. The distance is hereby computed based on the latent space representations of the samples in the trained DNN. The BatchBALD [107] approach computes the joint mutual information between the whole batch and the model parameters. The k-MEANS++ [288] approach utilizes the k-means approach to compute the b cluster centers representing the unlabeled data. The closest samples to the respective b cluster centers are chosen. Further methods include GLISTER [101] and ADS [96]. This work selects the Coreset method to represent batch-based methods due to its effectiveness, simplicity, and easy scalability to the segmentation task (see Section 2.4.4 for further details).

The problem with batch-based methods, generally speaking, is the combinatorial explosion that occurs when selecting a subset of b samples from a pool of $|\mathcal{X}_U|$ unlabeled samples. The number of possible batches that can be selected and hence have to

be scored is given by $\binom{|\mathcal{X}_U|}{b}$. Scoring each of these $\binom{|\mathcal{X}_U|}{b}$ combinations is computationally infeasible. Therefore, the described batch-based approaches estimate the optimal solution for the batch's cumulative information gain using greedy algorithms.

Active learning in semantic segmentation: When applied to semantic segmentation, active learning methods must choose which area of the image is to be considered for the acquisition: the full image [212], superpixels [13], polygons [64, 157], or each pixel [208]. There is no common understanding so far of which approach is cheaper and more effective. Thus, this work uses the straightforward image-wise selection and annotation procedure.

Most existing methods for segmentation are based on the model's uncertainty for the input image, where the average score over all pixels in the image is used to select top-b images. Entropy [205] (estimated uncertainty) is a widely used active learning baseline for selection. This function computes per-pixel entropy for the predicted output and uses the averaged entropy as the final score. EquAL [64] determines the per-pixel uncertainty based on the consistency of the prediction on the original image and its horizontally flipped version. The average value over all the pixels is used as the final score. BALD [76] is often used as a baseline in existing works. It is employed for segmentation by adding dropout layers in the decoder module of the segmentation model and then computing the pixel-wise mutual information using multiple forward passes. Coreset [203] is a batch-based approach that was initially proposed for image classification, but it can be easily modified for segmentation. The pooled output of the ASPP [19] module (part of the encoder) in the DeepLabv3+ [20] model can be used as the feature representation for computing the distance between the samples. Some other methods [103, 209, 212] use a GAN model to learn a combined feature space for labeled and unlabeled images and utilize the discriminator output to select the least represented images. The experiments include Entropy, EquAL, BALD, and Coreset approaches for the analysis, along with the random sampling baseline. In this work, these methods are also studied with the integration of semi-supervised learning.

2.3 Semi-Supervised Learning

Active learning is not the only way of reducing the labeling effort. Training supervised on the labeled subset and parallelly utilizing the unlabeled data pool through unsupervised learning has the same effect. This training strategy is called semi-supervised learning. Figure 2.3 on page 14 illustrates the clustering assumption of SSL. According to the clustering assumption of SSL, if two points belong to the same cluster, then their outputs are likely to be close and can be connected by a short curve [15]. Given a labeled sample for a cluster, the label information can be extended to the other samples of the same cluster (c.f. Figure 2.3).

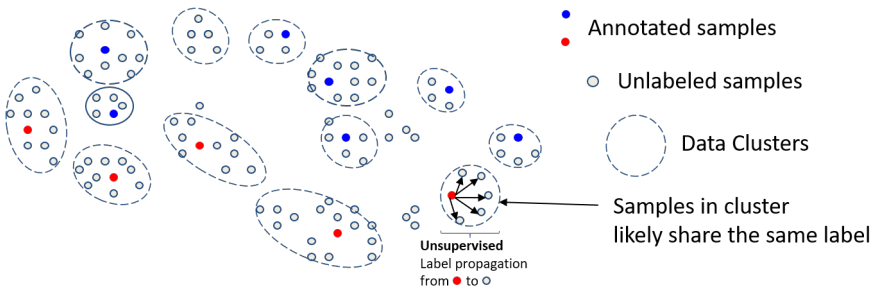


Figure 2.3: Illustration of the clustering assumption of semi-supervised learning: The data distribution consists of labeled and unlabeled data points (images). The assumption is that the data is distributed in clusters. If the label of one image is known, the labels of the other image from this cluster can be inferred. Unsupervised learning becomes possible by utilizing these pseudo-labels.

Different types of methods exist to compute this label extension. Self-training utilizes the predictions of a trained neural network on the unlabeled pool of data as so-called pseudo-labels. The difficulty is to figure out which pseudo-labels are correct and which must be ignored. The simplest way is to filter according to the classifier uncertainty, which could, for example, be the confidence or the entropy over the class probabilities. Wang et al. [250] further introduce a mean teacher network for computing more reliable pseudo-labels. Feature Space Alignment is another commonly used method for semi-supervised learning. Here, the aim is to match the feature representation of labeled and unlabeled data. The same class should have the same feature statistics in the labeled and unlabeled distribution. The difficulty here is to match the class distributions even though the class is unknown for the unlabeled set. Wang et al. [250] employ contrastive learning for the feature space alignment, and Mittal et al. [156] employ a generative adversarial optimization. For further details on SSL methods, refer to Chapter 3, which discusses SSL methods in the context of adapting between data domains.

Integration of semi-supervised learning into active learning: SSL offers the possibility of utilizing the unlabeled pool during AL. Therefore a combination of active learning and semi-supervised learning makes sense and has been e.g. applied in speech understanding (c.f. [45, 99]), in image classification (c.f. [56, 157, 161, 203]) or pedestrian detection (c.f. [187]). The combination of semi-supervised and active learning has recently been applied to segmentation. However, these works' scope was limited to special cases like subsampled driving datasets [186] or low labeling budget [157]. These approaches have in common the utilization of single-sample acquisition functions.

The previously described clustering assumption of SSL (c.f. Figure 2.3) indicates that the performance of the unsupervised training on the unlabeled data depends on the selection of the labeled data. If the labeled distribution includes labeled data from many unlabeled data clusters, applying SSL becomes possible. Therefore, newly

Table 2.1: This work explored active learning (AL) techniques for semantic segmentation across three key dimensions: dataset distribution, annotation budget, and the incorporation of semi-supervised learning (SSL-AL). Newly investigated scenarios are highlighted in green cells, while gray cells represent settings examined in prior AL research. This work aims to serve as a guide for utilizing AL across all depicted conditions.

Dataset↓	Annotation Budget			
	Low		High	
Supervision →	AL	SSL-AL	AL	SSL-AL
Diverse	✓	✓	✓	✓
Redundant	✓	✓	✓	✓

selected samples during AL must cover many unlabeled clusters not already covered by labeled set or already selected samples. Only acquisition functions that foster this coverage requirement have the potential to leverage the additional benefits that arise from the integration of semi-supervised learning. Batch-based methods like Coreset optimize for this property, whereas single-sample acquisition functions might fall into the mode-collapse problem. The experiments evaluate the integration of different types of SSL functions into AL.

2.4 Evaluating and Comparing Active Learning Methods

In the current literature, active learning approaches for semantic segmentation are usually compared and evaluated in very specific scenarios. Table 2.1 shows that mostly highly diverse benchmark datasets are chosen with a comparatively large annotation budget b (gray settings). This chapter analyses the current state of the art in active learning with three questions in mind:

1. How do different active learning methods perform when the dataset has many redundant samples? In this case, redundant samples refer to the redundant information between the images in a dataset. A video with many consecutive frames would, e.g., be considered redundant. Datasets like Cityscapes [36], CamVid [11] or BDD100k [272] are the results of a curation process to eliminate most of the redundancies, which can be viewed as a sort of manual annotation process, also. However, these diverse distributions are still commonly used to evaluate active learning approaches.

2. What happens when the initial unlabeled pool is also used for training along with annotated samples using semi-supervised learning (SSL)? The integration of SSL into active learning and how it interacts with different AL approaches and different types of distributions is understudied for the task of semantic segmentation.

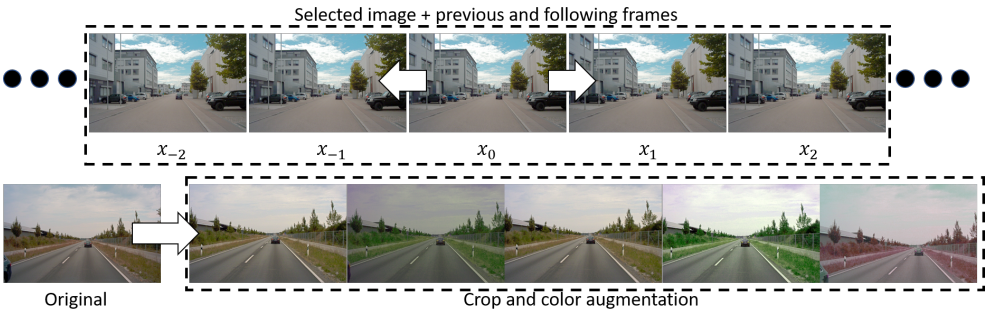


Figure 2.4: Top: creation of the A2D2 Pool-Xf by selecting consecutive frames. Bottom: creation of the Pool-Aug by cropping and color augmentation.

3. What happens when the annotation budget is low? Which methods scale best in such low-budget settings? Semantic segmentation is an especially expensive task to annotate. This high cost for annotation is especially true for applications requiring annotation specialists, such as medical image processing. Therefore, the annotation budget b is low in these scenarios, so it is important to understand this parameter's influence on the different AL approaches.

This work, therefore, creates specialized experimental settings for all of these questions. The following section shows how the respective dataset settings are created, what metrics are utilized, and how the AL and SSL settings are generally implemented.

2.4.1 Datasets

Cityscapes [36] is a driving dataset for benchmarking semantic segmentation approaches. It was compiled from videos recorded in 27 cities (see Figure 2.5). For annotation, a diverse selection of images was chosen. Due to this selection, it is considered "diverse," even though it was compiled from videos.

PASCAL-VOC [52] is another widely used segmentation dataset that is utilized in this study. The experiments use the extended version of the dataset, comprising 10,582 training images and 1,449 validation images. This dataset presents a broad spectrum of natural images featuring a mix of categories such as vehicles, animals, furniture, and more. It is the most diverse dataset examined in this study.

A2D2 [62] is a driving dataset comprising 41,277 annotated images sourced from 23 sequences. It includes highways, country roads, and scenes from three distinct cities. In the experiments, the 38 categories in A2D2 are mapped to the 19 classes of Cityscapes. Notably, A2D2 offers annotations for approximately every 10th frame within each sequence, resulting in significant redundancy between frames. For training data, 40,135 frames were extracted from 22 sequences, reserving one sequence containing 1,142 images for validation. The validation sequence, $'20180925\_112730'$, is selected based on achieving the maximum class balance among the available options.

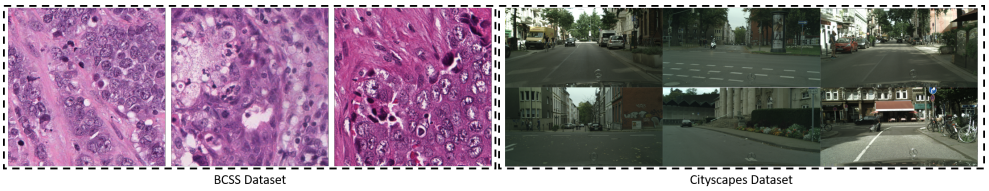


Figure 2.5: Left: examples of the BCSS dataset. Right: examples of the Cityscapes dataset.

A2D2-Pools: Subsampling the A2D2 dataset creates five smaller dataset pools to obtain a more continuous spectrum between diverse and redundant datasets. Each pool comprises 2640 images comparable to the Cityscapes training set. Four pools are curated by subsampling the original dataset, while the fifth pool is created by augmentation. Figure 2.4 shows the curation process. The first four pools, denoted by Pool-Xf (where X is 0, 5, 11, and 21), were created by randomly selecting samples and X consecutive frames for each randomly selected sample from the original A2D2 dataset. Pool-0f contains only randomly selected images. The assumption is that the consecutive frames contain highly redundant information. Therefore, the pool with more consecutive frames has higher redundancy and lower diversity. The fifth pool, Pool-Aug, contains augmented duplicates instead of consecutive frames. Five duplicates of each randomly selected frame are created by randomly cropping 85% of the image and adding color augmentation.

BCSS [2] Breast Cancer Semantic Segmentation is a dataset comprised of tissue regions from breast cancer images obtained from the Cancer Genome Atlas (TCGA). For annotation, 151 whole-slide images (WSIs) were used, cropped into RGB images with a resolution of 512x512 pixels for the experiments (see Figure 2.5). The resulting training set comprises 6,000 images, and the validation set comprises 2,768 images. The training dataset includes 22 initial classes that are mapped to 5 classes in the experiments, a common practice due to the sparse representation of many categories. The final classes are Tumor, Stroma, Inflammatory, Necrosis, and Other. The resulting distribution is redundant because the dataset is derived from only 151 WSI images. This redundancy is a typical scenario in medical datasets, where data pools for annotation are often obtained from only a few patients and generally show little variation in anatomy.

Diverse vs. Redundant datasets: PASCAL-VOC can be easily tagged as diverse, and the BCSS, the A2D2 original data, and A2D2-Pool-5f/11f/21f can be tagged as redundant. However, assigning a redundant/diverse tag to many datasets at the middle of the spectrum is difficult. Cityscapes and A2D2-Pool-0f fall in this spectrum since they are curated by sparsely selecting from large video data. The study considers these datasets diverse since they behave like diverse datasets. Quantifying dataset redundancy is a novel research question that emerged with this work and remains part of future work.

2.4.2 Metrics

The Intersection over Union (IoU), which is referred to as the Jaccard index in medical image processing, is a score for the segmentation quality. Given the class k, the Intersection over Union (IoU) is defined as:

$$\mathrm{IoU}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k + \mathrm{FN}_k} \tag{2.1}$$

Where TP_k is the number of true positive predictions, FP_k is the number of false positive predictions, and FN_k is the number of false negative predictions for class k. The Mean Intersection over Union (mIoU) is then defined as the mean IoU over the classes: $\mathrm{mIoU} = \frac{1}{K}\sum_{k=1}^{K}\mathrm{IoU}_k$, where K is the number of classes. The Dice score is a similar score for the segmentation quality and can be computed as follows:

$$\mathrm{Dice}_k = \frac{2 \cdot \mathrm{TP}_k}{2 \cdot \mathrm{TP}_k + \mathrm{FP}_k + \mathrm{FN}_k} \tag{2.2}$$

Compared to the IoU it puts a stronger emphasis on the correctly segmented regions.

During active learning, ρ percent of the unlabeled data $\mathcal{X}_U$, which is equivalent to a number of b samples, are acquired in each step of the active learning circle (c.f. Figure 2.1 page 10). This data is labeled and added to the training distribution. At each step, the mIoU of the retrained model is computed. An example for the curves that result can be seen in Figure 2.7 (see page 24). The area under the curve (AUC) is used to evaluate the performance of an active learning method. The AUC is computed after a certain percentage B of the unlabeled pool is acquired. This metric is abbreviated by **AUC@B**.

$$\mathrm{AUC@}B = \sum_{n=1}^{N_{AL}-1} \frac{(\rho_{n+1} - \rho_n)(\mathrm{mIoU}_n + \mathrm{mIoU}_{n+1})}{2}, \quad \rho_{N_{AL}} = B \tag{2.3}$$

N_{AL} is the number of AL acquisition steps. The model's mIoU at step n is expressed by mIoU_n. The percentage of the acquired data at AL step n is denoted by ρ_n. The final percentage $\rho_{N_{AL}}$ is hence equal to B. Usually, B is chosen as 50%. Alternatively, also the mIoU, after a certain percentage ρ_n (usually 30%) of the unlabeled dataset was acquired, is computed. This metric is termed $mIoU@B$.

2.4.3 Evaluation Scheme

In the experiments, different budget settings for AL are analyzed. ρ_0 denotes the initial label budget (percentage of $\mathcal{X}_U$) sampled randomly. ρ is the percentage of the dataset sampled from the unlabeled pool in each acquisition. Here, the AL functions are applied. Therefore, $\rho_0 - \rho$ denotes the setting of an experiment. For the experiments on the different A2D2 pools, the configuration of global budget $B = 50\%$ and $\rho_0 - \rho$ as

$10 - 10$ is chosen. The same setting was applied to the experiments on the Cityscapes dataset. For the PASCAL-VOC dataset, the focus is also on low-budget settings. B can, therefore, be 2,5 and 10 and $\rho_0 - \rho$ 2-2, 5-5, and 10-10 settings, respectively. Similarly, the experiments for the medical BCSS dataset focus on the low-budget setting. Therefore, B is 5, and $\rho_0 - \rho$ is 1-1 in this case.

2.4.4 Implementation

The experiments described in this chapter were done using the following setup. For the segmentation architecture, the experiments relied on the DeepLabv3+ [20] architecture with a WideResNet38[253] backbone. The WideResNet38 backbone is more efficient and yields better performance for segmentation. This setup yielded state-of-the-art performance for semantic segmentation. The network was initialized with ImageNet [38] pre-trained weights.

This work tests five active learning acquisition functions: Random, Entropy, EquAL, BALD, and Coreset. Here, the Entropy, EquAL, and BALD approaches represent single-sample, and Coreset represents the batch-based approach. All methods select the whole image for annotation. These methods are further described below, along with the segmentation-specific changes.

- *Random:* The samples are selected randomly for annotation from the unlabeled pool.

- *Entropy* [205]: This acquisition function uses per-pixel entropy as an estimation of the epistemic uncertainty (U) for the predicted output y. The entropy is computed over the class predictions $y_{k,i}$, where $k \in \mathcal{K}$ are the possible classes and K is the number of classes.

$$H(y_i) = -\sum_{k=1}^{K} y_{k,i} \log y_{k,i} \tag{2.4}$$

  The final score for selection is the average entropy over the number of pixels $i \in \mathcal{I}$.

$$U(y) = \frac{1}{N} \sum_{i=1}^{N} H(y_i) \tag{2.5}$$

  This method selects all b top-scoring images.

- *EquAL* [64]: The EquAL approach determines the uncertainty (U) based on the self-consistency between the prediction on the original image y and the prediction on its horizontally flipped version $\tilde{y}$. The average uncertainty value over all the pixels is used as the final score.

$$U(y_i, \tilde{y}_i) = \sum_{k=1}^{K} H(y_{k,i}) + \sum_{k=1}^{K} H(\tilde{y}_{k,i}) \tag{2.6}$$

The experiments use the EquAL implementation, which trains using only cross-entropy loss to keep the baselines comparable.

- *BALD:* [76] The BALD approach is based on a Monte Carlo Dropout network to compute the pixel-wise Mutual Information of the classification. The implementation employs dropout layers with a dropout ratio of 10% in the decoder layer and, during inference, compute 10 passes that result in y^r (where $r = 1...10$) predictions per image. The Mutual Information (MI) is then computed as follows:

$$\mathrm{MI}(y) = H(\mathbb{E}_r[y^{(r)}]) - \mathbb{E}_r[H(y^{(r)})], \tag{2.7}$$

  Where $\mathbb{E}_r[y^{(r)}]$ is the mean predicted probability over the *10* stochastic forward passes, and $\mathbb{E}_r[H(y^{(r)})]$ is the mean entropy over the individual forward passes.

- *Coreset:* [203] The Coreset approach selects a batch of samples that cover the whole data distribution. It formulates this batch selection as a robust b-center selection problem. Coreset implements a greedy algorithm that iteratively selects unlabeled samples with maximum distance to the nearest neighbor of the so far selected samples. This work utilizes the b-center greedy approach since it is much faster and only performs slightly worse than the robust formulation. The experiments use the ASPP module output in the DeepLabv3+ [20] model as the feature representation. Formally, the selection of a new sample x^* from the set of unlabeled images $\mathcal{X}_U$ can be defined as:

$$x^* = \arg\max_{x_u \in \mathcal{X}_U} \min_{x_s \in \mathcal{X}_S} d(f_{enc}(x_u), f_{enc}(x_s)) \tag{2.8}$$

  $d(f_{enc}(x_u), f_{enc}(x_s))$ is the distance between the feature representations $f_{enc}(x_u)$ and $f_{enc}(x_s)$. Hence, f_{enc} refers to the encoder mapping an image to the feature representations. The set $x_s \in \mathcal{X}_S$ represents the already selected images, and $x_u \in \mathcal{X}_U$ the unlabeled images.

- *MCD setting:* Since the BALD method requires the introduction of dropout layers into the architecture, the experiments segregate the methods into two categories: With Monte Carlo Dropout (MCD) and without Monte Carlo Dropout layers. Random, Entropy, EquAL, and Coreset are without MCD. BALD and Coreset-MCD are based on MCD. The experiments compare methods in each category separately due to different architectures. The analysis includes the fully-supervised performance, referred to as '100%' in the result tables, both with (100% MCD) and without MCD (100%) architectures.

Semi-supervised Learning To leverage the unlabeled samples, the experiments use the semi-supervised learning s4GAN method [156]. It uses adversarial training to align the labeled and unlabeled data distribution and further uses self-training based on the GAN discriminator score. This work paired all the active learning approaches

Table 2.2: Active learning results on Cityscapes, A2D2 Pool-0f, PASCAL-VOC. AUC@50 and mIoU@30 metrics are reported. A denotes the acquisition function type. S and B denote the single-sample and batch-based acquisition.

A	AL Method	SSL	Cityscapes		A2D2 Pool-0f		VOC 5-5		VOC 10-10	
	Metric →		mIoU	AUC	mIoU	AUC	mIoU	AUC	mIoU	AUC
S	Random	✗	58.90	23.29	48.48	19.20	70.70	13.92	72.13	28.85
S	Entropy	✗	61.83	24.25	52.40	**20.37**	70.38	13.94	73.72	29.10
S	EquAL	✗	62.41	24.32	**52.50**	20.35	69.14	13.82	73.40	29.03
B	Coreset	✗	60.89	23.89	51.14	19.88	70.85	13.96	73.63	29.06
S	Random-SSL	✓	60.72	23.85	49.69	19.60	72.57	14.36	75.33	29.87
S	Entropy-SSL	✓	60.61	23.93	50.80	19.90	73.36	14.51	**76.08**	30.01
S	EquAL-SSL	✓	60.26	23.96	51.08	20.02	**73.39**	**14.55**	75.89	**30.06**
B	Coreset-SSL	✓	**63.14**	**24.47**	51.49	20.02	72.88	14.46	75.91	30.03
-	100%	✗	68.42	27.37	56.87	22.75	77.00	15.40	77.00	30.80

with SSL using this approach. This is marked by the suffix '-SSL' in the experiments. In particular, the model is trained using an SSL objective, which impacts the resulting model and, hence, the acquisition function.

2.5 Results

This section analyzes the results of the experimental setup described in the previous section. The first subsection analyzes the influence of a dataset's redundancy on the effectiveness of different kinds of AL acquisition functions. The following subsection deals with the integration of SSL into the AL process and how the different types of AL acquisition functions interact with it under different types of distributions. Section 2.5.3 investigates the influences of small-scale annotation budgets, a scenario especially relevant for medical image processing. Finally, Section 2.5.4 derives a new benchmark from the insights of the preceding experiments to cover settings underexplored in the current AL literature.

2.5.1 Single-sample vs. Batch-based Active Learning

How do different active learning methods perform when the dataset has many redundant samples? For this comparison, the supervised-only setting is first considered. Table 2.2 and Figure 2.7 (see page 24) show the results for the Cityscapes datasets and the A2D2 Pool-0f. Given the supervised-only setting, the single sample (S) EquAL approach performs best in both scenarios. The results on the PASCAL-VOC 5-5 and 10-10 scenarios are also given in Table 2.2. For the 10-10 setting, the single sample acquisition functions obtain the best results. The batch-based (B) Coreset approach performs the best on the 5-5 setting. There, however,

Table 2.3: Active learning results on A2D2-Pool5f, A2D2-Pool11f, A2D2-Pool-21f, and A2D2-PoolAug. AUC@50 and mIoU@30 metrics are reported. S and B denote the single-sample and batch-based acquisition, respectively.

A	AL Method	SSL	Pool-5f		Pool-11f		Pool-21f		Pool-Aug	
	Metric →		**mIoU**	**AUC**	**mIoU**	**AUC**	**mIoU**	**AUC**	**mIoU**	**AUC**
S	Random	✗	47.58	18.69	44.61	17.76	44.52	17.67	43.80	17.15
S	Entropy	✗	49.96	19.48	47.43	18.52	46.08	18.21	44.51	17.33
S	EquAL	✗	49.50	19.29	47.14	18.44	46.32	18.18	44.24	17.29
B	Coreset	✗	50.08	19.44	47.72	18.69	46.68	18.38	44.70	17.54
S	Random-SSL	✓	47.92	19.03	45.25	18.02	46.27	18.19	44.17	17.29
S	Entropy-SSL	✓	48.78	19.31	47.53	18.56	46.93	18.43	44.50	17.47
S	EquAL-SSL	✓	48.80	19.28	46.50	18.39	47.11	18.54	44.81	17.56
B	Coreset-SSL	✓	**50.44**	**19.69**	**48.99**	**19.01**	**47.62**	**18.69**	**45.81**	**17.74**
-	100%	✗	53.25	21.30	48.85	19.54	49.23	19.69	46.03	18.41
With MC-Dropout decoder										
S	BALD	✗	50.40	19.29	47.85	18.74	46.78	18.57	45.53	17.80
S	BALD-SSL	✓	50.33	19.62	47.34	18.61	47.06	18.57	45.16	17.72
-	100%-MCD	✗	53.82	21.53	50.86	20.34	50.43	20.17	46.62	18.65

is only a marginal gap compared to the random baseline. The results on redundant datasets are shown in Table 2.3 and in the lower part of Figure 2.7. The batch-based Coreset approach consistently performs best in the supervised-only setting for all four datasets.

Given these results, it can be deduced that datasets with a diverse distribution already allow for single-sample-based methods. Datasets with a redundant distribution, however, require batch-based methods. This effect can be attributed to the mode collapse problem (see Section 2.2). A qualitative analysis can be seen in Figure 2.6. The figure shows that the redundant dataset possesses local clusters of data. The data in these clusters is redundant w.r.t. their information. If one sample of the cluster is selected, the rest of the cluster is likely to be selected, too. The batch-based corset method prevents the mode collapse problem. Diverse datasets have fewer such clusters. It is, hence, even a priori unlikely to select redundant information. Therefore, an acquisition function does not explicitly need to take the diversity of samples as an objective into account.

Since redundant datasets, such as the A2D2 Dataset, are a common result of measurement campaigns, the mode collapse (see Section 2.2) is likely to occur in real-world active learning applications. The AL literature mostly utilizes datasets like PASCAL-VOC or Cityscapes, which are considered diverse. The results show that testing and comparing AL methods on such distributions is an unrealistic setting.

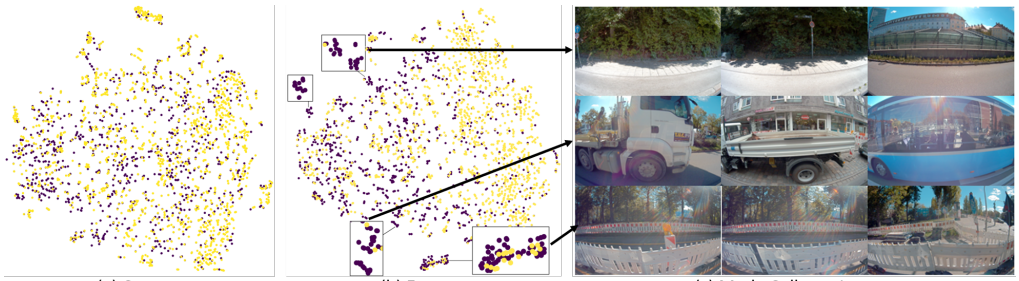
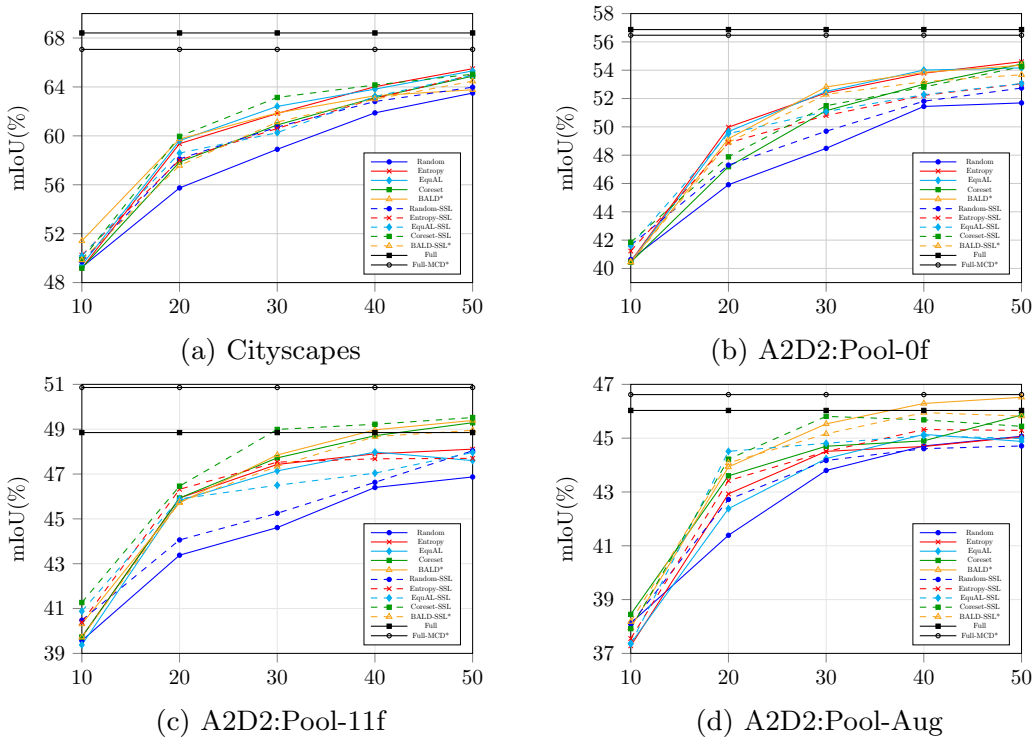


Figure 2.6: T-SNE plots of (a) Coreset and (b) Entropy functions for A2D2 Pool-21f. The yellow points are feature representations from the unlabeled set, and the violet points are the acquired points. The batch-based approach has good selection coverage, whereas the single-sample approach selects similar samples from clusters. Figure (c) shows acquired redundant samples from the violet clusters in (b).

2.5.2 Integration of Semi-Supervised Learning with AL

Table 2.2 and Table 2.3 show the results of integrating semi-supervised learning into the different active learning methods. For the redundant datasets (c.f. Table 2.3), Coreset-SSL achieves the best results. There is no consistently best active learning approach for the diverse datasets, but in general, the integration of SSL is helpful (c.f. Table 2.2). For the PASCAL-VOC dataset, the combination of single-sample-based methods with SSL achieve the best results. For the Cityscapes dataset, the batch-based Coreset-SSL is the best approach. In the case of the A2D2-Pool0f, Coreset-SSL is better than Coreset. However, the best results are obtained by the single-sample acquisition functions Entropy-SSL and EquAL-SSL.

Why is the integration of semi-supervised learning into batch-based active learning especially effective? The batch-based Coreset active learning approach always improves with the integration of SSL. The objective of creating a diverse and meaningful batch aligns well with SSL. According to the clustering assumption of SSL [15], if two points belong to the same cluster, then their outputs, or, in this case, labels, are likely to be similar. The selection and labeling of samples from many clusters allows for utilizing SSL to train unsupervised (e.g., by pseudo-labeling) on the other samples of these clusters. The Coreset approach creates a selection that better represents the global distribution. Since single-sample based acquisition functions often fall into the mode collapse problem and hence select samples from fewer clusters, fewer clusters can be utilized for semi-supervised learning. In the A2D2 pools, this effect is especially strong. Coreset-SSL is always better than Coreset and shows the best performance. Except for PASCAL-VOC, integrating SSL into single sample methods is either ineffective or harmful. In Pool-11f, Coreset-SSL with 30% of the data sampled is even better than the supervised training with 100% of the data. This observation



(a) Cityscapes (b) A2D2:Pool-0f

(c) A2D2:Pool-11f (d) A2D2:Pool-Aug

Figure 2.7: Results on diverse and redundant datasets. Active learning performance curves on diverse datasets - Cityscapes and A2D2 Pool-0f, and redundant datasets - A2D2 Pool-11f and Pool-Aug. The X-axis shows the percentage of labeled datasets. The methods that utilize MC-Dropout in their network architecture are marked with $*$.

indicates that adding labeled redundant samples to the training set can harm the performance. This effect could happen due to data imbalances. Coreset-SSL yields strong improvements over Coreset on the Cityscapes datasets. Here, the integration of SSL into Coreset even changes the rank order of the AL methods. Without SSL, the EquAL method is best with SSL Coreset-SSL. This effect is a slight anomaly since Cityscape is a diverse dataset. In this case, the synergies between SSL and batch-based AL methods outweigh the advantages of single-sample AL methods on diverse datasets. For the PASCAL-VOC dataset, SSL integrates well with all AL methods. Since the dataset is diverse, all AL methods return diverse batches, which integrate well with the SSL. Hence, no clear winner method for this dataset can be identified.

2.5.3 Small Annotation Budget Settings

Active learning is volatile with a low budget: As Table 2.4 shows, in the 2-2 budget setting on the PASCAL-VOC dataset, the integration of SSL into the random

Table 2.4: Active learning results on PASCAL-VOC, Cityscapes, and A2D2 Pool-11f in the 2-2 setting and BCSS in the 1-1 setting. AUC@10 and mIoU@6 metrics are reported for the PASCAL-VOC, Cityscapes, and A2D2 and AUC@5 and mIoU@3 for BCSS. A denotes acquisition method type. S and B denote single-sample and batch-based acquisition.

A	AL Method	SSL	PASCAL-VOC 2-2		Cityscapes 2-2		A2D2 Pool-11f 2-2		BCSS 1-1	
	Metric →		**mIoU@6**	**AUC@10**	**mIoU@6**	**AUC@10**	**mIoU@6**	**AUC@10**	**mIoU@3**	**AUC@5**
S	Random	✗	66.41	5.22	46.05	3.65	37.74	2.93	56.29	2.24
S	Entropy	✗	66.33	2.92	**51.24**	5.11	36.37	**4.00**	53.67	2.15
B	Coreset	✗	66.24	5.19	47.26	3.74	**39.63**	**3.10**	56.68	2.37
S	Random-SSL	✓	**68.60**	**5.37**	47.46	3.72	36.46	2.90	58.93	2.34
S	Entropy-SSL	✓	67.26	5.31	49.99	3.93	36.70	2.93	57.96	2.28
B	Coreset-SSL	✓	68.03	5.35	48.51	3.82	39.20	3.06	**60.10**	**2.38**
-	100%	✗	77.00	6.16	48.85	3.91	68.42	5.47	65.71	2.63

acquisition performs best. Hence, no AL approach was able to create an improvement. A possible explanation is that any selected sample is meaningful in this low-data setting, where the network could not learn from a meaningful distribution share. These observations provide more substantial support for similar findings made in the low-budget setting in Mittal et al. [157]. Table 2.4 also shows the A2D2 Pool-11f and the Cityscapes dataset in the 2-2 setting. In the case of the Cityscapes dataset, the single sample-based Entropy method performs best. For the comparatively more redundant A2D2 Pool-11f, the batch-based Coreset method yields the best results. For both datasets, however, the integration of SSL is detrimental. A likely explanation is that the low number of labeled samples acquired did not represent a sufficient share of the unlabeled distribution to support the SSL. Generally speaking, active learning is highly volatile in low-budget settings. However, batch-based acquisition is still the most effective for redundant datasets in low-budget settings.

In medical image processing, this use case of having a low budget due to the high cost of professionals for annotation and redundant data distributions is especially common. For the medical BCSS dataset, the integration of Coreset and SSL yields the best results. Even when comparing the AL approaches without integrating SSL, Coreset performs best. The single-sample acquisition method Entropy yields worse results than random acquisition, both with and without integrating SSL. These results indicate that batch-based AL approaches are especially important in the medical domain.

Overall, a highly volatile nature of active learning in conjunction with a low budget can be observed. As the dataset redundancy increases, the ideal policy transitions from random selection to batch-based acquisition.

Table 2.5: AL results on the proposed A2D2-3K task. S and B denote the single-sample and batch-based acquisition. Uniform refers to temporal subsampling selection process and (@5) means every 5^{th} frame.

A	AL Method	without SSL		with SSL	
		mIoU	AUC	mIoU	AUC
B	Uniform	57.75	—	58.93	—
S	Random	56.14	5.35	57.57	5.53
S	Entropy	60.16	5.53	59.91	5.61
B	Coreset	60.30	5.55	**61.13**	**5.72**
S	Uniform (@5) + Entropy	60.40	5.66	59.63	5.59
-	100%	66.65	6.64	—	—

2.5.4 New Realistic A2D2-3k Task

As Table 2.1 on page 15 shows, up until this work, active learning in semantic segmentation was mostly studied in diverse datasets with high annotation budgets. This work introduces the A2D2-3K, a new AL benchmark for segmentation. This novel benchmark covers the high-budget, highly redundant dataset pool case. This scenario is much more common in the real world, where for example, in the driving scenario, novel data is recorded in the form of videos. The A2D2-3K is based on the A2D2 dataset, which is a video dataset. The task is to select 3K images, which is a similar size as the Cityscapes dataset, in 3 AL cycle steps (1K images in each) from the A2D2 dataset pool (40K images). Given this new task, the random acquisition, Entropy, and Coreset are tested together with the respective integration of SSL. The manual sub-sampling methods commonly applied to the measurement campaigns are tested for further comparison. On the one hand, the subsampling is applied w.r.t. time by taking every fifth frame from the A2D2 dataset. This method is denoted by "Uniform" and is closer to previously used AL benchmarks like Cityscapes. On the other hand, the first 8K images are selected uniformly, and then Entropy selection (denoted by Uniform(@5) + Entropy) is applied. Table 2.5 shows these results. The Coreset SSL approach performs best for such a redundant dataset. The subsampling approaches are suboptimal, which indicates that active learning should be used instead of such data curation, which is an interesting insight for practitioners aiming to create new datasets.

2.6 Discussion and Conclusion

The results of this work show that active learning is a valuable tool for creating semantic segmentation datasets. The type of acquisition function that is effective depends,

.
Table 2.6: Overview showing the best performing AL method for each scenario. Single and Batch refer to single-sample and batch-based methods, and Random refers to random selection. The suffix -SSL refers to semi-supervised learning.

Dataset ↓	Annotation Budget			
	Low		High	
Sup. →	AL	SSL-AL	AL	SSL-AL
Diverse	Random	Random-SSL	Single	Single-SSL
Redundant	Batch	Batch	Batch	Batch-SSL

however, on the scenario. Table 2.6 gives an overview of the best practices for active learning, covering the dimensions of dataset diversity/redundancy, the annotation budget, and the integration of semi-supervised learning. Single-sample based acquisition functions perform best, given diverse distributions. Given high levels of redundancy in a dataset, batch-based acquisition functions with a diversity objective for sampling should be used. The importance of the diversity objective is especially true for creating datasets for medical image processing, where low-budget settings, are common and data pools are often redundant. Semi-supervised learning integrates well with batch-based methods, but SSL can harm single-sample-based active learning methods. Before this work, the method development and evaluation focused on only a few scenarios. This work contributes best practices for settings that are more likely to occur when creating new datasets. Therefore, these results allow for a broader view of active learning and positively affect many applications. As this work pointed out, the current active learning literature focuses on the specific high-budget and diverse dataset use case, a scenario that does not occur in many applications. Therefore, the chapter introduced a novel benchmark, contributing to the development of AL methods for redundant datasets.

Consequently, these novel best practices for integrating active learning and semi-supervised learning contribute to lowering labeling costs in the analyzed scenarios, thereby addressing challenge 1 from Chapter 1. The insights in this chapter help guide the AL research towards a more robust method development, which further contributes to addressing challenge 1.

Chapter 3

Reducing Manual Annotation Effort with Unsupervised Domain Adaptation

This chapter is partially derived from my previously published works, specifically "Domain Adaptation and Generalization: A Low-Complexity Approach" [166], "Combining Semantic Self-supervision and Self-training for Domain Adaptation in Semantic Segmentation" [165], "Overcoming the Sensor Delta for Semantic Segmentation in OCT Images"[168], and "Survey on Unsupervised Domain Adaptation for Semantic Segmentation for Visual Perception in Automated Driving"[201]. Some text, figures, and findings have been re-utilized or adapted from these publications. Co-Authors of these Publications are Jan Ehrhardt, Jörg P. Schäfer, Manuel Schwonberg, Jan-Aike Termöhlen, Nico M Schmidt, Hanno Gottschalk, Tim Fingscheidt, Timo Kepp and Heinz Handels.

