UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Generalizable Visual Understanding with Deep Neural Networks

by Guangrui Li

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Prof. Yi Yang

University of Technology Sydney
Faculty of Engineering and Information Technology

July 2023

Certificate of Authorship/Originality

I, Guangrui Li, declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science, FEIT at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 24 July 2023

ABSTRACT

Generalizable Visual Understanding

with Deep Neural Networks

by

Guangrui Li

Deep neural networks (DNNs) have transformed computer vision, advancing object recognition, scene understanding, and image synthesis. However, a critical challenge remains in their ability to generalize to unseen distributions and novel categories, limiting their full potential in real-world applications. This thesis endeavors to address this limitation and develops methodologies to bestow vision models with strong generalizability in diverse and changing environments. It delves into two crucial perspectives of generalizability in computer vision, *i.e.*, generalizing to novel structures and novel categories.

In addressing the challenge of generalizing to novel structures, the research endeavors to extract generalizable structural representations from diverse visual scenarios. These encompass 2D rigid scenes, 3D rigid scenes, and non-rigid structures. The study identifies obstacles to generalization, including discrepancies in layout distribution for 2D scenes, dropout noises disrupting 3D scene geometry, and variations in inter-joint relationships within non-rigid structures. To overcome these challenges, innovative methodologies are developed. These methodologies include layout-matching techniques to bridge layout distribution gaps, adversarial masking paradigms to enhance robustness against disruptive geometry noises, and a "decompose to generalize" paradigm that reinforces commonalities in inter-joint relationships among different species, thereby promoting generalization.

Regarding the generalizability with novel categories, this thesis is structured around two fundamental questions: (1) the ability to discern novel categories from

known ones, and (2) the aptitude to effectively classify each newly encountered category. The former challenge is addressed as the "category shift" problem, wherein only partial categories are shared between two correlated domains / datasets. To tackle this issue, a clustering algorithm is proposed to delineate the known from the unknown through cross-domain consensus knowledge. For the latter challenge, a solution is devised by leveraging cross-modality knowledge from Vision-Language Models (VLMs), wherein distinctions between known and novel categories are discerned through discriminative mappings in the latent text space. In pursuit of this objective, the thesis introduces a "decouple to contrast" methodology to alleviate ambiguities between visual and text latent spaces in a decoupled manner.

In conclusion, this thesis contributes to the advancement of generalizable visual understanding by proposing novel approaches and methodologies tailored for deep neural networks. The developed techniques enhance the network's ability to learn robust and transferable representations, enabling better generalization across diverse visual domains. These findings have implications for various real-world applications, including autonomous systems, robotics, and computer vision-based technologies.

Dissertation directed by Professor Yi Yang,
Australian Artificial Intelligence Institute, University of Technology Sydney

Acknowledgements

First, I would like to express my heartfelt gratitude to my supervisor, Prof. Yi Yang, for his unwavering support and guidance throughout my Ph.D. journey. I vividly recall the meeting with Prof. Yang on a cloudy afternoon in Chongqing, which undoubtedly changes my academic and personal trajectory. In the entire journey, his profound expertise and insightful suggestions guided me to achieve more than I ever thought possible. I extend my sincerest appreciation to Prof. Yang for being an exceptional mentor and for making an indelible impact on my life and career.

I extend my sincere thanks to my mentors, Dr. Yunchao Wei and Dr. Guoliang Kang, for their invaluable guidance and inspiration in my research. Their insightful discussions and suggestions have played a significant role in shaping my work, and I am truly appreciative of their support.

I would also like to thank my colleagues and friends in Sydney. Living in such a vibrant city has been a truly enjoyable journey, made even more fulfilling with these intellectual, interesting, and supportive minds. I would like to thank Dr. Linchao Zhu, Mr. Tianqi Tang, Dr. Yanbin Liu, Mr. Xuanmeng Zhang, Mr. Yaowei Li, Mr. Mingfei Han, Mr. Liulei Li, Dr. Peike Li, and many others. I was really fortunate to work with them and participate in intellectual conversations with them. I would also love to thank Dr. Shanshan Zhao, Dr. Bohan Hu, Dr. Feng Liu, and Dr. Weijian Deng, whose presence is a valuable addition to my life and research.