3.1 Introduction and Motivation

This chapter introduces the field of unsupervised domain adaptation (UDA). The distribution a DNN is trained on is sampled from a specific domain. A domain can be,

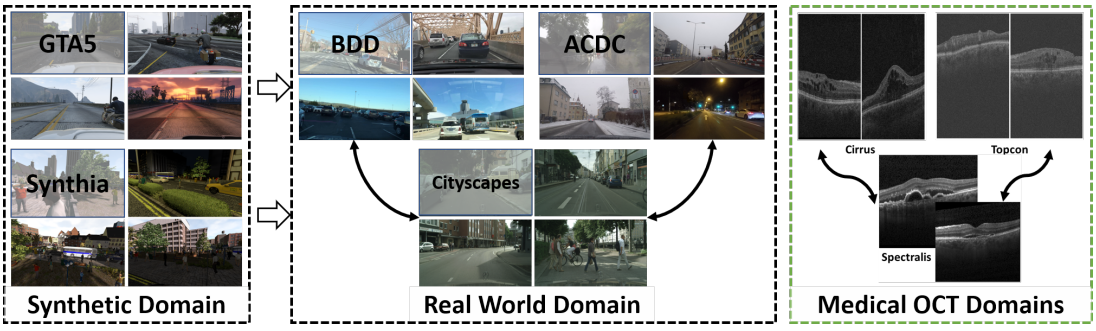


Figure 3.1: Domain adaptation scenarios: in autonomous driving, adapting from synthetic to real-world data and to varying weather and environmental conditions is essential; in the medical field, variations between images from different devices are critical.

e.g., a specific type of sensor or environment. A model trained on this distribution will perform well on novel data sampled from the same domain. If, however, this model is applied to data sampled from an unseen novel domain, the DNN will not yield good results (c.f. Figure 1.1 page 2). Unsupervised domain adaptation (UDA) aims to adapt neural networks from a particular source domain to the distribution of a specific target domain. Figure 3.1 describes many such adaptation scenarios. In autonomous driving, important domain shifts include the shift from synthetic to real data or between different environments (country, lighting, weather, etc.). Given a car manufacturer that has optimized the environment perception for a specific operational design domain, e.g., driving during clear conditions in the day in a specific country, but wants to extend to a new ODD like driving at night or during rain, a large amount of the labeling process would have to be redone. In Figure 3.1, this scenario would correspond to switching from the Cityscapes dataset to the ACDC dataset. Similarly, for companies providing DNN-based software for classifying or segmenting pathologies, a shift between training data and the domain to which the DNNs are applied could lead to accuracy losses. Given that imaging devices are often updated, this is a likely scenario. Figure 3.1 shows such a switch of imaging devices in the form of the domain difference of different OCT scanners. Labeling in such a scenario would be even more costly since it requires medical professionals. Therefore, this chapter deals with the field of unsupervised domain adaptation. The adaptation process between the training distribution (source domain) and the new application domain (target domain) is unsupervised. The supervised training is done on the source domain data, and the unsupervised training is done on the data from the target domain. This training procedure makes UDA a special case of semi-supervised learning. This chapter, therefore, tackles problem (challenge) 1: "Annotations are costly" from Chapter 1.

The chapter is divided into two main sections. The first Section 3.2 introduces an overview of the rapidly growing field of UDA and identifies and categorizes the methodologies. In the second part, i.e., Section 3.3 of the chapter, this work utilizes the knowledge of the research gaps identified in the survey to introduce novel methods. These methods are investigated in applications in the field of medical image processing and autonomous driving.

3.2 Unsupervised Domain Adaptation: A Survey

DNNs have shown remarkable results in the analysis of complex sensor data. They, therefore, enabled novel applications in medical image processing and autonomous driving. These advancements, however, come at the cost of annotating large amounts of data, a costly and work-intensive task. The trained DNNs additionally yield a poor generalization of out-of-domain data. This poor generalization introduces the need to redo the annotation for novel domains. Therefore, automatizing the adaptation process

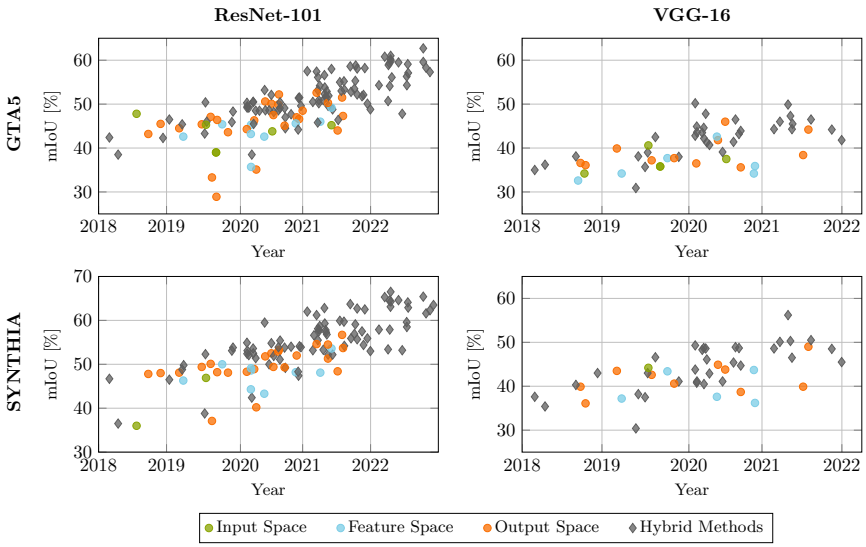


Figure 3.2: Performance (mIoU (%)) on the Cityscapes validation set after training on the source domains GTA5 (top row) or SYNTHIA (bottom row) with simultaneous adaptation to Cityscapes. The results are shown for models based on a ResNet-101 feature extractor (left column) or a VGG-16 feature extractor (right column). The reported values are taken from the respective papers.

is of high interest. As Figure 3.2 shows, this unsupervised domain adaptation sparked large interest in the computer vision community over the last years. The large amounts of research done in this domain make keeping track of the methodological developments difficult. Therefore, this chapter first provides a survey that categorizes and reflects the methodology. Since the car industry has strongly driven the research, most approaches deal with datasets from this domain and semantic segmentation as a task, which is crucial in this application. Furthermore, the synthetic to real adaptation application is of special interest in this scenario. The reason is that the synthetic source domain does not require human annotation, making the training process completely unsupervised. Therefore, most methods were developed in the context of UDA in autonomous driving between synthetic and real data for semantic segmentation. The relevant benchmarks in this context usually deal with the adaptation from the synthetic GTA5 [188] and SYNTHIA [191] datasets to the real Cityscapes dataset.

This survey introduces a simple taxonomy to group the different approaches. Unsupervised domain adaptation methods can be categorized w.r.t. the space the unsupervised learning is applied to. Figure 3.3 (page 32) illustrates this categorization. The adaptation can be applied to the *input space*, the *latent representations* (*feature representations*), and the *output* of the neural network. Approaches that fall into the category *input space* operate in the pixel-space and are described in Section 3.2.3. Section 3.2.4 describes the category *latent representations*, which is comprised of ap-

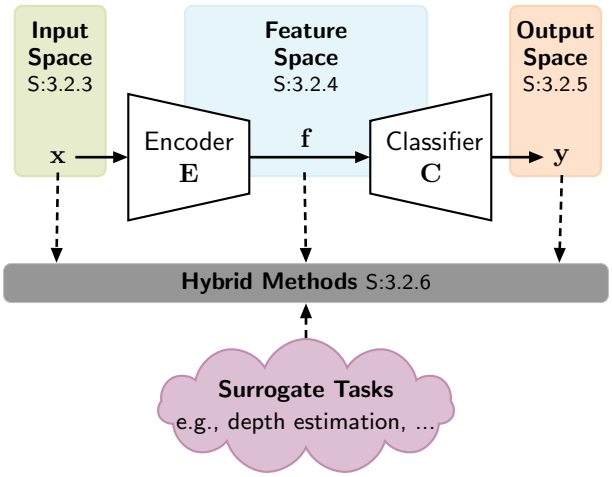


Figure 3.3: Overview of adaptation spaces that are covered in this survey. The subsections dealing with the respective space are indicated. Hybrid methods perform adaptation in at least two spaces or utilize surrogate tasks.

proaches that operate in the feature space, which is also known as the latent space. The feature space is the result of passing the input image through the encoder **E**. Based on it, the classifier **C** computes the segmentation map **m**, which is in the output space. Section 3.2.5 introduces UDA approaches operating in this output space. This chapter identifies several sub-clusters of approach types for each of these categories. The different sections follow these sub-clusters in their structure. The sub-clusters are presented as tables (see Table 3.1, Table 3.2 and Table 3.3) at the beginning of each section. Additionally, surrogate tasks like depth estimation can facilitate the adaptation in the different spaces. Since they can influence all spaces, they are not seen as an extra category. Many of the current approaches utilize *hybrid* approaches. Such approaches represent a combination of methods working in different spaces (see Figure 3.3). Section 3.2.6 analyzes how the different techniques operating in the *input space*, the *features space*, and the *output space* interact with each other.

Compared to other surveys like in Toldo et al. [228] and Csurka et al. [37], this survey provides a more recent overview with about three times more approaches covered (the most detailed survey up to date) and introduces further and more fine-grained categorizations. Additionally, this section provides a quantitative comparison of the performance (see Figure 3.2) w.r.t. the approach category, which allows for a discussion of current trends. Finally, this survey identifies common problems and gaps in the current state of the art and introduces future research ideas. The survey provides a publicly accessible project website including a beaverboard for the synthetic to real benchmark in semantic segmentation `https://uda-survey.github.io/survey/leaderboard`.

3.2.1 Mathematical Notation

This survey builds on the following mathematical notations for the unsupervised domain adaptation in semantic segmentation. The input to the segmentation network is an image, denoted as $\mathbf{x} \in \mathbb{G}^{H \times W \times Ch}$ defined, where $\mathbb{G}$ denotes the set of integer color intensity values, H and W the image height and width in pixels, and Ch the number of channels, respectively. A semantic segmentation network transforms an image into an output $\mathbf{y} = (y_{i,k}) \in \mathcal{P}^{H \times W \times K}$ with posterior probability (score) $y_{i,k} = P(k|i, \mathbf{x})$ for each class $k \in \mathcal{K}$ at pixel index $i \in \mathcal{I} = \{1, 2, ..., H \cdot W\}$. Here, $\mathcal{K} = \{1, 2, ..., K\}$ denotes the set of K classes and $\mathcal{P} = [0, 1]$. The final segmentation map $\mathbf{m} = (m_i) \in \mathcal{K}^{H \times W}$ is obtained with $\arg\max$ operating on each pixel i of the network output $\mathbf{y} = (\mathbf{y}_i)$ individually so that $m_i = \arg\max_{k \in \mathcal{K}} y_{i,k}$. Note that $\mathbf{y}_i = (y_{i,k})$ 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.

3.2.2 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. This work distinguishes between three distribution shifts to describe how the domains differ: the prior, covariate, and concept shift [110].

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 designed to overcome this issue and provide autonomous driving functions without needing a large-scale data selection and the corresponding data labeling effort.

3.2.3 Input Space Adaptation

Unsupervised domain adaptation in the input space refers to methods that aim to overcome the domain difference in the input space, i.e., the pixel space, of the deep neural networks (DNN). Approaches of this category try to find a function $\mathcal{F}: \mathcal{D}^S \mapsto \mathcal{D}^T$ that maps a given input image x from the source to the target domain or vice versa from target to source $\mathcal{F}: \mathcal{D}^T \mapsto \mathcal{D}^S$. Figure 3.4 on page 36 shows the three general categories such methods fall into: Style transfer, Data Augmentation, and Image Mixing. Table 3.1 shows sub-clusters in these general categories. They all have in common that they aim to align the image style of the source and target domain. This survey distinguishes the style and content of an image in the following way: The content is the semantic structure of an image, i.e., the geometry of an image w.r.t. shapes of the classes relevant to the segmentation task. Low-level properties of images, such as hue, saturation, contrast, brightness, image noise, depth of field, etc., define an image's style. However, the appearance of classes (like, e.g., cars) w.r.t. their shape or texture can also be counted as a style, even if the previously mentioned properties cannot express this. Feature space approaches can address this latter kind of domain difference (see Section 3.2.4). The function $\mathcal{F}: \mathcal{D}^S \mapsto \mathcal{D}^T$ is then applied in one of two ways. When $\mathcal{F}$ transforms the source domain images to the style of the target domain, the resulting images are used for supervised training. Since the original source domain labels are used, it is important that $\mathcal{F}: \mathcal{D}^S \mapsto \mathcal{D}^T$ keeps the image content consistent. If $\mathcal{F}$ transforms the target domain images to the source domain style $\mathcal{F}: \mathcal{D}^T \mapsto \mathcal{D}^S$, the aim is to use the $\mathcal{F}$ during inference. In that case, the segmentation model trained on the source domain should yield better results on the style-transferred target domain images. So far, no input space approaches have set a new state of the art (c.f. Figure

Table 3.1: 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	[48], [63], [134]
	Normalization	[31],[83], [144],[162], [224], [235],[245], [252]
	GAN-Based	[12], [16], [23], [24], [28], [29] [31], [33], [42], [58], [65], [74], [104], [114], [115], [116], [123], [126], [127], [128], [162], [185], [198], [199], [213], [229], [244], [263], [265], [266], [269], [271], [294]
	Frequency Domain	[267], [285]
	Histogram Matching	[87], [141][147],
Data Augmentation		[3][89], [102], [284], [296]
Image Mixing		[57],[92] [138], [152], [246], [297] [298],
Others		[90] [178]

3.2 page 31), but they are often an important building block for hybrid approaches, as section 3.2.6 will show.

3.2.3.1 Style Transfer

As already mentioned, style transfer is the primary input space adaptation technique. This section discusses 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

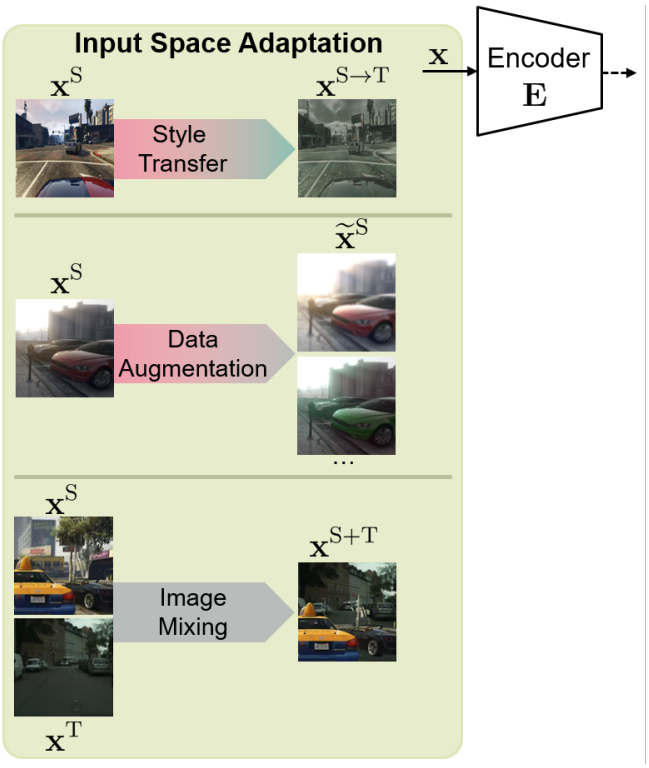


Figure 3.4: The three **main input space adaptation** methods are illustrated in this figure. Style transfer and data augmentation usually change the full appearance of the image, but style transfer does it in a more target domain-directed manner. Image mixing generates images consisting of source and target domain pixels.

to match the target domain during training (cf. Figure 3.4, 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.

Feature transforms: Style transfer using feature transforms must be distinguished from feature space domain adaptation (see Section 3.2.4). The feature transforms presented here are methods that convert the images of the source domain into the style of the target domain using 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 [122], 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) [120], which is called PhotoWCT. In PhotoWCT, the upsampling layers of the style transfer network are replaced by unpooling layers. Afterward, smoothing ensures that semantically similar regions are stylized consistently. The FastPhotoStyle method [122] was utilized by the domain stylization (DS) [48] and the mask-aware gated discriminator (MAGD) [134] methods, which both randomly match source and target domain samples. Restyling data (RD) [63] also employs FastPhotoStyle [122] 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.

Normalization methods: The efficacy of normalization methods for style transfer has been known for some time [235]. Adaptive instance normalization (AdaIN) [83] is particularly relevant in this context. The style transfer with AdaIN uses an encoder-decoder structure (usually based on a VGG-19 [210] 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 [252] employ AdaIN and assume that the mean and standard deviation of the feature maps in an image generator encode 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 later in this chapter elaborated, this idea is also significant for the distribution alignment in the feature space. The bi-directional style-induced domain adaptation (BiSIDA) [245] 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. [224] employs an encoder-decoder network with standard AdaIN layers for style transfer.

The adversarial style mining (ASM) method [144] 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 [31] employs a cycle-free generator network that is based on multimodal unsupervised image-to-image translation (MUNIT) [85]. 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 remain unchanged.

Frequency domain: Domain adaptation in the frequency domain is a relatively new field. Yang et al. [267] 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 [285] 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.

Histogram matching: Histogram matching is a long-established method [181] to match the style of images. However, only recently has there been research on its use for domain adaptation. Huang et al. [87] 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 inter-task consistency, which enforces consistent labels between a semantic segmentation and an instance segmentation network. They employ a histogram-matching algorithm [181] for the stylization. Ma et al. [147] propose a global photometric alignment for style transfer. They align the source and target image 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 [141].

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 [300] 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 $\mathcal{F} : \mathcal{D}^S \mapsto \mathcal{D}^T$ as well as an inverse mapping function $\mathcal{F}^{-1} : \mathcal{D}^T \mapsto \mathcal{D}^S$ 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 [85] 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. [198] was about the first to employ a standard GAN-based style transfer for domain adaptation. Also, Chen et al. [23] employed a GAN for style transfer. The domain invariant structure extraction (DISE) method [16] 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) [149] and can be used in both directions.

Li et al. [127] 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 [115] 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 [114] 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 [271] employs the ResNet generator proposed by Johnson et al. [97] 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. [12] propose a condition-guided style transfer by employing a standard conditional GAN [155] that is trained with a semantic consistency loss. They also utilize concepts from StarGAN [30] and BicycleGAN [301]. This way, preferred styles like `foggy` or `cloudy` can be added to the images as needed.

SUIT [126] 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. Content consistency is employed by comparing features of a pre-trained network for stylized and normal input images.

The stochastic image translation method by Chiou et al. [29] is based on MUNIT [85]. 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 [74] 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 [24], MSS [229], and CADA [269] employ a standard CycleGAN for their image translation. Zhou et al. [294] show that their ASANet+ is complementary to style transfer by combining their method with the image translation module from CyCADA [74].

The SE-GAN method [263] makes adversarial training more stable and employs a simple CycleGAN for style transfer. Yang et al. [266] 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 [65], 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 [74].

Another idea is to use a content invariant representation (CIR) [58], 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 [123], 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 [188] and SYNTHIA [191] datasets, which were used by many subsequent methods. For example, CDGA [104], SIM [244], MCSSF [33], and BDL+ESL [199] use this method or the already transferred images.

In contrast to previous works, the authors of LDR [265] 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 [123] 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 [213] 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. The style transfer is similar to that of LDR[265], but it replaces the transposed convolutions with bilinear upsampling and convolutions.

Ramirez et al. [185] 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 [116] is based on DISE [16] but adds a cycle consistency to the generator training. It also adapts the zero loss [5] 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) [28] 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. [42] 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 [162]. 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 [30] for image-to-image translation.

3.2.3.2 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 3.4 page 36, 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. [296] 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 [102] generates a stylized version of the commonly used GTA5 [188] and SYNTHIA [191] datasets to force the model to learn texture invariant representations, which are usually not learned from style-transferred images.

Huang et al. [89] 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 a form of style augmentation. Araslanov et al. [3] perform heavy data augmentation and then calculate output consistency using differently augmented images. The unsupervised contrastive domain adaptation (UCDA) method [284] also employs multiple augmentation techniques on source and target domain images.

3.2.3.3 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 3.4 page 36, lower part). One popular method is domain adaptation via cross-domain mixed sampling (DACS) [231], 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 training. It also applies color augmentation and Gaussian blurring to the training samples.

The RCCR approach [298] employs ClassMix [173] and CutMix [276] as proposed by DACS [231]. Also BAPA-Net [138] solely employs CutMiX [276]. Likewise, the CorDA method by Wang et al. [246] is based on DACS [231] and utilizes all of its input space adaptations. Zhou et al. propose a new image mixing method termed CAMix [297], where they leverage contextual information on relationships to guide the image mixing. It can be seen as an improved version of DACS.

DBST [14] 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 [283]. The dual soft-paste (DSP) method [57] improves on DACS [231] 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 [152] employs consistency training with two different perturbations added to the images in its best working model. The authors show that Fourier domain and CutMix [276] perturbations yield the best results.

3.2.4 Feature Space 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 3.5 page 44). Hence, this section discusses approaches that try to adapt the model from

Table 3.2: Adaptation methods in the **feature space**. The papers clustered and sub-clustered according to similar methodology.

Technique		Approach
Distribution Divergence	Adversarial Training	[9] [18], [21], [24], [25], [27], [28], [40], [41], [46], [73], [74], [81], [84], [90], [124], [142], [143], [145], [178], [194], [229], [234], [242], [251], [269], [271], [274], [280], [289]
	Instance Norm/ Gaussian	[84]
	L2 Distance	[21]
	Max Classifier Discrepancy	[113], [131], [195]
	Max Mean Discr. (MMD)	[57]
	Shared Style GAN	[75], [116], [193], [198], [302]
Distribution Norm		[93], [109], [121], [189], [252]
Self-Supervised	Contrastive	[98], [136], [150], [207], [254], [284], [298]
	Semantic Clustering	[33], [40], [91], [104], [132], [138], [165], [224], [230], [247]
	Depth & Ego Motion	[70], [194], [246]
	Augmentation	[221], [257]
	Weak Supervision	[146], [178]

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 represents 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. One can identify different subclusters in the methods for distribution alignment in the feature space (see Table 3.2) that the following subsections discuss.

3.2.4.1 Distribution Divergence

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

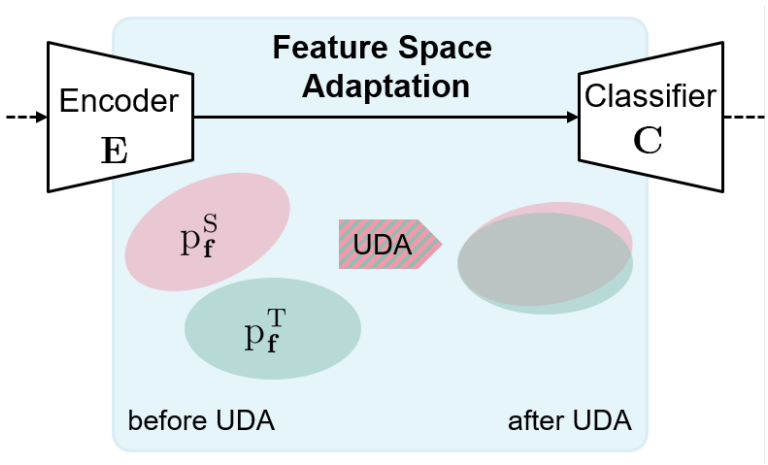


Figure 3.5: Feature space adaptation methods: The encoder of a CNN projects the input image to the feature space. Here, before domain adaptation, the source and target distributions are not aligned. Hence, the source domain-trained classifier does not generalize to the target domain. After feature space adaptation methods are used, the feature distributions are generally aligned much better, which improves performance on the target domain.

Adversarial adaptation: Adversarial training in the feature space for aligning the source and target domain distributions is one of the earliest approaches. Ganin et al. [55] introduced the first method for adversarial training for domain adaptation. 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. [55] 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. Ganin et al. applied this concept to the classification task but inspired many approaches for unsupervised domain adaptation for segmentation, also.

FCNs in the Wild [73] is the earliest example of adversarial learning in the feature space applied to semantic segmentation. The proposed method applies the adversarial loss function on the pre-logit feature map (the last representation before the classifier). Building on FCNs in the Wild [73], many approaches use adversarial training as an

additional tool in their domain adaptation strategy, e.g., WDC [289], RPT [280], DPL [28], CAA-Net [193], and SWLS [40]. Many works introduced new strategies to improve adversarial learning of FCNs in the Wild.

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.[143]). Sub-distributions in this 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. [242] 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. [25] speculated that another issue with global distribution alignment through GANs is non-contributing ambiguous features.

Chen et al. [18] and Du et al. [46] introduced early approaches to align the class-conditional sub-distributions. Their idea is to introduce 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. [46] improves the approach of Chen et al. [18] by addressing the inconsistent adaptation issue. CCDA [251] addresses the alignment of the class-conditional sub-distributions of classes with different frequencies. Wang et al. [251] 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 pixel-wise. 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 have a high classification uncertainty. FADA [242] and CCDA [251] follow a very similar idea.

CCD [25] 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 [21] 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. Hence,

the adversarial distribution alignment matches similar sub-distributions.

Adversarial training with attention: The following approaches aim to guide 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. [124] 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 is to align the distributions of the source and target domain of the vector.

DAST [274] 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. [27] do not directly model the attention via a measure for the distribution alignment or a spatial attention module. Instead, they 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 implicitly guides the attention to the class boundaries.

Adversarial training on style: As described in Section 3.2.3, 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. [74] and as described in Section 3.2.3 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 representations of the stylized source domain and the target domain images.

CrDoCo [24] 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 [229] follow a similar approach as in CrDoCo [24]. The main difference to CrDoCo [24] is that the encoder computing the feature representations is shared between the domains.

LWC [271] 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 [271] 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.

Shared style GAN: Most approaches that use a shared style GAN rely on the original GAN principle presented by Goodfellow et al. [67]. 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 $\mathbf{f_{dec}}$ that reconstructs the input image (mostly trained by $L1$ loss); And finally a discriminator **D** that tries to classify the image output of $\mathbf{f_{dec}}$ into either being "fake", or "real".

The approach presented in LSD [198] 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 $\mathbf{f_{dec}}$ 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. [193] follows a similar direction as LSD [198].

The architecture in PTP [302] consists of the elements and again follows a similar principle as CAA-Net [193] and LSD [198]. Similar to LSD [198], 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 [75] 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 $\mathbf{f_{dec}}$ 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. [116] 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.

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. [195] 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.[113] 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 [195]. Further follow-up work is presented in Li et al.[131], 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.

Other methods for distribution divergence minimization: MMD [57] uses the soft paste algorithm, combining two images by a weighted overlay (see Section 3.2.3). 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) [84] 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 [279] 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. [60]. The $L2$ distance is minimized at different feature levels.

3.2.4.2 Distribution Normalization

Normalizing the feature space distribution presents an alternative to aligning the source to the target domain feature distributions (or vice versa). Such approaches often try to normalize the channelwise mean and standard deviation (or variance) with source and target domain statistics, so that similar images in two different domains are mapped to a similar representation. Concepts like batch normalization and instance normalization are used in this context. The former computes the normalization over entire datasets, whereas the latter normalizes based on single instances of data.

Klinger et al. [109] utilize batch normalization to adapt the feature statistic to the target domain. The batch normalization, apart from scaling and translation parameters that are learned in a supervised way through backpropagation, normalizes the feature space at a certain layer by the channelwise mean and variance of that layer. These mean and variance values are determined unsupervised over the whole dataset by passing the images through the network. Klinger et al. utilize this circumstance to adapt to the target domain by replacing the source domain mean and variance values per layer with the target domain mean and variance values.

The approaches presented by Ioffe et al. [93] and Li et al. [121] follow a similar idea. Here, batch normalization is trained independently for both domains. The batch normalization layers in a neural network try to solve the covariate shift problem by setting the mean of the neuron activations to 0 and the standard deviation to 1. Hence, during training, the channel-wise mean and standard deviation of the CNN are obtained for normalization. Since the batch norm parameters are independently updated for the source and target domain, the feature distributions are aligned so far that the mean and standard deviations of the source and target domain match.

Although this approach was developed for classification, it is also used for semantic segmentation. (see Lian et al. [133]).

The method of instance normalization introduced in DCAN [252] uses reference images from the target domain to calibrate the feature distribution of a source domain image. The idea is that the channel-wise mean and standard deviation contain the style information and that both can be aligned between the source and target domain. However, the same idea is used to align the segmentation network's feature distribution. The method aligns the mean and standard deviation of the source domain feature space to those of the calibration image. Based on the aligned feature space, the segmentation loss is minimized. Given that the segmentation head is trained based on a feature space with target domain statistics, the segmentation also generalizes to the target domain.

The normalization of the feature distribution to a shared representation tends to be more robust than the alignment between the source and target distributions. The alignment requires that the class wise sub distributions of the source and target domain are mapped to the same feature representations. Given that the class-wise distribution of the target domain can only be estimated, this requirement is hard to fulfill. The normalization parameters, i.e., the mean and standard deviations, can be determined unsupervised and rather represent global translation and scaling parameters. This robustness, however, comes with the drawback that the transformation is less meaningful. This is because the normalization doesn't capture the finer nuances of class-wise distribution shifts, making the transformation less sensitive to domain-specific features. This, in practice, causes the alignment methods to yield a higher potential for improvement.

3.2.4.3 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 this case, it is an essential method for unsupervised training on the unlabeled target domain, too. In the case of unsupervised domain adaptation, there are three main sub-clusters of approaches: Augmentation and depth-based SSL, Contrastive learning SSL, and semantic clustering through SSL.

Augmentation and depth: SSL-UDA [220] and SSDA [257] 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 of image rotation, image flipping, and location prediction of image crops. These tasks are jointly trained on the source and the target domain. 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 [220] 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 [70], CTRL [194], and CorDA [246] show that depth prediction and ego-motion estimation are meaningful pretext tasks. GUDA [70] 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 [246] and CTRL [194], in contrast to GUDA [70], 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. [246], another difference can be found in how depth estimation is incorporated via spatial attention into semantic segmentation.

Contrastive learning: Unlike the implicit alignment of the feature distributions, self-supervised approaches are also used for direct alignment. DACL [207], SPCL [254], UCDA [284], PWCL [136], RCCR [298], and CLST [150] 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. The way positive and negative pairs are constructed depends on the task and method. A positive pair is, 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. [207] uses a CycleGAN-generated style transfer from the source to the target domain. The resulting pseudo-target domain images with ground truth labels yield pixel-wise class information. The contrastive loss can be computed based on the so-constructed positive and negative pairs.

The authors of CLST [150] 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 [254] is similar to CLST [150]. 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 using the pseudo-labels created by the network.

RCCR [298] combines contrastive learning with knowledge distillation and introduces both a teacher and a student for the projection head. Different 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 [284] follows SimCLR [26]. 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 [136] determines positive and negative pairs between source and target domain image patches. The utilization of image patches is a distinct feature in PWCL and is done through multi-level spatial pyramid matching. Their contrastive approach is close to the idea of MoCo [26] and utilizes the cosine similarity.

SCDA [132] 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.

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 [196] CAM [247], CFContra [224], SCDA [132], BAPA-Net [138], SWLS[40] , and SSS+ST [165] which all aim to cluster the pre-logit feature space directly towards so-called class prototypes. These class prototypes are vectors that represent the pre-logit feature representations of their respective class.

By advancing the method proposed by DANCE [196], 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 [224] 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 [247] 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 [230] 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 [33] is also based on clustering. The authors introduce a dictionary containing the correctly classified feature representations in the source domain. The target domain feature representations of the current batch are also stored in a dictionary. 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 [4] and BAPA-Net [138].

Similarly, the authors of LSR [4] apply non-contrastive clustering to prototypes. The prototypes are computed for the 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 [230]) and the norm of the target and source domain features to be the same. They assume, according to recent research by Xu et al. [258], 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 [244] 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. [138] (BAPA-Net) assume that near-boundary pixels are hard to classify and propose special handling of the boundary regions, different from the previous approaches. The authors employ the CutMix [276] 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. 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 [91] 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.

3.2.5 Output Space Adaptation

Output space adaptation methods can be formally defined by the distinctive property that the pixel-wise logits or softmax probability output $y_{i,k} = P(k|i, \mathbf{x})$ of the network are utilized for the adaptation. Output space adaptation methods can be subdivided into different subcategories, which are shown in Table 3.3. 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.

Table 3.3: Adaptation methods in the **output space**. The papers are clustered and sub-clustered according to similar methodology.

Technique		Approach
Self-Training	Global Thresholding	[22], [28], [102], [116], [123], [136], [162], [194], [216], [231], [242], [256], [263], [269], [294]
	Adaptive Training	[3], [33], [34], [40], [41], [42], [87], [94], [129], [132], [147], [151], [213], [244], [247], [254], [274], [283], [290] [304]
	Image-Level	[69], [86], [133], [137], [146], [178], [251], [277]
	Entropy-Based	[27], [90], [165], [177], [218], [224], [232], [249], [264], [297]
	Ensemble Learning	[4], [22], [128], [131], [262], [267], [291]
	Discriminator Confidence	[153], [206], [215], [289]
	Others	[90], [141], [246]
Adversarial	Basic	[16], [28], [41], [58], [104], [139], [151], [233], [237], [248], [265], [268], [274], [281]
	Depth	[23], [237], [279], [289]
	Entropy-Based	[29], [35], [225], [238]
	Modified Input	[234]
	Bi-Classifier	[113], [143], [195]
	Intra-Domain	[177], [268]
	Multi-Level	[90], [204]
	Multi Discriminator	[21], [134]
	Class-Wise	[46], [193]
Consistency	Augmentations	[152], [285]
	Style Transfer	[24], [229]
	Knowledge Distillation	[31], [259], [282], [291], [296]
	Others	[34], [280]
Depth		[14], [70], [239]
Contrastive Learning		[98], [150], [298]
Others		[27] [138]

3.2.5.1 Self-Training

The general idea is to retrain the network on labels that are generated by itself (see Figure 3.6 page 56). In the unsupervised domain adaptation setting, the network $\mathbf{M}(\cdot)$ 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 $\mathbf{y} = \mathbf{M}(x^{\mathrm{T}}; \boldsymbol{\theta})$. 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

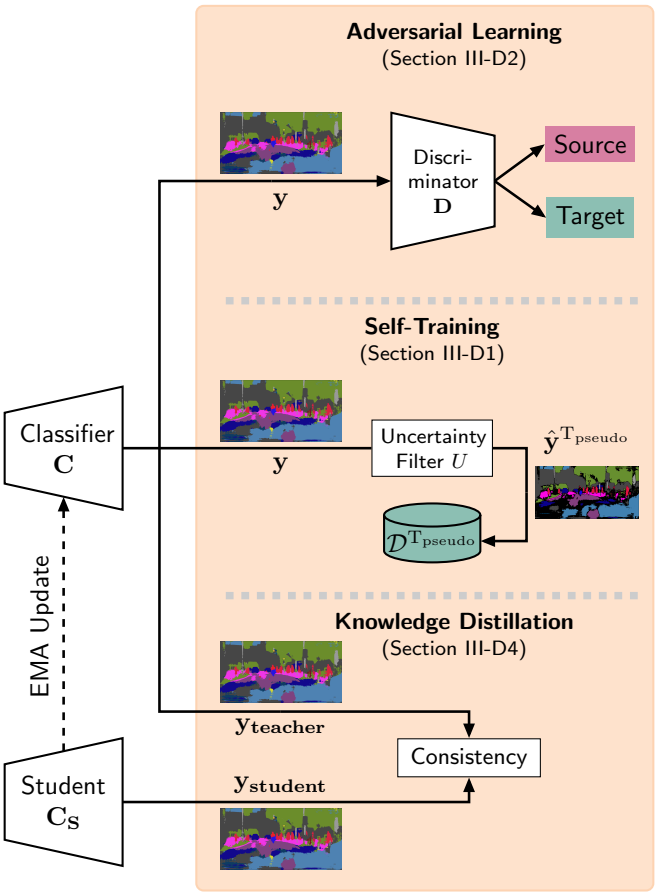


Figure 3.6: Overview of output space adaptation methods: Simplified and exemplary schematic visualization. Deviations of certain approaches from this basic scheme are possible.

is mostly the filter operation $\overline{\mathbf{y}}^{\mathrm{T}_{\mathrm{pseudo}}} = \mathbf{U}(\mathbf{y})$, which removes predictions with low confidence. A small sub-taxonomy of self-training methods is given in the second column of Table 3.3 on page 55. These methods are particularly often used in hybrid approaches and only rarely as stand-alone adaptation methods. 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 [150, 151, 278], 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, often multiple iterations or stages of self-training are performed to iteratively increase the performance of the pseudo-labels [151, 283, 304, 305].

Global Thresholding: Global maximum softmax thresholding is the simplest method employed by several approaches (see Table 3.3 page 55). These approaches take the softmax probability distribution $P(k|i, \mathbf{x})$ 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.

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) [304] introduced these class-wise thresholds as one of the first works. It combines output normalization and class-specific quantile-guided thresholding. The best ρ_k percent of the pixels per class k are chosen as a pseudo-label, and ρ_k is increased over the self-training iterations. 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 [125], CSCL [41], APL [213], PA+CCR [147], and SCDA [132]. Additionally, regularization methods for CBST were proposed.

Entropy-Based: Computing the entropy over the pixel-wise softmax probabilities presents another way of determining a measure for uncertainty. The entropy is high if all class probabilities are approximately equally distributed and low if only one class has the probability of one. Several methods apply the above-mentioned thresholding to this uncertainty measure to determine the reliable labels for training. Examples for this are e.g. [87, 165, 297] Apart from being used to create a supervision signal for the unlabeled target domain directly, the entropy is used implicitly, also. Approaches like, e.g., [125, 224, 238] minimize the entropy of the pixel-wise predictions. This optimizes the network to create more confident predictions on the target domain.

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 [277] utilizes the prediction of global image class distribution and superpixel class distribution. 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) [146] utilizes multi-nomial logistic regression. It is trained on the source domain with the image-level class distribution to train multiple region expansion units.

Ensemble Learning: Ensemble learning is a commonly applied method for various applications for uncertainty quantification [1]. 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 [128] 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 [291] method is different by only using a second classifier head, which takes features from a different layer for weighted prediction summation. EPS-UDA [262] 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, which is 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 [267], 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 [29]] follows a similar scheme. However, instead, it uses a stochastic style transfer 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 [28] 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 [140] 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 [143]. The authors of CorDA [246] 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.

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 [215] 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 [206] applies a very similar principle but combines it with the softmax probabilities.

3.2.5.2 Adversarial Learning

The basic idea of this kind of approach is to attach a discriminator after the segmentation probability output. This discriminator is trained to predict if a predicted semantic segmentation originates from a source or target input on the image-level. A simple visualization of this idea is shown in Figure 3.6 (top) on page 56. 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 becomes adapted to the target domain. This principle also works for two different discriminators as proposed by AdaptSegNet [233]. 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 [16, 102, 134, 244, 265, 281]. 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 [162, 213, 263]. 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. SASP [204] 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 also reason that the earlier layers receive a strong learning signal from this multi-layer fusion. MLAN [90] 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.

3.2.5.3 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 [280] 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 of that, an LSTM is used to enforce a similar spatial structure for the source and target.

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, such as image augmentations, and learnable, such as GANs and CycleGANs. SUDA [285] creates two different spectral views of the same target images and applies an $L1$ consistency loss to obtain similar predictions.

A popular and simple way are image augmentations which can severely change image characteristics such as sharpness, contrast, hue, etc. PA+CCR [147] augments the target images with color jitter and enforces the prediction to be similar to the prediction without augmentation. 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 [152] applies 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 [231] 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.

Style transfer: Another line of work uses a GAN or CycleGAN to obtain different image styles. SUIT [126] 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 [128] follows a similar idea but trains two distinct networks and enforces consistency using an $L2$ loss. CrDoCo [24] 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 [229] 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 [264] 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.

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 [259] 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.

UACR [296] 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 networks 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 [291] 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.

The original domain mixture idea from DACS is further extended by DSP [57]. 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 predictions from the student model, so it also enforces consistency between these two models.

SAC [3] 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 [245] 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.

CDGA [104] 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.

SAM [34] 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.

3.2.5.4 Depth

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.

SPIGAN [114] 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 [70] 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 [14] 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 [231]. 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.

3.2.5.5 Contrastive Learning

Contrastive learning is mainly applied directly in the feature space, 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 [136], CLST [150], SDCA [132], RCCR [298], and UCDA [284] all have in common that the contrastive adaptation operates in the feature space. 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 [98], but it is the only work that conducts multi-level contrastive learning on both the feature maps and the semantic predictions. For the

Table 3.4: Adaptation methods for input, feature, and output spaces using different techniques. The papers are grouped based on their use of techniques across these spaces.

Adaptation Space			Approach
Input	Feature	Output	
✓	✓		[65], [74], [144], [207], [252], [271]
✓		✓	[16], [23], [28], [29], [31], [34], [70], [87], [90], [102], [104], [114], [123], [128], [134], [162], [193], [199], [213], [231], [234], [245], [263], [265], [266], [267], [268], [285], [297], [294], [296], [302]
	✓	✓	[17], [27], [40], [41], [86], [90], [91], [98], [129], [132], [137], [145], [150], [165], [166] [206], [225], [233], [237], [247], [249], [254], [256], [257], [259], [262], [264], [274], [278], [279], [280], [281], [283], [284], [291]
✓	✓	✓	[3], [12], [14], [24], [33], [42], [57], [58], [92], [104], [116], [136], [138], [141], [147], [178], [229], [244], [246], [269], [282], [298]

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.

3.2.6 Hybrid Methods

Early in the research, it became evident that the methods of the different adaptation spaces can be combined to increase performance. A large group of research works has emerged from this idea, referred to as *hybrid* domain adaptation approaches. The complexity of different approaches and ways to combine techniques is large. Therefore, a two-level grouping is provided to ease the overview. The first-level grouping is done according to the variations of how the different spaces can be combined. This results in four different fields, as shown in Table 3.4.

The terms mutually independent and mutually dependent approaches are introduced for the second-level grouping. 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 a style transfer method with multiple loss functions for input space alignment. Softmax-based self-training can be "attached" for increased output alignment 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.

3.2.6.1 Input and Feature Space Adaptation

Mutually independent approaches: CyCADA [74] 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 [65] works similarly and only extends the style transfer by a domainness factor for higher style diversity. Closely related to that, GAM [84] utilizes CycleGAN-transferred images for pre-training and independently applies deep activation matching afterward. Likewise, the idea of DACL [207] is similar but applies contrastive learning in the feature space.
Mutually dependent approaches: LWC [271] combines input-level and feature-level adversarial learning within one framework. However, both techniques interact. The input style transfer enables feature-level adversarial learning, forming a mutually dependent approach.