At this moment, I am surprised that I can successfully survive the Ph.D. grind in an easier style than I initially imagine. This journey is so enjoyable and swift that I am actually starting to reminisce about it now, in which I learned how to comprehend the world from a scientific perspective, how to design my life and its balance with work, and the unforgettable memory with Sydney. I am grateful to

myself, for my continuous endeavor, wise management of life and research, and most important, the open mind that actively explores and adapts to the unseen domains/distributions.

Lastly, thank my parents for their support and love throughout the years.

Guangrui Li

Sydney, Australia

July, 2023

List of Publications

Conference Paper:

C-1 **Guangrui Li**, Guoliang Kang, Wu Liu, Yunchao Wei, Yi Yang. "Content-Consistent Matching for Domain Adaptive Semantic Segmentation", In *European Conference on Computer Vision (ECCV 2020).*

C-2 **Guangrui Li**, Guoliang Kang, Yi Zhu, Yunchao Wei, Yi Yang. "Domain Consensus Clustering for Universal Domain Adaptation", In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021).*

C-3 Jiaxu Miao, Yunchao Wei, Yu Wu, Chen Liang, **Guangrui Li**, Yi Yang. "VSPW: A Large-scale Dataset for Video Scene Parsing in the Wild", In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021).*

C-4 **Guangrui Li**, Guoliang Kang, Xiaohan Wang, Yunchao Wei, Yi Yang. "Adversarially Masking Synthetic to Mimic Real: Adaptive Noise Injection for Point Cloud Segmentation Adaptation", In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2023).*

C-5 **Guangrui Li**, Yifan Sun, Zongxin Yang, Yi Yang. "Decompose to Generalize: Species-Generalized Animal Pose Estimation", In *The Eleventh International Conference on Learning Representations (ICLR 2023).*

C-6 **Guangrui Li**, Guoliang Kang, Yunchao Wei, Yi Yang. "Construct to Associate: Cooperative Context Learning for Domain Adaptive Point Cloud Segmentation", In *submission.*

C-7 **Guangrui Li**, Yifan Sun, Zongxin Yang, Yi Yang. "Decouple to Contrast: Orthogonalized Ambiguity Reduction for Open-Vocabulary Object Detection", In *submission.*

Contents

List of Figures

Chapter 1

Introduction

In recent years, deep neural networks (DNNs) have revolutionized the field of computer vision, enabling significant advancements in computer vision tasks such as object recognition, scene understanding, and image synthesis. Despite their impressive capabilities, DNNs often exhibit limited generalizability when confronting unseen distributions, *e.g.*, adapting to adverse weather conditions. This deficiency causes visual understanding algorithms to lag far behind the seamless adaptability and generalizability of humans to new environments. To address this limitation, this thesis aims to bestow the vision models with strong generalizability in diverse and changing environments, thereby minimizing the gap between human perception and visual understanding algorithms.

In computer vision, generalization refers to the ability of a model to effectively adapt and perform well on new, unseen data that is not part of the training dataset. This thesis primarily investigates two critical perspectives of generalizability: (1) learning to generalize to unseen distributions, and (2) learning to generalize to unseen categories. These two properties span two critical aspects of generalizability when dealing with dynamic and changing environments.

For generalizing to unseen distributions, I first investigate how to extract informative structural knowledge from diverse array of structures, encompassing 2D rigid scenes, 3D rigid scenes, and non-rigid structures. Then I delve into key factors that impede the generalization across visual domains. For 2D scenes, I found that the discrepancy in layout distribution hinders the generalizable scene understand-

ing. This motivates me to propose an effective layout-matching strategy to bridge the gap. In the context of 3D scenes, I identify the dropout noises as one of the causes that harm the adaptability through disrupting the scene geometry. Then I propose an adversarial masking paradigm to improve the robustness towards these geometry-disruptive noises and enhance generalizability. For more challenging non-rigid poses from diverse animal species, I discover the discrepancy arises from the varying inter-joint relationships, involving both geometry and appearance. Driven by this insight, I formulate a "divide and conquer" scheme to mine and reinforce the commonalities of inter-joint relationships among different species, thereby promoting the generalization.