ASM [144] is different because it utilizes an autoencoder-based style transfer to generate mini-batches with different stylized versions of the same image. This style transfer is necessary to enforce feature consistency across the mini-batch.

3.2.6.2 Input & Output Space Adaptation

Mutually independent approaches: APL [213] and DISE [16] 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 [102] 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 [266] 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, with input space adaptation added as an independent subcomponent. The three methods PixIntraDA [268], MAGD [134], and MLAN [90] 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+ [294] 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 [114] and GUDA [70] 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. Unlike that, GUDA combines depth prediction with a view synthesis module.

PTP [302], and CAA-Net [193] 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.

Mutually dependent approaches: The mutually dependent combination of style transfer and self-training closely relates to ensemble-like learning. FDA [267], SAC [128], and SIT [29] 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 [28] 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 [231] is more straightforward because it computes pseudo-labels only based on mixed images from both domains. CVRN [87] and SUDA [285] 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 [162] combines these three methods, while BDL [123] 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.

Knowledge distillation from a teacher to a student network is another combination of input and output adaptation that can be categorized as a mutually dependent framework. TGCF-DA [31] proposes an exemplary framework where the source images are translated to a target-like style and used as input for the student network. UACR [296] and CAMix [297] follow a similar scheme, but CAMix inputs domain-mixed images to the student network instead of a style transfer. In contrast, BiSIDA [245] employs style transfer in both directions; therefore, student and teacher networks receive images from both domains with a shared style.

3.2.6.3 Feature and Output Space Adaptation

Mutually independent approaches: AdaptSegNet [233], SEDA [27], MLAN [90], CGDA [145], and CrCDA [86] all utilize the adversarial learning for distribution alignment in the output and the feature space. Similarly, CLS [137] and DAST [274] 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 3.2.4 and self-training approaches described in Section 3.2.5. The approaches SSS+ST [165] and SePiCo [256] apply contrastive clustering as described in Section 3.2.4 and self-training in the output space. SWLS [40] 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 utilizing feature-level adversarial learning [233] in addition to another output technique. Several approaches follow this idea. RPT [280] proposes output patch consistency. SSDA [257] combines adversarial learning with self-supervised pretext tasks. JAL [281] adds a weight transfer, while CRA [249] is proposed as an additional technique to any UDA method and can be combined with adversarial learning. VAE-UDA [129] applies an autoencoder-based output space alignment and adversarial alignment. PFR [279] and SRDC [225] 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 [259] instead combine a self-attention mechanism in the feature space with an output consistency loss.

The approaches DADA [237] and CTRL [194] 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 [70] and CorDA [246] utilize depth regression as self-supervised training but use self-training in the output space.

Mutually dependent approaches: The approaches MCD [195], SWD [113], and BCDM [131], 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 3.2.4) 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 creates a close mutual dependence between feature and output space. Different approaches such as ProDA [283], CLST [150], SPCL [254], and EPS-UDA [262] 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 method of assigning pseudo-labels creates a strong interdependency of both spaces since the quality of pseudo-labels directly relies on the extracted prototypes. SCDA [132] is an exemplary work for this, and also UCDA [284], CAM [247], and CAG [278] utilize this idea.

The approach presented in MADA [206] 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 [41] 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.

3.2.6.4 Input, Feature & Output Space Adaptation

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 [298], DAP [92], and BAPA-Net [138] are building upon this mechanism that DACS initially proposed; DISE-CT [116] 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 [138] instead uses the domain mixture to enforce the boundary consistency on feature and pseudo-label-level. DAP [92] must be highlighted in this context since it introduces a novel extension at the intersection of feature and output space, including input space alignment. As the only currently known UDA approach for semantic segmentation, it introduces another modality by using word2vec embeddings [154] as domain invariant priors and projects them together with the mixed semantic output to enforce similarity between the priors and actual network features.

CrDoCo [24] and MSS [229] formed another group of mutually dependent approaches. 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 [57] and SSAC [3] 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 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 [244]. 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 [12] also has an independent input adaptation module. However, attention-based feature adaptation and self-training output adaptation interact using attention weights for pseudo-labels and an attention discriminator. BiSMAP [141] 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 [42] 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.

Mutually independent adaptation: Some works mainly vary in feature space adaptation methods among these approaches. An exemplary approach for this is CIR [58], 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 [269] 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 [33], which employs standard input and output space methods but uses a cosine similarity-based feature centroid alignment. WDA [178] 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 [224] and PWCL [136] 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 [246] and DBST [14] 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 [147] 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.

3.2.7 Discussion

The previous sections provided an introduction and overview of the categories of unsupervised domain adaptation visualized in Figure 3.2 on page 31. Figure 3.3 on page 32 provides insight into the quantitative evaluation of UDA approaches over time,

while the approach categories are color-coded. The scenario that is depicted is the adaptation from the synthetic datasets GTA5 and Synthia to the real-world dataset Cityscapes. This scenario is the most commonly used benchmark for determining the adaptation performance for semantic segmentation.

If the different categories of approaches are compared in Figure 3.3 it becomes evident that feature and input space methods do not cross the 50% mark. Output space adaptation approaches work best among the nonhybrid approaches. However, hybrid approaches are necessary to reach the current state of the art. Especially recently, most of the approaches seem to fall into the hybrid category.

The improvements hybrid models yield are most likely the result of the combination of the different sub-methods. E.g., ProDA [283] utilized knowledge distillation, contrastive learning, self-supervised learning, and self-training, which obtained a new state of the art performance of 57.5% mIoU for CNN networks. Apart from the fact that such methods seldom introduce larger methodological novelties, the combination of different methods introduces a larger number of hyperparameters. The increased number of hyperparameters introduces the need for finetuning them. This finetuning is usually done based on a target domain validation set, which is not always available. Therefore, a low-complexity training structure with fewer hyperparameters would be ideal. Later, this chapter shows (c.f. Section 3.3) how to use synergistic effects between the sub-methods to create low-complexity hybrid methods.

Advanced network structures yield practical domain adaptation advantages through better generalization capabilities. Better generalization, e.g., means that pseudo-labels are more reliable. Vision Transformers, e.g., used in the HRDA [78] or DAFormer [77] approaches yield such improved generalization properties. They have three potential advantages over conventional CNN architectures. 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 [255] and enables the transformer networks to incorporate more global contextual information at the early layers [184], which might be one reason why occlusions cause a smaller performance drop than for CNNs [163]. Second, CNNs are considered sensitive to texture [61], 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 [163]. Third, Raghu et al. [184] 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.

Figure 3.2 on page 31 demonstrates the relevance of the UDA field in recent years by the number of papers published. Given a simulation environment, synthetic data is free. Therefore, UDA approaches allow the elimination of manual annotation by adaptation to real-world data. Hence, UDA from synthetic to real-world datasets can save a lot of costs and effort for real-world applications and is, therefore, commercially interesting. UDA approaches that are relevant for real-world applications should be low

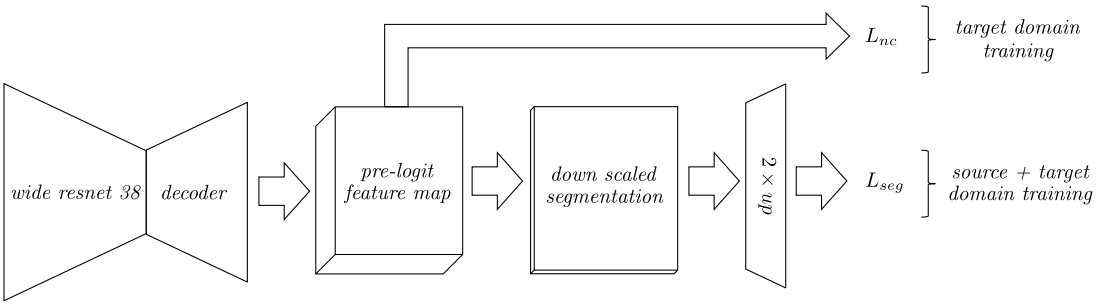


Figure 3.7: This figure shows the integration of the domain adaptation into DeepLabV3+ [20]. The semantic segmentation is trained in a supervised manner on the source domain. The semantic clustering is applied to the target domain data. Semantic clustering is performed on the pre-logit feature map, which is 1/4 the size of the original image.

in complexity and introduce generalization. The following presents a low-complexity approach and shows its effectiveness in the synthetic to real use case and multiple real to real adaptation use cases, including an application to medical datasets.

3.3 Domain Adaptation and Generalization: A Low-Complexity Approach

As the previous section discusses, most approaches that make up the current state of the art in UDA fall under the hybrid model category. Since these hybrid models combine several approaches, they often consist of a complex architecture. The papers "Combining Semantic Self-Supervision and Self-Training for Domain Adaptation in Semantic Segmentation" [165], "Overcoming the Sensor Delta for Semantic Segmentation in OCT Images" [168] and "Domain Adaptation and Generalization: A Low-Complexity Approach" [166] by Niemeijer et al. presented low-complexity approaches for unsupervised domain adaptation. These works introduced the category of semantic clustering for unsupervised domain adaptation. Building on the success of the semantic clustering the work discovered a synergistic effect of self-training and semantic clustering that allowed for low-complexity architecture for domain adaptation. The following part of the chapter first introduces the methodology of the approaches and then applies and evaluates them in unsupervised domain adaptation scenarios on medical datasets and driving datasets. Additionally, the generalization introduced by the adaptation to third unseen domains is analyzed.

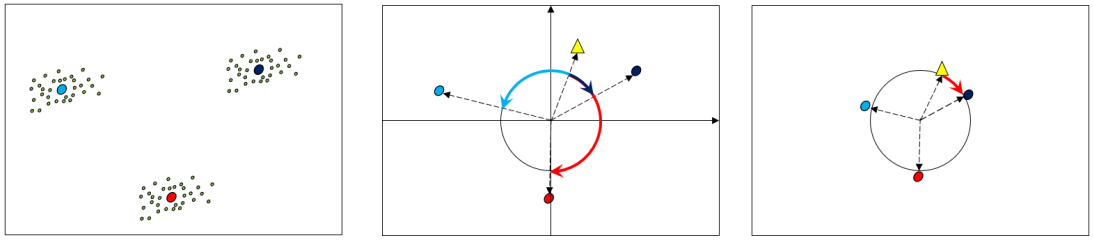


Figure 3.8: This figure shows feature representations and class centroids of a source domain batch (left), the normalized similarity of a target feature representation (yellow triangle) to all centroids (center), and the effect of minimizing the entropy (cf. Section 3.3.1), which causes the target representation to move closer to its closest centroid (right).

3.3.1 Semantic Clustering

The aim of semantic clustering, which is a variant of self-supervised learning (see Section 3.2.4.3), is to align the class-wise feature space distribution of the target domain with the class-wise distributions of the source domain. The pre-logit feature representations are utilized in the provided implementation, as shown in Figure 3.7. Each element of the feature space $\mathbf{f}_j$ represents a certain part of the input image (corresponding to the receptive field). In the case of a semantic segmentation architecture like the one shown in Figure 3.7, the number of pre-logit feature representations N_t depends on the spatial size of the image. For example, the DeepLabV3+ architecture yields $N_t = \frac{width \times height}{4}$. Each entry $\mathbf{f}_j$ is a vector that is L_2 normalized. The tuple f

$$f = [\mathbf{f}_1, \cdots, \mathbf{f}_{N_t}] \tag{3.1}$$

contains the feature representations $\mathbf{f}_j$. The general idea of semantic clustering can be understood from Figure 3.8. The respective mean representation of all the elements belonging to the source domain class is chosen to represent the source domain class distributions. The aim is to cluster the target domain feature representation to the closest class centroid. It is implicitly assumed that the closest centroid belongs to the same class as the given target domain feature representation. Given that this assumption is true, the semantic clustering process aligns the class-wise distributions from the target domain to the source domain.

The approach for clustering is inspired by the method presented in [196]. However, the class prototypes are not represented using the normalized weights of the classification neurons. The reason is that these *classification normals* are strongly affected by outliers, which causes the normals to fluctuate a lot during the training process. Clustering towards a noisy cluster center makes convergence hard for the algorithm. Eventually, this leads to misrepresented centroids of the class representations in the pre-logit layer. Instead, the centroids should be computed in a probabilistic way.

In every training step, first, the pre-logit representations f^{src} of the batch of source domain images $\mathcal{B}^S$ are computed. Since the approach is implemented for the semantic segmentation architecture shown in Figure 3.7, every image yields a number of $\frac{width \times height}{4}$ pre-logit representations f^{src} (cf. Chen et al. [20]). The feature representations are partitioned regarding the classification results of the model to compute the expected values $\mathbb{E}_K(f^{src})$ for each of the K classes as shown in Figure 3.8 (left). Each expected value $\mathbb{E}_k$ is a centroid for the cluster of class k, respectively. Why not use the ground truth instead of the classification results? Representations of wrongly classified pixels are within another centroid's area instead of being close to their class prototype. Therefore, they corrupt their centroid's position by being an outlier.

Since the current batch $\mathcal{B}^S$ only represents a subset of the entire dataset, a running average of the class centroids is computed iteratively.

$$c_k = c_k * \alpha + \mathbb{E}_k * (1 - \alpha) \tag{3.2}$$

Intuitively, α affects the accuracy of rare classes. Section 3.5.1 discusses how to choose this hyperparameter α.

Given the updated centroids c_k, the clustering method presented by Saito et al. [196] is applied for the samples in the target domain Batch $\mathcal{B}^T$. As suggested there, first, an L_2 normalization is performed on the class centroids c_k and the target domain pre-logit feature representations f^t that result from the images present in $\mathcal{B}^T$. Each element of the similarity matrix $p_{j,k}$ for every $\mathbf{f}_j \in f^t$ and class centroid c_k is computed as:

$$p_{j,k} = \frac{exp(c_k^\top \mathbf{f}_j^t / \tau)}{Z_j} \tag{3.3}$$

The neighborhood parameter τ is set to 0.05 according to the best practice by Saito et al. [196]. Since c_k and $\mathbf{f}_j^t$ are L_2 normalized, computing the dot product yields the cosine similarity as illustrated in Figure 3.8 (center). The soft-max normalization parameter Z_j is

$$Z_j = \sum_{k=1}^{K} exp(c_k^\top \mathbf{f}_j^t / \tau). \tag{3.4}$$

Finally, the loss function computes the overall entropy of all elements $p_{j,k}$:

$$L_{nc} = -\frac{1}{|\mathcal{B}^T|} \sum_{j \in \mathcal{B}^T} \sum_{k=1}^{K} p_{j,k} log(p_{j,k}) \tag{3.5}$$

$|\mathcal{B}^T|$, in this case, represents the number of feature representations in the target domain batch $\mathcal{B}^T$. Figure 3.8 (right) helps to understand the effect of minimizing this loss function: Intuitively, each element of f^t moves closer to the most similar centroid in terms of the cosine similarity computed in $p_{j,k}$.

3.3.1.1 Uncertainty Weights

The loss function described in Section 3.3.1 implicitly assumes that target samples are closer to the centroid of their respective ground truth class than to the others. Since this assumption does not hold in every case, the original method is extended with a measure of uncertainty. A standard measure for the uncertainty of the network predictions is the entropy of the class confidences. This measure is used to weigh the loss function

$$L_{nc} = -\frac{1}{|\mathcal{B}^T|} \sum_{j \in \mathcal{B}^T} \sum_{k=1}^{K} p_{j,k} log(p_{j,k}) \cdot \frac{1}{1 + H_j}, \tag{3.6}$$

where H_j is the class confidences' entropy with respect to each $\mathbf{f}_j$. The entropy H_j is computed over the average prediction of all pixels $\mathbf{f}_j$ influences during prediction. The weight factor $1/(1 + H_j)$ mitigates the influence of target samples that yield a high classification uncertainty. The intuition is that the assumption above is more likely not to hold when this uncertainty is larger. Unfortunately, by introducing the entropy as a weighting factor, the loss function can be minimized by increasing the term H_j. This problem is addressed by scaling down the gradients produced by this objective with a factor of 0.01; thus, the objective is to cluster the pre-logit representations around the most similar centroid dominates. This factor is explicitly not chosen as zero since the objective acts as an additional counter loss for those cases where the maximum similarity to a centroid is low.

3.3.2 Self-Training

The unsupervised domain adaptation introduced by semantic clustering yields a network that can produce more reliable pseudo-labels. Therefore, applying a simple self-training approach makes sense. Before training a model on the target domain utilizing the self-training approach, pseudo-labels must be created for the target-domain images. The pixel-wise entropy of the prediction is a measure of the uncertainty of the given model. The entropy H is computed for the prediction $y_{i,k}$ (pixel $i \in \mathcal{I}$) as follows:

$$H(y_i) = -1 \cdot \sum_{k=1}^{K} y_{i,k} log(y_{i,k}) \tag{3.7}$$

If the entropy of the prediction vector exceeds the threshold $\beta \cdot \log K$, the corresponding label is excluded. K is the number of classes, $\log K$ is the maximum entropy and $1 > \beta > 0$ is the uncertainty threshold (see Section 3.4 for the parametrization). Implementing an iterative adaptation process can improve the quality and impact of the pseudo-labels for self-training. For example, each adaptation step must create new pseudo-labels before resetting the model's weights and (re-) starting the training.

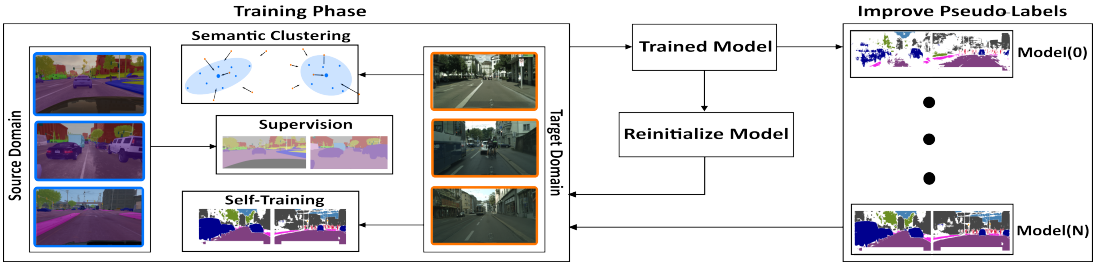


Figure 3.9: An overview of EasyAdapt's internal dependencies (dashed lines): The source domain training depends on the source domain data and labels; the self-training of the target domain training depends on the target domain data and its previously created pseudo-labels; the semantic clustering (a self-supervision variant) depends on the target domain data and the feature clusters of the source domain data.

3.3.3 Source Training

Self-training approaches require models with a sufficient understanding of the data to provide valuable pseudo-labels. Thus, this approach initializes the model with a sophisticated supervised training method on the source domain to provide valuable pseudo-labels. This supervised training includes a rich data augmentation and a class-uniform sampling strategy. Zhu et al. [303] showed that a class-uniform sampling strategy can significantly improve supervised training. Their class-uniform sampling is adapted in the following way. First, a list of objects from the dataset is gathered, i.e., a list of polygons[1], each with a class attribute and a centroid. During the training, samples regarding uniformly distributed class attributes are generated from this list. Eventually, each training batch combines randomly chosen crops and crops that are centered around these polygons. A parameter $0 \leq \gamma \leq 1$ controls the random and class-uniformly selected sample ratio. This uniform sampling is paired with a strong data augmentation consisting of Gaussian blurring, color jittering, and random scaling[2].

3.3.4 Iterative Training

This work introduces a nested iterative training approach for domain adaptation based on repeated application of self-training and semantic clustering (see Figure 3.9). The process is initialized with training on the source domain (see Section 3.3.3). The inner loop of the training process consists of training on the source and target domain, stopping early after a short number of epochs. The training on the target domain comprises semantic clustering and self-training, implemented as described in Section 3.3.2 and 3.3.1. Upon finishing such an inner iteration, the outer loop of the training pro-

[1]The COCO format stores segmentation masks as polygons.
[2]These simple augmentations are available in the Tensorflow and PyTorch API.

cess creates new pseudo-labels and repeats the process. Section 3.3.5 describes the algorithm for the training process.

3.3.5 EasyAdap: Assembling the Bricks

Figure 3.9 describes the training process. First, the supervised training is performed on the source domain data with class-uniform sampling and strong data augmentation to gain an initial model $\mathbf{M}(\mathbf{0})$. Then, the process enters the domain adaptation loop. Each adaption step first (re-) creates the pseudo-labels using the currently available model $\mathbf{M}(\mathbf{n})$ (i. e., $\mathbf{M}(\mathbf{0})$ in the first iteration). After resetting the model's weights (to weights pre-trained on ImageNet[38]), the adaptation step starts training a new model. The list of samples is rebuilt regarding random and class-uniform samples in each training epoch. While iterating over the smaller (source- or target domain) dataset, as many batches as the larger dataset allows are sampled, i.e., the sampling is restarted from the smaller dataset until the processing of the larger dataset is finished. The approach computes the semantic segmentation loss for each training batch using ground truth and pseudo-labels for the source- and target domain data. The features of the source domain data are used to update the running average of the class centroids. The similarities between these centroids and the target domain features yield the clustering loss. The model's weights are updated using both loss functions.

3.4 Implementation and Experimental Settings

The previous part presented two approaches for unsupervised domain adaptation: semantic clustering (see Section 3.3.1) and the EasyAdap approach (see Section 3.3.5). The following part of this chapter presents the experimental evaluation of these approaches for the unsupervised domain adaptation in visual perception in automated driving and medical image analysis. Therefore, the following section introduces the general settings of the experiments and the concrete implementation of the approaches in these experimental settings.

3.4.1 Experimental Setting

As described at the beginning of this chapter, different domain changes are important (see Section 3.2). The **real to real** domain change refers to the difference between two domains that consist of data from the real world. It occurs when, e.g., the sensor of the training datasets and the sensor of the target data are different, or environment conditions change between the source and target domain. **The simulation to real** domain gap occurs when adapting from a synthetic dataset created from, e.g., a simulation engine like Carla [43] to a real-world dataset like Cityscapes [36].

The semantic clustering was primarily developed for the real to real domain change. Therefore, the experiments are focused on scenarios that fall into this category. For the area of autonomous driving experiments, adapting between the Cityscapes and the BDD [272] datasets are presented. Experiments for adapting between different OCT sensor setups are presented for medical image processing. Section 3.5 of this chapter contains these real to real evaluations.

The simulation to real domain gap is particularly significant in the development of autonomous vehicles. The EasyAdap method presented in Section 3.3.5, which employs a semantic clustering approach combined with self-training in an iterative adaptation system, is especially effective in addressing this domain shift. This method was specifically developed with this scenario in mind. Hence, the experimental evaluation focuses on the adaptation from the synthetic GTA5 [188] and Synthia [191] dataset towards the real-world Cityscapes dataset. Given that the general objective of synthetic data is to yield a network that generalizes well to many domains of the real world, experiments regarding the domain generalization capabilities of the system are presented. Section 3.6 of this chapter contains these synthetic to real evaluations.

The evaluation of the UDA approaches for the domain change contains three models: the source-only model is trained on the source domain only, the Oracle model is trained both on the source and the target domain and the respective UDA model that is trained, supervised on the source domain, and unsupervised on the target domain. The evaluation is done on the target domain. The Oracle is, therefore, the upper bound of the performance and the source-only the lower bound.

3.4.2 Implementation

All of the experiments are based on an implementation of DeepLabV3+[3] with a WideResNet38 [253] backbone. All target domain training images are scaled to match the size of the source-domain images. While the training of the model is done on crops of 400×400 pixels in batches of 24 source- and 24 target domain images, the validation is done on the original-sized images. The training process implements a stochastic gradient descent optimizer with momentum 0.9 and a weight decay of 10^{-4}. The initial learning rate of 0.007 is decayed with a factor $1 - \frac{epoch}{epoch_{max}}$ after each epoch. The parametrization differs slightly depending on the domain change. In the case of the **real to real domain change**, the training is done for 180 epoch. In the case of the **synthetic to real domain change**, the training is done in the source-only setups for 45 epochs. The iterative adaptation steps are stopped early after 20 epochs. The self-training threshold, which is now set to $\beta = \frac{1}{16}$ (see Section 3.3.2). In the real to real scenario, the empirical study showed that $\alpha = 0.9$ promises the best results (α is the centroid update factor). Therefore, $\alpha = 0.9$ was kept for the synthetic to real experiments.

[3]Implementation based on `https://github.com/NVIDIA/semantic-segmentation/tree/sdcnet`

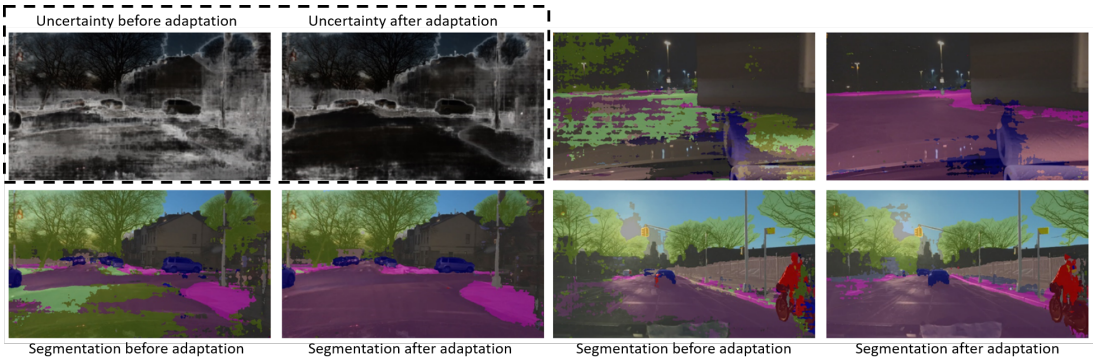


Figure 3.10: The image pairs show the results on the same target domain image before (left) and after (right) adaptation. The adaptation process shows benefits in difficult conditions (e.g., glare and on different road surfaces) and during the shift from day to night. The top left image pair additionally shows the effect of the adaptation on the pixel-wise uncertainty. The adaptation reduced the uncertainty on the target domain, providing a better foundation for inferring pseudo-labels.

3.5 Adaptation In the Real to Real Setting

The following sections present the experimental evaluation of the semantic clustering approach for unsupervised domain adaptation explained in Section 3.3.1. The experiments first analyze a real to real use case in the environment perception for automated driving. The Cityscapes dataset represents the source domain, and the BDD represents the target domain. This scenario represents a domain change between countries and camera setups. The second part of this section analyzes the adaptation between three different OCT scanners for the medical domain. Data from two of them always represent the source domain, and the remaining sensor represents the target domain. This results in three different adaptation scenarios.

3.5.1 Driving Domain

Table 3.5 on page 78 shows the results when adapting from the real-world Cityscapes dataset to the real-world Berkeley Deep Drive dataset (BDD). The challenge, in this case, lies within the variety of weather conditions (rain, snow, sunny), lighting conditions (normal sunlight, direct sunlight causing glaring, twilight, and night), camera types, camera positions, and locations (rural, urban) of the BDD data. On the other hand, the Cityscapes dataset only consists of images from urban parts of Germany recorded under constant weather conditions with a constant camera setup. The domain difference can be observed in Figure 3.1 on page 29.

Table 3.5: Cityscapes to BDD: Evaluation is done on the target domain BDD using the mIoU.

	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bicycle	mIoU
source-only	88.56	49.25	70.57	10.24	22.61	37.86	41.96	42.75	73.29	36.52	89.59	58.36	34.32	84.28	24.27	26.35	0.01	46.51	41.65	46.26
memory bank	84.75	39.51	74.64	14.26	17.93	33.37	31.66	34.61	76.98	33.76	87.85	51.77	26.56	75.94	17.02	41.77	0.0	26.43	21.42	41.59
prototype	89.96	48.27	76.68	17.86	26.74	41.44	42.91	44.17	80.31	26.74	90.05	58.69	41.37	84.53	22.87	37.95	1.01	46.4	37.28	48.17
semantic α0.1	90.21	49.75	74.45	14.65	24.68	37.61	42.91	46.17	79.02	31.07	90.27	61.11	48.97	85.1	31.08	30.64	0.06	55.31	53.89	49.84
semantic α0.9918	90.97	52.21	**78.48**	17.87	30.34	42.7	45.07	44.04	82.6	37.79	90.81	56.93	36.39	81.55	30.54	47.12	0.1	38.79	32.4	49.3
semantic α0.9	90.85	51.44	77.04	14.89	25.98	38.63	43.63	46.19	81.14	36.72	90.65	61.34	48.27	84.63	31.39	49.29	0.6	**57.31**	49.83	51.54
self-training	92.13	55.33	78.28	**18.45**	**34.53**	42.44	45.64	46.1	81.68	40.69	91.13	61.54	47.13	87.13	37.1	43.73	0.01	51.46	44.38	52.58
1. semantic self-training	**93.06**	**57.01**	77.98	17.85	34.43	**42.76**	**48.37**	47.48	**82.25**	**41.17**	91.69	63.66	**54.02**	87.01	**38.08**	50.73	0.08	52.69	49.45	54.2
2. semantic self-training	92.37	56.54	78.12	15.02	33.26	41.95	48.06	**48.53**	81.64	40.69	**91.72**	**64.25**	52.58	**87.48**	37.76	**56.65**	**1.79**	55.99	**55.11**	**54.71**
Oracle	94.24	61.45	84.54	31.13	49.47	50.49	54.92	51.06	86.17	47.48	94.54	66.0	32.51	89.33	47.9	74.41	0.026	53.64	45.86	58.72

These deltas between the datasets cause the performance of a model trained on the Cityscapes dataset to drop when evaluated on the validation set of the BDD dataset. Table 3.5 shows that the mIoU value decreases from 58.72% when trained on both datasets (row: "Oracle") to 46.26% when trained on the source domain only (row: "source-only"). In terms of absolute mIoU values, this is a difference of 12.46%. For example, Figure 3.10 shows that the source-only model has poor quality on twilight and night images. Applying the semantic clustering (self-supervision) method presented in Section 3.3.1 eliminates 43.06% of this gap, i. e., the method achieves an *mIoU* value of 51.54% (row: "semantic α0.9"). These improvements are visible in Figure 3.10. Based on these results, the method outperforms the source-only training for the deltas introduced by glaring, twilight, and nighttime. The uncertainty maps (i. e., the entropy of the class predictions per pixel) in Figure 3.10 show that the method reduces the overall uncertainty as expected with clustering to the class centroids.

The improved results and the more interpretable uncertainty maps are, in turn, a good state for generating high-quality pseudo-labels. The experiments, therefore, include self-training on pseudo-labels generated according to the method explained in Section 3.3.2. Table 3.5 shows that the mIoU on the target domain increases by 1.04% with simple self-training applied (row: "self-training"). Adding the semantic clustering loss into the optimization process increases the improvement (row: "1. and 2. semantic self-training"). Expressed in absolute terms, the mIoU improves by 3.17% in comparison to the training with semantic clustering; and by 2.13% in comparison to the simple self-training. The combination of semantic clustering and self-training closes 64.45% of the original gap of 12.46% mIoU between the source-only and the Oracle model. The two methods (1. and 2. semantic self-training) achieve the best results in 13 out of 19 classes and are even better than the combined training on the source and target domain for the classes *train*, *bicycle*, *rider*, and *motorbike*.

Introducing the Uncertainty weights: The difference between the two versions is the introduction of the uncertainty weights in the semantic clustering introduced in Section 3.3.1.1. In the experiment "2. semantic self-training", the clustering is normalized by the uncertainty weights. In the version "1. semantic self-training", the normalization is not applied. The normalization yields a gain of 0.51% in terms of

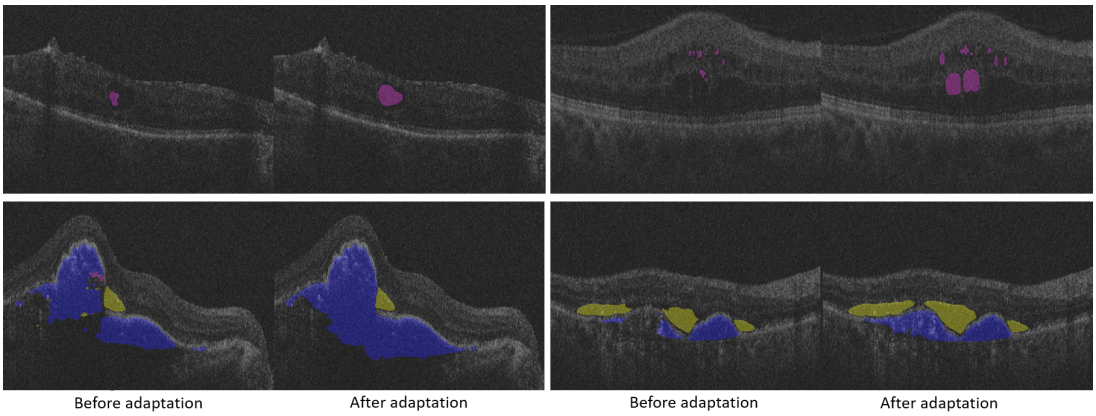


Figure 3.11: The image pairs show qualitative results for the domain shift from the Cirrus + Spectralis source domain to the Topcon target domain. The left image of each pair shows the results of the source-only training, and the right image shows the results of the domain adaptation. The class depicted in purple is the IRF, the blue is the PED, and the yellow is the SRF.

mIoU. This indicates that introducing uncertainty weights can mitigate the influence of cases where the pre-logit target domain representations are clustered to the wrong centroids.

Finetuning the centroid updates: The parameter α, introduced in Section 3.3, controls the update of the centroids representing the classes. As a value between 0 and 1, it weights the old state of the centroids while $1 - \alpha$ weights the update. Different values for α were tested, as shown in Table 3.5. The empirical study shows that $\alpha = 0.9$ promises the best results. The assumption is that a low value like 0.1 causes the centroids to adapt strongly to the updates. This strong influence of the local updates introduces noise into the clustering process, making it hard for the optimization process to converge. If α, on the other hand, is chosen too high (e. g., $\alpha = 0.9918$), then the centroids do not represent the current state of the network, in which case the data is clustered to the wrong centroids. Hence, a value of $\alpha = 0.9$ seems to be a good trade-off as it yields the best results in Table 3.5 compared to the other parameterizations ($\alpha = 0.1$ or $\alpha = 0.9918$).

3.5.2 Medical Domain

This section presents the application of the semantic clustering introduced in Section 3.3.1 to an application in the medical domain (published by Niemeijer et al. [168]). The RETOUCH challenge training dataset was used for these medical image processing experiments. More specifically, the segmentation dataset for the disease-related classes: subretinal fluid (SRF), intraretinal fluid (IRF), and pigment epithelial detach-

Table 3.6: Jaccard (J) and Dice (D) on the target domain for the adaptation from the Cirrus + Spectralis domain to the Topcon domain. For comparison, results of the RETOUCH challenge (mean and winning approach) are given. Note that challenge participants used all domains for training. The best and the second-best approaches are highlighted (**bold** and *italic*).

Approach	IRF (J)	IRF (D)	SRF (J)	SRF (D)	PED (J)	PED (D)	Mean (J)	Mean (D)
source-only	51.3%	67.8%	65.1%	78.9%	*62.5%*	*76.9%*	59.6%	74.5%
semantic-clustering	*53%*	*69.3%*	**70.5%**	**82.7%**	**70.5%**	**82.7%**	**64.6%**	**78.2%**
improvement	1.7%	1.5%	5.4%	3.8%	8%	5.8%	3.8%	3.7%
challenge mean	43.9%	61%	60%	75%	49.3%	66%	51.1%	67%
challenge winner	**56.3%**	**72%**	*68.1%*	*81%*	61.3%	76%	*61.7%*	*76.3%*

ment (PED). This dataset contains annotated subsets of image data acquired by three different sensors: Cirrus, Spectralis, and Topcon (see Figure 3.1 page 29). The Cirrus and Spectralis subsets contain scans of 24 patients, and the Topcon subset 22. No patient is present in more than one sensor dataset. Each scan is subdivided into multiple 2D B-Scans. The number of B-scans depends on the sensor: Cirrus / Topcon 128 B-scans, Spectralis 49 B-scans. The available data is split according to their sensor domains to test the domain adaptation approach. In each split, two sensors comprised the source domain. The remaining sensor represented the target domain. This partitioning leaves us with the three setups, of which the results are presented in Table 3.6 (source domain: Cirrus/Spectralis, target domain: Topcon), Table 3.7 (source domain: Cirrus/Topcon, target domain: Spectralis) and Table 3.8 (source domain: Spectralis/Topcon, target domain: Cirrus).

Results for Adapting between the OCT domain gaps: The three tables present the results of the semantic clustering (see Section 3.3.1) for unsupervised domain adaptation between the OCT sensor deltas and the source-only training, i.e., the training only on the respective source domain. The segmentation quality is reported with the Dice and the Jaccard score (Jaccard is equivalent to IoU). Both of these quality metrics for the segmentation are introduced in Section 2.4.2. The RETOUCH challenge [7] evaluated the mean performance of the approaches on different domains as well. Therefore, the tables also report the result of the best method in the challenge ("challenge winner") and the mean of the methods in the challenge ("challenge mean"). However, one must consider that these approaches were trained on the whole training dataset, containing examples of all domains, and tested on a different test set than the approaches presented in this chapter. Although the results are not directly comparable, even the source-only training on only the respective source domains in the OCT adaptation scenarios achieves higher Dice and Jaccard scores in almost all classes in

Table 3.7: Jaccard (J) and Dice (D) on the target domain for the adaptation from the Topcon + Cirrus domain to the Spectralis domain. For comparison, results of the RETOUCH challenge (mean and winning approach) are given. Note that challenge participants used all domains for training. The best and the second-best approaches are highlighted (**bold** and *italic*).

Approach	IRF (J)	IRF (D)	SRF (J)	SRF (D)	PED (J)	PED (D)	Mean (J)	Mean (D)
source-only	40.2%	57.4%	*83%*	*90.7%*	*65%*	*78.8%*	62.7%	75.6%
semantic-clustering	49.9%	66.6%	**84.5%**	**91.6%**	**67.4%**	**80.5%**	**67.3%**	**79.5%**
improvement	9.7%	9.2%	1.5%	0.9%	2.4%	1.7%	4.6%	3.9%
challenge mean	*52.7%*	*69%*	39.9%	57%	51.5%	68%	48%	65%
challenge winner	**77%**	**87%**	57.5%	73%	61.3%	76%	*64.7%*	*78.6%*

Table 3.8: Jaccard (J) and Dice (D) on the target domain for the adaptation from the Topcon + Spectralis domain to the Cirrus domain. The RETOUCH challenge results (mean and winning approach) are given for comparison. Note that challenge participants used all domains for training. The best and the second-best approaches are highlighted (**bold** and *italic*).

Approach	IRF (J)	IRF (D)	SRF (J)	SRF (D)	PED (J)	PED (D)	Mean (J)	Mean (D)
source-only	59.6%	74.7%	*71.1%*	*83.1%*	*72.3%*	*83.9%*	*67.6%*	*80.6%*
semantic-clustering	*61.9%*	*76.5%*	**76.2%**	**86.5%**	**75.8%**	**86.2%**	**71.3%**	**83.1%**
improvement	2.3%	1.8%	5.1%	3.4%	3.5%	2.3%	3.7%	2.5%
challenge mean	57.5%	73%	46%	63%	58.7%	74%	54.1%	70%
challenge winner	**74%**	**85%**	56.3%	72%	69%	82%	66.1%	79.6%

all settings compared to the challenge mean and is only slightly worse than the winner. This comparison is especially interesting since the source-only model, unlike the challenge participants, was not trained on data from the sensor type the evaluations were performed on. This at least indicates that the network architecture and training strategy employed are superior to the approaches presented in the RETOUCH challenge (approaches from 2019). The segmentation is worse only for one class (the IRF) in only one configuration. Topcon+Cirrus comprise the source in this configuration, and Spectralis the target domain.

Apart from improving architecture and training strategy, this work aimed to adapt unsupervised learning to new sensor domains. Table 3.6, 3.7, and 3.8 show that the semantic clustering algorithm improves the performance of the source-only training in

all domain changes in all classes. Figure 3.11 on page 79 shows qualitative examples of the improvement achieved by the domain adaptation approach. This improvement is especially strong in the domain change from Topcon and Cirrus to Spectralis. The results for IRF, e.g., are 9.7% in terms of Jaccard and 9.2% in terms of Dice better. The adapted model outperforms even the challenge winner in almost all classes, even though no manually annotated data of the evaluation OCT sensor was used during training. Only the IRF class is better for the challenge winner. This is likely due to the low performance of the source-only training, even though the adaptation method's improvements are strong.

These results show the effectiveness of the semantic clustering for unsupervised domain adaptation, even in the medical domain, which proves the approach's versatile applicability. In some scenarios, the adaptation that does not require any manual annotations even outperformed approaches that were trained on target domain data. Therefore, the semantic clustering in this scenario is an interesting alternative to the costly manual annotation by medical professionals.

3.6 Adaptation from Synthetic to Real Data

This section evaluates the EasyAdap approach (see Section 3.3.5) regarding its domain adaptation and domain generalization quality. To that, an evaluation of its performance on the domain shift from synthetic to real-world data via adaptations from GTA5 [188] and Synthia [191] to Cityscapes [36] is performed, respectively. The synthetic to real domain shift is especially interesting since synthetic data usually provides labels automatically. Given the supervised training on the synthetic domain and the unsupervised domain adaptation to the real world, the whole training process is free of manual annotation. Furthermore, the section explores the generalization capabilities to real-world data that EasyAdap introduces.

3.6.1 Synthetic to Real Domain Change

The most common benchmark used to compare approaches in the synthetic to real domain shift is the adaptation from the GTA5 or Synthia dataset to the real-world Cityscapes dataset.

The **GTA5** [188] dataset comprises frames extracted from the Grand Theft Auto V game and provides semantic segmentation annotations. This game mainly represents a virtual environment set in California. The game was sampled by taking every 40th frame while simultaneously introducing various lighting and weather conditions. The aim was to create a diverse synthetic distribution. The **Synthia** [191] dataset was created similarly to provide a synthetic training dataset for semantic segmentation. It includes different textures and lighting conditions, as well. Additionally, it contains a larger variety of different viewing angles. It is, however, based on the Unity platform.

Table 3.9: GTA5 to Cityscapes: Evaluation is done on Cityscapes using the mIoU.

	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bicycle	mIoU
source-only DLv2	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
AdapSeg [233]	86.5	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	7.2	29.5	32.5	41.4
CyCADA [74]	86.7	35.6	80.1	19.8	17.5	38.0	39.9	41.5	82.7	27.9	73.6	64.9	19.0	65.0	12.0	28.6	4.5	31.1	42.0	42.7
CLAN [143]	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
APODA [264]	85.6	32.8	79.0	29.5	25.5	26.8	34.6	19.9	83.7	40.6	77.9	59.2	28.3	84.6	34.6	49.2	8.0	32.6	39.6	45.9
PatchAlign [234]	92.3	51.9	82.1	29.252	25.1	24.5	33.8	33.0	82.4	32.8	82.2	58.6	27.2	84.3	33.4	46.3	2.2	29.5	32.3	46.5
ADVENT [238]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5
sem. self-train. [165]	82.5	43.9	76.4	31.7	24.7	45.2	45.6	22.5	87.1	30.9	82.6	71.0	41.8	86.5	28.0	27.8	0.01	25.5	27.3	46.4
BDL [123]	91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	43.6	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5
CBST [128]	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
MRKLD [305]	91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	26.9	26.0	42.3	47.1
FADA [242]	91.0	50.6	86.0	43.4	29.8	36.8	43.4	25.0	86.8	38.3	87.4	64.0	38.0	85.2	31.6	46.1	6.5	25.4	37.1	50.1
CAG [278]	90.4	51.6	83.8	34.2	27.8	38.4	25.3	48.4	85.4	38.2	78.1	58.6	34.6	84.7	21.9	42.7	41.1	29.3	37.2	50.2
Seg-Uncertainty [291]	90.4	31.2	85.1	36.9	25.6	37.5	48.8	48.5	85.3	34.8	81.1	64.4	36.8	86.3	34.9	52.2	1.7	29.0	44.6	50.3
CLST [150]	92.8	53.5	86.1	39.1	28.1	28.9	43.6	39.4	84.6	35.7	88.1	63.9	38.3	86.0	41.6	50.6	0.1	30.4	51.7	51.6
SAC [3]	90.4	53.9	86.6	42.4	27.3	45.1	48.5	42.7	87.4	40.1	86.1	67.5	29.7	88.5	49.1	54.6	9.8	26.6	45.3	53.8
Coarse2Fine [147]	92.5	58.3	86.5	27.4	28.8	38.1	46.7	42.5	85.4	38.4	91.8	66.4	37.0	87.8	40.7	52.4	44.6	41.7	59.0	56.1
ProDA [283]	87.8	56.0	79.7	46.3	44.8	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1.0	48.9	56.4	57.5
source-only	46.44	10.75	62.2	1.21	16.75	22.16	18.93	4.65	72.76	3.73	63.95	50.91	7.44	68.55	26.1	4.18	0	3.75	1.66	25.58
source-only aug	58.39	25.44	68.01	25.55	26.77	40.25	44.62	19.32	84.37	30.78	56.56	69.17	36.64	74.29	24.13	10.86	0.9	29.34	21.07	38.8
source-only uni 50%	57.87	32.03	58.53	23.62	25.06	42.38	45.17	28.22	83.36	26.02	81.01	70.16	40.24	80.07	20.05	15.95	1.02	32.47	23.32	41.4
source-only uni 100%	76.03	34.02	75.52	29.25	29.72	46.55	45.91	27.96	82.52	21.47	78.8	69.51	34.58	86.14	25.98	24.64	0	32.56	23.95	44.5
Self-Train	63.4	40.99	60.85	41	37.21	45.28	51.06	38.45	87.34	33.52	79.02	70.38	35.67	90.63	42.02	47.93	13.77	37.92	18.03	49.18
Easy Adap [166]	87.84	56.1	80.68	37.21	40.12	49.39	55.04	47.18	86.87	39.54	85.35	69.93	42.13	90.65	52.12	61.45	0	42.13	46.39	56.32
Oracle	98.01	84.41	92.07	49.66	59.69	64.43	68.76	78.22	92.36	63.49	94.3	82.17	62.3	94.82	80.36	85.76	79.74	65.99	76.93	77.55

The Cityscapes dataset [36] represents a subset of the relevant real-world data. It is recorded under constant conditions in German cities. Therefore, this work additionally investigates the generalization of the adapted models to the **BDD** [273] and **ACDC** [197] datasets in Section 3.6.2.2. The former contains images crowd-sourced from dash cams and multiple weather and environmental conditions in the USA. The latter (ACDC) dataset was specifically designed to enhance the robustness of semantic segmentation models for driving scenes under adverse weather conditions, such as fog, nighttime, rain, and snow. It was recorded using a constant sensor setup in Zurich, Switzerland.

3.6.2 EasyAdap: Synthetic to Real Results

The following sections analyze EasyAdap's performance on the synthetic to real domain change. First, it is compared to the existing state-of-the-art in terms of its UDA performance. Following up, the resulting domain generalization features are compared, which represents an important feature of UDA methods that is often overlooked. Finally, an ablation study provides insights into the interaction of EasyAdap's building blocks.

3.6.2.1 Comparison to Existing Approaches

GTA5 to Cityscapes: EasyAdap achieves near state-of-the-art performance in terms of mIoU on the GTA5 to Cityscapes domain change: EasyAdap achieves 56.32% mIoU,

Table 3.10: Synthia to Cityscapes: Evaluation is done on Cityscapes using the mIoU.

	road	sidewalk	building	wall*	fence*	pole*	traffic light	traffic sign	vegetation	sky	person	rider	car	bus	motorbike	bicycle	mIoU	mIoU*
source-only DLv2	64.3	21.3	73.1	2.4	1.1	31.4	7.0	27.7	63.1	67.6	42.2	19.9	73.1	15.3	10.5	38.9	34.9	40.3
AdapSeg [233]	79.2	37.2	78.8	-	-	-	9.9	10.5	78.2	80.5	53.5	19.6	67.0	29.5	21.6	31.3	-	45.9
PatchAlign [234]	82.4	38.0	78.6	8.7	0.6	26.0	3.9	11.1	75.5	84.6	53.5	21.6	71.4	32.6	19.3	31.7	40.0	46.5
CLAN [143]	81.3	37.0	80.1	-	-	-	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	-	47.8
APODA [264]	86.4	41.3	79.3	-	-	-	22.6	17.3	80.3	81.6	56.9	21.0	84.1	49.1	24.6	45.7	-	53.1
ADVENT [238]	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	41.2	48.0
BDL [123]	86.0	46.7	80.3	-	-	-	14.1	11.6	79.2	81.3	54.1	27.9	73.7	42.2	25.7	45.3	-	51.4
FADA [242]	84.5	40.1	83.1	4.8	0.0	34.3	20.1	27.2	84.8	84.0	53.5	22.6	85.4	43.7	26.8	27.8	45.2	52.5
CBST [245]	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	78.3	60.6	28.3	81.6	23.5	18.8	39.8	42.6	48.9
MRKLD [305]	67.7	32.2	73.9	10.7	1.6	37.4	22.2	31.2	80.8	80.5	60.8	29.1	82.8	25.0	19.4	45.3	43.8	50.1
CAG UDA [278]	84.7	40.8	81.7	7.8	0.0	35.1	13.3	22.7	84.5	77.6	64.2	27.8	80.9	19.7	22.7	48.3	44.5	51.5
Seg-Uncertainty [291]	87.6	41.9	83.1	14.7	1.7	36.2	31.3	19.9	81.6	80.6	63.0	21.8	86.2	40.7	23.6	53.1	47.9	54.9
Coarse2Fine [147]	75.7	30.0	81.9	11.5	2.5	35.3	18.0	32.7	86.2	90.1	65.1	33.2	83.3	36.5	35.3	54.3	48.2	55.5
CLST [150]	88.0	49.2	82.2	16.3	0.4	29.2	31.8	23.9	84.1	88.0	59.1	27.2	85.5	46.6	28.9	56.5	49.8	57.8
SAC [147]	89.3	47.2	85.5	26.5	1.3	43.0	45.5	32.0	87.1	89.3	63.6	25.4	86.9	35.6	30.4	53.0	52.6	59.3
ProDA [283]	87.8	45.7	84.6	37.1	0.6	44.0	54.6	37.0	88.1	84.4	74.2	24.3	88.2	51.1	40.5	45.6	55.5	62.0
source-only	8.6	11.58	32.01	1.45	0	30.25	18.55	9.1	74.57	68.74	56.63	8.16	66.98	12.05	3.41	9.69	26.31	29.24
source-only aug	55.14	30.54	69.12	4.69	0	40.35	25.78	23.55	80.15	76.93	61.92	21.59	40.78	18.6	17.05	27.6	37.13	42.21
source-only uni 50%	64.39	28.34	72.2	3.35	1.19	40.69	27.51	20.18	79.93	59.2	64.77	24.07	79.04	24.1	17.76	20.48	39.2	43.22
source-only uni 100%	79.0	33.18	68.75	3.16	0.38	42.07	25.79	26.05	78.67	76.76	61.28	24.76	80.72	26.64	18.73	28.94	42.18	48.41
Easy Adap [166]	84.48	46.12	74.69	0.16	0.04	47.14	49.77	31.94	77.84	85.11	73.33	36.14	86.96	46.06	28.89	23.01	49.48	57.25
Oracle	98.01	84.41	92.07	49.66	59.69	64.43	68.76	78.22	92.36	94.3	82.17	62.3	94.82	85.76	65.99	76.93	78.12	92.77

which is comparable to the currently best approach, ProDA, with 57.5% mIoU. Similar to ProDA, the improvement over other approaches is gained from classes like *traffic light, traffic sign, fence, rider,* and *motorbike*, which are difficult to learn. The EasyAdap method outperforms approaches that have similarities to the presented method on this domain change. In particular, this includes CLST and CAG with 50.2% and 51.6% mIoU, respectively. Except for ProDA, the same holds for all hybrid methods. ProDA, however, is hard to tune due to its complex architecture (four structurally different training stages) and hyperparameter tuning. Advancing the method from [165] into an iterative domain adaptation process improves the results from 46.4% mIoU to 56.32% due to synergistic effects between the semantic clustering and self-training.

Synthia to CS: Table 3.10 shows the performance of unsupervised domain adaptation methods from Synthia to the Cityscapes dataset. Without changing any hyperparameters tuned for the GTA5 to Cityscapes domain change, the approach still performs well, ranking behind ProDA, SAC, and CLST, all of which are also well-performing in the adaptation from GTA5 to the Cityscapes dataset.

3.6.2.2 Domain Generalization

This section evaluates the domain generalization capabilities of EasyAdap and compares them with those of several other UDA approaches. To that, Table 3.11 shows the performance of several models on the datasets Cityscapes, BDD [273], and four different domains of the dataset ACDC [197]. No model has seen samples of BDD

Table 3.11: Domain generalization: Methods trained on GTA5 (source-only) and methods adapted from GTA5 to Cityscapes tested on different real-world domains.

Model	Cityscapes	BDD	rain	fog	snow	night
source-only DLv2 [233]	36.6	36.5	33.6	40.2	33.4	8.6
source-only DLv2 [3]	40.8	35.1	32.9	31.3	28.7	7.1
source-only DLv3+	25.58	28.45	28.37	24.69	23.39	3.7
source-only aug DLv3+	38.9	31.3	32.02	29.04	27.6	5.75
source-only uni 100% DLv3+	44.06	36.77	33.27	35.75	28.28	7.63
AdapSeg [233]	42.4	37.4	30.8	35.4	27.9	7.4
Seg-Uncertainty [291]	50.3	35.7	35.9	41.4	37.4	14.0
SAC [3]	53.8	41.55	39.6	44.7	34.9	15.6
ProDA [283]	57.5	47.5	43.1	49.2	40.7	15.4
EasyAdap [166]	56.6	46.69	43.02	48.56	38.87	14.38
Oracle (Cityscapes model)	77.55	46.26	45.58	61.22	47.65	17.87

or ACDC during the training. Furthermore, the source-only models did not see the Cityscapes samples during training.

BDD differs from the target domain of the adaptation (Cityscapes) by containing a diverse set of weather and lighting conditions and by being recorded across the USA instead of Germany. ACDC, which was recorded in Europe, defines the specific partitions *rain, fog, snow,* and *night*. Hence, regarding the models trained on GTA5 (and Cityscapes without ground truth), these datasets include domain shifts regarding the environment, weather, and sensors.

The improvements of the source-only models due to data augmentation and additional class-uniform sampling also transfer to the unseen domains of BDD and ACDC. Considering the DeepLabV2 [3, 233] and DeepLabV3+ source-only models, there is no clear superior model. Note that the "Oracle" model (trained on the Cityscapes dataset) yields similar results on BDD as the adaptation approaches ProDA and EasyAdap from GTA5 to Cityscapes. Apart from that, the "Oracle" model always outperforms the domain adaptation approaches significantly. While ProDA leads the board of domain adaptation approaches, EasyAdap achieves very close results. Furthermore, EasyAdap outperforms the other domain adaptation approaches (see Table 3.11). Section 3.6.3 discusses these results in the context of transferring knowledge from the synthetic to the real-world data and the generalization the adaptation introduces.

3.6.2.3 Source-Only Training

The quality of the source-only training is crucial for the later creation of the first pseudo-labels. To study the design decision of the source-only training, Table 3.9 and Table 3.10 (page 83 and 84) show the performance of different source-only models: trained without any augmentation (source-only); trained with color jitter, Gaussian

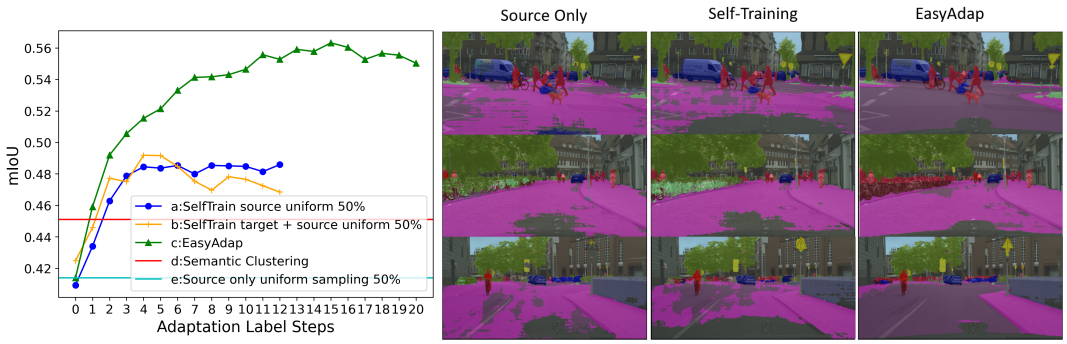


Figure 3.12: Left: Best mIoU per adaptation step; self-training (a); self-training+uniform sampling on target domain (b); self-training+uniform sampling+semantic clustering (c); uniform sampling+semantic clustering (d); uniform sampling source only (e). Right: Qualitative examples.

blurring, and random scaling (source-only aug); and trained with augmentation and class-uniform sampling (source-only uni 50% and source-only uni 100%). On both datasets, GTA5 and Synthia, the augmentation improves the plain source-only model's quality by over 10% mIoU. An additional class-uniform sampling improves the model's quality further by over 5% mIoU. Sampling the complete batch via class-uniform sampling (source-only uni 100%) outperforms a mixed batch of random and class-uniformly sampled images (source-only uni 50%). Section 3.6.3 discusses why uniform sampling introduces this generalization.

3.6.2.4 Impact of the Adaptation Features

This work conducted several experiments to study the impact of EasyAdap's features on the model's quality. Figure 3.12 shows the best mIoU values on the validation set per adaptation step, which consists of 20 training epochs. This work successively enabled self-training, class-uniform sampling and the semantic clustering on the target domain (see Section 3.3.1 and 3.3.2). Figure 3.12 shows that self-training yields similar improvements as a combination of self-training with an additional class-uniform sampling on the target domain. Additionally, introducing semantic clustering to the training process shows a steeper learning curve in the first few adaptation steps and yields a more durable learning behavior. For example, the two self-training learning processes without semantic clustering stagnate after four adaptation steps, but the model trained with additional semantic clustering still increases in quality. Section 3.6.3 discusses this synergistic effect and provides a possible explanation for it.

3.6.3 Discussion of EasyAdap's Features

Does domain adaptation transfer knowledge? Table 3.11 shows that the target model trained on Cityscapes (*"Oracle"* model) achieves similar performance on BDD as the ProDA and EasyAdap models adapted to the Cityscapes dataset. While the Cityscapes model has to overcome the domain change from sunny to rainy weather included in BDD, the adaptation methods need to overcome the domain change from synthetic to real-world. The Cityscapes model also achieves a similar performance on ACDC's rain domain to that of the ProDA and EasyAdap models. In this case, the target domain consists of rainy images only, which challenges the Cityscapes model even more, while the adapted models from GTA5 seem to play out their advantage of having seen synthetic rainy images during their training. On the other hand, the Cityscapes model outperforms the adapted models on the foggy and snowy domains, which are unseen for all adapted models. A possible interpretation is that the additional domain gap from the synthetic to the real world for the adapted models is the reason for the performance drop of the adapted models compared to the Cityscapes model. Hence, the adapted models show the behavior of knowledge transfer (here, rainy images) from synthetic to real-world domains. This observation supports approaches adapting from synthetic to real-world domains. Moreover, it supports the demand for richer synthetic datasets regarding different domains, such as sensor, weather, lighting, and environmental domains.

How to support domain generalization through adaptation? Since domain adaptation to every possible encountered domain is infeasible in real-life applications, this work explicitly addresses and studies domain generalization effects of unsupervised domain adaptation methods, including the EasyAdap method introduced in this chapter. Section 3.6.2.2 and Table 3.11 (page 85) show stronger generalization effects for some of the adaptation methods. In particular, SAC, ProDA, and EasyAdap show significant improvements in the unseen domains of BDD and ACDC. These methods apply self-training while AdapSeg does not, and Seg-Uncertainty only applies self-training without improving the pseudo-labels through recreation. These observations open the question of whether self-training, which mimics supervised training in both domains, yields these strong generalization effects.

What are the reasons for synergistic effects? The experiments in Section 3.6.2.4 and Figure 3.12 show a synergistic effect since enabling self-training and semantic clustering gains 15% mIoU while enabling only one of the mechanisms gains only 8% and 4%, respectively. Hence, simultaneously enabling both mechanisms gains another 3% atop each single mechanism. This supports the interpretation that there is a synergy between self-training and semantic clustering, i.e., they support each other. On the one hand, self-training aims to mimic a supervised training on the source and target domain, which helps the training process to generalize upon both domains. This generalization effect yields a better alignment of the feature distribution in both

domains, which in turn helps the semantic clustering to attract the target-domain features to the correct source-domain class centroids. On the other hand, the semantic clustering reduces the number of correct but ignored pseudo-labels by sharpening the feature space and therefore reducing the uncertainty of correctly classified pixels.

How does the sampling strategy affect generalization? Sections 3.6.2.2 and 3.6.2.3 show that the source-only models generalize well to unseen real-world data. Combining a strong data augmentation with a class-uniform sampling improves the domain generalization by avoiding overfitting. On the one hand, rare classes gain weight in the training by oversampling, but the strong augmentation avoids overfitting. On the other hand, large-area classes such as vegetation and terrain lose weight compared to rare classes, which again avoids overfitting the training data.

3.7 Conclusion and Outlook

Structural changes between the training distribution of a neural network and the distribution during the application to a target domain are a common scenario in real-world applications. Common examples are e.g. the adaptation between different environments (countries, weather, lighting, ...) in autonomous driving or the adaptation between different sensors in medical image processing. This introduces the need to adapt the neural network from the source to the target domain. Unsupervised domain adaptation is especially attractive since it closes the delta between the domains without human intervention. Given its unsupervised training, UDA provides an approach for bringing down the cost of annotation (c.f. challenge 1 from Chapter 1).

The resulting interest in this field of research is evident from Figure 2.7 on page 24, showing a large number of published papers. This chapter provides a comprehensive survey of the UDA approaches, which covers significantly more approaches than previous survey papers (by a factor of three). The presented work resulted in a publicly accessible website, which includes a leaderboard for the important synthetic to real use-case (`https://uda-survey.github.io/survey/leaderboard`). Thus, this survey helps to break the complexity that is introduced by a large amount of research. The chapter introduces a categorization into input, feature, and output space adaptation approaches and identifies sub-clusters inside these categories. It further points out the recent trend to use methods that apply multiple of these approaches (hybrid methods) and critically reflects the complexity that is therefore introduced. This complexity hinders the application to real-world scenarios in which the finetuning of hyperparameters is difficult.

Apart from presenting a survey, this chapter introduces novel methods for UDA. A clustering approach is introduced to align the feature space distributions between the source and target domain. The aim is to cluster the target domain feature space representations to their corresponding class-based source domain centroids. These

class-based source domain centroids represent the average representation of a class in the source domain. The effectiveness of this method is demonstrated in medical image processing by adapting between different OCT sensors and in autonomous driving by adapting between different camera types and environments. For the application in the environment perception of autonomous cars, this work additionally presents the application from synthetic to real-world data. In this scenario, the feature space clustering is combined with self-training, introducing a synergistic effect. This synergistic effect allows for the low complexity in the architecture results that are very close to the state of the art in the competitive synthetic to real-world domain change. Extensive research has been conducted on adapting synthetic data to real-world scenarios. This interest stems from the fact that synthetic data eliminates the need for manual labeling, significantly reducing human effort. Synthetic data however has got the problem of a large domain gap to the real world, which usually introduces the need for complex approaches, as the survey shows. The EasyAdap approach, therefore, is especially relevant.

Finally, the chapter investigates, how knowledge from the synthetic domain can be transferred to the real-world during the adaptation process and how UDA can introduces domain generalization this way. Therefore the next logical step is the targeted generation of synthetic data, which is studied in the next chapter. By understanding and optimizing the generation of synthetic data to facilitate adaptation, this thesis moves closer to models that generalize well to many domains and distributions (see Chapter 5.).

Chapter 4

Optimizing Synthetic Datasets by Targeted Image Generation

This chapter is partially derived from my previously published works, specifically "Synthetic Dataset Acquisition for a Specific Target Domain"[169], and "TSynD: Targeted Synthetic Data Generation for Enhanced Medical Image Classification" [170]. Some text, figures, and findings have been re-utilized or adapted from these publications. Co-Authors of these Publications are Jan Ehrhardt, Hristina Uzunova, Sudhanshu Mittal, Thomas Brox and Heinz Handels.

4.1 Introduction and Motivation

Chapter 2 described the intelligent creation of real-world datasets. However, even when the AL circle from Chapter 2 is followed, there are two major drawbacks to creating real-world datasets. Even though real-world images are generally easy to collect, it is hard to collect relevant images or sensor data (c.f. problem 2 in the introduction). Of course, this is the prerequisite for AL since the acquisition function can only choose relevant data from the unlabeled pool if this pool contains such data. The work that is presented in Chapter 2 shows that data obtained from measurement campaigns is redundant. Relevant data is hence very sparse, introducing the need for large measurement campaigns to capture such data. Given an unlabeled pool containing relevant data and the correct dataset selection was performed, labeling remains a problem (c.f. problem 1 in the introduction).

Synthetic data can resolve both of these issues. Labeling does not need human intervention since one has almost perfect information about the rendered scene. The labeling time or cost is hence negligible. What is more, one has almost perfect control over the generated scene. That means it is theoretically possible to create relevant data by choosing the simulation parameters correctly. As autonomous vehicle datasets become larger, it becomes more difficult to encounter relevant novel data. However, self-driving cars must be robust against rare critical scenarios to be used in the real world. Synthetic data provides a solution by simulating such scenarios. In medical image processing, datasets are smaller. The image acquisition process is not as con-

trollable as in the case of autonomous vehicles since it is often produced as a byproduct of clinical practice and hindered by data privacy regulations. This lack of controllability makes deliberate measurement campaigns difficult. Simulated data for medical images can also represent edge cases that have not been recorded yet.

Synthetic data, however, brings its own challenges. First, given the current simulation engines, the sensor data produced still has a large appearance gap from real-world images. Second, even though a large degree of control over the scene generation is given, choosing the simulation parameters to produce relevant data remains a problem. Third, the simulation's diversity still depends very much on the human modeling of the scene.

This chapter offers solutions to these challenges. The first part of this chapter is based on "Synthetic Dataset Acquisition for a Specific Target Domain" by Niemeijer et al. [169] and shows how to generate/select relevant data from the synthetic world of a simulation engine. The domain of application in this work is autonomous driving. In this context the chapter includes the work "Generalization by Adaptation: Diffusion-Based Domain Extension for Domain-Generalized Semantic Segmentation" by Niemeijer et al. [171], which shows how to bridge the domain gap between the synthetic data from such simulation engines and the real world using generative models and offers methods to extend the distribution, effectively mitigating the need for human modeling of 3D worlds. The second part of this chapter is based on the paper "TSynD: Targeted Synthetic Data Generation for Enhanced Medical Image Classification" by Niemeijer et al. [170] and extends the idea of synthetic data generation through generative models to the domain of medical image processing. In this work, the focus is on guiding the generation process towards generating high epistemic uncertainty data. Hence, the chapter contributes solutions for creating relevant synthetic data while reducing the need to manually model the given synthetic environment.

4.2 Intelligent Generation and Selection

Synthetic data represents a promising building block for training perception systems. Simulation is a crucial alternative for automatically creating labeled data, especially for tasks like semantic segmentation, which require about 90 minutes per frame to annotate [36]. Hence, the creation of simulation engines has been the focus of the computer vision community. However, the number of possible scenarios and configurations such simulators can produce is still uncountably large. Due to the constraints in storage space and training hardware, intelligent sampling from the simulation is crucial. Generating all necessary cases would require more memory than is typically available, and even if sufficient memory were available, training on such a dataset would be infeasible. This infeasibility is due to the slow convergence of the training caused by the redundancy in the generated dataset, i.e., important data points occur

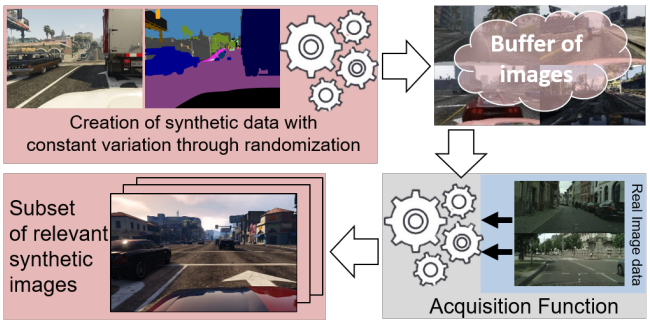


Figure 4.1: A system for creating a synthetic dataset. The simulation creates images while constantly introducing large variations. The images are written into a buffer. The acquisition function selects the synthetic images relevant to the real-world target domain. As a result, a smaller subset of relevant synthetic images is obtained.

rarely during training. Some parts of synthetic data could even introduce a bias that negatively affects real-world performance.

This chapter introduces the research direction of creating intelligent acquisition functions for synthetic data. Such acquisition functions are challenging to construct. Other than in active learning, the domain from which these functions sample is unequal to that to which the trained model is applied. Therefore, the objective is to find synthetic data to train a model that performs well on real data. Performing the selection process based only on the synthetic data is thus insufficient, yet most synthetic datasets in the literature were created this way. For example, the Synthia [191] and the GTA5 [188] dataset both selected synthetic images either randomly or uniformly while only integrating knowledge about certain types of relevant scenes (day, night, weather, ...). However, they did not explicitly optimize their acquisition for selecting synthetic data that best represents the given real-world data and did not explicitly optimize for selecting correspondences for difficult scenes in the real world.

This work addresses the limitations of such existing acquisitions. The objective is to score the value of a simulated frame by estimating its usefulness in training perception models to perform well in the real-world target domain. More specifically, the focus is on the challenging setting of adapting from the created synthetic dataset to the real-world use case via unsupervised domain adaptation. Integration of unsupervised domain adaptation aims to leverage the capabilities of adaptation techniques and minimize the need for simulation of aspects that can be effectively covered by domain adaptation. The performance of the acquisition function is hence defined by how well it facilitates the adaption. This work seeks to develop an intelligent acquisition strategy that selects synthetic data points based on their potential to enhance adaptation.

Figure 4.2: Left: examples from the GTA5 dataset. Middle: examples from the Synthia dataset. Right: examples from the CARLA simulation environment.

The following sections showcase the performance of several strategies. On the one hand, sampling approaches based on objectives that are independent of the real-world target domain are analyzed. Class uniform sampling and active learning acquisition functions like entropy and coreset are utilized to assign a value to the synthetic images. On the other hand, sampling approaches are analyzed, which take advantage of a key assumption underlying many semi-supervised learning methods: "Unsupervised learning performs best if correspondences between the labeled and unlabeled sets exist". Therefore, challenging images are identified in the real world, and their counterparts are searched for in the synthetic world. For repeatability, the experiments utilize the GTA-5 dataset to represent the synthetic world in which sampling can be performed. The Cityscapes and ACDC datasets represent the target domain. This setting simulates a scenario that is demonstrated in Figure 4.1, where the assumption is that a simulation engine (e.g., CARLA [43]) produces simulation data constantly and writes it to a buffer. The aim is to acquire the optimal subset of synthetic images from this buffer.

4.2.1 Common Simulation Engines and Acquisition Strategies

Several existing synthetic simulation engines and acquisition strategies have been used to create synthetic datasets. These engines are crucial in generating diverse and realistic synthetic data for various applications. The following discusses some notable examples:

The **GTA5** [188] dataset was created by extracting frames from the game Grand Theft Auto V (GTA5) and annotating them with semantic segmentation labels. Frames were recorded during gameplay. The authors sampled the synthetic frames by taking every 40th frame. The aim was to capture diverse scenes within the game world. The dataset mainly aims to provide a large-scale, pixel-level annotated dataset for training semantic segmentation models.

The **SYNTHIA** [191] dataset provides synthetic images with semantic segmentation annotations. The dataset was generated using a virtual city created with the Unity development platform. The synthetic frames were sampled from multiple cam-

era viewpoints. The authors introduced various illumination conditions and textures to create a diverse distribution. The dataset aims to improve segmentation performance on real-world data by combining SYNTHIA with real-world data, just like the GTA5 dataset.

The **SHIFT** [222] dataset is built to analyze autonomous driving domain shifts. It is built on data from CARLA [43] and comprises various domain shifts, including various driving environments and weather conditions. The dataset provides annotations for tasks relevant to the environment perception of self-driving cars, such as semantic segmentation, detection, and depth estimation. The dataset contains 2.500.000 of such annotated frames with a continuous shift in the given domain, which allows for analyzing and training segmentation models under different domain shifts.

Acquisition strategies: The Synthia, GTA5, and SHIFT datasets were sampled randomly or uniformly. For example, the GTA5 dataset chose every 40th frame. They all tried to capture various weather, viewpoints, and lighting conditions. However, they did not choose frames w.r.t. objectives like novelty uniqueness or relevance, as active learning approaches do (c.f. Chapter 2.2). As argued in the introduction of this section ideally synthetic dataset should be sampled by acquisition functions that select frames that are relevant to the real-world target domain. That means they should represent unknown and, therefore, critical information about the target domain and ideally be similar to the real-world target domain. There are some approaches that present solutions for acquiring corner case data in the synthetic world.

For instance, Kowol et al. [111] focus on augmenting training data in automated driving by generating safety-critical driving situations called “corner cases”. These “corner cases” were simulated using CARLA [43]. A test rig is employed, in which one operator observes the virtual image while another monitors the output of a semantic segmentation network applied to the same image. When the network’s prediction is unsafe for driving the car, the first operator intervenes, and the image is saved as a corner case. The objective is to generate corner cases that challenge the network’s perception in safety-critical scenarios.

This acquisition strategy does not explicitly consider the actual real-world target domain when evaluating a frame. I.e. the reason the image is difficult to segment and interpret could be due to image information that occurs in the synthetic but not the real world. There are, however, some “sampling” strategies that have been developed for other tasks that take the real-world target domain into account.

Sun et al. [219] introduce a method for transfer learning in semantic segmentation relevant to this chapter. This work, however, uses annotations both in the real world and the synthetic world. The aim is to select regions of the synthetic images that yield a high similarity to the real-world images. They utilize hierarchical weighting networks to score similarity between pixels, regions, and the whole image. They improve training the segmentation model on the combined real and subsampled synthetic domain.

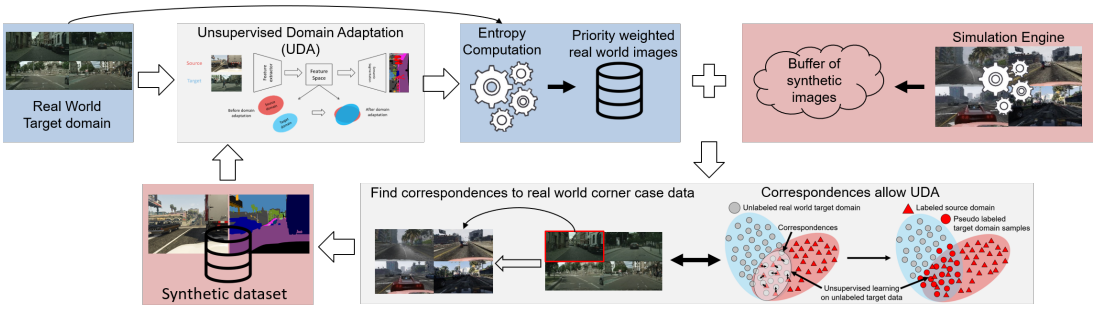


Figure 4.3: An initial synthetic training set is provided. Unsupervised domain adaptation (UDA) is applied to adapt the model to the real-world target domain. Priority scores are computed using the trained model for real-world images. Real-world images where the segmentation performs poorly receive a high priority score. Synthetic images with the highest correspondence to the prioritized real-world images are selected. The chosen synthetic images are incorporated into the training dataset. The images represent information about the real-world failure cases.

Kim et al. [105] present a method for semi-supervised domain adaptation, i.e., a small amount of data from the real-world target domain is given. They introduce a sampling of the synthetic source domain. This sampling is done on the pixel level. The method first trains a network on the small real-world target domain set. This pre-trained network is then applied to the synthetic source domain. The labels are discarded if the network's prediction does not agree with the ground truth. The idea is that the remaining pixels are similar to the target domain. Additionally, the method conducts an image-level subsampling based on class balance.

Both approaches have in common that they try to eliminate information from the synthetic source domain that does not correspond to the given real-world target domain. However, synthetic data should ideally represent unknown information about the real-world target domain. The approaches by Sun et al. [219] and Kim et al. [105] would eliminate such information from the synthetic images since it would not match the subset of data sampled from the real world that makes up the target domain distribution. The following section will introduce an approach to scoring images based on their similarity to the target domain and the criticality of the data.

4.2.2 Synthetic Dataset Acquisition for a Specific Target Domain

This section develops an acquisition function to intelligently select important images from the buffer (see Figure 4.1 page 93) filled constantly by the simulation engine with synthetic images of a large variety. With a specified budget B of synthetic images to be acquired, the function aims to improve the performance of the semantic segmentation

network on the target domain in a scenario where methods of unsupervised domain adaptation (UDA)–see Section 3–are utilized during training. All acquisition functions share a similar generic structure. Each sample from the currently buffered synthetic data receives a score. The score should reflect the value of the corresponding synthetic image. The value is high if training on the image greatly improves the segmentation performance on the real-world target domain. In the end, the B images with the largest score are selected from the synthetic world, irrespective of the approach used. After these B images are added to the set of synthetic data, the DNN is trained again, and UDA is utilized to adapt to the real-world target domain. Due to the retraining, the uncertainties of the network are updated, and the knowledge that is represented by the B images is incorporated. As Figure 4.3 shows, the next B images are selected by their score. Since the uncertainties of the network are updated, the next B images that are sampled will represent novel information.

This section proposes two types of acquisition functions, which can be distinguished by the computation of the scores.

- ***Correspondence-based*** *acquisition functions* leverage synthetic and real-world data to select synthetic images representing meaningful correspondences to the real-world target domain. These acquisition functions select synthetic samples that align well with important data clusters in the real-world domain.

- ***Simulation-only*** *acquisition functions* focus solely on information and statistics from the synthetic world.

Section 4.2.2.1 explains how the different versions of simulation-only acquisition functions this chapter considers score the images, and Section 4.2.2.2 introduces the correspondence-based acquisition functions.

4.2.2.1 Simulation-only Acquisition Functions

Class uniform sampling: This acquisition function aims to achieve a balanced synthetic set regarding the class statistics in the segmentation labels. A histogram is built by incrementing a certain class if it is present in the chosen image's segmentation label to track how often that class appears in the already chosen data. The class uniform sampling is initialized with a random selection of an image. After updating the histogram, the class with the lowest value is computed, and an image with a label map containing that class is chosen. The 'score' of an image is maximized if the rarest class in the histogram is present in the synthetic image. If many images fulfill that requirement, a random selection is applied among them.

Entropy: Entropy [205] acquisition is based on ideas presented in active learning literature of Chapter 2. The semantic segmentation is computed for a given simulated image together with the corresponding pixel-wise entropy over the a-posteriori class

distributions (c.f. Section 2.4.4). The pixel-wise entropy represents the 'score' value of a synthetic image and is an estimate of the epistemic uncertainty. The images with the largest averaged entropy are chosen until B is full.

CoreSet: The Coreset [203] approach selects a batch of samples that cover the whole data distribution of the simulated world. It formulates this batch selection as a robust B-center selection problem and tackles the redundancy in the simulated world (c.f. Section 2.4.4). The 'score' of an image represents how well the distribution of all synthetic images is represented by the acquired batch when this image was added.

4.2.2.2 Correspondence-based Acquisition Functions

The *clustering assumption* of semi-supervised learning (SSL) yields that if two points belong to the same cluster, their outputs are likely to be close and can be connected by a short curve (c.f. Section 2.3). In other words, if the synthetic samples are aligned with clusters of the unlabeled real-world samples, it follows from the cluster assumption of semi-supervised learning that a valid learning signal can be created for these real-world clusters. In such a case, UDA methods, should yield good performance since they are a variant of semi-supervised learning (the target domain is the unlabeled data). Therefore, newly selected synthetic samples must cover the unlabeled real-world clusters not yet covered by synthetic samples to maximize semi-supervised learning performance.

Figure 4.3 shows how the approach utilizes the cluster assumption. The figure represents the general feedback loop for selection that is further described in Algorithm 1. The algorithm starts with a small initial synthetic set and trains a DNN for segmentation while using UDA to the real-world target domain. The UDA method aligns the distributions of the source and target domain in the feature space of the DNN.

Synthetic image selection based on real-world images: The aim is to select B synthetic images $\mathcal{X}_S^* = \{ x_1^{S*}, \ldots, x_B^{S*}\}$ from the N synthetic source domain images $\mathcal{X}_S = \{x_1^S, \ldots, x_N^S\}$ based on their correspondence to a subset $\mathcal{X}_R^*$ of the set of M real-world target domain images $\mathcal{X}_R = \{x_1^R, \ldots, x_M^R\}$. This aim follows the *clustering assumption*, i.e., that the correspondences between the labeled source domain and unlabeled target domain facilitate the unsupervised learning on the target domain. Therefore, three methods for selecting the subset $\mathcal{X}_R^*$ of real-world images as targets for correspondence with the synthetic world are analyzed first. In Algorithm 1 this is the 'sort_by_priority' function. This function sorts the real-world images by their priority. The priority defines for which of the real-world images the algorithm selects a matching synthetic image first. To compute the value of 'sort_by_priority', there are different options:

- **Corner Case (CC):** The real-world images are scored and sorted by the entropy over the prediction of the trained network (see Chapter 2). This scoring aims to find the synthetic correspondence for these real-world corner case images.

Algorithm 1: Greedy Synthetic Image Selection via Correspondence

Data: synthetic image set $\mathcal{X}_S = \{x_n^S\}_{n=1}^N$,
real image set $\mathcal{X}_R = \{x_m^R\}_{m=1}^M$,
budget B
Result: selected synthetic set $\mathcal{X}_S^* = \{x_i^{S*}\}_{i=1}^B \subseteq \mathcal{X}_S$
$\mathcal{X}_R^* \leftarrow$ sort_by_priority($\mathcal{X}_R$, B); `// top-`B `or all if Global`
$\mathbf{C}_{n,m}^{\text{corr}} \leftarrow$ correspondence(x_n^S, x_m^{R*}); `// for all` $n = 1 \dots N$, $m = 1 \dots |\mathcal{X}_R^*|$
$\mathcal{X}_S^* \leftarrow \emptyset$;
foreach $x_m^{R*} \in \mathcal{X}_R^*$ **do**
  $n^* \leftarrow$ match_synthetic(x_m^{R*}; $\mathbf{C}^{\text{corr}}$); `// match via Eq. 4.2`
  if $x_{n^*}^S \notin \mathcal{X}_S^*$ **then**
    $\mathcal{X}_S^* \leftarrow \mathcal{X}_S^* \cup \{x_{n^*}^S\}$;
    $\mathbf{C}_{n^*,:}^{\text{corr}} \leftarrow -\infty$; `//` ∞ `if distance; prevents re-selection`
  end
  if $|\mathcal{X}_S^*| = B$ **then**
    break; `// budget reached`
  end
end

- **Arbitrary (Arb):** The real-world images are scored and sorted by a random score. This arbitrary scoring aims at finding the synthetic correspondence for a random subset of real-world images.