For generalizing to novel categories, this thesis is structured around two fundamental questions: how to distinguish novel categories from the known, and how to classify each novel category. For the former, I consider this as the category shift problem, where only partial categories are shared by training sets and test sets. I propose a clustering pipeline to separate the known from the unknown with cross-domain consensus knowledge. As for the latter, I devise a solution by drawing the cross-modality knowledge from Vision-Language Models (VLMs), where known and novel categories are distinguished by their discriminative mapping in the text latent space. With this objective, I propose a "decouple to contrast" to calibrate ambiguities between the visual space and text latent space in a decoupled manner.

The contributions of the thesis are listed as follows.

- I explore how to model dynamic visual structures and enhance their generalizability with unseen distributions. Effective techniques and methodologies are developed to deal with various forms of structures, including 2D rigid scenes, 3D rigid scenes, and non-rigid structures. Experiments on representative benchmarks verify that the proposed methods can effectively improve the

adaptability and generalizability with diverse environments/distributions.

- I explore how to generalize the vision models to novel categories. To achieve this, I propose to mine the agreement between different domains/modalities and leverage their consensus to differentiate novel categories from known classes, as well as the identification and localization of them. Comprehensive experiments and analysis validated that the proposed approaches can greatly enhance the generalizability to novel categories.

The following introduces the background and developed methodologies for generalizable visual understanding with novel structures and novel categories.

1.1 Generalization to Novel Structures

Humans interact with diverse structures daily, effortlessly recognizing and understanding them across different environments. In contrast, visual understanding algorithms remain vulnerable to intrinsic (*e.g.*, color, shape) and extrinsic (*e.g.*, viewpoint, lighting) variations. This thesis aims to enable machines to overcome these variations and adapt to unseen distributions. By enhancing generalizability, the proposed approaches bridge the gap between human-like perception and machine perception, empowering machines to interpret and understand intricate structures in diverse and changing environments adaptively.

In Chapter 3, I study the domain adaptation problem for 2D scene segmentation, which adapts the model trained on a labeled source domain to an unlabeled target domain under distribution shifts. Unlike previous solutions treating all source samples equally, I observe that not all source samples contribute to the adaptation, while some may impede the transfer, namely negative transfer. To solve this, I propose a new scheme, Content-Consistent Matching (CCM), that actively compares and matches source and target samples encompassing two perspectives: layout dis-

tribution and semantic distribution. The scenes are decomposed as the semantic frequencies over different spatial locations, and the discrepancy in layout distributions is derived by comparing the frequencies along two directions, *i.e.*, vertical and horizontal. Then the mined discrepancy is leveraged to select source samples with higher content consistency, thus enabling the adaptation with fewer source samples and avoiding negative transfer. On two representative benchmarks, CCM yields consistent improvements over the baselines and achieves the new state-of-the-art.

In Chapter 4, I consider the domain adaptation problem in point cloud semantic segmentation, where 3D scenes are more dynamically organized with complex scene geometries. Due to environmental factors (*e.g.*, occlusions, glasses), point clouds may contain irregular and random noises (missing points), which harms the generalizability by ruining the scene geometries and contexts. In this chapter, I aim to adaptively inject noises into synthetic point clouds to imitate the irregular patterns of these noises, thereby enhancing the robustness and generalizability towards these noises. To achieve this, I propose adaptive spatial masking, where the adversarial training and learnable masking module imitate real-world noises in a cooperative manner. Specifically, the masking module combines geometrical and feature cues to derive learnable masks as noises while the domain adversarial learning paradigm urges domain-invariant representations. Consequently, the domain adversarial paradigm implicitly drives the masking module to imitate real-world noises, thus easing the domain gap induced by the noises. Experiments validate that the proposed ASM significantly improves the robustness to real-world noises.

In Chapter 5, I explore a much more challenging generalization problem with novel structures, *i.e.*, species-generalizable pose estimation, which requires the generalizable representations from multiple animal species for the dynamic non-rigid structures. To tackle this, I dive into the inter-joint relationships and reveal the varying consistency across different species, *i.e.*, some joint relationships are consis-

tent for different species while some are not. With this insight, I propose a novel scheme, Decompose-to-Generalize (D-Gen), which decomposes joints into groups that hold species-consistent relationships and then reinforces these groups in a parallel style. To be more specific, I devise an attention module to effectively model the inter-joint relationships implicitly combining both geometric and visual cues, then divide the body joints into different groups according to the inter-joint interactions. Experiments with diverse animal species and families verify that the proposed approach can effectively enhance generalizability.