- **Global (Gl):** No priority is assigned. This strategy aims to choose the synthetic images that correspond the most closely with any real-world image.

Given the budget B the algorithm computes a set of selected real-world images $\mathcal{X}_R^* = \{x_1^{R*}, \dots, x_B^{R*}\}$ where $\mathcal{X}_R^* \subseteq \{x_1^R, \dots, x_M^R\}$ that are the B real-world images with the highest priority. This, at least, is the case for **CC** and **Arb**. However, the last case **Gl** does not perform a selection, and all real-world images are considered. Next, the algorithm computes a similarity or distance matrix that describes the correspondence of each of the selected real-world images with each synthetic image:

$$\mathbf{C}_{n,m}^{\text{corr}} = \text{correspondence}(x_n^S, x_m^{R*}), \quad x_m^{R*} \in \mathcal{X}_R^*,\ x_n^S \in \mathcal{X}_S. \tag{4.1}$$

A later part of this section shows the computation of such a correspondence matrix. The correspondence can be a distance or a similarity. If it is a distance, the set of synthetic images that minimize the sum of all correspondence values with the real-world

target images is computed. Vice-versa, if the correspondence represents similarity, the sum of the correspondence values is maximized.

$$n^* = \text{match\_synthetic}(x_m^R;\, \mathbf{C}^{\text{corr}}) = \begin{cases} \underset{n=1\ldots N}{\arg\min}\ \mathbf{C}^{\text{corr}}_{n,m}, & \text{if } \mathbf{C}^{\text{corr}}_{n,m} \text{ is a distance,} \\ \underset{n=1\ldots N}{\arg\max}\ \mathbf{C}^{\text{corr}}_{n,m}, & \text{if } \mathbf{C}^{\text{corr}}_{n,m} \text{ is a similarity.} \end{cases} \tag{4.2}$$

The 'match_synthetic' function in Algorithm 1 computes the index n^* of the image from the set of non selected synthetic images, with the minimum distance or maximum similarity (see Equation 4.2) to the current $x_m^{R*} \in \mathcal{X}_R^*$. After a matching synthetic image $x_{n^*}^S$ has been computed for each $x_m^{R*} \in \mathcal{X}_R^*$ the B synthetic images $x_1^{S*}, \ldots, x_B^{S*}$ have been found. This way of selecting the B synthetic images $x_1^{S*}, \ldots, x_B^{S*}$ represents a greedy approximation to minimizing or maximizing the sum of correspondences. The global solution would, however, involve matching $\binom{N}{B}$ sets of synthetic data with the B prioritized real images, a computationally infeasible task.

Finding a good measure for correspondence: to compute the correspondence matrix, either a measure of similarity or dissimilarity is needed (see Equation (4.2)) that captures the relevant features of the real and synthetic images for semi-supervised learning. For this, the current trained model is taken to compute the feature space representation of the real and synthetic images. A synthetic image "corresponds" to a real-world image if the embedding is similar or if the feature space distance is small.

The cosine similarity is used as a measure of similarity. A global average pooling over the spatial dimensions reduces the feature space to one vector. The cosine similarity is computed between the vector representation of the images. In case the contents of the correspondence matrix are distances, the Hausdorff [47] distance or the Euclidean distance between the Gram matrices [59] is used as a measure of the distance between the feature space representations of real-world or synthetic images. The aim of using the Gram matrix is to represent the style and texture of an image. The distance of two images then focuses on global image properties. An essential aspect for both is the encoder that is being used. In this work, the feature space of the trained segmentation network is utilized in most cases. Some experiments, also employ a VGG-19 [210] ImageNet [38] encoder.

Deciding on final methods: Finally, this work develops and applies different methods from the building blocks that were described:

- "CC min/max sim": Indicates the minimization or maximization of the cosine similarity with real-world corner case images
- "max sim I-NET": Uses the the VGG-19 ImageNet encoder to compute feature space representation/similarities.
- "Arb max sim": Instead of maximizing the similarity with corner case images, a random corner case score is assigned to each real-world image.

- "Gl max sim": The synthetic images that represent the pairs in the correspondence matrix with the largest similarity
- "Gram Matrix"/"Hausdorf": The distance of Gram matrices or the symmetric Hausdorf distance is used
- "uni": This prefix implies that the class uniform sampling is introduced as a requirement

As described above, some variations include adding class uniformity as an additional optimization criterion ("uni"). In the end, the selection of synthetic images is added to the synthetic training set, and the model is retrained, starting the cycle again.

4.2.3 Experiments

This section briefly describes the experimental setup and compares the proposed methods introduced in Section 4.2.2.

4.2.3.1 Experimental Setup

During the experiments, the GTA5 dataset represents the synthetic world. It consists of 24966 synthetic frames (source domain). In each acquisition step, a budget of $B = 250$ frames is sampled by Algorithm 1 in each iteration of the process displayed in Figure 4.3 on page 96 . This budget $B = 250$ represents roughly 1% of the data in the GTA5 dataset. The sampling in the experiments is done five times, which accumulates up to 5% of the whole GTA5 dataset. This setup would correspond to a system that constantly creates synthetic data with many variations and saves the resulting images into a buffer. The GTA5 dataset represents the current state of the buffer of synthetic images (c.f. Figure 4.1 page 93). The acquisition functions select the images from this buffer and add the resulting frames to the final set. The current buffer would be extended or replaced between the selection cycles in a practical application. However, given its large size, the GTA5 dataset approximates such a setup well.

The experiments employ the state-of-the-art UDA approach "DA-Former" [77] to adapt to the real world. The model is trained for 20,000 iterations with a batch size of two images. The real-world target domain is represented by the Cityscapes [36] and ACDC [197] dataset. The Cityscapes dataset represents a very "clean" distribution with limited variations w.r.t. illumination, weather, and season conditions. The ACDC dataset represents a distribution with large variability w.r.t. weather (e.g., fog, rain, snow) and illumination (day, night) conditions. Experiments on multiple target domains are important since different acquisition functions might work in different scenarios.

As introduced in Section 4.2.2, this work employs different acquisition functions for the synthetic world. Entropy, Coreset, and Class uniform acquisitions only depend on

Table 4.1: Domain change from GTA5 to Cityscapes. Maximum mean Intersection over Union (mIoU) in % and the area under the curve (AUC) after sampling from 1%–5% of the data. Simulation-only (S) and Correspondence-based (S+R) acquisition functions are compared. Best results **bold**, second-best in *emphasis*.

	Acq. Function	max	AUC
-	Random	64.53	2.36
S	Entropy on GTA5	62.01	2.38
S	Coreset on GTA5	62.93	2.39
S+R	Hausdorff	62.26	2.37
S+R	Gram Matrix	61.94	2.33
S+R	CC min sim	65.78	2.55
S+R	CC max sim	63.57	2.47
S	uni sampling	**66.17**	**2.56**
S+R	uni CC max sim	*65.87*	*2.55*
S+R	uni CC max sim I-NET	65.47	2.54
S+R	uni CC min sim	65.55	2.53
S+R	uni Gl max sim	65.55	*2.55*
S+R	uni Arb max sim	65.35	2.54
–	100% data	67.80	–

Table 4.2: GTA5 to ACDC domain change. Maximum mean Intersection over Union (mIoU) in % and the area under the curve (AUC) after sampling from 1%–5% of the data. Simulation-only (S) and Correspondence-based (S+R) acquisition functions are compared. Best results **bold**, second-best in *emphasis*.

	Acq. Function	max	AUC
-	Random	46.91	1.82
S+R	CC min sim	46.68	1.80
S+R	CC max sim	47.83	1.84
S	uni sampling	47.50	1.88
S+R	uni CC max sim	**49.30**	**1.98**
S+R	uni Gl max sim	*48.89*	*1.91*
–	100% data	46.37	–

the synthetic world, while selecting synthetic data based on real-world data takes the target domain into account. The variations presented in Section 4.2.2 are evaluated for the latter. Each variation function is evaluated by the maximum Mean Intersection over Union (mIoU) on the Cityscapes and ACDC dataset and the area under the mIoU curve (AUC see chapter 2) that results from the acquisition steps.

4.2.3.2 Quantitative Evaluation

The quantitative evaluation assesses the performance of different acquisition functions for the synthetic world as introduced in Section 4.2.2. These acquisition functions are analyzed on the two domain changes (GTA5 to Cityscapes and GTA5 to ACDC).

GTA5 to Cityscapes Shift: Table 4.1 presents these values to evaluate the segmentation trained on the GTA5 dataset and adapted to the Cityscapes dataset.

Most of the performance can be achieved with a small amount of synthetic data. A max $mIoU = 66.17\%$ is achieved with 5% of synthetic data only, which is 95.6% of the performance achieved with 100% of the data (max $mIoU = 67.8\%$). This indicates that the synthetic world contains a lot of redundancy. Except for Entropy, Coreset, and Gram Matrix, all acquisition functions performed better than random sampling. Class uniform sampling proved crucial in the GTA5 to Cityscapes domain change, as having samples from all classes was particularly important for unsupervised domain adaptation (UDA). Notably, the class uniform sampling does not utilize information from the real-world target domain but still achieves the best results. However, the Entropy and Coreset acquisition functions that utilize uncertainty or distribution features from the synthetic domain perform worse than functions that utilize real-world information. Regarding the latter class of acquisition functions, the Hausdorff and Gram matrix distance performed worse than utilizing the cosine similarity as a measure of correspondence. Interestingly, similarity minimization works better than maximization on GTA5 to Cityscapes domain change if corner case images are chosen for computing correspondences and class uniformity is not enforced. Introducing the class uniformity requirement does help all acquisition strategies to be on par with the class uniform sampling, which is the best strategy for this domain change. Additionally, to class uniformity, introducing a priority for real-world corner case images during the computation of correspondences does not improve the results. Finally, utilizing the VGG-19 ImageNet encoder does not result in a more meaningful correspondence score as the results do not improve (e.g., 'uni CC max sim I-NET' compared to uni 'uni CC max sim').

GTA5 to ACDC Shift: Table 4.2 presents the GTA5 to ACDC use case results. For this domain shift, using 100% of the data seems to decrease performance compared to all acquisition functions, including the random sampling, even though only 1%-5% of the data is sampled. That even indicates that some images may introduce a bias into the model that makes unsupervised learning on the real world more difficult. On the GTA5 to ACDC domain change, uniform sampling did not perform best but still offered an improvement in combination with similarity maximization. Finally, one can observe that, other than in the GTA5 to Cityscapes shift, assigning priority for real-world corner case images during correspondence computation ('uni CC max sim') is introducing a benefit in the GTA5 to ACDC domain change (better than 'uni CC max sim').

Comparison: Very little data achieves similar or better performance on the respective real-world target domain for both domain shifts. These results highlight the need for more effective acquisition strategies, saving memory and training time, and avoiding a bias in the data that introduces a worse performance despite having more data. The latter problem, which is present in the GTA5 to ACDC shift, is probably due to the fact that the dataset distributions do not overlap as well as the GTA5 and Cityscapes datasets (see Figure 4.5 page 105). The assumption is that either 'wrong'

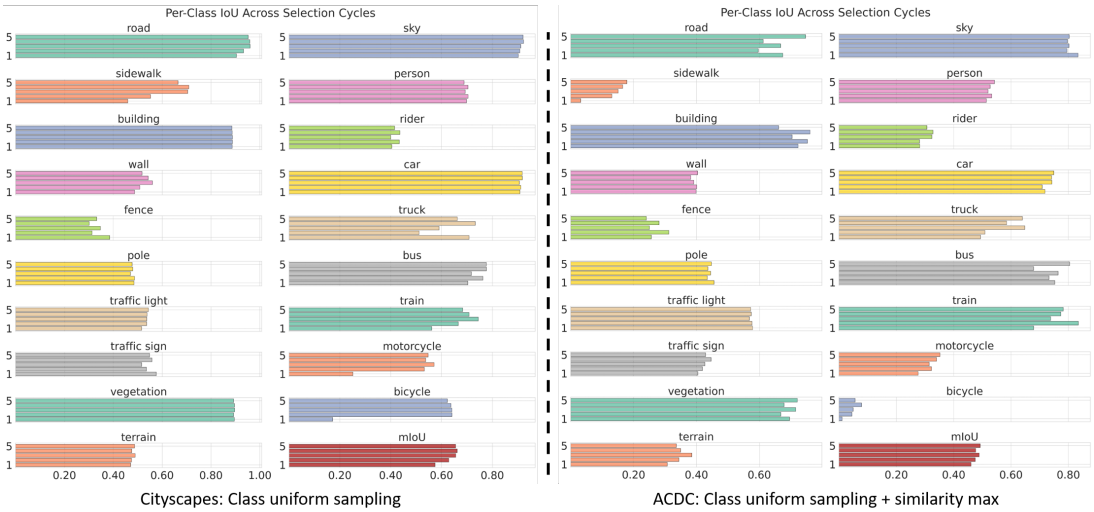
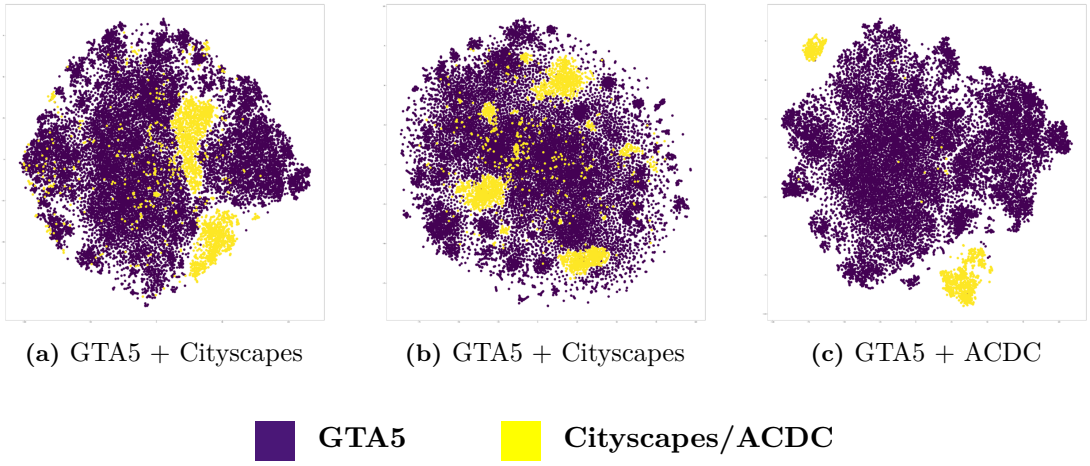


Figure 4.4: Left: class-wise IOU over the acquisition steps. The bottom bar is the first iteration, and the top bar is the last. The Cityscapes dataset is the target domain. Class uniform sampling was used. Right: class-wise IOU over the acquisition steps. The bottom bar is the first iteration, and the top bar is the last. ACDC is the target domain: class uniform sampling + similarity max.

information is introduced (image information represents a different label) or that the imbalance of little corresponding relevant and lots of non-corresponding irrelevant data causes the model to assign less weight to the former. A further interpretation is that this is part of the reason why similarity maximization combined with class uniformity has the advantage in this domain shift. Given that it picks only corresponding information between the domains, the non-corresponding irrelevant data is being eliminated.

4.2.3.3 Qualitative Evaluation

Figure 4.5 displays the t-SNE representations of the synthetic and the real-world images in the feature space of VGG-19 encoders and the trained segmentation net. The t-SNE [148] algorithm computes a dimensionality reduction of the high dimensional feature space representation into two dimensions, allowing for a qualitative comparison of the image distributions. The t-SNE plots are generated for both the GTA5 to Cityscapes domain change and the GTA5 to ACDC domain change. One can observe that the number of correspondences, i.e., yellow and purple data points close to each other, is sparse in all domain changes and for all encoders. This effect is especially strong in the case of the GTA5 to ACDC domain change. This can be interpreted in different ways. For one, it could be that the encoders do not capture the features



(a) GTA5 + Cityscapes **(b)** GTA5 + Cityscapes **(c)** GTA5 + ACDC

Figure 4.5: T-SNE plots of feature representations: (a) GTA5 to Cityscapes domain change based on a segmentation network and a (b) VGG-19 Image-Net encoder. (c) GTA5 to ACDC domain change t-SNE plots are based on feature representations from a segmentation network. Feature representations of GTA5 are indicated in purple, while those of Cityscapes and ACDC are in yellow.

important for similarity well. On the other hand, it could be a hint that the GTA5 dataset is not ideal for capturing the Cityscapes or ACDC domain. There appears to be more overlap between the Cityscapes and the GTA-5 than between the GTA5 and the ACDC datasets, which could further explain the lower absolute mIoU values obtained in the latter domain change. Given that most of the samples do not overlap between the two domains, this could also be a reason why sampling 100% of the GTA5 data introduces a bias and works worse than sampling 1-5%.

When comparing the ImageNet VGG-19 encoder with the segmentation network encoder based on their feature representations, the overlap seems to be slightly larger when using the VGG19 encoder. Generally speaking, using the feature representations of the segmentation network encoder makes sense since the displayed distribution is the same that the classification is performed on. Hence, similarity in this feature space indicates the similarity of two images with respect to the segmentation task.

4.2.3.4 Limitations and Discussion

Future work should build on the analysis of the different performances of the acquisition functions introduced in this chapter. An interesting factor is the kind of domain change, i.e., the real-world target domain representing the target domain. Relevant factors are different environment conditions or sensor setups and how the content/ correspondences in the synthetic world influence the performance of the acquisition functions. The similarity function is a likely building block of the system that can be

improved. The quality of the similarity function is largely dependent on the encoder that is used to compute the latent space. The encoder function, therefore, determines the meaning of the distance or the cosine similarity, identifying the most relevant correspondences. In this context, another interesting future analysis is to determine why specific synthetic images negatively impact the training and the performance of the trained model on the target domain. Figure 4.4 on page 104 shows that the class IoU values alternate with the increase in iterations, which indicates that not all of the added information has a positive impact. Following the proposed analysis points, a direction for improving the acquisition functions can be identified.

4.2.4 Conclusion

This work addresses and introduces the research question of how to construct a synthetic dataset that facilitates unsupervised adaptation to a specific target domain. The experiments demonstrated that by sampling a small portion of the synthetic data (1-5%), a performance quite close to or better than achieved with the entire dataset can be obtained. Moreover, the work discovered that training on the complete synthetic dataset can introduce a bias and degrade performance when the synthetic domain poorly represents the real-world domain. Through an extensive study of different acquisition functions, the work identified their benefits in enhancing adaptation. The findings presented in this work can be leveraged to create a synthetic dataset tailored to specific real-world target domains.

Given the results, one can propose utilizing the CARLA simulation engine to generate a stream of data and iteratively select the most relevant samples, as demonstrated in this work. The insights gained from this research contribute to advancing the field of synthetic data utilization and have implications for improving perception systems in real-world applications.

Future work should focus on optimizing the building blocks of the presented acquisition functions in real-world scenarios. Among these building blocks, improving the similarity function is especially interesting. A possible point for improvement would be the latent space in which the similarity is computed. For example, an analysis of different encoder architectures and training strategies would be warranted. The proposed acquisition function should also be validated in a wide variety of real-world applications spanning different camera systems and environments. This validation could be a foundation for the next step towards piratical use.

4.3 Utilization of Data-Driven Generative Models

Even though simulation engines like CARLA offer a cost-effective and easy way to create annotated data, they come with challenges. One of the most prominent issues is the domain gap between the generated synthetic data and the real-world data to

Figure 4.6: Example of utilizing diffusion models to improve realism and quality of a synthetic image. Φ_n denotes the prompt that, together with the GTA-5 image, serves as input for the diffusion model. A big problem are the semantic inconsistencies created by the generative models (in this case, ControlNet [287]).

which the trained DNNs should be applied. Additionally, the synthetic data from these simulation engines often are limited w.r.t. diversity. That is because the quality and variety of the generated data largely depend on the human modeling of the elements in the synthetic world. This human effort can be interpreted as a kind of annotation in itself and, therefore, contradicts the original purpose of utilizing simulation engines for data generation. Furthermore, the human effort to model synthetic elements complicates the adaptation of simulation engines to new application domains. This challenge is especially evident in complex, specialized domains like medical image processing. In medical image processing, many sensor domains and medical applications would require specialization or adaptation of the given simulation environment. Therefore, data-driven generative models based on DNNs offer a promising alternative. Given that such models are data-driven, no human modeling is necessary. They can then, e.g., be used to transform synthetic images from simulation engines to more realistic versions to reduce the gap between the synthetic and the real-world domain. Modern diffusion models like Stable Diffusion [190] or ControlNet [287] even offer the ability to alter the image content and style by control through textual descriptions, i.e., text prompts. As Figure 4.6 shows, this allows for introducing higher degrees of realism and diversity without human modeling of synthetic elements.

This work, therefore, investigates two applications of data-driven generative models:

1. It utilizes diffusion models to create large training data distributions to train models that generalize well to many unseen data domains. This approach described in the

next Chapter 5 hence augments the capabilities of simulation engines like CARLA by transforming the synthetic data to more realistic and diverse versions.

2. But first, the remaining part of this chapter describes how to utilize data-driven models on their own for the application of medical image processing, where pre-existing simulation engines are rare. The presented approach utilized the differentiability of DNN-based generative models to produce high epistemic uncertainty images. The epistemic uncertainty of an image is high if it is not part of the training distribution of a given discriminative model, which makes the generation of such images especially interesting. The following section shows that a model trained on such data yields better generalization capabilities and improved robustness.

4.4 Targeted Synthetic Data Generation

Creating imaging datasets for training deep neural networks consists of three major steps: data acquisition, data selection, and data labeling. These steps are especially challenging in the domain of medical image processing. Data acquisition is often limited and data delivery is impaired by privacy regulations. Also, relevant image data might further be bound by the frequency of certain medical scenarios (e.g. rare diseases). Another main obstacle is the costly and time-intensive data labeling, which often requires medical professionals.

This work addresses these problems by utilizing generative models to extend the distribution of the given training data. More specifically, it aims to create data points representing missing parts of the relevant distribution. Such data points are marked by a high epistemic uncertainty (see chapter 2) when processed by a discriminative model (i.e. a classifier network). This section presents a novel approach called TSynD (***T****argeted* ***Syn****thetic* ***D****ata generation*): a method specifically designed to steer the generation process in order to synthesize data points from the missing parts of the training distribution and utilize them during the training of downstream tasks (here: classifier). For data generation, TSynD employs an autoencoder model that is able to reconstruct existing images of the training distribution. The autoencoder consists of an encoder that transforms the image into the latent space and a decoder that reconstructs the input image from the latent space. TSynD aims to optimize the latent space representations of the autoencoder in a way that the decoded images maximize the epistemic uncertainty in a given classifier. By further training the classifier on these images, classification models that generalize well to unseen data are obtained. This feature is especially important in medical image processing. We, therefore, show the performance of the TSynD method on several medical classification datasets. In order to simulate the smaller training datasets, typical for the medical image community, as well as recreate cases of out-of-distribution samples, this work primarily considers a low-data training setting. This work provides experiments to investigate the out-of-

distribution performance through random test time augmentations and investigate the robustness to adversarial attacks. Further, the robustness of the presented approach is investigated visually by applying class activation explanation approaches. This section additionally shows that a classifier trained with TSynD utilizes more meaningful image information.

4.4.1 Generative Models and Augmentation for Generalization

This work presents a novel method for training networks that generalize to out-of-distribution samples. An adaptive data generation process based on generative models is employed in this context.

Data augmentation is a commonly used way of extending the given training distribution. However, the distribution extension is mostly untargeted for these methods. As stated by Zhou et al. [299], there are four different types of data augmentation: firstly, there are image transformations, e.g., random flipping, rotation, or color augmentations. Secondly, model-based augmentations, which, e.g., consist of random convolutions [260] or other augmentation networks like style transfer networks [10] or learnable image generators [171, 292]. Thirdly, latent space augmentations directly augment the latent space distributions of the tasks (e.g., classification) model as in Zhou et al. [293]. Finally, some approaches utilize adversarial gradients.

Adversarial gradient augmentation and, more specifically, task adversarial augmentations are the most similar categories to the approach presented in this work. The approaches of Sinha et al. [211], Volpi et al. [236], and Qiao et al. [183] utilize adversarial attacks by computing adversarial gradients w.r.t. to the task network in order to alter the training images. In this context, the alternation is done by optimizing the pixel values of the image as parameters themselves. Such methods are often accused of introducing noise perturbations instead of larger image alternations, e.g., representing domain shifts (Zhou et al. [299]).

In contrast, the approach presented in this work optimizes the latent space of a generative model as parameters of the image generation. The intuition behind this is that latent space is a more abstract image representation; thus, altering it would lead to more complex and meaningful image changes. The work of Stutz et al. [217] is, therefore, the most related to the approach presented in this work approach. They employ a VAE-GAN model [192] to represent the manifold and, similar to this work, compute perturbations on the latent space to create adversarial examples. In contrast to us, they do not maximize the uncertainty but rather maximize the cross-entropy loss. The optimization effectively changes the predicted label and thus introduces the need for constraints to maintain the true image class. The approach presented in this work is inspired by active learning (see chapter 2) and puts the main focus on generating images that maximize the epistemic uncertainty of the given classifier. This brings the advantage of not requiring any additional constraints. The work of Li et al. [130]

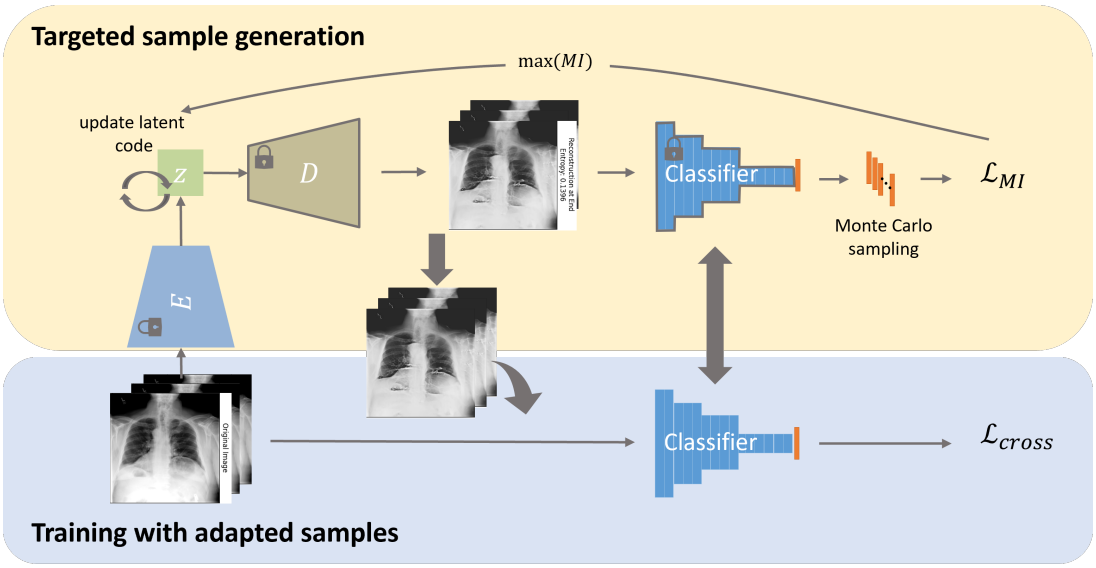


Figure 4.7: The overall framework of TSynD (***T**argeted **Syn**thetic **D**ata generation*) for the robust training of a classifier: The autoencoder is pre-trained unsupervised; then, its weights are frozen. During classifier training, the latent spatial representations of original images are optimized to maximize the classifier's epistemic uncertainty in the decoded images. The new images then serve as additional training data.

also utilizes an autoencoder. Similar to this work, they compute perturbations on the latent space by, e.g., using random noise. This work utilizes the randomly perturbed latent space as a starting point for the optimization to increase data diversity.

4.4.2 Guiding the Generation Process

Given a labeled data set $\mathcal{D} = \{(x_n, t_n)\}_{n=1}^{N_{data}}$, where $x_n \in \mathcal{X}$ and $t_n \in \{1, \ldots, K\}$ (K is the number of classes), the approach aims to use the generative model in a targeted way to make a classification network $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ more robust to missing parts of the data distribution that are not included in the labeled set $\mathcal{D}$. The generative model, e.g. an autoencoder, consists of an encoding function $f_{\text{enc}} : \mathcal{X} \to \mathcal{Z}$ and a decoder $f_{\text{dec}} : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{Z}$ is the latent space. It can be trained in an unsupervised way using a larger amount of unlabeled data from the domain $\mathcal{X}$. Inspired by active learning strategies, the approach utilizes the generative model to create images that maximize the epistemic uncertainty of the classification network $\mathcal{M}$. Samples yielding a high epistemic uncertainty represent missing parts of the learned distribution, and training on such samples can make the classification network more robust. Figure 4.7 shows an overview of the approach: Starting with the encoded labeled images, the latent code z is optimized to reconstruct new images that locally maximize the

epistemic uncertainty of the classifier $\mathcal{M}$. The newly generated samples are now used together with the labeled images for the classifier training.

4.4.2.1 Estimation of the Epistemic Uncertainty

Given the classifier $\mathcal{M}$ with model parameters θ, the predictive class probability distribution for a decoded image $\hat{x} = f_{\text{dec}}(z)$ with latent code z is computed by

$$P(k \mid \hat{x}, \theta) = P(k \mid z, \theta) = \sigma(\mathcal{M}(f_{\text{dec}}(z); \theta)), \tag{4.3}$$

where the function $\sigma(\cdot)$ transfers the classifier logit outputs into probabilities of the classes $k \in \mathcal{K}$. Here, $\sigma(\cdot)$ is the softmax function in the case of a multilabel classification or a sigmoid function in the case of a binary classification. The primary objective is to guide the reconstruction process $\hat{x} = f_{\text{dec}}(z)$ in a manner that the resulting sample $\hat{x}$ contributes meaningfully to the training of the classifier $\mathcal{M}$. This guidance involves modifying a latent variable $z \in \mathcal{Z}$ of the autoencoder with the aim of generating samples with high epistemic uncertainty in the classifier $\mathcal{M}$.

The uncertainty of the predictive distribution is defined by the entropy

$$\mathbf{U}_H(z) = H(P(k \mid z, \theta)) = -\sum_{k=1}^{K} P(k|z, \theta) \log(P(k|z, \theta)), \tag{4.4}$$

however, the epistemic uncertainty associated with a data sample $\hat{x} = f_{\text{dec}}(z)$ stems from uncertainty in model parameters. This can be quantified by the expected change in entropy of the model parameter posterior distribution, expressed by the conditional mutual information [135]:

$$\mathbf{U}_{MI}(z) = MI(z; \theta^{(r)}) = H(\mathbb{E}_\theta(P(k \mid z, \theta^{(r)}))) - \mathbb{E}_\theta(H(P(k \mid z, \theta^{(r)}))), \tag{4.5}$$

where the expectation is computed over Monte Carlo Dropouts [107]. Mutual information is considered to be a better measure of epistemic uncertainty [107]. To keep the additional computational effort low, the approach only iterates over the last layers of $\mathcal{M}$ with R_{MC} dropout masks to compute samples of $P(k|z, \theta^{(r)})$, with $\{\theta^{(r)}\}_{r=1}^{R_{MC}}$ and $R_{MC} = 10$.

4.4.2.2 Targeted Synthetic Data Generation

An optimization-based approach is used to find latent codes z that locally maximize the given measure for uncertainty $\mathbf{U}(z)$. As shown in Fig. 4.7, starting with the latent code $z_n = f_{\text{enc}}(x_n)$ of a random image of the training distribution, a local maximum is searched as

$$z^* = \arg\max_z \ \mathbf{U}(z). \tag{4.6}$$

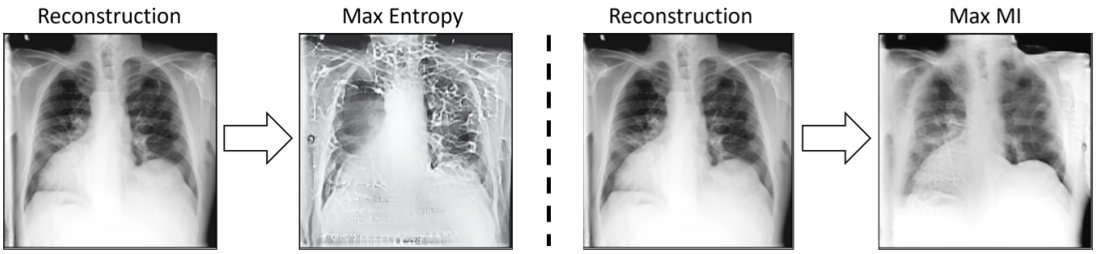


Figure 4.8: Left: the result of guiding the generative model by maximizing the entropy. Right: the result of guiding the generative model by maximizing the Mutual Information Additionally, the figure provides the non-optimized reconstruction of the image.

Since $\mathcal{M}$ and f_{dec} both are differentiable, $\mathbf{U}(z)$ can be maximized by standard back-propagation. The resulting sample $\hat{x} = f_{\text{dec}}(z^*)$ is added to the training set, assuming that it belongs to the same class as x_n, but lies in missing parts of the learned data distribution, as indicated by the high uncertainty. Figure 4.8 shows the effects that choosing the MI or the entropy as a measure for the uncertainty U has on the generated data.

Latent space noise. Apart from the uncertainty that a sample yields, the diversity w.r.t. the training distribution is crucial. To introduce further varieties into the reconstructed image, samples are generated by adding uniform noise to latent codes $z_n = f_{\text{enc}}(x_n)$:

$$\hat{x} = f_{\text{dec}}(z_n + \epsilon), \quad \epsilon \sim N(0, \sigma\mathbf{I}). \tag{4.7}$$

The resulting image $\hat{x}$ is an alternative representation of x_n and, therefore, increases the diversity in the dataset. Latent space noise can be used as a stand-alone augmentation or as an initial augmentation before optimization according to Eq. 4.6 to generate even more diverse samples.

4.4.2.3 The Training Process.

In each training iteration, the batch (batch size $\mathcal{B}$) is divided into two halves: The first half consists of original image-label pairs $\{(x_n, t_n)\}_{n=1}^{\frac{\mathcal{B}}{2}}$, and the second half consists of the optimized reconstructions $\hat{x}_n = f_{\text{dec}}(z_n^*)$ resulting from Eq. (4.6), along with their corresponding labels $\{(\hat{x}_n, t_n)\}_{n=1}^{\frac{\mathcal{B}}{2}}$. Since the maximization of the epistemic uncertainty depends on the current state of the classifier $\mathcal{M}$, the generation process of $\hat{x}_n$ needs to be redone after each training iteration. This also prevents the so-called mode collapse problem (see chapter 2) that would occur if the image generation was performed only once. The resulting images would be similar since similar images are likely to maximize the uncertainty of the given classifier. However, since retraining and generation are done in an alternating way, the network is updated, and the generation process yields new alternations.

Table 4.3: Accuracy results of different MedMNIST datasets with a subsampling of the training dataset to 1% and 10%. The results are reported for the respective test set of the datasets and two augmented versions of the test sets (Gaussian Noise and adversarial attacks).

	BreastMNIST		**DermaMNIST**		**OCTMNIST**		**OrganaMNIST**		**OrgansMNIST**		**PathMNIST**	
	1%	**10%**	**1%**	**10%**	**1%**	**10%**	**1%**	**10%**	**1%**	**10%**	**1%**	**10%**
Baseline	70.7	75.2	66.8	65.2	59.9	67.5	71.9	89.3	49.6	67.8	67.4	77.2
Noise	72.0	**78.2**	66.7	65.8	61.0	**71.7**	73.8	87.8	52.9	68.6	64.7	**83.5**
TSynD	**73.3**	77.8	**66.9**	**66.7**	**61.4**	66.7	**77.2**	**89.4**	**54.2**	**71.4**	**73.1**	78.5
Gaussian Noise Augmentation during Test												
Baseline	62.6	**73.1**	66.8	65.1	24.9	29.9	44.5	78.9	37.1	52.6	12.6	10.6
Noise	**73.5**	**73.1**	66.7	65.7	24.5	34.9	44.1	65.3	37.3	52.4	13.5	11.5
TSynD	73.3	**73.1**	**66.9**	**66.7**	**28.7**	**36.4**	**63.5**	**85.1**	**45.6**	**66.4**	**28.2**	**12.8**
Adversarial Attacks during Test												
Baseline	65.6	7.1	66.4	48.8	5.3	3.3	34.4	68.4	13.8	25.6	28.5	21.1
Noise	68.6	21.6	**66.7**	53.0	8.1	4.1	39.2	71.6	25.1	25.6	31.7	26.1
TSynD	**71.4**	**28.2**	**66.7**	**64.1**	**12.8**	**42.8**	**53.5**	**83.9**	**27.1**	**51.3**	**43.5**	**47.8**

Optimizing latent codes vs. pixel values as parameters. Optimizing the pixel values like in [183, 211, 236] likely results in salt and pepper noise. Altering abstract representations introduces more substantial alternations since each element of $z \in \mathcal{Z}$ represents larger receptive fields in the image. Additionally, the autoencoder is learned on the distribution of relevant images. The reconstruction process is, therefore, already constrained w.r.t. this distribution. Constraints that need to be introduced when optimizing the image pixels directly, like in the approaches of [183, 211, 236], are not needed. However, it is important to maximize epistemic uncertainty (model uncertainty) rather than aleatoric uncertainty (data uncertainty). Maximizing aleatoric uncertainty would result in ambiguous data, such as altering an image so that it can no longer be classified (e.g., to a noise image). By solely optimizing the epistemic uncertainty, the optimization process is implicitly constrained to generate meaningful, unambiguous data.

4.4.3 Experiments

The experiments aim to show the effect of TSynD on the generalization performance and robustness of classification networks. Since the test and validation sets of available datasets are often drawn from similar distributions as the training distribution, the generalization of networks is hard to measure. For that reason, a sub-sampling of the training dataset to 1% and 10% of the respective datasets is introduced. This introduces a sampling bias and makes it more likely that the test and validation dis-

Table 4.4: Comparison between TSynD maximizing mutual information (MI) and Entropy based on the accuracy values obtained on the validation sets of the respective datasets.

	OrgansMNIST		**Chest-XRay**		**OCTMNIST**	
	1%	10%	1%	10%	1%	10%
Entropy	73.5	86.0	60.8	67.9	79.5	82.9
MI	68.1	84.1	61.5	68.0	81.6	89.6

tributions contain out-of-distribution data. This also mirrors the common scenario in medical data, where training datasets are often small. The experiments concentrate on two main questions: 1) Does the proposed TSynD improve classification results when training in a low-data setting? 2) Is the training using the proposed approach more robust, e.g., against random test data augmentations and test time adversarial attacks? To investigate 1) training and evaluation are done using three different settings: baseline classifier without any additional training time augmentations; augmentation through random latent space noise during the training (see Section 4.4.2); and training using TSynD. For research question 2), the three previously trained settings are used and tested in three scenarios: no test data augmentation; Gaussian noise with $\sigma = 0.2$ added to the test data; and the test data is altered using adversarial attacks as described in [68].

The datasets used in the experiments are MedMNIST v2 [270] datasets and the Chest-Xray [241] dataset for classification since they are openly available and suitable for establishing a baseline. The experiments utilize the commonly used ResNet-18 [71] and DenseNet [82] as classifiers, and a state-of-the-art autoencoder VQ-VAE [51, 174] trained unsupervised on the full training set as the generative model. In each experiment, the classifier was trained for 100 epochs, and the model with the best validation performance was selected. The training was repeated three times, and the averaged values were reported. The TSynD process is influenced by the learning rate (chosen as 0.1) and the number of iterations (either 100 or 50) for the optimizer to maximize the epistemic uncertainty. The noise factor that is added to the feature space is chosen empirically (either 0.1 or 1.0 in the experiments).

4.4.3.1 Classification and Robustness Results

Table 4.3 shows the classification results across different MedMNIST datasets using a ResNet-18 model to compare baseline training without augmentation, augmentation with latent space noise (Noise), and TSynD. On the standard test sets, the TSynD models outperformed the baseline model and even the Noise models in almost all tested

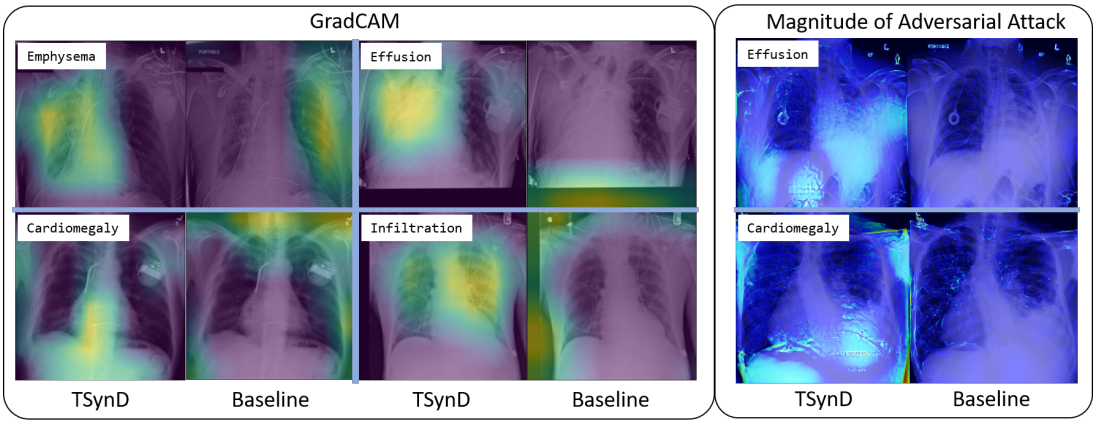


Figure 4.9: Left: EigenGradCAM maps of the baseline classifier and classifier trained with TSynD. Right: perturbation of images to minimize the probability of the given class. Depicted is the difference of the images at the start and end of the minimization.

low-data scenarios. This shows that TSynD is an effective method for training models that generalize well in such low-data settings. It further shows the advantage of the targeted optimization-based generation of new samples compared to random sampling. When Gaussian noise is applied to the test set or introduced adversarial attacks one can observe that the TSynD models are always better than the baseline models and even improve over the noise models, as well. This indicates that the samples that were generated by TSynD made the resulting model more robust against these out-of-distribution samples.

4.4.3.2 Uncertainty Maximization

Table 4.4 presents an ablation study w.r.t. to the uncertainty that is maximized during the image generation. Section 4.2.2 introduced the entropy and the mutual information (MI) as measures for the uncertainty. One can see that the MI performs better than the entropy. Figure 4.8 on page 112 further shows that the MI introduces more realistic variation. This also aligns with the theory since entropy is often viewed as a measure of aleatoric uncertainty, and the MI is viewed as a measure of epistemic uncertainty. However, the difference between maximizing the MI and the entropy is not large, indicating that the entropy is not a strict measure of the aleatoric uncertainty, and the MI is not a strict measure of the epistemic uncertainty. Improving the measure of epistemic uncertainty could further constrain the image generation process to produce more meaningful and unambiguous data (see Section 4.2.2).

4.4.3.3 Qualitative Robustness Evaluation

A classifier is trained on the Chest-Xray [241] dataset with and without TSynD. On average, an AUC improvement of about 1% is obtained using TSynD on the validation set (both on the 1% and 10% subsampling of the training dataset). However, this experiment does not concentrate on performance gain but investigates the robustness of the proposed training mechanism. The reasoning process of the classifier is explored by applying a commonly used explanation approach – EigenGradCAM [160]. The results can be seen on the left-hand side in Figure 4.9. It can be observed that the classifier trained using TSynD utilizes more relevant regions of the image than the baseline classifier trained without TSynD. The experiments additionally employed the synthetic data generation process to create adversarial examples by minimizing class probabilities instead of maximizing the classifier uncertainty. The magnitude of the difference between the original reconstruction and the optimized adversarial image can be seen on the right-hand side of Figure 4.9. One can observe that in order to minimize the probability for the classifier trained with TSynD, much larger and more relevant image regions must be altered, further indicating the increased robustness introduced by TSynD.

4.4.4 Conclusion

This work has shown how to utilize generative models to create synthetic data that explores unknown and relevant parts of the training distribution. Thus, future directions could include extending the method to generate new samples that yield high epistemic uncertainty and are relevant to the training process. The work showed that training on this synthetic data yields a model that generalizes better to out-of-distribution samples and is more robust against adversarial attacks.

In the current state, the generation method only augments the samples given. Augmenting only existing samples is not ideal from a distribution diversity standpoint. As a future direction, the method can be extended to generate new samples that yield a high epistemic uncertainty and are, therefore, relevant to the training process.

4.5 Conclusion and Outlook

The previous Chapters 2 and 3 dealt with the challenges of data collection and labeling. This chapter showed how synthetic data can be used to alleviate the need for human intervention in both of these challenges. Human labeling can be reduced since it is possible to automatically determine the labels for the synthetic data. Given control over the simulation, rare critical scenarios can be simulated directly. To record such data in the real world would require large measurement campaigns (c.f. challenge 2 in Chapter 1).

This chapter offers two perspectives for generating the training data distributions: 1. using simulation engines like CARLA and 2. using generative models. For scenario one, this chapter focused on creating methods for selecting the most meaningful images in a synthetic distribution. Such images have to fulfill three requirements:

- 1. They need to represent information that yields high epistemic uncertainty in the given discriminative model
- 2. They need to be novel w.r.t. to the already selected images
- 3. They need to be relevant to the given target domain to which the resulting discriminative model should be applied.

This chapter, therefore, proposed an acquisition function to score the relevance of a synthetic image w.r.t. these requirements. This work showed that selecting images according to this function performs better than random selection and that selecting irrelevant images can even introduce biases during training that reduce performance on the target domain. Given limited resources for training and image generation, this research provides an important solution for creating synthetic data in real-world applications.

For the application of medical image processing, simulation engines are less developed. In this domain, generative models based on DNNs are more common. These models are differentiable, which allows for optimizing the generated images w.r.t. to metrics. This chapter showed how to optimize the generation process to create images that yield high epistemic uncertainty when processed by a given discriminative model. This generation process represents an efficient way of creating rare but critical data.

Therefore, this chapter has provided two approaches for the efficient generation of synthetic data that can alleviate the need for real-world measurement campaigns (c.f. challenge 2 in Chapter 1). The next chapter investigates how synthetic data from simulation engines and the abilities of modern generative models to create highly realistic data can be combined to create large and diverse training distributions. Given such training data, the next chapter shows how to train discriminative models that generalize to unseen distributions, resulting in state-of-the-art generalization capabilities.

Chapter 5

Knowledge-Based Optimization of Synthetic Data for Real-World Domain Generalization

This chapter is partially derived from my previously published work, specifically *"Generalization by Adaptation: Diffusion-Based Domain Extension for Domain-Generalized Semantic Segmentation"* [171]. Some text, figures, and findings have been re-utilized or adapted from this publication. Co-Authors of these Publications are Manuel Schwonberg, Jan-Aike Termöhlen, Nico M Schmidt and Tim Fingscheidt.

5.1 Introduction and Motivation

Previous chapters explored various strategies for intelligently selecting training distributions from existing measurement campaign data. They discussed methods to reduce labeling efforts when adapting to new domains and distributions. Additionally, these chapters demonstrated how leveraging already annotated simulation data can decrease the need to record real-world data by filling in missing parts of the training distribution.

Therefore, the presented research has approached the challenge of generalization from multiple angles. These efforts aim to develop discriminative models, such as semantic segmentation models, that can generalize effectively to novel domains and distributions. The field of domain generalization focuses on evaluating and developing methods that enable models trained on a source domain to generalize well to entirely new domains, including those with unknown sensor data. The domains of application for trained DNNs are rarely known during the developmental phase. Even if they are known, carrying out large measurement campaigns to record the relevant data from these target domains is often challenging. Therefore, active learning or semi-supervised learning methods can not be directly applied. For the environment perception of self-driving cars, obtaining data from every relevant location, weather, and lighting scenario is difficult, and even more importantly, many of the relevant scenarios are unknown. I.e., since one is unaware of these scenarios, no data can be recorded. Medical image processing additionally faces the problem of the relevant domains of application not

being accessible for recording data. This constraint on data access could be due to data privacy regulations or the release of novel scanner versions. Therefore, the objective of domain generalization without access to all the relevant target domain data is crucial in these applications and aligns with the two main challenges introduced in Chapter 1. Given a network that generalizes well, the need for annotation is reduced, and critical scenarios that need to be recorded are reduced.

This chapter presents the DIDEX method for extending the original training distributions to develop models that generalize effectively across as many target domains as possible. DIDEX utilized symbolic knowledge about relevant target domains. Such knowledge is represented in the form of text, which allows for the utilization of previously inaccessible sources of information. The result is a distribution of images representing a good approximation of relevant domains. The DIDEX utilizes this distribution to train segmentation networks that generalize well without access to sensor data from these real-world domains (see Figure 5.1 page 122).

5.2 Generalization Without Accessing Real Data

In recent years, the success of deep learning has led to significant advancements in the field of computer vision, e.g., for semantic segmentation. For this task, the usage of synthetic data is particularly interesting as manual data labeling is time- and cost-intensive. Synthetic data can be valuable for training and validation since it allows the simulation of rare and dangerous events. However, deploying models trained on synthetic data in real-world settings with varying data distributions is still challenging due to large domain shifts towards real-world data [169, 233]. One approach to overcome this problem is by unsupervised domain adaptation (UDA) (c.f. Chapter 3), where unlabeled data from a real target domain is available to adapt to [9, 74, 109, 150, 165, 166, 226, 256, 267]. Some approaches also perform this task source-free [109] or in a continual manner [108, 226]. Recently, vision transformer models [44, 255] caused a significant increase in performance and reduced the domain gap, with DAFormer [77] being the initial work. All these methods rely on access to target domain data to adapt to this particular domain. In practice, this data might not always be available due to various reasons. Data collection can be difficult because, e.g., adverse weather conditions such as fog, rain, and snow do not constantly occur, and sometimes, the target domain cannot be anticipated at all. Consequently, the field of domain generalization (DG) emerged where no data from the target domain is available at all, and the task is to generalize from only a single, usually synthetic, source domain to unseen and unknown target domains [6, 88, 117, 175, 180, 275]. Style transfer methods such as AdaIN [83] are widely used in UDA, especially in combination with other techniques [74, 244, 252]. These methods cannot be used for domain generalization since no target domain guidance is available for the style transfer. For this reason,

the majority of DG methods perform style randomization or augmentation to alter the visual appearance of the source domain [88, 106, 117, 179, 223, 275]. However, these methods reveal two major issues. First, many of them require additional real data from auxiliary domains [88, 106, 117, 179], which disrupts the idea of domain generalization. Second, the style randomization is difficult to control and often leads to limited diversity, as mostly only textures are changed.

This work proposes a novel method that tackles the domain generalization problem by diffusion-based domain extension (DIDEX). Diffusion models have demonstrated remarkable capabilities in capturing complex distributions and semantics and generating realistic samples with high quality [39, 72, 190, 214]. The introduced method not only allows the generation of realistic pseudo-target images from synthetic images but also alters important parameters such as location, time, weather conditions, and semantic content via text prompts. The approach uses these capabilities to generate a realistic-looking pseudo-target domain, as shown in Figure 5.4 on page 126. Diffusion models have severe limitations in the semantic consistency of the generated output [287] w.r.t. the input image. To address these limitations, inspiration is drawn from the field of *unsupervised* domain adaptation (UDA). In this context, the diffusion data is defined as the target domain for the adaptation process. The key contributions of this work are as follows:

- A new method leveraging diffusion models for domain transfer, enhancing the model's ability to generalize across domains without accessing real data.
- By utilizing UDA techniques, the semantic inconsistency limitation of diffusion models is overcome, ensuring improved domain generalization performance.
- The results are reported on common benchmarks, where the method outperforms the state-of-the-art domain generalization methods by a large margin.

5.3 Domain Generalization: An Overview

Domain generalization methods for semantic segmentation try to overcome the domain gap by either constraining learned feature distributions or randomizing distributions by augmentation or extension during training.

Pan et al.proposed IBN-Net [175] as one of the first DG approaches and employed instance-batch normalization (IBN), which is more robust w.r.t. appearance changes. Further approaches utilize instance whitening to decorrelate features from different layer channels, thereby removing style-related information. Pan et al.[176] proposed an approach that can switch between different whitening and normalization techniques depending on the task. RobustNet [32], DIRL [261], and SAN+SAW [180] all proposed improvements such as better guidance of the whitening process by a so-called

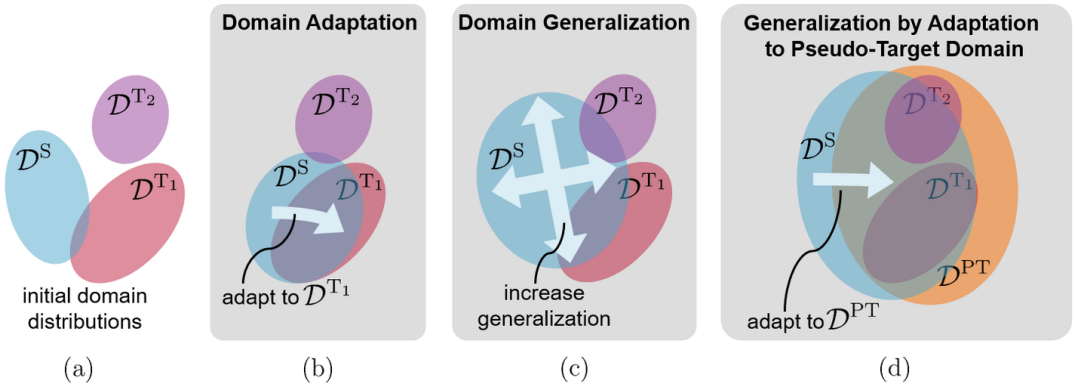


Figure 5.1: Simplified representation of different domain distributions and the effects of domain adaptation and generalization methods. Shown in (a) are the initial distributions for a source domain $\mathcal{D}^S$ and two target domains $\mathcal{D}^{T_1}$ and $\mathcal{D}^{T_2}$. In (b), the effect of domain adaptation from $\mathcal{D}^S$ to a specific target domain $\mathcal{D}^{T_1}$ is shown. In (c), the effect of commonly used domain generalization methods is shown. These methods broaden the source distribution, e.g., via data augmentation, but in a less directed manner. In (d), the effect of the proposed method is shown, where a domain extension through the generation of a pseudo-target domain is performed $\mathcal{D}^{PT}$ that also covers the distributions of multiple target domains ($\mathcal{D}^{T_1}$ and $\mathcal{D}^{T_2}$). For the purpose of clarity, only $Q=2$ target domains are shown here. In the remainder of this chapter, by default, $Q=4$ target domains for the evaluation are chosen.

sensitivity-aware prior module [261] or semantic-aware whitening and normalization [180].

Domain randomization can be separated into approaches that require additional real-world data (auxiliary domains) and those that do not need it. Yue et al. [275] proposed one of the first domain randomization methods combining domain randomization and pyramid consistency (DRPC), enforcing semantic feature consistency between differently stylized images with transferred textures from ImageNet. Similarly, Huang et al. [88] with FSDR employ style randomization in the frequency space, Peng et al. [179] use painting for their style randomization and WildNet by Lee et al. [117] applies style randomization in conjunction with contrastive and consistency learning. Kim et al. [106] utilize style randomization with internet-sampled images and self-training similar to UDA approaches. In contrast to these methods, the proposed method can be seen as a more structured distribution randomization without the need for real-world auxiliary domains.

Only a few works perform style randomization without additional real data. AugFormer by Schwonberg et al.[200] showed that simple augmentations can significantly increase domain generalization. Similarly, Sun et al. [223] proposed a strong style randomization in the CIELAB color space. Zhong et al. [295] proposed to alter the

channel-wise mean and standard deviation in an adversarial manner to generate stylized images which are hard to segment for the model. SHADE by Zhao et al. [286] introduces a style hallucination module based on farthest point sampling and generates new diverse samples by a linear combination of the basic styles. Bi et al.[6] proposed CMFormer, the first DG approach tailored towards vision transformers. It fuses low and original resolution in a so-called content-enhanced mask attention. In contrast, the approach presented in this chapter is independent of the architecture. Gong et al.[66] proposed PromptFormer as the first DG approach utilizing diffusion models. However, their method fundamentally differs from the DIDEX approach since they employ a diffusion backbone for domain-invariant pre-training, where scene and category text prompts help the model to disentangle domain-variant and invariant information. They also employ a consistency loss to enforce the same predictions under different input prompts.

Diffusion Models: diffusion models for image synthesis recently emerged and outperform the established standard using GANs [39]. Several improvements were proposed to accelerate the image generation, namely denoising diffusion probabilistic models (DDPMs) [72] and denoising diffusion implicit models (DDIMs) [214]. Rombach et al. proposed latent diffusion models (LDMs), sometimes referred to as stable diffusion models, which reduce the training and inference time and enable text-to-image and image-to-image generation. Based on this, several extensions and improvements were proposed, such as stable diffusion XL (SDXL) [182] or ControlNet [287], which offers the possibility to constrain the generation process with different additional inputs such as depth or semantics.

5.4 The DIDEX Approach

Given the capabilities of diffusion models to create controllable data distributions, a big problem in the previous domain generalization literature can be addressed: the unknown target domain distributions. Although the general distributions of the target domain remain unknown, it is reasonable to assume that certain a priori knowledge about these distributions is known. Given the symbolic control through text prompts over the outputs of a diffusion model, this a priori knowledge can be utilized to generate relevant image distributions. This chapter shows how to utilize these distributions to create networks that generalize well. Figure 5.2 on page 124 shows how to utilize the generated distributions. The general aim is to train a segmentation model $\mathbf{M}$ that generalizes to the relevant target domains $\mathcal{D}^{\mathrm{T}_q}, \quad q \in \mathcal{Q} = \{1, \ldots, Q\}$. As Figure 5.1 indicates, other than with UDA during training, no data from $\mathcal{D}^{\mathrm{T}_q}$ is available. This chapter shows how to create synthetic data estimating images from $\mathcal{D}^{\mathrm{T}_q}$ utilizing the properties diffusion models to create images based on symbolic prior information.

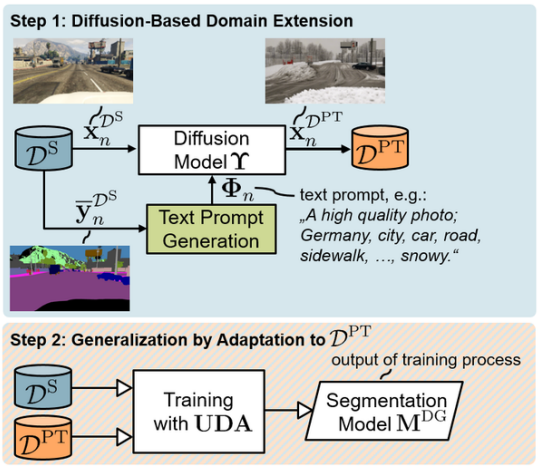


Figure 5.2: Overview showing a **block diagram** of the diffusion-based domain extension (DIDEX) method (step 1), followed by the **generalization-by-adaptation process** to the newly generated pseudo-target domain $\mathcal{D}^{\mathrm{PT}}$ (step 2).

The crucial advantage of diffusion models is the prompt-based control of the generated image. That enables the definition of the content and style of the generated images and the creation of a pseudo-target domain $\mathcal{D}^{\mathrm{PT}}$. Two different strategies for the generation of pseudo-target domains are possible. First, a specialized pseudo-target domain could be created if some information regarding the target domains is available, such as location, weather, etc. In the presented approach, however, the aim is to achieve a domain generalization without any assumptions about the target domains and generate $\mathcal{D}^{\mathrm{PT}}$ with high diversity to cover a wide range of possible target domain distributions as shown in Figure 5.1 (d). The objective of the generation is to create a pseudo-target domain, which is the union of all relevant target domains. This objective is formulated as

$$\mathcal{D}^{\mathrm{PT}} \approx \bigcup_{q \in \mathcal{Q}} \mathcal{D}^{\mathrm{T}_q}, \tag{5.1}$$

where $\mathcal{D}^{\mathrm{T}_q}$ denotes the relevant target domains and $\mathcal{Q} = 1, ..., Q$ and Q is the number of target domains. The domain generalization setting only provides a single source domain distribution to obtain $\mathcal{D}^{\mathrm{PT}}$. The diffusion model is used to extend the initial source distribution into the more diverse and wider pseudo-target distribution. A single image of $\mathcal{D}^{\mathrm{PT}}$ can be obtained by:

$$\mathbf{x}_n^{\mathrm{PT}} = \boldsymbol{\Upsilon}(\mathbf{x}_n, \boldsymbol{\Phi}_n) \tag{5.2}$$

where $\boldsymbol{\Upsilon}$ denotes the diffusion model and $\boldsymbol{\Phi}_n$ the text prompt and $\mathbf{x}_n$ is the input image and $n = 1, \ldots, N_{\mathrm{data}}$ is the index over the images in the dataset. The generated pseudo-target domain provides the crucial advantage of having a target domain available that can be used for adapting the segmentation network, as shown in Figure 5.1 (d).

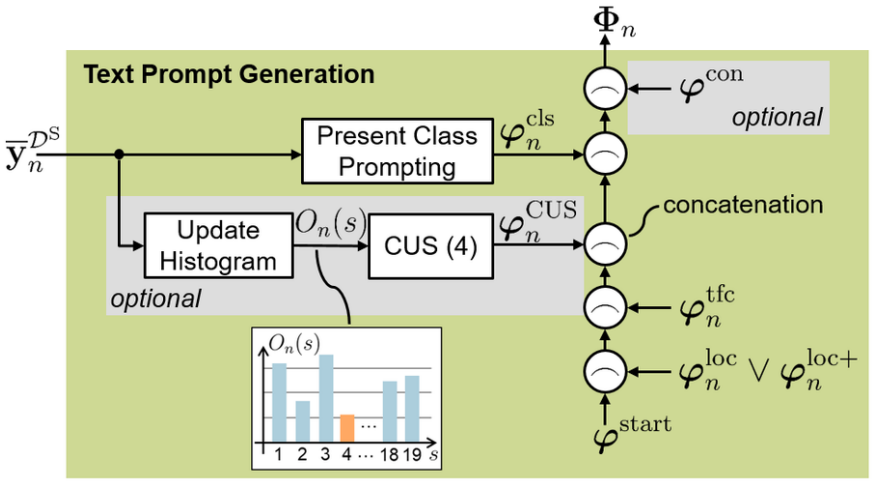


Figure 5.3: Block diagram of the **text prompt generation** from Figure 5.2 providing $\boldsymbol{\Phi}_n$. The operation in the circles denotes a concatenation. Concatenation of φ^{CUS} and φ^{con} is optional. Class-uniform sampling (CUS) is described in (5.4). The rarest class in the example histogram $O_n(k)$ is marked in orange.

Compared to other DG approaches, this is a novel and unique feature of the DIDEX method and enables obtaining generalization by adaptation.

Semantic consistency: Semantic or structural consistency of the image-to-image generation is often not given for complex urban street scenes, i.e., pedestrians or cars can disappear, and the semantic content of the pixels is changed (see Figure 5.4 page 126). Directly using the synthetic labels for the pseudo-target domain is, therefore, not possible. However, one can assume that a certain structural and semantic consistency is also beneficial for the adaptation in step two, and for this reason, depth-guided Stable Diffusion 2.0 [190] and ControlNet [287] are employed, which offer a higher structural and semantic consistency. Both extend the image generation to

$$\mathbf{x}_n^{\text{PT}} = \boldsymbol{\Upsilon}(\mathbf{x}_n, \mathbf{G}(\mathbf{x}_n), \boldsymbol{\Phi}_n)\,, \tag{5.3}$$

where $\mathbf{G}$ is a constraint (or guidance) based on the image $\mathbf{x}_n$, e.g., by canny edge extraction, depth, or semantic prediction.

5.4.1 Text Prompt Generation

Text prompts are crucial for the class distribution of the pseudo-target domain since the prompts determine the style and content of the generated images. There are no existing strategies for designing text prompts for diffusion models in the context of complex urban street scenes. For this reason, a new systematic and modular text prompt generation is developed, which is visualized in Figure 5.3. The text prompt

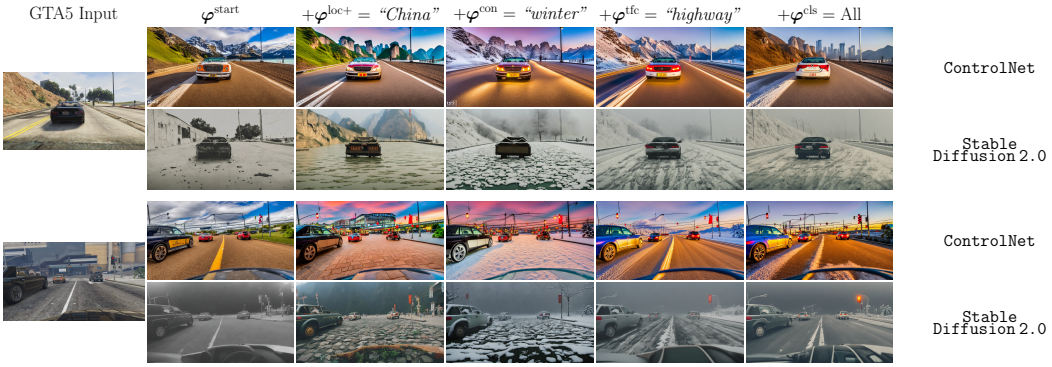


Figure 5.4: Visualization of the prompt ablation study showing the impact of the prompt on the images of $\mathcal{D}^{\text{PT}}$ for ControlNet [287] and SD 2.0 [190]. Each prompt has a corresponding impact on the content of the generated images. However, semantic inconsistencies can be seen, such as trees turning into mountains or buildings that disappear. More examples are provided in the supplement.

strategy aims at varying the image content and style at different levels, introducing as much diversity as possible into the pseudo-target domain. To design the building blocks of the prompts, this work considers major causes of real-to-real domain shifts as guidance.

The start of each prompt is $\boldsymbol{\varphi}_n^{\text{start}}$ = *"A high quality photo;"* Particularly emphasizing photorealism with prompts such as "real" or "photorealistic" does not improve the realism of the output.
The location shift between different countries or continents is important in automated driving. Therefore, the approach includes a set of different countries and regions to include their style and content in D^{PT}. The approach concatenates a geographic location to the start prompt by choosing a location $\boldsymbol{\varphi}_n^{\text{loc}} \in \{$*"Europe"*, *"Germany"*$\}$ or a location from an extended set $\boldsymbol{\varphi}_n^{\text{loc+}} \in \{$*"Europe"*, *"Germany"*, *"USA"*, *"China"*, *"India"*$\}$. This set of locations is not globally comprehensive and may have a significant impact on the domain generalization performance. An extensive study with more regions of the world can be done in future work to improve generalization. Subsequently, the approach concatenates a traffic location $\boldsymbol{\varphi}_n^{\text{tfc}} \in \{$*"_"*(blank), *"highway"*, *"city"*$\}$. The blank denotes that no traffic location is provided.

Present class prompting: The work utilizes the GTA5 segmentation label map $\overline{\mathbf{y}}_n^{\mathcal{D}^{\text{S}}}$ to identify and concatenate all classes that are present in the current image. This serves as additional guidance on which classes $\boldsymbol{\varphi}_n^{\text{cls}} \subseteq \mathcal{K}$ ($\mathcal{K} = \{1, ..., K\}$ the set of classes) the diffusion model should include in the output.

Class-uniform sampling (CUS): Optionally, a class-uniform sampling (CUS) can be introduced to mitigate the unbalanced class distribution by tracking the number of times a class occurred in the synthetic source images and concatenating the least

often seen class to the prompt. For each image with index n, the segmentation labels from the respective source image are used $\overline{\mathbf{y}}_n^{\mathcal{D}^\mathrm{S}}$ to update a histogram with the class occurrences $O_n(k)$. Then, the rarest class name is concatenated to the prompt:

$$\boldsymbol{\varphi}_n^{\mathrm{CUS}} = \mathrm{class}(\arg\min_{k \in \mathcal{K}}(O_n(k))). \tag{5.4}$$

After concatenating the rarest class to the prompt, the counter for this class is increased. This leads to a more uniform class distribution in the domain extension.

Image conditions: Adverse weather or visibility conditions are a challenge for autonomous vehicles. Therefore, these shifts are represented in the pseudo-target domain by adding one of the following words to the prompt:

$$\begin{aligned}\boldsymbol{\varphi}_n^{\mathrm{con}} \in \{&\textit{“rain”}, \textit{“fog/mist”}, \textit{“snowy”}, \textit{“sunny”},\\ &\textit{“overcast”}, \textit{“stormy”}, \textit{“overexposure”},\\ &\textit{“underexposure”}, \textit{“evening”}, \textit{“morning”},\\ &\textit{“night/darkness”}, \textit{“backlighting”},\\ &\textit{“artificial lighting”}, \textit{“harsh light”},\\ &\textit{“dappled light”}, \textit{“sun flare”}, \textit{“hazy/haze”},\\ &\textit{“spring”}, \textit{“autuum”}, \textit{“winter”}, \textit{“summer”}\}\end{aligned}$$

By concatenating all the different building blocks, the text prompt is created according to Figure 5.3 (page 125) as:

$$\boldsymbol{\Phi}_n = \boldsymbol{\varphi}^{\mathrm{start}} \frown (\boldsymbol{\varphi}_n^{\mathrm{loc}} \vee \boldsymbol{\varphi}_n^{\mathrm{loc+}}) \frown \boldsymbol{\varphi}^{\mathrm{tfc}} \frown \boldsymbol{\varphi}^{\mathrm{CUS}} \frown \boldsymbol{\varphi}^{\mathrm{cls}} \frown \boldsymbol{\varphi}^{\mathrm{con}}, \tag{5.5}$$

with $\frown$ symbolizing concatenation and $\boldsymbol{\varphi}^{\mathrm{CUS}}$ and $\boldsymbol{\varphi}^{\mathrm{con}}$ being optional. The base prompt is denoted without the optional strings as $\boldsymbol{\Phi}_n^{\mathrm{base}}$. A sample prompt can become, e.g., *“A high quality photo; Europe, highway, road, car, building, vegetation, winter”*. If there are multiple options within one of the building blocks, the prompt is selected randomly. $\mathcal{D}^{\mathrm{PT}}$ is generated before the adaptation step.

5.4.2 Generalization by Adaptation

Even with strong semantic guidance by depth or the semantic ground truth, the stable diffusion models generate partially highly inconsistent outputs. However, the area of unsupervised domain adaptation provides powerful methods that are capable of handling datasets without labels effectively. These methods are utilized to adapt the segmentation model towards the pseudo-target domain that was generated in a 1$^{\mathrm{st}}$ step (Section 5.4). UDA methods usually obtain an adapted segmentation model by $\mathbf{UDA}(\mathcal{D}^S, \mathcal{D}^T) \rightarrow \mathbf{M}^{\mathrm{UDA}}$, where $\mathcal{D}^T$ is usually a real data target domain without labels and $\mathbf{M}$ the domain-adapted model. The DIDEX method changes this setting slightly

but notably to $\mathbf{UDA}(\mathcal{D}^S, \mathcal{D}^{\mathrm{PT}}) \rightarrow \mathbf{M}^{\mathrm{DG}}$, where model $\mathbf{M}$ is adapted to the pseudo-target domain $\mathcal{D}^{\mathrm{PT}}$ and therefore also generalizes without accessing any real data. In this case, UDA strategies create this supervision signal by aligning the distributions of the source $\mathcal{D}^{\mathrm{S}}$ and pseudo-target domain $\mathcal{D}^{\mathrm{PT}}$.

5.5 Experimental Setup

The following introduces the employed datasets, metrics, and diffusion and segmentation network architectures. Afterward, the UDA methods utilized are introduced.

5.5.1 Datasets and Metrics

Datasets: The experiments follow the common domain generalization standard setting [66, 117, 179, 180, 223] and employ the two synthetic datasets SYNTHIA [191] (SYN) and GTA5 [188] as the source domains. The datasets comprise 9400 and 24966 images, respectively, which are denoted as $\mathcal{D}^{\mathrm{SYN}}_{\mathrm{train}}$ and $\mathcal{D}^{\mathrm{GTA5}}_{\mathrm{train}}$.
To evaluate the domain generalization capabilities, the experiments use Cityscapes (CS) [36], BDD100k (BDD) [272], Mapillary Vistas (MV)[164] as the real-world target domains with 500, 1000 and 2000 validation images, respectively. The training sets of these datasets remain unused in the presented generalization method. All experiments are evaluated on the validation sets $\mathcal{D}_{\mathrm{val}}$ of the target domains and compute the domain generalization (DG) mean over these sets, as is common practice in domain generalization [88, 180, 275]. Additionally to this domain generalization benchmark, some experiments include the ACDC dataset [197]. ACDC has 406 validation images, which contain adverse weather conditions. It is excluded from the DG mean because most other approaches do not report ACDC performance.

Metrics: For evaluation the mean intersection over union (mIoU) of $S = 19$ segmentation classes [36, 188, 197] is used for GTA5 trained models and $S = 16$ classes for models trained on SYNTHIA, as it is common practice [109].

5.5.2 Network Architectures

Diffusion models: For the data generation process, Stable Diffusion 2.0 (SD2.0) [190] is utilized as the standard diffusion model. ControlNet [287], which provides several options to constrain the output generation, is also used for comparison in the ablation studies. During data creation, the image-to-image prompting strategy described in Section 5.4 is used. The ablation studies w.r.t. the data generation process that is described in Section 5.6 analyze the different prompting elements.

Segmentation models: As the network architectures, the common standards in the area of domain generalization are chosen. The DeepLabV2 [19] architecture with

Table 5.1: Domain generalization performance (mIoU (%)) of several methods employing two different encoder networks. **Training** was performed on the synthetic **GTA5** ($\mathcal{D}^S = \mathcal{D}^{\text{GTA5}}_{\text{train}}$) dataset. **Evaluation** is performed on various real-world **validation sets** ($\mathcal{D}^T = \mathcal{D}_{\text{val}}$). Prior work results are either cited from [117] (marked with °) or from the respective paper (marked with *).

Enc.	DG Method	No real data	mIoU (%) on			
			$\mathcal{D}^{\text{CS}}_{\text{val}}$	$\mathcal{D}^{\text{BDD}}_{\text{val}}$	$\mathcal{D}^{\text{MV}}_{\text{val}}$	**DG mean**
ResNet	Baseline	✓	36.1	36.6	43.8	38.8
	IBN-Net° [175]	✓	37.7	36.7	36.8	37.1
	RobustNet° [32]	✓	37.3	38.7	38.1	38.0
	DRPC* [275]	✗	42.5	38.7	38.1	39.8
	SW* [176]	✓	36.1	36.6	32.6	35.1
	FSDR* [88]	✗	44.8	41.2	43.4	43.1
	SAN+SAW* [180]	✓	45.3	41.2	40.8	42.4
	WEDGE* [106]	✗	45.2	41.1	48.1	44.8
	GTR* [179]	✗	43.7	39.6	39.1	40.8
	SHADE* [286]	✓	46.7	43.7	45.5	45.3
	WildNet° [117]	✗	45.8	41.7	47.1	44.9
	RICA* [223]	✓	48.0	**45.2**	46.3	46.5
	DIDEX with MIC [80]	✓	**52.4**	40.9	**49.2**	**47.5**
Transformer	Baseline	✓	46.6	45.6	50.1	47.4
	ReVT* [227]	✓	50.0	48.0	52.8	50.3
	DAFormer* [77]	✓	52.7	47.9	54.7	51.7
	HRDA* [79]	✓	57.4	49.1	61.2	55.9
	CMFormer* [6]	✓	55.3	49.9	60.1	55.1
	PromptFormer* [66]	✓	52.0	-	-	-
	DIDEX with MIC [80]	✓	**62.0**	**54.3**	**63.0**	**59.7**

a ResNet-101 is used to evaluate the generalization on a CNN-based architecture, and for the recently emerged vision transformer, the DAFormer [77] network is used.

5.5.3 Employed UDA Methods

Five different state-of-the-art domain adaptation approaches are utilized for the generalization by adaptation step: DACS [231], DAFormer [77], HRDA [78], MIC [80] and SePiCo [256]. As described in Section 5.4, these UDA methods are utilized to adapt

Table 5.2: Domain generalization performance (mIoU (%)) of several methods employing two different encoder networks. **Training** was performed on the synthetic **SYNTHIA** ($\mathcal{D}^S = \mathcal{D}^{\text{SYN}}_{\text{train}}$) dataset. **Evaluation** is performed on real-world **validation sets** ($\mathcal{D}^T = \mathcal{D}_{\text{val}}$). Prior work results are either cited from [117] (marked with °) or from the respective paper (marked with *).

Enc.	DG Method	No real data	mIoU (%) on			
			$\mathcal{D}^{\text{CS}}_{\text{val}}$	$\mathcal{D}^{\text{BDD}}_{\text{val}}$	$\mathcal{D}^{\text{MV}}_{\text{val}}$	**DG mean**
ResNet	Baseline	✓	34.3	27.8	38.0	33.4
	IBN-Net° [175]	✓	34.2	32.6	36.2	34.3
	DRPC* [275]	✗	37.6	34.4	34.1	35.4
	SW* [176]	✓	36.1	36.6	32.6	35.1
	FSDR* [88]	✗	40.8	37.4	39.6	39.3
	SAN+SAW* [180]	✓	40.9	36.0	37.3	38.1
	WEDGE* [106]	✗	40.9	38.1	43.1	40.7
	GTR* [179]	✗	39.7	35.3	36.4	37.1
	RICA* [223]	✓	45.0	36.3	41.6	41.0
	DIDEX with MIC [80]	✓	**53.1**	**41.8**	**50.3**	**48.4**
Transformer	Baseline	✓	41.4	36.2	42.4	40.0
	ReVT* [227]	✓	46.3	40.3	44.8	43.8
	CMFormer* [6]	✓	44.6	33.4	43.3	40.4
	PromptFormer* [66]	✓	49.3	-	-	-
	DIDEX with MIC [80]	✓	**59.8**	**47.4**	**59.5**	**55.6**

from the synthetic source domain to the pseudo-target domain. Previous experiments have shown that the adaptation of UDA methods to other domain shifts can diminish the performance [197]. In this context, *it has to be noted that no finetuning of any hyperparameters of the employed UDA methods was done.* All the methods are employed as provided by the respective authors.

5.6 Evaluation and Discussion

This section first compares the DIDEX approach to state-of-the-art domain generalization methods. Afterward, the impact of different prompting strategies is investigated along with UDA approaches, semantic consistency constraints, and the quantity of generated images.

5.6.1 Comparison with State of the Art

For GTA5-trained models, for both the ResNet-based and the Transformer-based backbones, DIDEX clearly achieves a new state-of-the-art (SOTA) performance for the DG mean. For the ResNet-based models, DIDEX improves significantly on Cityscapes and Mapillary. For SYNTHIA-trained models (Table 5.2), the method provides an even larger improvement. With a ResNet-based backbone, DIDEX outperforms other approaches by 7.4% absolute for the DG mean and by 11.8% abs. with a transformer backbone. On Mapillary Vistas, the method provides 14.7% abs. mIoU improvement compared with the best performing prior method [227]. With 59.8% mIoU on Cityscapes, DIDEX performs competitively with the UDA method DAFormer [77] with 60.9% mIoU *without using any real data during training.*

5.6.2 Influence of Prompting Strategy

Table 5.3 shows the results of the resulting domain generalization when varying the prompts w.r.t. location, environment condition, and class-uniform sampling. Overall, the most important part of the text prompts seems to be the class-uniform sampling ($+\boldsymbol{\varphi}^{\text{CUS}}$) as it results in the highest domain generalization mIoU. But increasing the variation in each of the dimensions seems to have a positive effect, although the increase in performance is rather small for $\boldsymbol{\varphi}^{\text{loc}} \rightarrow \boldsymbol{\varphi}^{\text{loc+}}$ and $+\boldsymbol{\varphi}^{\text{con}}$. Increasing the variety of locations has the biggest effect on the performance on the Mapillary Vistas dataset [164], which consists of images from various locations all over the world.

However, increasing the number of conditions does not increase performance on MV or ACDC, which are very diverse w.r.t. the conditions. This might be related to the insufficiently realistic generation of such conditions by the diffusion model or related to the fact that the HRDA [78] method has difficulties adapting to these conditions. Finally, one can observe that using all text prompts does not improve over only class-uniform sampling.

Table 5.3: Influence of the different text prompt building blocks on the generalization performance (mIoU(%)). **Training** was performed on the synthetic **GTA5** ($\mathcal{D}^S = \mathcal{D}^{\text{GTA5}}_{\text{train}}$) dataset. **Evaluation** is performed on various real-world **validation sets** ($\mathcal{D}^T = \mathcal{D}_{\text{val}}$). The adaptation was performed with the HRDA method [78] and a transformer-based encoder.

Base Prompt ($\boldsymbol{\Phi}^{\text{base}}$)	**Add. Location** ($\boldsymbol{\varphi}^{\text{loc}} \rightarrow \boldsymbol{\varphi}^{\text{loc+}}$)	**+ Conditions** ($+\boldsymbol{\varphi}^{\text{con}}$)	**+ CUS** ($+\boldsymbol{\varphi}^{\text{CUS}}$)	**mIoU (%) on** $\mathcal{D}^{\text{CS}}_{\text{val}}$	$\mathcal{D}^{\text{BDD}}_{\text{val}}$	$\mathcal{D}^{\text{MV}}_{\text{val}}$	$\mathcal{D}^{\text{ACDC}}_{\text{val}}$	**DG mean**
✓				58.5	52.2	62.9	46.9	57.9
✓	✓			58.7	52.5	63.4	46.7	58.2
✓		✓		59.4	52.7	62.7	46.8	58.3
✓			✓	**61.2**	52.5	**63.7**	**48.8**	**59.1**
✓	✓		✓	58.6	51.8	62.8	45.2	57.7
✓		✓	✓	60.1	**53.7**	63.5	46.6	**59.1**
✓	✓	✓	✓	58.8	52.7	63.2	47.4	58.3

Table 5.4: Influence of DIDEX combined different UDA methods on the generalization performance (mIoU(%)). **Training** was performed on the synthetic **GTA5** ($\mathcal{D}^S = \mathcal{D}^{\text{GTA5}}_{\text{train}}$, upper part) or **SYNTHIA** ($\mathcal{D}^S = \mathcal{D}^{\text{SYN}}_{\text{train}}$, lower part) dataset. **Evaluation** is performed on various real-world **validation sets** ($\mathcal{D}^T = \mathcal{D}_{\text{val}}$). Text prompts comprised the base prompt $\boldsymbol{\Phi}^{\text{base}}$ and the CUS $\boldsymbol{\varphi}^{\text{CUS}}$. * indicates additional usage of the masked image consistency loss [80].