1.2 Generalization to Novel Categories

To achieve human-like adaptability to expanding environments, it is of vital importance to identify novel categories/objects that differ from the known distributions. While recognizing seen samples or objects from known distributions is easy for both machines and humans, the challenge lies in generalizing to novel instances from unseen or unknown categories. The presence of these unknown samples can significantly undermine the performance of vision models, especially in human-critical systems. For example, consider an autonomous driving model trained with normal driving scenes, including trees, roads, buildings, *etc.* However, when a deer unexpectedly jump across the road, the model may struggle to capture and identify it without having seen this before, potentially leading to severe consequences. To address this issue, this thesis tackles two fundamental problems in generalizing to novel categories: how to distinguish the novel categories from the known, and how to identify and classify the novel categories.

In Chapter 6, I formulate the novel category discovery as the category shift problem between two correlated domains, encompassing the universal domain adaptation (UniDA) challenge. The main challenge of UniDA lies in how to separate common classes (*i.e.*, classes shared across domains) from private classes (*i.e.*, classes only

exist in one domain). Previous works treat the private samples in the target as one generic class but ignore their intrinsic structure. Consequently, the resulting representations are not compact enough in the latent space and can be easily confused with common samples. To better exploit the intrinsic structure of the target domain, we propose Domain Consensus Clustering (DCC), which exploits the domain consensus knowledge to discover discriminative clusters on both common samples and private ones. Specifically, we draw the domain consensus knowledge from two aspects to facilitate the clustering and the private class discovery, *i.e.*, the semantic level consensus, which identifies the cycle-consistent clusters as the common classes, and the sample-level consensus, which utilizes the cross-domain classification agreement to determine the number of clusters and discover the private classes. Based on DCC, we are able to separate the private classes from the common ones, and differentiate the private classes themselves. Experiments on four representative benchmarks demonstrate DCC significantly outperforms previous state-of-the-arts.

In Chapter 7, inspired by tight text-image correspondence in Vision-Language Models (VLMs), I explore how to leverage the cross-modal consensus to identify the novel categories. As the known and novel categories are unified in the discriminative text latent space of VLMs, the main challenge here is how to transfer this into the specialized vision models, *i.e.*, detector. Further investigation reveals two remaining obstacles between the detector and VLMs, *i.e.*, semantic and positional ambiguities. Semantic ambiguity refers to the cross-modal gap that object features may confuse with closely related text concepts/descriptions. Analogously, the text features can fail to discriminate object features with slight positional offsets, due to the position insensitivity in VLMs. To tackle this, I propose a "Decouple-to-Contrast" (DeCo) scheme to suppress these two ambiguities in a unified contrastive learning framework. Concretely, DeCo first decouples the object query into two parts, *i.e.*, semantics and positions, then reduces their ambiguities with two contrasting

branches. Given queries in the detector, DeCo applies position and semantic distractions separately, and then the ambiguities are alleviated through parallel contrasting paradigm, *i.e.*, contrasting with semantic-disturbed negative queries to reduce the semantic ambiguities. Experiments on representative benchmarks show that our method achieves the new state-of-the-art.

1.3 Thesis Organization

This thesis is structured as follows:

In Chapter 2, a survey of learning generalization in structured visual understanding is presented, which include discussions on generalizing to novel structures and categories.

Chapters 3 to 5 sequentially explore the generalization problem across diverse visual structures. More specifically, Chapter 3 focuses on domain adaptation in 2D rigid scenes, Chapter 4 on domain adaptation in 3D rigid scenes, and Chapter 5 on domain generalization with diverse non-rigid structures. In these chapters, I identify the primary obstacles among different structures and develop methodologies to model and derive generalizable representations for them.

In Chapters 6 and 7, the investigation shifts towards generalizability with novel categories. Chapter 6 proposes a novel clustering algorithm designed to differentiate between novel and known categories under various scenarios of category shift. Taking this a step further, Chapter 7 leverages visual-text correspondence to identify and classify novel categories.

Finally, Chapter 8 provides a brief summary of the thesis and discusses potential directions for future exploration.

Chapter 2

Literature Review

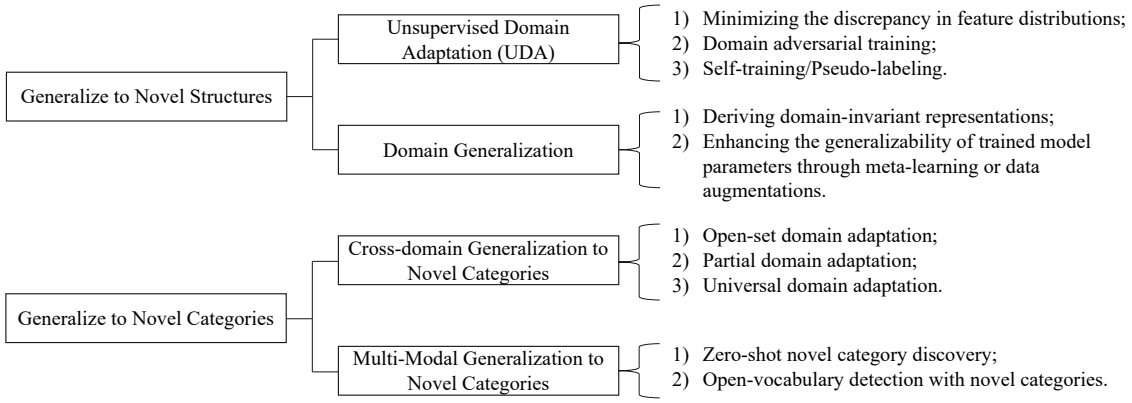


Figure 2.1 : An overview of methodologies in generalization learning with novel structures and novel categories.

This chapter introduces a survey of related work in generalization learning in computer vision, mainly encompassing the generalization to novel visual structures and novel categories.

2.1 Generalization to Novel Visual Structures

2.1.1 Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) aims to minimize the distribution discrepancy between the source domain and the target domain. Specifically, UDA tries to train a model on a labeled domain (source domain) and an unlabeled target domain, so as to adapt to the target domain by overcoming the distribution shift between two domains. To achieve this goal, earlier [127, 126, 179, 189] methods

proposed to learn domain invariant features via directly minimizing the discrepancy of feature distribution. More recent approaches [63, 78, 238, 90] employed adversarial training in image level. [24, 57, 125, 187, 196] leveraged adversarial training to learn domain-invariant representations in feature level. There are some works [170, 225, 24, 204] using self-training to mitigate the domain gap via assigning labels to the most confident samples in the target domain.

2.1.2 Generalization of 2D Rigid Scenes

UDA for semantic segmentation Unsupervised domain adaptation for semantic segmentation is the task that applies domain adaptation for semantic segmentation tasks [236, 106, 134]. Semantic segmentation aims to classify pixels in images into one of the predefined semantic categories, thereby obtaining segments depicting the objects/scenes with semantic meanings, *e.g.*, chairs. Many approaches [227, 226, 24, 36, 63] have been proposed. There are mainly two ways to tackle this problem, *i.e.* via adversarial training or self-training. The works that exploited adversarial training can be categorized into the feature-level adaptation and the image level adaptation. Some works [186, 192, 200, 95, 46] adopted adversarial training at feature level to learn domain-invariant features to reduce the discrepancy across domains. [72, 31, 111] applied adversarial training at the image level to make features invariant to illumination, color and other style factors. Some recent approaches adopted self-training to perform adaptation. [242] proposed to assign pseudo labels in a curriculum way and [224, 243, 113, 232] combined self-training with other constraints to improve the quality of pseudo labels.