	Enc.	DIDEX + ...	mIoU (%) on				
			$\mathcal{D}^{\text{CS}}_{\text{val}}$	$\mathcal{D}^{\text{BDD}}_{\text{val}}$	$\mathcal{D}^{\text{MV}}_{\text{val}}$	$\mathcal{D}^{\text{ACDC}}_{\text{val}}$	**DG mean**
$\mathcal{D}^S$: **GTA5**	ResNet	DACS* [231]	46.9	40.0	45.2	31.9	44.0
		SePiCo [256]	44.9	36.4	38.8	30.4	40.0
		DAFormer* [77]	50.4	**41.8**	47.1	33.9	46.4
		MIC [80]	**52.4**	40.9	**49.2**	**36.1**	**47.5**
	Transformer	DACS* [231]	52.0	49.0	53.7	41.9	51.8
		SePiCo [256]	57.4	49.7	56.4	44.5	54.5
		DAFormer* [77]	57.7	**56.6**	60.7	46.4	58.3
		MIC [80]	**62.0**	54.3	**63.0**	**50.1**	**59.7**
$\mathcal{D}^S$: **SYNTHIA**	ResNet	DACS* [231]	47.5	38.6	41.5	30.1	42.5
		SePiCo [256]	43.3	32.6	40.6	27.7	38.8
		DAFormer* [77]	49.8	40.0	45.5	**33.7**	45.1
		MIC [80]	**53.1**	**41.8**	**50.3**	33.3	**48.4**
	Transformer	DACS* [231]	52.1	38.4	48.0	36.4	46.2
		SePiCo [256]	54.4	45.5	52.3	37.7	50.7
		DAFormer* [77]	53.3	44.5	52.3	38.6	50.0
		MIC [80]	**59.8**	**47.4**	**59.5**	**43.5**	**55.6**

5.6.3 Influence of UDA Approaches

Table 5.4 shows the influence of the UDA approach that is used for the generalization by adaptation step to the pseudo-target domain. Recent UDA methods, such as DAFormer [77] and MIC [80], adapt better to the pseudo-target domain and thus generalize better across domains. It should be noted that even comparably simple UDA methods such as DACS [231] obtain a high domain generalization and outperform previous SOTA methods for SYNTHIA as the source dataset. However, SePiCo [256] with a ResNet-101 backbone performs worse than the other methods, which might be caused by the lack of hyperparameter optimization.

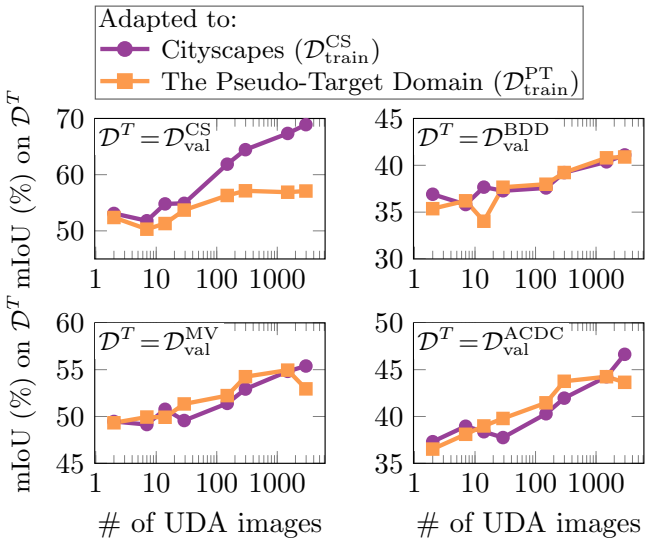


Figure 5.5: Influence of the # of UDA images (target domain) on the generalization performance (mIoU (%)). **Training** was performed on the synthetic **GTA5** ($\mathcal{D}^S = \mathcal{D}^{\text{GTA5}}_{\text{train}}$) dataset. **Evaluation** is performed on various real-world **validation sets** ($\mathcal{D}^T = \mathcal{D}_{\text{val}}$). DAFormer [77] was used for UDA, and text prompts comprised only the base prompt $\boldsymbol{\Phi}^{\text{base}}$ and the CUS block $\boldsymbol{\varphi}^{\text{CUS}}$.

5.6.4 Influence of Image Quantity & Consistency

Figure 5.5 shows the results of randomly sampled subsets of either Cityscapes (purple) or the pseudo-target domain (orange) used in the UDA approach. For each step, DAFormer [77] is applied for the adaptation from GTA5 to the subset. One can observe that the model adapted to Cityscapes is on par with the pseudo-target domain model on Cityscapes up until a subset size of 29 UDA images but gets increasingly better with more sampled images. This divergence of performance does, however, not occur for the other domains (BDD, MV, ACDC), where the models are mostly on par with each other. This indicates that the increase in performance on Cityscapes is caused by a specialization of the model to the target domain but does not increase its generalization capabilities relative to the model adapted to the pseudo-target domain. It further indicates that the images in the pseudo-target domain are of a similar "functional quality" for generalizing to other real-world domains as the real-world Cityscapes data.

5.7 Conclusions

This chapter introduced a novel diffusion-based domain extension (DIDEX) method for domain generalization, which utilizes the generative capabilities of diffusion models. This method projects the problem of domain generalization to the problem of domain

adaptation, which opens the possibility of utilizing powerful adaptation methods for domain generalization. The presented diffusion-based domain extension outperforms previous state-of-the-art methods by a large margin across datasets and architectures, with GTA5 as the source dataset by 3.8% abs. mIoU and with SYNTHIA even by 11.8% abs. mIoU on average. The text prompt ablation study has shown that information about present classes is beneficial for the pseudo-target domain. A remarkable result is that the functional quality of the diffusion-generated data for the purpose of domain generalization is comparable to real Cityscapes data, highlighting the potential of using diffusion models for domain generalization.

DIDEX addresses the two main challenges introduced in Chapter 1. On the one hand, it provides efficient data acquisition by employing diffusion models to extend the training data distribution with functionally valuable synthetic data. Given that prior knowledge can be used for the targeted generation/ acquisition of data, challenge 2 from Chapter 1 is addressed. On the other hand, DIDEX contributes to challenge 1 from Chapter 1 by introducing strong generalization of the trained discriminative models to reduce the need for additional labeling. This is especially true since even the source domain is synthetic.

5.7.1 Outlook

DIDEX is based on a symbolic (text-based) description of image content. This symbolic description comes with two challenges. First, the expressiveness of text-based descriptions is limited and can be ambiguous regarding the content. Additionally, failure cases (images) of DNNs are often hard to describe symbolically. For example, such failure cases often consist of image patterns. Second, the current DIDEX utilizes prior knowledge about relevant distributions to create the pseudo-target domain. Such knowledge represents information that is known to be unknown. However, it would also be important to create parts of the distribution that are not yet known to be difficult (unknown unknowns). Generative models can generate such images but must be guided towards creating this information.

The previous chapter (c.f. Section 4.4) presented the TSynD approach. TSynD optimizes the sub-symbolic latent space of a generative model to create unknown unknowns in the images. The optimization is guided by maximizing the epistemic uncertainty that the generated image yields when processed by a discriminative model, such as a classification or segmentation network. This concept can be extended to more expressive generative models like stable diffusion. Therefore, an interesting extension of DIDEX is optimizing prompt embeddings to create images that maximize epistemic uncertainty in the semantic segmentation network.

Finally, UDA needs corresponding source domain samples to adapt the segmentation model to challenging aspects of $\mathcal{D}^{\mathrm{PT}}$. Therefore, parts of $\mathcal{D}^{\mathrm{PT}}$ remain unutilized. Employing UDA is necessary due to the semantic inconsistency the generative model

introduces in the image-to-image use cases. The location of the objects in the generated image does not stay consistent with the original image. Therefore, the training can not utilize the label of the source domain image. The DIDEX approach would benefit from research into creating this semantic consistency in diffusion models. The semantic consistency would enable the utilization of the original label and, therefore, the supervised training. Thus, also the most challenging parts of $\mathcal{D}^{\mathrm{PT}}$ could be used for training.

Therefore, future research should focus on creating distributions by combining DIDEX and TSynD and on creating generative models that can create semantically consistent images corresponding to a label. Such a system would allow supervised training on meaningful distributions.

Chapter 6

Summary and Conclusion

Training DNNs that generalize well to novel domains and distributions is challenging. To generalize, DNNs require large training distributions that are good approximations of the desired application domains. As introduced in Chapter 1, the creation of such datasets comes with two main challenges:

1. **Annotations are costly:** This problem arises due to the large amounts of data that need to be labeled and the complex nature of the annotation.

2. **Data acquisition can be challenging and expensive:** Depending on the field of application, the acquisition process of relevant data is costly, or data privacy regulations impede it. In most applications, the acquisition processes yield data that is redundant. Adding value to the training dataset requires annotating data that is novel w.r.t. the existing distribution.

Therefore, this work tackled these challenges through active learning, semi-supervised learning, and synthetic data. Chapters 2, 3, 4 and 5 present approaches for efficient annotation, unsupervised learning, and the targeted generation of synthetic data. Combined, they can be viewed as a framework that minimizes the need for human effort to label and record data.

Chapter 2 presented new best practices for active learning while integrating semi-supervised learning. This work identified the bias of the AL literature toward very specific scenarios that do not represent most real-world scenarios. Therefore, it contributed an analysis exploring the dimensions of low and high acquisition batch sizes, low and high diversity in distributions and the integration of SSL. The chapter showed that batch-based and single-sample acquisition functions are suited to different scenarios and that the effectiveness of integrating semi-supervised learning into AL depends on the type of distribution and the type of acquisition function. This chapter contributed a new benchmark covering more redundant distributions, a scenario understudied by the common benchmarks, to steer AL research in a more realistic direction. Considering problem 1, this chapter contributes to making the selection of samples for labeling more efficient. Additionally, it shows an optimal way of integrating SSL to mitigate human labeling efforts.

Chapter 3 further addressed problem 1 by focusing on unsupervised domain adaptation, which represents a special case of semi-supervised learning. In UDA, the labeled set (the source domain) is structurally different from the unlabeled target domain. This work explored different scenarios, such as real to real domain changes in the environment perception of self-driving cars (e.g., overcoming a delta between different countries or weather conditions) or domain changes between different kinds of medical imaging sensors (OCT gadgets with different features and quality). An important setting was the adaptation from the synthetic to the real world, given that synthetic data has the attractive property of not needing to be labeled. For all these applications, the chapter contributed novel approaches and experiments. Furthermore, this work added a detailed survey that helped to structure the large complexity of this rapidly growing field. The survey's discussion of the field identified research gaps, which the novel approaches of this chapter addressed.

Chapter 4 addressed both problem 1 and 2 by focusing on the targeted creation of synthetic data. Simulated data comes nearly without the need for manual annotation. Additionally, one can parameterize the simulation world to create relevant images. This work introduced a system for generating data representing difficult scenes in a specific real-world target domain. Such synthetic data allows training a model that generalizes well to that specific real-world target domain. The presented acquisition function helps to utilize nondifferentiable simulation engines effectively. The second part of the chapter showed how to utilize DNN-based generative models to create training data for applications where simulation engines are rare. Medical image processing represents such a case. Given the differentiability of these models, the work provided an approach for guiding the generation process toward generating rare but critical scenarios.

Chapter 5 addresses and utilizes ideas and insights of the previous chapters to train models that generalize well to many unseen domains. The work presents an approach that meaningfully augments a given distribution of synthetic data. The chapter presents the DIDEX approach, which utilizes prior symbolic knowledge about important distributions to create a pseudo-target domain containing such data. The approach utilizes the capabilities of modern diffusion models, which are guided by text prompts to achieve this knowledge-based data generation. The augmentation process leads to semantic inconsistencies between the ground truth labels and the generated image, which are overcome by utilizing unsupervised domain adaptation. The chapter shows that the generated data has a similar functional quality as real images, which indicates that many real-world measurement campaigns could be replaced by approaches similar to DIDEX.

Given the contributions of this work in the fields of active learning, semi-supervised learning, and simulation data, several conclusions can be derived that are relevant for the efficient training of models that generalize well:

- AL learning methods (image acquisition functions) in most scenarios allow for a more efficient selection of training data for labeling than random acquisition.
- The choice of the correct acquisition function depends on the distribution of the unlabeled data and the annotation budget. A redundant distribution introduces the need for a batch-based acquisition function, i.e., to score the cumulative information of the selected data. Diverse unlabeled distributions do not have this requirement. Given a diverse distribution and large annotation budgets, it is best to utilize acquisition functions that score the images independently of each other. AL methods often struggle to perform better than random selection for small annotation budgets in diverse distributions. Given a small budget and high levels of redundancy (medical datasets), batch-based acquisition functions should be used.
- The Integration of semi-supervised learning into AL is helpful in reducing the need for manual annotation further. The SSL allows for training on the part of the data pool that has not been annotated yet.
- The way SSL integrates into AL depends on the choice of acquisition function. Redundant datasets favor the integration of batch-based active learning and semi-supervised learning. SSL propagates the knowledge learned from the training set to the unlabeled data. Batch-based acquisition represents the unlabeled data better. Thus, the classifier can generate a supervision signal for the unlabeled data.
- If structural changes exist between the labeled subset (source domain) and the unlabeled subset (target domain), SSL can unsupervised adapt a given DNN to the target domain distribution. Such unsupervised domain adaptation (UDA) methods try to align the distributions of the source and target domain in the input, the feature, or the output space of the DNN.
- Most of the best-performing methods are complex, which introduces the need for validation sets that are often missing in real-world applications. The thesis presents a low-complexity approach that yields almost equal performance to the current state of the art but is more usable for real-world applications due to its low complexity.
- The thesis presents a novel approach for aligning the distributions of the source and target domains in the latent space by unsupervised clustering towards the class centroids of the source domain. The work proved its effectiveness in the driving and medical domains.
- A critical use case for SSL is the adaptation from synthetic data to real-world data since the synthetic data does neither require image recording nor labeling. The adaptation from synthetic data to real-world data should introduce a benefit on the real-world target domain, i.e. introduce an improved generalization.

- Given the constraints on memory and training time, synthetic distributions can not be arbitrary in size. Therefore, this thesis introduces the research question of constructing an acquisition function for synthetic data.
- The work introduced an approach to creating correspondences for failure cases recorded in the real world for the application of autonomous vehicles. Such data can be viewed as known unknowns. The experiments show that such a targeted generation is crucial since irrelevant synthetic data can create a negative bias when the trained model is applied in the real world.
- For the application in medical image processing, simulation engines rarely exist, and the use of generative models is common for generating synthetic data. This work showed how to formulate the generation of synthetic data as an optimization process to create images with high epistemic uncertainty. The epistemic uncertainty indicates whether the generated image is novel w.r.t. the training distribution. This approach does not require explicit descriptions of the generation goal through examples and, therefore, is a possibility to create unknown unknowns.
- Finally, the thesis contributes an approach to leverage modern diffusion models to create training distributions based on symbolic prior knowledge. The presented DIDEX approach creates data for various target domains without actually accessing them. The presented work achieved a new state of the art in the competitive domain generalization benchmark.
- The thesis shows that the data created by the DIDEX approach possesses a similar functional quality as real-world data. This insight opens up future opportunities to replace real-world data recording with the targeted utilization of generative foundation models.

Overall, the approaches and analysis presented in this work facilitate reducing the manual effort for annotation and data acquisition to train DNNs that generalize to new domains and distributions. Given the recent advent of diffusion models that allow for the controlled generation of high-quality data, the potential for future work in creating synthetic data is probably the largest. The approaches introduced in this thesis, DIDEX (c.f. Chapter 5) and TSynD (c.f. Chapter 4), can be further developed to create targeted data that represent missing parts of the distribution. Further development of the TSynD approach seems especially interesting since it creates synthetic data with the same objectives as those used in AL, i.e., high epistemic uncertainty. Applying this concept to the large expressiveness of diffusion models would lead to the creation of meaningful training data.

In conclusion, this work has explored active learning, semi-supervised learning, and synthetic data for adapting and generalizing to new domains and distributions.

References

[1] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al. "A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges". In: *Information Fusion* 76 (2021), pp. 243–297.

[2] M. Amgad et al. "Structured crowdsourcing enables convolutional segmentation of histology images". In: *Bioinformatics* 35.18 (Feb. 2019), pp. 3461–3467.

[3] N. Araslanov and S. Roth. "Self-supervised Augmentation Consistency for Adapting Semantic Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2021, pp. 15379–15389.

[4] F. Barbato, M. Toldo, U. Michieli, and P. Zanuttigh. "Latent Space Regularization for Unsupervised Domain Adaptation in Semantic Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2021, pp. 2835–2845.

[5] S. Benaim, M. Khaitov, T. Galanti, and L. Wolf. "Domain Intersection and Domain Difference". In: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 3445–3453.

[6] Q. Bi, S. You, and T. Gevers. "Learning content-enhanced mask transformer for domain generalized urban-scene segmentation". In: *AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 38. 2. 2024, pp. 819–827.

[7] H. Bogunović et al. "RETOUCH: The Retinal OCT Fluid Detection and Segmentation Benchmark and Challenge". In: *IEEE Transactions on Medical Imaging* 38.8 (2019), pp. 1858–1874.

[8] C. Böker, J. Niemeijer, N. Wojke, C. Meurie, and Y. Cocheril. "A System for Image-Based Non-Line-Of-Sight Detection Using Convolutional Neural Networks". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 535–540.

[9] 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Long Beach, CA, USA, June 2019, pp. 1404–1413.

[10] F. C. Borlino, A. D'Innocente, and T. Tommasi. "Rethinking domain generalization baselines". In: *International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 9227–9233.

[11] G. J. Brostow, J. Fauqueur, and R. Cipolla. "Semantic Object Classes in Video: A High-Definition Ground Truth Database". In: *Pattern Recognition Letters* 30.2 (Jan. 2009), pp. 88–97.

[12] B. Cai, H. Fu, R. Jia, B. Zhao, H. Li, and Y. Xu. "Exploiting Diverse Characteristics and Adversarial Ambivalence for Domain Adaptive Segmentation". In: *AAAI Conference on Artificial Intelligence (AAAI)* 35.8 (May 2021), pp. 6850–6858.

[13] L. Cai, X. Xu, J. H. Liew, and C. S. Foo. "Revisiting Superpixels for Active Learning in Semantic Segmentation With Realistic Annotation Costs". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[14] 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: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA, Jan. 2022, pp. 1129–1139.

[15] O. Chapelle, B. Schölkopf, and A. Zien. "Analysis of Benchmarks". In: *Semi-Supervised Learning*. Ed. by O. Chapelle, B. Schölkopf, and A. Zien. The MIT Press, 2006, pp. 376–393.

[16] 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, June 2019, pp. 1900–1909.

[17] C.-H. Chao, B.-W. Cheng, and C.-Y. Lee. "Rethinking ensemble-distillation for semantic segmentation based unsupervised domain adaption". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 2610–2620.

[18] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. Frank Wang, and M. Sun. "No More Discrimination: Cross City Adaptation of Road Scene Segmenters". In: *International Conference on Computer Vision (ICCV)*. Venice, Italy, Oct. 2017, pp. 1992–2001.

[19] L.-C. 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". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (Apr. 2017), pp. 834–848.

[20] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *European Conference on Computer Vision (ECCV)*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Vol. 11211. Lecture Notes in Computer Science. Springer, 2018, pp. 833–851.

[21] Y. Chen, W. Li, and L. Van Gool. “Road: Reality Oriented Adaptation for Semantic Segmentation of Urban Scenes”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, June 2018, pp. 7892–7901.

[22] M. Chen, H. Xue, and D. Cai. “Domain Adaptation for Semantic Segmentation With Maximum Squares Loss”. In: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 2090–2099.

[23] Y. Chen, W. Li, X. Chen, and L. V. Gool. “Learning Semantic Segmentation From Synthetic Data: A Geometrically Guided Input-Output Adaptation Approach”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, June 2019, pp. 1841–1850.

[24] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang. “CrDoCo: Pixel-Level Domain Transfer With Cross-Domain Consistency”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, June 2019, pp. 1791–1800.

[25] T. Chen, J. Zhang, G.-S. Xie, Y. Yao, X. Huang, and Z. Tang. “Classification Constrained Discriminator for Domain Adaptive Semantic Segmentation”. In: *IEEE International Conference on Multimedia & Expo (ICME)*. virtual, July 2020, pp. 1–6.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the International Conference on Machine Learning (ICML)*. virtual, July 2020, pp. 1597–1607.

[27] H. Chen, C. Wu, Y. Xu, and B. Du. “Unsupervised Domain Adaptation for Semantic Segmentation via Low-level Edge Information Transfer”. In: *CoRR* abs/2109.08912 (2021). arXiv: 2109.08912.

[28] Y. Cheng, F. Wei, J. Bao, D. Chen, F. Wen, and W. Zhang. “Dual Path Learning for Domain Adaptation of Semantic Segmentation”. In: *International Conference on Computer Vision (ICCV)*. virtual, Oct. 2021, pp. 9082–9091.

[29] E. Chiou, E. Panagiotaki, and I. Kokkinos. “Beyond Deterministic Translation for Unsupervised Domain Adaptation”. In: *British Machine Vision Conference (BMVC)*. BMVA Press, 2022, p. 501.

[30] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, June 2018, pp. 8789–8797.

[31] J. Choi, T. Kim, and C. Kim. "Self-Ensembling With GAN-Based Data Augmentation for Domain Adaptation in Semantic Segmentation". In: *International Conference on Computer Vision (ICCV)*. Oct. 2019, pp. 6830–6840.

[32] 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. virtual, June 2021, pp. 11580–11590.

[33] I. Chung, D. Kim, and N. Kwak. "Maximizing cosine similarity between spatial features for unsupervised domain adaptation in semantic segmentation". In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2022, pp. 1351–1360.

[34] I. Chung, J. Yoo, and N. Kwak. "Exploiting inter-pixel correlations in unsupervised domain adaptation for semantic segmentation". In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2023, pp. 12–21.

[35] 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: *Asian Conference on Computer Vision (ACCV)*. Kyoto, Japan, Dec. 2020.

[36] 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA, June 2016, pp. 3213–3223.

[37] G. Csurka, R. Volpi, and B. Chidlovskii. "Unsupervised Domain Adaptation for Semantic Image Segmentation: a Comprehensive Survey". In: *CoRR* abs/2112.03241 (2021). arXiv: `2112.03241`.

[38] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. "ImageNet: A large-scale hierarchical image database". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2009, pp. 248–255.

[39] P. Dhariwal and A. Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems (NeurIPS)*. virtual, Dec. 2021, pp. 8780–8794.

[40] J. Dong, Y. Cong, G. Sun, and D. Hou. "Semantic-Transferable Weakly-Supervised Endoscopic Lesions Segmentation". In: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 10712–10721.

[41] J. Dong, Y. Cong, G. Sun, Y. Liu, and X. Xu. “CSCL: Critical Semantic-Consistent Learning for Unsupervised Domain Adaptation”. In: *European Conference on Computer Vision (ECCV)*. virtual, Aug. 2020, pp. 745–762.

[42] 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”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.3 (2021), pp. 1664–1681.

[43] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. “CARLA: An Open Urban Driving Simulator”. In: *Conference on Robot Learning (CoRL)*. Mountain View, CA, USA, Nov. 2017, pp. 1–16.

[44] 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 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021.

[45] T. Drugman, J. Pylkkönen, and R. Kneser. “Active and Semi-Supervised Learning in ASR: Benefits on the Acoustic and Language Models”. In: *Annual Conference of the International Speech Communication Association (Interspeech)*. Ed. by N. Morgan. ISCA, 2016, pp. 2318–2322.

[46] 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: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 982–991.

[47] M.-P. Dubuisson and A. K. Jain. “A modified Hausdorff distance for object matching”. In: *International Conference on Pattern Recognition (ICPR)*. Vol. 1. IEEE. 1994, pp. 566–568.

[48] A. Dundar, M. Liu, T. Wang, J. Zedlewski, and J. Kautz. “Domain Stylization: A Strong, Simple Baseline for Synthetic to Real Image Domain Adaptation”. In: *CoRR* abs/1807.09384 (2018). arXiv: 1807.09384.

[49] S. Engelson, J. Ehrhardt, T. Kepp, J. Niemeijer, and H. Handels. “LNQ Challenge 2023: Learning Mediastinal Lymph Node Segmentation with a Probabilistic Lymph Node Atlas”. In: *Machine Learning for Biomedical Imaging* 2 (MICCAI 2023 LNQ challenge special issue 2024), pp. 817–833.

[50] S. Engelson, J. Ehrhardt, T. Kepp, J. Niemeijer, S. Schierholz, L. Berkel, Y. Elser, M. M. Sieren, and H. Handels. “Comparison of anatomical priors for learning-based neural network guidance for mediastinal lymph node segmentation”. In: *Medical Imaging 2024: Computer-Aided Diagnosis*. Ed. by W. Chen

and S. M. Astley. Vol. 12927. International Society for Optics and Photonics. SPIE, 2024, 129271K.

[51] P. Esser, R. Rombach, and B. Ommer. “Taming transformers for high-resolution image synthesis”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 12873–12883.

[52] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. “The Pascal Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision (IJCV)* 88.2 (2010), pp. 303–338.

[53] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proceedings of the International Conference on Machine Learning (ICML)*. Ed. by M. Balcan and K. Q. Weinberger. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 1050–1059.

[54] Y. Gal, R. Islam, and Z. Ghahramani. “Deep bayesian active learning with image data”. In: *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR. 2017, pp. 1183–1192.

[55] Y. Ganin and V. Lempitsky. “Unsupervised Domain Adaptation by Backpropagation”. In: *Proceedings of the International Conference on Machine Learning (ICML)*. Lille, France, July 2015, pp. 1180–1189.

[56] M. Gao, Z. Zhang, G. Yu, S. Ö. Arik, L. S. Davis, and T. Pfister. “Consistency-Based Semi-supervised Active Learning: Towards Minimizing Labeling Cost”. In: *European Conference on Computer Vision (ECCV)*. Ed. by A. Vedaldi, H. Bischof, T. Brox, and J. Frahm. Vol. 12355. Lecture Notes in Computer Science. Springer, 2020, pp. 510–526.

[57] L. Gao, J. Zhang, L. Zhang, and D. Tao. “DSP: Dual Soft-Paste for Unsupervised Domain Adaptive Semantic Segmentation”. In: *IEEE International Conference on Multimedia & Expo (ICME)*. Chengdu, China, Oct. 2021, pp. 2825–2833.

[58] L. Gao, L. Zhang, and Q. Zhang. “Addressing Domain Gap via Content Invariant Representation for Semantic Segmentation”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 35. 9. 2021, pp. 7528–7536.

[59] L. A. Gatys, A. S. Ecker, and M. Bethge. “A Neural Algorithm of Artistic Style”. In: *CoRR* abs/1508.06576 (2015). arXiv: `1508.06576`.

[60] L. A. Gatys, A. S. Ecker, and M. Bethge. “Image Style Transfer Using Convolutional Neural Networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA, June 2016, pp. 2414–2423.

[61] 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”. In: *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019.

[62] 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. “A2D2: Audi Autonomous Driving Dataset”. In: *CoRR* abs/2004.06320 (2020). arXiv: 2004.06320.

[63] V. Gkitsas, A. Karakottas, N. Zioulis, D. Zarpalas, and P. Daras. “Restyling Data: Application to Unsupervised Domain Adaptation”. In: *CoRR* abs/1909.10900 (2019). arXiv: 1909.10900.

[64] S. A. Golestaneh and K. Kitani. “Importance of Self-Consistency in Active Learning for Semantic Segmentation”. In: *British Machine Vision Conference (BMVC)*. BMVA Press, 2020.

[65] R. Gong, W. Li, Y. Chen, and L. V. Gool. “DLOW: Domain Flow for Adaptation and Generalization”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, June 2019, pp. 2477–2486.

[66] R. Gong, M. Danelljan, H. Sun, J. D. Mangas, and L. V. Gool. “Prompting Diffusion Representations for Cross-Domain Semantic Segmentation”. In: *CoRR* abs/2307.02138 (2023). arXiv: 2307.02138.

[67] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems (NIPS)*. Montréal, Canada, Dec. 2014, pp. 2672–2680.

[68] I. J. Goodfellow, J. Shlens, and C. Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *International Conference on Learning Representations (ICLR)*. Ed. by Y. Bengio and Y. LeCun. 2015.

[69] D. Guan, J. Huang, S. Lu, and A. Xiao. “Scale Variance Minimization for Unsupervised Domain Adaptation in Image Segmentation”. In: *Pattern Recognition* 112 (2021), p. 107764.

[70] V. Guizilini, J. Li, R. Ambruș, and A. Gaidon. “Geometric Unsupervised Domain Adaptation for Semantic Segmentation”. In: *International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 8537–8547.

[71] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA, June 2016, pp. 770–778.

[72] J. Ho, A. Jain, and P. Abbeel. “Denoising Diffusion Probabilistic Models”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. virtual, Dec. 2020, pp. 6840–6851.

[73] J. Hoffman, D. Wang, F. Yu, and T. Darrell. “FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation”. In: *CoRR* abs/1612.02649 (2016). arXiv: 1612.02649.

[74] 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: *Proceedings of the International Conference on Machine Learning (ICML)*. Stockholm, Sweden, July 2018, pp. 1989–1998.

[75] W. Hong, Z. Wang, M. Yang, and J. Yuan. “Conditional Generative Adversarial Network for Structured Domain Adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, June 2018, pp. 1335–1344.

[76] N. Houlsby, F. Huszar, Z. Ghahramani, and M. Lengyel. “Bayesian Active Learning for Classification and Preference Learning”. In: *CoRR* abs/1112.5745 (2011). arXiv: 1112.5745.

[77] L. Hoyer, D. Dai, and L. Van Gool. “DAFormer: Improving Network Architectures and Training Strategies For Domain-Adaptive Semantic Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA, June 2022, pp. 9924–9935.

[78] L. Hoyer, D. Dai, and L. Van Gool. “HRDA: Context-Aware High-Resolution Domain-Adaptive Semantic Segmentation”. In: *European Conference on Computer Vision (ECCV)*. Ed. by S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner. Cham: Springer Nature Switzerland, 2022, pp. 372–391.

[79] L. Hoyer, D. Dai, and L. Van Gool. “Domain adaptive and generalizable network architectures and training strategies for semantic image segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.1 (2023), pp. 220–235.

[80] L. Hoyer, D. Dai, H. Wang, and L. Van Gool. “MIC: Masked image consistency for context-enhanced domain adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 11721–11732.

[81] D. Hu, J. Liang, Q. Hou, H. Yan, and Y. Chen. “Adversarial Domain Adaptation With Prototype-Based Normalized Output Conditioner”. In: *IEEE Transactions on Image Processing* 30 (2021), pp. 9359–9371.

[82] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. “Densely Connected Convolutional Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[83] X. Huang and S. Belongie. “Arbitrary Style Transfer in Real-Time With Adaptive Instance Normalization”. In: *International Conference on Computer Vision (ICCV)*. Venice, Italy, Oct. 2017, pp. 1501–1510.

[84] H. Huang, Q. Huang, and P. Krahenbuhl. “Domain Tansfer Through Deep Activation Matching”. In: *European Conference on Computer Vision (ECCV)*. Munich, Germany, Sept. 2018, pp. 590–605.

[85] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. “Multimodal Unsupervised Image-to-Image Translation”. In: *European Conference on Computer Vision (ECCV)*. Munich, Germany, Sept. 2018, pp. 1–18.

[86] J. Huang, S. Lu, D. Guan, and X. Zhang. “Contextual-Relation Consistent Domain Adaptation for Semantic Segmentation”. In: *European Conference on Computer Vision (ECCV)*. Glasow, UK, Aug. 2020, pp. 705–722.

[87] J. Huang, D. Guan, A. Xiao, and S. Lu. “Cross-View Regularization for Domain Adaptive Panoptic Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 10133–10144.

[88] J. Huang, D. Guan, A. Xiao, and S. Lu. “FSDR: Frequency Space Domain Randomization for Domain Generalization”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. virtual, June 2021, pp. 6891–6902.

[89] J. Huang, D. Guan, A. Xiao, and S. Lu. “RDA: Robust Domain Adaptation via Fourier Adversarial Attacking”. In: *International Conference on Computer Vision (ICCV)*. virtual, Oct. 2021, pp. 8988–8999.

[90] J. Huang, D. Guan, A. Xiao, and S. Lu. “Multi-level adversarial network for domain adaptive semantic segmentation”. In: *Pattern Recognition* 123 (2022), pp. 108–384.

[91] J. Huang, D. Guan, A. Xiao, S. Lu, and L. Shao. “Category Contrast for Unsupervised Domain Adaptation in Visual Tasks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 1203–1214.

[92] X. Huo, L. Xie, H. Hu, W. Zhou, H. Li, and Q. Tian. “Domain-Agnostic Prior for Transfer Semantic Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 7075–7085.

[93] S. Ioffe. “Batch Renormalization: Towards Reducing Minibatch Dependence in Batch-Normalized Models”. In: *Advances in Neural Information Processing Systems (NIPS)*. Long Beach, CA, USA, Dec. 2017, pp. 1945–1953.

[94] J. Iqbal and M. Ali. “MLSL: Multi-Level Self-Supervised Learning for Domain Adaptation With Spatially Independent and Semantically Consistent Labeling”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Aspen, CO, USA, Mar. 2020, pp. 1864–1873.

[95] K. Jahan, J. Niemeijer, N. Kornfeld, and M. Roth. “Deep Neural Networks for Railway Switch Detection and Classification Using Onboard Camera Images”. In: *IEEE Symposium Series on Computational Intelligence (SSCI)*. 2021, pp. 01–07.

[96] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. J. Spanos. “Towards Efficient Data Valuation Based on the Shapley Value”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Ed. by K. Chaudhuri and M. Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1167–1176.

[97] Justin Johnson and Alexandre Alahi and Li Fei-Fei. “Perceptual Losses for Real-Time Style Transfer and Super-Resolution”. In: *European Conference on Computer Vision (ECCV)*. Amsterdam, Netherlands, Oct. 2016, pp. 694–711.

[98] G. Kang, Y. Wei, Y. Yang, Y. Zhuang, and A. Hauptmann. “Pixel-Level Cycle Association: A New Perspective for Domain Adaptive Semantic Segmentation”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 3569–3580.

[99] S. Karlos, C. Aridas, V. G. Kanas, and S. Kotsiantis. “Classification of acoustical signals by combining active learning strategies with semi-supervised learning schemes”. In: *Neural Computing and Applications* (2021), pp. 1–18.

[100] A. Kendall and Y. Gal. “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” In: *Advances in Neural Information Processing Systems (NIPS)*. Long Beach, CA, USA, Dec. 2017, pp. 5574–5584.

[101] K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, and R. K. Iyer. “GLISTER: Generalization based Data Subset Selection for Efficient and Robust Learning”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2021, pp. 8110–8118.

[102] M. Kim and H. Byun. “Learning Texture Invariant Representation for Domain Adaptation of Semantic Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, June 2020, pp. 12975–12984.

[103] K. Kim, D. Park, K. I. Kim, and S. Y. Chun. “Task-Aware Variational Adversarial Active Learning”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 2021, pp. 8166–8175.

[104] M. Kim, S. Joung, S. Kim, J. Park, I.-J. Kim, and K. Sohn. “Cross-Domain Grouping and Alignment for Domain Adaptive Semantic Segmentation”. In: *AAAI Conference on Artificial Intelligence (AAAI)* 35.3 (May 2021), pp. 1799–1807.

[105] D. Kim, M. Seo, J. Park, and D. Choi. “Source Domain Subset Sampling for Semi-Supervised Domain Adaptation in Semantic Segmentation”. In: *CoRR* abs/2205.00312 (2022). arXiv: 2205.00312.

[106] N. Kim, T. Son, J. Pahk, C. Lan, W. Zeng, and S. Kwak. “WEDGE: Web-Image Assisted Domain Generalization for Semantic Segmentation”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 9281–9288.

[107] A. Kirsch, J. van Amersfoort, and Y. Gal. “BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett. 2019, pp. 7024–7035.

[108] M. Klingner, M. Ayache, and T. Fingscheidt. “Continual BatchNorm Adaptation (CBNA) for Semantic Segmentation”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.11 (2022), pp. 20899–20911.

[109] M. Klingner, J.-A. Termöhlen, J. Ritterbach, and T. Fingscheidt. “Unsupervised BatchNorm Adaptation (UBNA): A Domain Adaptation Method for Semantic Segmentation Without Using Source Domain Representations”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACVW)*. Waikoloa, HI, USA, Jan. 2022, pp. 210–220.

[110] W. M. Kouw. “An introduction to domain adaptation and transfer learning”. In: *CoRR* abs/1812.11806 (2018). arXiv: 1812.11806.

[111] K. Kowol, S. Bracke, and H. Gottschalk. “A-Eye: Driving with the Eyes of AI for Corner Case Generation”. In: *International Conference on Computer-Human Interaction Research and Applications (CHIRA)*. Ed. by H. P. da Silva, J. Vanderdonckt, A. Holzinger, and L. L. Constantine. SCITEPRESS, 2022, pp. 41–48.

[112] B. Lakshminarayanan, A. Pritzel, and C. Blundell. “Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles”. In: *Advances in Neural Information Processing Systems (NIPS)*. Long Beach, CA, USA, Dec. 2017, pp. 6402–6413.

[113] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht. "Sliced Wasserstein Discrepancy for Unsupervised Domain Adaptation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, June 2019, pp. 10285–10295.

[114] K.-H. Lee, G. Ros, J. Li, and A. Gaidon. "SPIGAN: Privileged Adversarial Learning From Simulation". In: *International Conference on Learning Representations (ICLR)*. New Orleans, LA, USA, Apr. 2019, pp. 1–14.

[115] S. Lee, S. Cho, and S. Im. "DRANet: Disentangling Representation and Adaptation Networks for Unsupervised Cross-Domain Adaptation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. virtual, June 2021, pp. 15252–15261.

[116] S. Lee, J. Hyun, H. Seong, and E. Kim. "Unsupervised Domain Adaptation for Semantic Segmentation by Content Transfer". In: *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2021, pp. 8306–8315.

[117] S. Lee, H. Seong, S. Lee, and E. Kim. "WildNet: Learning Domain Generalized Semantic Segmentation From the Wild". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA, June 2022, pp. 9936–9946.

[118] A. Leich, J. Fuchs, G. Srinivas, J. Niemeijer, and P. Wagner. "Traffic Safety at German Roundabouts—A Replication Study". In: *Safety* 8.3 (2022).

[119] A. Leich, N. Kornfeld, J. Niemeijer, M. Kaiser, and M. Jäckle. "Erkennung von Rissen mittels maschinellen Lernens". In: *EI-Der Eisenbahningenieur* (2023), pp. 38–43.

[120] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. "Universal Style Transfer via Feature Transforms". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Long Beach, CA, USA, Dec. 2017, pp. 386–396.

[121] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu. "Adaptive Batch Normalization for Practical Domain Adaptation". In: *Pattern Recognition* 80 (Aug. 2018), pp. 109–117.

[122] Y. Li, M. Liu, X. Li, M. Yang, and J. Kautz. "A Closed-Form Solution to Photorealistic Image Stylization". In: *European Conference on Computer Vision (ECCV)*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Vol. 11207. Lecture Notes in Computer Science. Springer, 2018, pp. 468–483.

[123] Y. Li, L. Yuan, and N. Vasconcelos. "Bidirectional Learning for Domain Adaptation of Semantic Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, June 2019, pp. 6936–6945.

[124] C. Li, D. Du, L. Zhang, L. Wen, T. Luo, Y. Wu, and P. Zhu. “Spatial Attention Pyramid Network for Unsupervised Domain Adaptation”. In: *European Conference on Computer Vision (ECCV)*. virtual, Aug. 2020, pp. 481–497.

[125] G. Li, G. Kang, W. Liu, Y. Wei, and Y. Yang. “Content-Consistent Matching for Domain Adaptive Semantic Segmentation”. In: *European Conference on Computer Vision (ECCV)*. virtual, Aug. 2020, pp. 440–456.

[126] R. Li, W. Cao, Q. Jiao, S. Wu, and H.-S. Wong. “Simplified Unsupervised Image Translation for Semantic Segmentation Adaptation”. In: *Pattern Recognition* 105 (2020), p. 107343.

[127] R. Li, W. Cao, S. Wu, and H.-S. Wong. “Generating Target Image-Label Pairs for Unsupervised Domain Adaptation”. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 7997–8011.

[128] Z. Li, R. Togo, T. Ogawa, and M. Haseyama. “Unsupervised Domain Adaptation for Semantic Segmentation With Symmetric Adaptation Consistency”. In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Barcelona, Spain, May 2020, pp. 2263–2267.

[129] Z. Li, R. Togo, T. Ogawa, and M. Haseyama. “Variational Autoencoder Based Unsupervised Domain Adaptation for Semantic Segmentation”. In: *IEEE International Conference on Image Processing (ICIP)*. Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 2426–2430.

[130] P. Li, D. Li, W. Li, S. Gong, Y. Fu, and T. M. Hospedales. “A simple feature augmentation for domain generalization”. In: *International Conference on Computer Vision (ICCV)*. 2021, pp. 8886–8895.

[131] S. Li, F. Lv, B. Xie, C. H. Liu, J. Liang, and C. Qin. “Bi-classifier determinacy maximization for unsupervised domain adaptation”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 35. 10. 2021, pp. 8455–8464.

[132] S. Li, B. Xie, B. Zang, C. H. Liu, X. Cheng, R. Yang, and G. Wang. “Semantic Distribution-aware Contrastive Adaptation for Semantic Segmentation”. In: *CoRR* abs/2105.05013 (2021). arXiv: 2105.05013.

[133] Q. Lian, F. Lv, L. Duan, and B. Gong. “Constructing Self-Motivated Pyramid Curriculums for Cross-Domain Semantic Segmentation: A Non-Adversarial Approach”. In: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 6758–6767.

[134] Y.-X. Lin, D. S. Tan, W.-H. Cheng, and K.-L. Hua. “Adapting Semantic Segmentation of Urban Scenes via Mask-Aware Gated Discriminator”. In: *IEEE International Conference on Multimedia & Expo (ICME)*. Shanghai, China, July 2019, pp. 218–223.

[135] H. Linander, O. Balabanov, H. Yang, and B. Mehlig. “Looking at the posterior: accuracy and uncertainty of neural-network predictions”. In: *Machine Learning: Science and Technology* 4.4 (2023), p. 45032.