2.1.3 Generalization of 3D Rigid Scenes

Point Cloud Semantic Segmentation aims to perform semantic segmentation on the point clouds, in which each point, storing the Cartesian coordinate, is classified into predefined semantic categories. Previous researches in this area

could be categorized into three streams: 1) *point-based methods* propose to handle this task in a point-wise manner and aggregate the contextual information through MLP (Multiple Layer Perception) [153, 154, 50], GCN (Graph Convolutional Network) [195, 207], or newly designed convolutions [201, 184, 206, 229]. These methods typically require massive computation, making them hard to satisfy the latency constraint in real-world applications. 2) *Voxel-based methods* [94] try to convert point clouds into 3D voxels and employ 3D convolutions to learn the geometric distributions. Some researchers [241, 228] study the partition strategy in the 3D space, while some researchers [182, 70] propose new convolution architectures to handle the sparse 3D voxels. Also, the heavy computation cost of 3D CNN hampers their applicability to real-world applications. 3) *projection-based solution* is another routine focusing on transforming 3D point clouds into 2D grids so that 2D convolutions can be utilized directly. Various architectures[197, 198, 205, 135, 38] in this direction have been proposed to cope with the projected 2D images, and have been proven to be effective and efficient. Moreover, projection-based methods also receive increasing attention on other tasks [116, 181] for their low computation cost. In this paper, we choose the projection-based architecture to perform our adaptation task as they strike a better balance between performance and efficiency.

Domain Adaptation for Point Cloud Semantic Segmentation aims to adapt a segmentation model from a labeled source domain to an unlabeled target domain.

There are several works dealing with the 3D segmentation adaptation scenario [197, 165, 201, 172]. SqueezeSegV2 [198] fills the missing intensity channel for the synthetic point cloud and adopts geodesic correlation alignment [138] to perform point-wise feature alignment. As SynLiDAR [203] provides the intensity information, we directly utilize the intensity information along with coordinate information to train our model in this paper. Wu et al. [197] notice the domain shift is largely caused

by the target noise, and propose to impose noise masks on source samples, where the mask is denoted as the point-wise frequency of noise over the whole target dataset. Analogously, Rochan et al. [165] perform masking on source samples with a randomly selected noise mask from the target domain. However, the points from the same spatial positions of two point clouds may have different surroundings or contexts and thus they should not share the same probability to be noise. Zhao et al. [231] employ CycleGAN [238] to perform target noise inpainting which is then used to learn synthetic noise generation module. However, the noise generation module is trained by target inpainting results and thus may still be affected by domain shift. Moreover, different from ours, the noise generation of [231] cannot be optimized towards reducing domain shift.

2.1.4 Generalization of Non-Rigid Poses

2D pose estimation for human and animals. 2D pose estimation refers to identifying the position and orientation of the anatomical joints of bodies in images. There are mainly two paradigms, top-down and bottom-up. The top-down paradigm [149, 219, 14, 141, 137] first detects the person and then localizes the joints for each detected person. The bottom-up paradigm [194, 10, 193, 183, 91] first identifies joints in an identity-agnostic manner, then derive poses with different grouping strategies. The top-down solutions are more accurate while requiring higher computation due to the extra detection process, while the bottom-up solutions are more popular for their lower computation cost. Here we adopt the bottom-up solution for its efficiency and simplicity.

In terms of single-species pose estimation, [183] performs joint decomposition to strengthen the part with close correlations. However, they only employ the geometric clue inside single species while neglecting the structure variations across species and other forms of joint relations (*e.g.*, visual). Empirical experiments prove it is sub-

optimal for the cross-species scenario.

Domain adaptation and domain generalization. Domain adaptation aims to transfer the knowledge learned from a labeled source domain to an unlabeled target domain, which has attained remarkable progress in the past decades. Researches in domain adaptation overcome the distribution shift via alignment in the feature space directly [179, 189], or employing adversarial training in the input space [72, 63, 102] or feature space [130, 24, 57, 103].

Domain generalization extends to a more universal scenario, where multiple source domains and target domains are available and only source domains are accessible during the training. Research in this area can be categorized into two streams: one stream seeks to derive domain-invariant representations [140, 60, 105, 220], and the other attempts to enhance the generalizability of trained model parameters [199, 80] via meta-learning [45, 98, 223], or data augmentations [235, 149, 107, 30].