[136] W. Liu, D. Ferstl, S. Schulter, L. Zebedin, P. Fua, and C. Leistner. “Domain Adaptation for Semantic Segmentation via Patch-Wise Contrastive Learning”. In: *CoRR* abs/2104.11056 (2021). arXiv: 2104.11056.

[137] 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: *International Conference on Computer Vision (ICCV)*. virtual, Oct. 2021, pp. 10367–10376.

[138] Y. Liu, J. Deng, X. Gao, W. Li, and L. Duan. “BAPA-Net: Boundary Adaptation and Prototype Alignment for Cross-Domain Semantic Segmentation”. In: *International Conference on Computer Vision (ICCV)*. virtual, Oct. 2021, pp. 8801–8811.

[139] Y. Liu, W. Zhang, and J. Wang. “Source-Free Domain Adaptation for Semantic Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. virtual, June 2021, pp. 1215–1224.

[140] Z. Lu, Y. Yang, X. Zhu, C. Liu, Y.-Z. Song, and T. Xiang. “Stochastic classifiers for unsupervised domain adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 9111–9120.

[141] 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”. In: *ACM International Conference on Multimedia (ACM MM)*. MM ’22. Lisboa, Portugal: Association for Computing Machinery, 2022, pp. 1405–1415.

[142] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang. “Significance-Aware Information Bottleneck for Domain Adaptive Semantic Segmentation”. In: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 6778–6787.

[143] 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, June 2019, pp. 2507–2516.

[144] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang. “Adversarial Style Mining for One-Shot Unsupervised Domain Adaptation”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. virtual, Dec. 2020, pp. 20612–20623.

[145] Y. Luo, Z. Wang, D. Huang, N. Ge, and J. Lu. “Get away from Style: Category-Guided Domain Adaptation for Semantic Segmentation”. In: *CoRR* abs/2103.15467 (2021). arXiv: 2103.15467.

[146] F. Lv, T. Liang, X. Chen, and G. Lin. “Cross-Domain Semantic Segmentation via Domain-Invariant Interactive Relation Transfer”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, June 2020, pp. 4334–4343.

[147] H. Ma, X. Lin, Z. Wu, and Y. Yu. “Coarse-to-Fine Domain Adaptive Semantic Segmentation With Photometric Alignment and Category-Center Regularization”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 4051–4060.

[148] L. van der Maaten and G. Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[149] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. “Least Squares Generative Adversarial Networks”. In: *International Conference on Computer Vision (ICCV)*. Venice, Italy, Oct. 2017, pp. 2794–2802.

[150] R. A. Marsden, A. Bartler, M. Döbler, and B. Yang. “Contrastive Learning and Self-Training for Unsupervised Domain Adaptation in Semantic Segmentation”. In: *International Joint Conference on Neural Networks (IJCNN)*. 2022, pp. 1–8.

[151] K. Mei, C. Zhu, J. Zou, and S. Zhang. “Instance Adaptive Self-training for Unsupervised Domain Adaptation”. In: *European Conference on Computer Vision (ECCV)*. Ed. by A. Vedaldi, H. Bischof, T. Brox, and J. Frahm. Vol. 12371. Lecture Notes in Computer Science. Springer, 2020, pp. 415–430.

[152] L. Melas-Kyriazi and A. K. Manrai. “PixMatch: Unsupervised Domain Adaptation via Pixelwise Consistency Training”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. virtual, June 2021, pp. 12435–12445.

[153] U. Michieli, M. Biasetton, G. Agresti, and P. Zanuttigh. “Adversarial Learning and Self-Teaching Techniques for Domain Adaptation in Semantic Segmentation”. In: *IEEE Transactions on Intelligent Vehicles* 5.3 (2020), pp. 508–518.

[154] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems (NIPS)*. Ed. by C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger. 2013, pp. 3111–3119.

[155] M. Mirza and S. Osindero. “Conditional Generative Adversarial Nets”. In: *CoRR* abs/1411.1784 (2014). arXiv: 1411.1784.

[156] S. Mittal, M. Tatarchenko, and T. Brox. “Semi-supervised semantic segmentation with high-and low-level consistency”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.4 (2019), pp. 1369–1379.

[157] S. Mittal, M. Tatarchenko, Ö. Çiçek, and T. Brox. “Parting with Illusions about Deep Active Learning”. In: *CoRR* abs/1912.05361 (2019). arXiv: 1912.05361.

[158] S. Mittal, J. Niemeijer, J. P. Schäfer, and T. Brox. “Best Practices in Active Learning for Semantic Segmentation”. In: *German Conference on Pattern Recognition (GCPR)*. Ed. by U. Köthe and C. Rother. Vol. 14264. Lecture Notes in Computer Science. Springer, 2023, pp. 427–442.

[159] S. Mittal, J. Niemeijer, O. Cicek, M. Tatarchenko, J. Ehrhardt, J. P. Schaefer, H. Handels, and T. Brox. “Realistic Evaluation of Deep Active Learning for Image Classification and Semantic Segmentation”. In: *International Journal of Computer Vision (IJCV)* (Feb. 2025), pp. 1–23.

[160] M. B. Muhammad and M. Yeasin. “Eigen-cam: Class activation map using principal components”. In: *International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–7.

[161] P. Munjal, N. Hayat, M. Hayat, J. Sourati, and S. Khan. “Towards Robust and Reproducible Active Learning using Neural Networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 223–232.

[162] L. Musto and A. Zinelli. “Semantically Adaptive Image-to-image Translation for Domain Adaptation of Semantic Segmentation”. In: *British Machine Vision Conference (BMVC)*. BMVA Press, 2020.

[163] M. M. Naseer, K. Ranasinghe, S. H. Khan, M. Hayat, F. Shahbaz Khan, and M.-H. Yang. “Intriguing Properties of Vision Transformers”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. virtual, Dec. 2021, pp. 23296–23308.

[164] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder. “The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes”. In: *International Conference on Computer Vision (ICCV)*. Venice, Italy, Oct. 2017, pp. 4990–4999.

[165] J. Niemeijer and J. P. Schäfer. “Combining Semantic Self-Supervision and Self-Training for Domain Adaptation in Semantic Segmentation”. In: *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. 2021, pp. 364–371.

[166] J. Niemeijer and J. P. Schäfer. “Domain Adaptation and Generalization: A Low-Complexity Approach”. In: *Conference on Robot Learning (CoRL)*. Ed. by K. Liu, D. Kulic, and J. Ichnowski. Vol. 205. Proceedings of Machine Learning Research. PMLR, 14–18 Dec 2022, pp. 1081–1091.

[167] J. Niemeijer, F. Battistella, G. Srinivas, and A. Leich. “An Approach for Fusing Two Training-Datasets with Partially Overlapping Classes”. In: *IEEE International Conference on Semantic Computing (ICSC)*. Laguna Hills, CA, USA: IEEE, Feb. 2023, pp. 73–79.

[168] J. Niemeijer, J. Ehrhardt, T. Kepp, J. P. Schäfer, and H. Handels. “Overcoming the sensor delta for semantic segmentation in OCT images”. In: *Medical Imaging 2023: Computer-Aided Diagnosis*. Ed. by K. M. Iftekharuddin and W. Chen. Vol. 12465. SPIE Proceedings. SPIE, 2023.

[169] J. Niemeijer, S. Mittal, and T. Brox. “Synthetic Dataset Acquisition for a Specific Target Domain”. In: *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. Paris, France: IEEE, Oct. 2023, pp. 4057–4066.

[170] J. Niemeijer, J. Ehrhardt, H. Uzunova, and H. Handels. “TSynD: Targeted Synthetic Data Generation for Enhanced Medical Image Classification”. In: *Proc. of MICCAI Workshops - International Workshop on Simulation and Synthesis in Medical Imaging*. Marrakesch, Morocco, Oct. 2024, pp. 69–78.

[171] J. Niemeijer, M. Schwonberg, J.-A. Termöhlen, N. M. Schmidt, and T. Fingscheidt. “Generalization by Adaptation: Diffusion-Based Domain Extension for Domain-Generalized Semantic Segmentation”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2024, pp. 2830–2840.

[172] J. Niemeijer, J. Ehrhardt, H. Uzunova, and H. Handels. “Abstract: TSynD Targeted Synthetic Data Generation for Enhanced Medical Image Classification”. In: *German Conference on Medical Image Computing (BVM)*. Ed. by C. Palm, K. Breininger, T. Deserno, H. Handels, A. Maier, K. H. Maier-Hein, and T. M. Tolxdorff. Wiesbaden: Springer Fachmedien Wiesbaden, 2025, pp. 157–157.

[173] V. Olsson, W. Tranheden, J. Pinto, and L. Svensson. “ClassMix: Segmentation-Based Data Augmentation for Semi-Supervised Learning”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA, Jan. 2021, pp. 1369–1378.

[174] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. “Neural discrete representation learning”. In: *Advances in Neural Information Processing Systems (NIPS)*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6309–6318.

[175] X. Pan, P. Luo, J. Shi, and X. Tang. “Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net”. In: *European Conference on Computer Vision (ECCV)*. Munich, Germany, Sept. 2018, pp. 464–479.

[176] X. Pan, X. Zhan, J. Shi, X. Tang, and P. Luo. "Switchable Whitening for Deep Representation Learning". In: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 1863–1871.

[177] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon. "Unsupervised Intra-Domain Adaptation for Semantic Segmentation Through Self-Supervision". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, June 2020, pp. 3764–3773.

[178] S. Paul, Y.-H. Tsai, S. Schulter, A. K. Roy-Chowdhury, and M. Chandraker. "Domain adaptive semantic segmentation using weak labels". In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 571–587.

[179] D. Peng, Y. Lei, L. Liu, P. Zhang, and J. Liu. "Global and Local Texture Randomization for Synthetic-to-Real Semantic Segmentation". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 6594–6608.

[180] D. Peng, Y. Lei, M. Hayat, Y. Guo, and W. Li. "Semantic-Aware Domain Generalized Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA, June 2022, pp. 2594–2605.

[181] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. T. H. Romeny, and J. B. Zimmerman. "Adaptive Histogram Equalization and Its Variations". In: *Computer Vision, Graphics, and Image Processing* 39.3 (Sept. 1987), pp. 355–368.

[182] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis". In: *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2024.

[183] F. Qiao, L. Zhao, and X. Peng. "Learning to learn single domain generalization". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12556–12565.

[184] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy. "Do Vision Transformers See Like Convolutional Neural Networks?" In: *Advances in Neural Information Processing Systems (NeurIPS)*. virtual, Dec. 2021, pp. 12116–12128.

[185] P. Z. Ramirez, A. Tonioni, and L. D. Stefano. "Exploiting semantics in adversarial training for image-level domain adaptation". In: *IEEE International Conference on Image Processing, Applications and Systems (IPAS)*. IEEE, 2018, pp. 49–54.

[186] A. Rangnekar, C. Kanan, and M. Hoffman. “Semantic Segmentation With Active Semi-Supervised Learning”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2023, pp. 5966–5977.

[187] P. K. Rhee, E. Erdenee, S. D. Kyun, M. U. Ahmed, and S. Jin. “Active and semi-supervised learning for object detection with imperfect data”. In: *Cognitive Systems Research* 45 (2017), pp. 109–123.

[188] S. Richter, V. Vineet, S. Roth, and V. Koltun. “Playing for Data: Ground Truth From Computer Games”. In: *European Conference on Computer Vision (ECCV)*. Amsterdam, Netherlands, Oct. 2016, pp. 102–118.

[189] R. Romijnders, P. Meletis, and G. Dubbelman. “A Domain Agnostic Normalization Layer for Unsupervised Adversarial Domain Adaptation”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, Hawaii, Jan. 2019, pp. 1866–1875.

[190] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. “High- Resolution Image Synthesis with Latent Diffusion Models”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA, June 2022, pp. 10684–10695.

[191] 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA, June 2016, pp. 3234–3243.

[192] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed. “Variational Approaches for Auto-Encoding Generative Adversarial Networks”. In: *CoRR* abs/1706.04987 (2017). arXiv: 1706.04987.

[193] C. Ruan, W. Wang, H. Hu, and D. Chen. “Category-Level Adversaries for Semantic Domain Adaptation”. In: *IEEE Access* 7 (2019), pp. 83198–83208.

[194] 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. virtual, July 2021, pp. 8197–8207.

[195] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. “Maximum Classifier Discrepancy for Unsupervised Domain Adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, June 2018, pp. 3723–3732.

[196] K. Saito, D. Kim, S. Sclaroff, and K. Saenko. “Universal Domain Adaptation through Self Supervision”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 16282–16292.

[197] C. Sakaridis, D. Dai, and L. Van Gool. "ACDC: The Adverse Conditions Dataset With Correspondences for Semantic Driving Scene Understanding". In: *International Conference on Computer Vision (ICCV)*. virtual, Oct. 2021, pp. 10765–10775.

[198] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa. "Learning From Synthetic Data: Addressing Domain Shift for Semantic Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, June 2018, pp. 3752–3761.

[199] A. Saporta, T. Vu, M. Cord, and P. Pérez. "ESL: Entropy-guided Self-supervised Learning for Domain Adaptation in Semantic Segmentation". In: *CoRR* abs/2006.08658 (2020). arXiv: 2006.08658.

[200] M. Schwonberg, F. El Bouazati, N. M. Schmidt, and H. Gottschalk. "Augmentation Based Domain Generalization for Semantic Segmentation". In: *IEEE Intelligent Vehicles Symposium Workshops (IVW)*. Anchorage, AK, USA, June 2023, pp. 1–8.

[201] M. Schwonberg, J. Niemeijer, J.-A. Termöhlen, J. P. Schäfer, N. M. Schmidt, H. Gottschalk, and T. Fingscheidt. "Survey on Unsupervised Domain Adaptation for Semantic Segmentation for Visual Perception in Automated Driving". In: *IEEE Access* 11 (May 2023), pp. 54296–54336.

[202] M. S. Seibel, J. Niemeijer, M. Rowedder, H. Sudkamp, T. Kepp, G. Hüttmann, and H. Handels. "Reducing the impact of domain shift in deep learning for OCT segmentation using image manipulations". In: *Medical Imaging 2024: Computer-Aided Diagnosis*. Ed. by W. Chen and S. M. Astley. Vol. 12927. International Society for Optics and Photonics. SPIE, 2024, p. 1292719.

[203] O. Sener and S. Savarese. "Active Learning for Convolutional Neural Networks: A Core-Set Approach". In: *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2018.

[204] Y. Shan, C. M. Chew, and W. F. Lu. "Semantic-Aware Short Path Adversarial Training for Cross-Domain Semantic Segmentation". In: *Neurocomputing* 380 (2020), pp. 125–132.

[205] C. E. Shannon. "A mathematical theory of communication". In: *Bell Syst. Tech. J.* 27.3 (1948), pp. 379–423.

[206] T. Shen, D. Gong, W. Zhang, C. Shen, and T. Mei. "Regularizing Proxies with Multi-Adversarial Training for Unsupervised Domain-Adaptive Semantic Segmentation". In: *CoRR* abs/1907.12282 (2019). arXiv: 1907.12282.

[207] D. Shim and H. J. Kim. "Learning a Domain-Agnostic Visual Representation for Autonomous Driving via Contrastive Loss". In: *CoRR* abs/2103.05902 (2021). arXiv: 2103.05902.

[208] G. Shin, W. Xie, and S. Albanie. “All you need are a few pixels: semantic segmentation with PixelPick”. In: *International Conference on Computer Vision Workshops (ICCVW)*. IEEE, 2021, pp. 1687–1697.

[209] C. Shui, F. Zhou, C. Gagné, and B. Wang. “Deep Active Learning: Unified and Principled Method for Query and Training”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Ed. by S. Chiappa and R. Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1308–1318.

[210] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations (ICLR)*. San Diego, CA, USA, May 2015, pp. 1–27.

[211] A. Sinha, H. Namkoong, and J. Duchi. “Certifiable Distributional Robustness with Principled Adversarial Training”. In: *International Conference on Learning Representations (ICLR)*. 2018.

[212] S. Sinha, S. Ebrahimi, and T. Darrell. “Variational Adversarial Active Learning”. In: *International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 5971–5980.

[213] L. Song, Y. Xu, L. Zhang, B. Du, Q. Zhang, and X. Wang. “Learning From Synthetic Images via Active Pseudo-Labeling”. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 6452–6465.

[214] J. Song, C. Meng, and S. Ermon. “Denoising Diffusion Implicit Models”. In: *International Conference on Learning Representations (ICLR)*. virtual, May 2021, pp. 1–20.

[215] T. Spadotto, M. Toldo, U. Michieli, and P. Zanuttigh. “Unsupervised Domain Adaptation with Multiple Domain Discriminators and Adaptive Self-Training”. In: *International Conference on Pattern Recognition (ICPR)*. 2021, pp. 2845–2852.

[216] S. Stan and M. Rostami. “Unsupervised Model Adaptation for Continual Semantic Segmentation”. In: *AAAI Conference on Artificial Intelligence (AAAI)* 35.3 (May 2021), pp. 2593–2601.

[217] D. Stutz, M. Hein, and B. Schiele. “Disentangling Adversarial Robustness and Generalization”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 2019, pp. 6976–6987.

[218] M. N. Subhani and M. Ali. “Learning From Scale-Invariant Examples for Domain Adaptation in Semantic Segmentation”. In: *European Conference on Computer Vision (ECCV)*. Glasgow, UK, Aug. 2020, pp. 290–306.

[219] R. Sun, X. Zhu, C. Wu, C. Huang, J. Shi, and L. Ma. "Not All Areas Are Equal: Transfer Learning for Semantic Segmentation via Hierarchical Region Selection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 2019, pp. 4360–4369.

[220] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros. "Unsupervised Domain Adaptation Through Self-Supervision". In: *CoRR* abs/1909.11825 (2019). arXiv: 1909.11825.

[221] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros. "Unsupervised Domain Adaptation through Self-Supervision". In: *CoRR* abs/1909.11825 (2019). arXiv: 1909.11825.

[222] T. Sun, M. Segu, J. Postels, Y. Wang, L. Van Gool, B. Schiele, F. Tombari, and F. Yu. "SHIFT: a synthetic driving dataset for continuous multi-task domain adaptation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 21371–21382.

[223] Q. Sun, P. Melnyk, M. Felsberg, and Y. Tang. "Augment Features Beyond Color for Domain Generalized Segmentation". In: *CoRR* abs/2307.01703 (2023). arXiv: 2307.01703.

[224] S. Tang, P. Tang, Y. Gong, Z. Ma, and M. Xie. "Unsupervised domain adaptation via coarse-to-fine feature alignment method using contrastive learning". In: *CoRR* abs/2103.12371 (2021). arXiv: 2103.12371.

[225] 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". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (2022), pp. 6517–6533.

[226] 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: *IEEE Intelligent Transportation Systems Conference (ITSC)*. virtual, Sept. 2021, pp. 2881–2888.

[227] J.-A. Termöhlen, T. Bartels, and T. Fingscheidt. "A Re-Parameterized Vision Transformer (ReVT) for Domain-Generalized Semantic Segmentation". In: *International Conference on Computer Vision Workshops (ICCVW)*. Paris, France, Oct. 2023, pp. 4376–4385.

[228] M. Toldo, A. Maracani, U. Michieli, and P. Zanuttigh. "Unsupervised Domain Adaptation in Semantic Segmentation: A Review". In: *Technologies* 8.2 (2020), pp. 1–35.

[229] M. Toldo, U. Michieli, G. Agresti, and P. Zanuttigh. “Unsupervised Domain Adaptation for Mobile Semantic Segmentation Based on Cycle Consistency and Feature Alignment”. In: *Image and Vision Computing* 95 (2020), pp. 103889–103899.

[230] M. Toldo, U. Michieli, and P. Zanuttigh. “Unsupervised Domain Adaptation in Semantic Segmentation via Orthogonal and Clustered Embeddings”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. virtual, Jan. 2021, pp. 1358–1368.

[231] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson. “DACS: Domain Adaptation via Cross-Domain Mixed Sampling”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA, Jan. 2021, pp. 1379–1389.

[232] 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: *International Conference on Computer Vision (ICCV)*. virtual, Oct. 2021, pp. 8548–8557.

[233] 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, June 2018, pp. 7472–7481.

[234] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker. “Domain Adaptation for Structured Output via Discriminative Patch Representations”. In: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 1456–1465.

[235] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. “Instance Normalization: The Missing Ingredient for Fast Stylization”. In: *CoRR* abs/1607.08022 (2016). arXiv: 1607.08022.

[236] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Savarese. “Generalizing to Unseen Domains via Adversarial Data Augmentation”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018, pp. 5339–5349.

[237] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. P. Perez. “ DADA: Depth-Aware Domain Adaptation in Semantic Segmentation ”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2019, pp. 7363–7372.

[238] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez. "ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, June 2019, pp. 2517–2526.

[239] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez. "DADA: Depth-Aware Domain Adaptation in Semantic Segmentation". In: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 7364–7373.

[240] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin. "Cost-Effective Active Learning for Deep Image Classification". In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (2017), pp. 2591–2600.

[241] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2097–2106.

[242] 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: *European Conference on Computer Vision (ECCV)*. virtual, Aug. 2020, pp. 642–659.

[243] S. Wang, Y. Li, K. Ma, R. Ma, H. Guan, and Y. Zheng. "Dual Adversarial Network for Deep Active Learning". In: *European Conference on Computer Vision (ECCV)*. Ed. by A. Vedaldi, H. Bischof, T. Brox, and J. Frahm. Vol. 12369. Lecture Notes in Computer Science. Springer, 2020, pp. 680–696.

[244] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W. Hwu, T. S. Huang, and H. Shi. "Differential Treatment for Stuff and Things: A Simple Unsupervised Domain Adaptation Method for Semantic Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, June 2020, pp. 12635–12644.

[245] K. Wang, C. Yang, and M. Betke. "Consistency Regularization with High-dimensional Non-adversarial Source-guided Perturbation for Unsupervised Domain Adaptation in Segmentation". In: *AAAI Conference on Artificial Intelligence (AAAI)* 35.11 (May 2021), pp. 10138–10146.

[246] Q. Wang, D. Dai, L. Hoyer, L. Van Gool, and O. Fink. "Domain Adaptive Semantic Segmentation With Self-Supervised Depth Estimation". In: *International Conference on Computer Vision (ICCV)*. virtual, Oct. 2021, pp. 8515–8525.

[247] 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". In: *CoRR* abs/2105.03151 (2021). arXiv: 2105.03151.

[248] Y. Wang, J. Peng, and Z. Zhang. "Uncertainty-Aware Pseudo Label Refinery for Domain Adaptive Semantic Segmentation". In: *International Conference on Computer Vision (ICCV)*. virtual, Oct. 2021, pp. 9092–9101.

[249] Z. Wang, X. Liu, M. Suganuma, and T. Okatani. "Cross-Region Domain Adaptation for Class-level Alignment". In: *CoRR* abs/2109.06422 (2021). arXiv: 2109.06422.

[250] Y. Wang, H. Wang, Y. Shen, J. Fei, W. Li, G. Jin, L. Wu, R. Zhao, and X. Le. "Semi-Supervised Semantic Segmentation Using Unreliable Pseudo-Labels". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 4238–4247.

[251] Y. Wang, Y. Li, J. H. Elder, R. Wu, and H. Lu. "Class-conditional domain adaptation for semantic segmentation". In: *Computational Visual Media* 10.5 (2024), pp. 1013–1030.

[252] 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: *European Conference on Computer Vision (ECCV)*. Munich, Germany, Sept. 2018, pp. 535–552.

[253] Z. Wu, C. Shen, and A. van den Hengel. " Wider or Deeper: Revisiting the ResNet Model for Visual Recognition ". In: *Pattern Recognition* 90 (2019), pp. 119–133.

[254] B. Xie, K. Yin, S. Li, and X. Chen. "SPCL: A New Framework for Domain Adaptive Semantic Segmentation via Semantic Prototype-based Contrastive Learning". In: *CoRR* abs/2111.12358 (2021). arXiv: 2111.12358.

[255] 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: *Advances in Neural Information Processing Systems (NeurIPS)*. virtual, Dec. 2021, pp. 12077–12090.

[256] B. Xie, S. Li, M. Li, C. H. Liu, G. Huang, and G. Wang. "Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.7 (2023), pp. 9004–9021.

[257] J. Xu, L. Xiao, and A. M. López. "Self-Supervised Domain Adaptation for Computer Vision Tasks". In: *IEEE Access* 7 (2019), pp. 156694–156706.

[258] R. Xu, G. Li, J. Yang, and L. Lin. "Larger Norm More Transferable: An Adaptive Feature Norm Approach for Unsupervised Domain Adaptation". In: *International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 1426–1435.

[259] Y. Xu, B. Du, L. Zhang, Q. Zhang, G. Wang, and L. Zhang. "Self-Ensembling Attention Networks: Addressing Domain Shift for Semantic Segmentation". In: *AAAI Conference on Artificial Intelligence (AAAI)*. Honolulu, HI, USA, Jan. 2019, pp. 5581–5588.

[260] Z. Xu, D. Liu, J. Yang, C. Raffel, and M. Niethammer. "Robust and Generalizable Visual Representation Learning via Random Convolutions". In: *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021.

[261] Q. Xu, L. Yao, Z. Jiang, G. Jiang, W. Chu, W. Han, W. Zhang, C. Wang, and Y. Tai. "DIRL: Domain-Invariant Representation Learning for Generalizable Semantic Segmentation". In: *AAAI Conference on Artificial Intelligence (AAAI)*. virtual, June 2022, pp. 2884–2892.

[262] W. Xu, Z. Wang, and W. Bian. "Unsupervised Domain Adaptation with Implicit Pseudo Supervision for Semantic Segmentation". In: *International Joint Conference on Neural Networks (IJCNN)*. 2022, pp. 1–10.

[263] Y. Xu, F. He, B. Du, D. Tao, and L. Zhang. "Self-Ensembling GAN for Cross-Domain Semantic Segmentation". In: *IEEE Transactions on Multimedia* 25 (2023), pp. 7837–7850.

[264] 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: *AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 34. 07. 2020, pp. 12613–12620.

[265] J. Yang, W. An, S. Wang, X. Zhu, C. Yan, and J. Huang. "Label-Driven Reconstruction for Domain Adaptation in Semantic Segmentation". In: *European Conference on Computer Vision (ECCV)*. Glasgow, UK, Aug. 2020, pp. 480–498.

[266] Y. Yang, D. Lao, G. Sundaramoorthi, and S. Soatto. "Phase Consistent Ecological Domain Adaptation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, June 2020, pp. 9011–9020.

[267] Y. Yang and S. Soatto. "FDA: Fourier Domain Adaptation for Semantic Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, June 2020, pp. 4085–4095.

[268] Z. Yan, X. Yu, Y. Qin, Y. Wu, X. Han, and S. Cui. "Pixel-Level Intra-Domain Adaptation for Semantic Segmentation". In: *IEEE International Conference on Multimedia & Expo (ICME)*. 2021, pp. 404–413.

[269] J. Yang, W. An, C. Yan, P. Zhao, and J. Huang. “Context-Aware Domain Adaptation in Semantic Segmentation”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. virtual, Jan. 2021, pp. 514–524.

[270] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni. “Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification”. In: *Scientific Data* 10.1 (2023), Article 41.

[271] 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. virtual, June 2020, pp. 13736–13745.

[272] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. “BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling”. In: *CoRR* abs/1805.04687 (2018). arXiv: `1805.04687`.

[273] 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: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. virtual, June 2020, pp. 1–14.

[274] 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: *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2021, pp. 10754–10762.

[275] 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: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 2100–2110.

[276] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. “Cutmix: Regularization Strategy to Train Strong Classifiers With Localizable Features”. In: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 6023–6032.

[277] K. Zhang, W. Zuo, S. Gu, and L. Zhang. “Learning Deep CNN Denoiser Prior for Image Restoration”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Honulu, HI, USA, July 2017, pp. 3929–3938.

[278] Q. ZHANG, J. Zhang, W. Liu, and D. Tao. “Category Anchor-Guided Unsupervised Domain Adaptation for Semantic Segmentation”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019, pp. 433–443.

[279] B. Zhang, S. Zhao, and R. Zhang. “Towards Adaptive Semantic Segmentation by Progressive Feature Refinement”. In: *IEEE International Conference on Image Processing (ICIP)*. Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 2221–2225.

[280] Y. Zhang, Z. Qiu, T. Yao, C.-W. Ngo, D. Liu, and T. Mei. “Transferring and Regularizing Prediction for Semantic Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. virtual, June 2020, pp. 9621–9630.

[281] Y. Zhang and Z. Wang. “Joint Adversarial Learning for Domain Adaptation in Semantic Segmentation”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 34. 04. 2020, pp. 6877–6884.

[282] K. Zhang, Y. Sun, R. Wang, H. Li, and X. Hu. “Multiple Fusion Adaptation: A Strong Framework for Unsupervised Semantic Segmentation Adaptation”. In: *British Machine Vision Conference (BMVC)*. BMVA Press, 2021, p. 42.

[283] 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”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 12414–12424.

[284] F. Zhang, V. Koltun, P. H. S. Torr, R. Ranftl, and S. R. Richter. “Unsupervised Contrastive Domain Adaptation for Semantic Segmentation”. In: *CoRR* abs/2204.08399 (2022). arXiv: 2204.08399.

[285] J. Zhang, J. Huang, Z. Tian, and S. Lu. “Spectral Unsupervised Domain Adaptation for Visual Recognition”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 9829–9840.

[286] Y. Zhao, Z. Zhong, N. Zhao, N. Sebe, and G. H. Lee. “Style-Hallucinated Dual Consistency Learning for Domain Generalized Semantic Segmentation”. In: *European Conference on Computer Vision (ECCV)*. Tel Aviv, Israel, 2022, pp. 535–552.

[287] L. Zhang, A. Rao, and M. Agrawala. “Adding conditional control to text-to-image diffusion models”. In: *International Conference on Computer Vision (ICCV)*. 2023, pp. 3836–3847.

[288] F. Zhdanov. “Diverse mini-batch Active Learning”. In: *CoRR* abs/1901.05954 (2019). arXiv: 1901.05954.

[289] Q. Zheng, J. Chen, Z. Wang, J. Jiang, and C. Liang. “Deep Segmentation Domain Adaptation Network With Weighted Boundary Constraint”. In: *IEEE Access* 7 (2019), pp. 93909–93918.

[290] Z. Zheng and Y. Yang. “Unsupervised Scene Adaptation with Memory Regularization in vivo”. In: *International Joint Conference on Artificial Intelligence (IJCAI)*. Ed. by C. Bessiere. ijcai.org, 2020, pp. 1076–1082.

[291] Z. Zheng and Y. Yang. “Rectifying Pseudo Label Learning via Uncertainty Estimation for Domain Adaptive Semantic Segmentation”. In: *International Journal of Computer Vision (IJCV)* (2021), pp. 1–15.

[292] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang. “Learning to generate novel domains for domain generalization”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 561–578.

[293] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang. “Domain Generalization with MixStyle”. In: *CoRR* abs/2104.02008 (2021). arXiv: 2104.02008.

[294] W. Zhou, Y. Wang, J. Chu, J. Yang, X. Bai, and Y. Xu. “Affinity Space Adaptation for Semantic Segmentation Across Domains”. In: *IEEE Transactions on Image Processing* 30 (2021), pp. 2549–2561.

[295] Z. Zhong, Y. Zhao, G. H. Lee, and N. Sebe. “Adversarial Style Augmentation for Domain Generalized Urban-Scene Segmentation”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. New Orleans, LA , USA, Dec. 2022, pp. 338–350.

[296] Q. Zhou, Z. Feng, Q. Gu, G. Cheng, X. Lu, J. Shi, and L. Ma. “Uncertainty-aware consistency regularization for cross-domain semantic segmentation”. In: *Computer Vision and Image Understanding* 221 (2022), p. 103448.

[297] 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”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 33.2 (2022), pp. 804–817.

[298] Q. Zhou, C. Zhuang, R. Yi, X. Lu, and L. Ma. “Domain Adaptive Semantic Segmentation via Regional Contrastive Consistency Regularization”. In: *IEEE International Conference on Multimedia & Expo (ICME)*. IEEE, 2022, pp. 1–6.

[299] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy. “Domain Generalization: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.4 (2023), pp. 4396–4415.

[300] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *International Conference on Computer Vision (ICCV)*. Venice, Italy, Oct. 2017, pp. 2223–2232.

[301] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. “Toward Multimodal Image-to-Image Translation”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Long Beach, CA, USA, Dec. 2017, pp. 465–476.

[302] Y. Zhu, H. Zhou, C. Yang, J. Shi, and D. Lin. "Penalizing Top Performers: Conservative Loss for Semantic Segmentation Adaptation". In: *European Conference on Computer Vision (ECCV)*. Munich, Germany, Sept. 2018, pp. 568–583.

[303] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. D. Newsam, A. Tao, and B. Catanzaro. "Improving Semantic Segmentation via Video Propagation and Label Relaxation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. Computer Vision Foundation / IEEE, 2019, pp. 8856–8865.

[304] Y. Zou, Z. Yu, B. V. K. Vijaya Kumar, and J. Wang. "Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training". In: *European Conference on Computer Vision (ECCV)*. Munich, Germany, Sept. 2018, pp. 289–305.

[305] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang. "Confidence Regularized Self-Training". In: *International Conference on Computer Vision (ICCV)*. Seoul, Korea, Oct. 2019, pp. 5982–5991.

Own Publications and Awards

Best Paper Awards

- **German Conference on Pattern Recognition (GCPR) 2023:** For the paper "Best Practices in Active Learning for Semantic Segmentation"
- **MICCAI - International Workshop on Simulation and Synthesis in Medical Imaging 2024:** For the paper "TSynD: Targeted Synthetic Data Generation for Enhanced Medical Image Classification"

Best Poster Awards

- **German Conference on Medical Image Computing (BVM) 2025:** For "TSynD: Targeted Synthetic Data Generation for Enhanced Medical Image Classification"

First Author Papers

[1] **Joshua Niemeijer**, Jörg P. Schäfer. "Combining Semantic Self-Supervision and Self-Training for Domain Adaptation in Semantic Segmentation". In: *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. 2021, pp. 364–371

[2] **Joshua Niemeijer**, Jörg Peter Schäfer. "Domain Adaptation and Generalization: A Low-Complexity Approach". In: *Conference on Robot Learning (CoRL)*. ed. by K. Liu, D. Kulic, and J. Ichnowski. Vol. 205. Proceedings of Machine Learning Research. PMLR, 14–18 Dec 2022, pp. 1081–1091

[3] Manuel Schwonberg, **Joshua Niemeijer**, Jan-Aike Termöhlen, Jörg P. Schäfer, Nico M. Schmidt, Hanno Gottschalk, Tim Fingscheidt. "Survey on Unsupervised Domain Adaptation for Semantic Segmentation for Visual Perception in Automated Driving". In: *IEEE Access* 11 (May 2023), pp. 54296–54336 *** equal contribution first author**

[4] **Joshua Niemeijer**, Jan Ehrhardt, Timo Kepp, Jörg P. Schäfer, Heinz Handels. "Overcoming the sensor delta for semantic segmentation in OCT images". In:

Medical Imaging 2023: Computer-Aided Diagnosis. Ed. by K. M. Iftekharuddin and W. Chen. Vol. 12465. SPIE Proceedings. SPIE, 2023

[5] Sudhanshu Mittal, **Joshua Niemeijer**, Jörg P. Schäfer, Thomas Brox. "Best Practices in Active Learning for Semantic Segmentation". In: *German Conference on Pattern Recognition (GCPR)*. ed. by U. Köthe and C. Rother. Vol. 14264. Lecture Notes in Computer Science. Springer, 2023, pp. 427–442 *** equal contribution first author**

[6] Sudhanshu Mittal, **Joshua Niemeijer**, Oezguen Cicek, Maxim Tatarchenko, Jan Ehrhardt, Joerg P Schaefer, Heinz Handels, Thomas Brox. "Realistic Evaluation of Deep Active Learning for Image Classification and Semantic Segmentation". In: *International Journal of Computer Vision (IJCV)* (Feb. 2025), pp. 1–23 *** equal contribution first author**

[7] **Joshua Niemeijer**, Sudhanshu Mittal, Thomas Brox. "Synthetic Dataset Acquisition for a Specific Target Domain". In: *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. Paris, France: IEEE, Oct. 2023, pp. 4057–4066

[8] **Joshua Niemeijer**, Manuel Schwonberg, Jan-Aike Termöhlen, Nico M. Schmidt, Tim Fingscheidt. "Generalization by Adaptation: Diffusion-Based Domain Extension for Domain-Generalized Semantic Segmentation". In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2024, pp. 2830–2840

[9] **Joshua Niemeijer**, Jan Ehrhardt, Hristina Uzunova, Heinz Handels. "TSynD: Targeted Synthetic Data Generation for Enhanced Medical Image Classification". In: *Proc. of MICCAI Workshops - International Workshop on Simulation and Synthesis in Medical Imaging*. Marrakesch, Morocco, Oct. 2024, pp. 69–78

[10] **Joshua Niemeijer**, Jan Ehrhardt, Hristina Uzunova, Heinz Handels. "Abstract: TSynD Targeted Synthetic Data Generation for Enhanced Medical Image Classification". In: *German Conference on Medical Image Computing (BVM)*. ed. by C. Palm, K. Breininger, T. Deserno, H. Handels, A. Maier, K. H. Maier-Hein, and T. M. Tolxdorff. Wiesbaden: Springer Fachmedien Wiesbaden, 2025, pp. 157–157

[11] **Joshua Niemeijer**, Federico Battistella, Gurucharan Srinivas, Andreas Leich. "An Approach for Fusing Two Training-Datasets with Partially Overlapping Classes". In: *IEEE International Conference on Semantic Computing (ICSC)*. Laguna Hills, CA, USA: IEEE, Feb. 2023, pp. 73–79

Co-Author Papers

[12] Sofija Engelson, Jan Ehrhardt, Timo Kepp, **Joshua Niemeijer**, Heinz Handels. "LNQ Challenge 2023: Learning Mediastinal Lymph Node Segmentation with a Probabilistic Lymph Node Atlas". In: *Machine Learning for Biomedical Imaging* 2 (MICCAI 2023 LNQ challenge special issue 2024), pp. 817–833

[13] Sofija Engelson, Jan Ehrhardt, Timo Kepp, **Joshua Niemeijer**, Stefanie Schierholz, Lennart Berkel, Yannic Elser, Malte Maria Sieren, Heinz Handels. "Comparison of anatomical priors for learning-based neural network guidance for mediastinal lymph node segmentation". In: *Medical Imaging 2024: Computer-Aided Diagnosis*. Ed. by W. Chen and S. M. Astley. Vol. 12927. International Society for Optics and Photonics. SPIE, 2024, 129271K

[14] Marc S. Seibel, **Joshua Niemeijer**, Marc Rowedder, Helge Sudkamp, Timo Kepp, Gereon Hüttmann, Heinz Handels. "Reducing the impact of domain shift in deep learning for OCT segmentation using image manipulations". In: *Medical Imaging 2024: Computer-Aided Diagnosis*. Ed. by W. Chen and S. M. Astley. Vol. 12927. International Society for Optics and Photonics. SPIE, 2024, p. 1292719

[15] Andreas Leich, Julian Fuchs, Gurucharan Srinivas, **Joshua Niemeijer**, Peter Wagner. "Traffic Safety at German Roundabouts—A Replication Study". In: *Safety* 8.3 (2022)

[16] Clarissa Böker, **Joshua Niemeijer**, Nicolai Wojke, Cyril Meurie, Yann Cocheril. "A System for Image-Based Non-Line-Of-Sight Detection Using Convolutional Neural Networks". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 535–540

[17] Andreas Leich, Nils Kornfeld, **Joshua Niemeijer**, Max Kaiser, Marcel Jäckle. "Erkennung von Rissen mittels maschinellen Lernens". In: *EI-Der Eisenbahningenieur* (2023), pp. 38–43

[18] Kanwal Jahan, **Joshua Niemeijer**, Nils Kornfeld, Michael Roth. "Deep Neural Networks for Railway Switch Detection and Classification Using Onboard Camera Images". In: *IEEE Symposium Series on Computational Intelligence (SSCI)*. 2021, pp. 01–07

Joshua Niemeijer

Joshua.Niemeijer@dlr.de

Education

01/2020–11/2025 **PH.D.**, *PhD student at University of Lübeck in cooperation with German Aerospace Center (DLR)*
Optimizing Synthetic and Real Training Data Distributions for Deep Learning in Image Recognition

01/2015–11/2017 **Master**, *University of Lübeck*,
Medical Informatics

10/2011–12/2014 **Bachelor**, *University of Lübeck*,
Medical Informatics

05/2011 **Abitur**,

Academic and Professional Positions

since 10/2024 **Deputy Group Leader**, *German Aerospace Center (DLR)*, Braunschweig

since 02/2018 **Researcher**, *German Aerospace Center (DLR)*, Braunschweig

Research Experience

google scholar `https://scholar.google.com/citations?user=SK0mAJ0AAAAJ&hl=de`

award *Best paper award at GCPR 2023 for the paper "Best Practices in Active Learning for Semantic Segmentation"*

special issue *The paper "Best Practices in Active Learning for Semantic Segmentation"got selected for an extension publication in a special issue of the IJCV*

award *Best paper award at MICCAI - International Workshop on Simulation and Synthesis in Medical Imaging 2024 for the paper "TSynD: Targeted Synthetic Data Generation for Enhanced Medical Image Classification"*

award *Best poster award at BVM 2025 for the abstract "TSynD: Targeted Synthetic Data Generation for Enhanced Medical Image Classification"*

reviewer *27th and 28th International Conference on Medical Image Computing - MICCAI*

reviewer International Conference on Computer Vision - ICCV 2025