In terms of animal pose estimation, there are already some works considering the domain adaptation problem for it. [139] and [97] focus on the synthetic-to-real domain adaptation inside the same species. Despite making progress, these methods typically mitigate the domain gap induced by the difference in appearance and background, while the structural discrepancies inside the same species are negligible and are not considered as the main obstacle. [17] made the first attempt to mitigate the structural discrepancies between different species and realize it with an adversarial training paradigm and pseudo labeling. However, we argue it is suboptimal because their work mainly alleviates the impact resulting from the parts that exhibit large variation across species, while ignoring the parts holding stable inter-joint relationships.

2.2 Generalization to Novel Categories

2.2.1 Cross-domain Generalization to Novel Categories

Closed Set Domain Adaptation, also known as unsupervised domain adaptation (UDA), assumes two domains share identical label set. The main focus lies in how to minimize the distribution shift. Some methods minimize the discrepancy in the feature space directly [127, 126, 208, 54]. Some recent works take advantage of adversarial training to promote the alignment in the input space [40, 72, 63, 39] or feature space [187, 130, 24, 57]. Moreover, there are also some works performing adaptation via clustering in the target domain [88]. However, they could not trivially generalize to the unaligned label space.

Partial Domain Adaptation (PDA) holds an assumption that private classes only lie in the source domain, which has received wide attention recently. SAN [18] employs class-wise domain discriminators to align the distributions in a fine-grained manner. IWAN [222] proposes to identify common samples with the domain similarities from the domain discriminator, and utilizes the similarities as weights for adversarial training. Recently, ETN [19] proposes a progressive weighting scheme to estimate the transferability of source samples, while BA^{3}US [115] incorporates an augmentation scheme and a complement entropy to avoid negative transfer and uncertainty propagation, respectively.

Open Set Domain Adaptation (OSDA). Different settings [171, 147, 11, 51] have been investigated for the open set domain adaptation. In this paper, we mainly focus on the setting proposed by [171], where the target domain holds private classes that are unknown to the source. OSBP [171] proposes an adversarial learning framework that enables the feature generator to learn representations to achieve common-private separation. Recent works [125, 53] follow this paradigm to draw the knowledge from the domain discriminator to identify common samples that share the

semantic classes across domains. ROS [11] employs self-supervised learning technique to achieve the known/unknown separation and domain alignment.

Universal Domain Adaptation (UniDA), as a more challenging scenario, allows both domains having their own private classes. UAN [213] proposes a criterion to quantify sample-level uncertainty based on entropy and domain similarity. Then samples with lower uncertainty are encouraged for adaptation with higher weight. However, as pointed by [56], this measurement is not discriminative and robust enough. Fu *et al.* [56] designs a better criterion that combines entropy, confidence, and consistency from auxiliary classifiers to measure sample-level uncertainty. Similarly, a class-wise weighting mechanism is applied for subsequent adversarial alignment. However, they both treat private samples as one general class while ignoring the intrinsic structure of private samples.

2.2.2 Multi-Modality Generalization to Novel Categories

Closed-vocabulary object detection. Object detection [15, 7] is traditionally formulated as a closed-vocabulary detection task that performs object localization and classification under identical class set between training and inference. Mainstream algorithms are initially dominated by CNN-based frameworks (convolutional neural networks), which can be simply divided into one-stage and two-stage directors. The two-stage detectors [61, 161] first generate the region proposals and then selectively perform bounding box regression on them. Instead, one-stage detectors [124, 160] seek to learn bounding box offset relative to the anchor directly. More recently, the DETR-style detectors [20, 240] reformulate detection as a set prediction problem and achieve competitive performance with a more simple pipeline, *i.e.*, without hand-designed modules like anchor design. Especially, recent studies [101, 221, 123] introduce a denoising process on ground truth boxes and classes to accelerate the convergence of DETR with superior performance.

Zero-shot / Open-vocabulary object detection. Zero-shot detection [158, 1, 83, 239] aims to generalize the detector from seen object classes to unseen classes. Earlier works [3, 112] propose to replace the classifier with text embeddings of corresponding classes. The latter studies [157] explore how to associate the object region features with the text descriptions. Nevertheless, Zero-shot detection is still impractical for the real world, *i.e.*, not having any example of the unseen objects except the word embeddings.

In light of this, OVR-CNN [218] relaxes the constraint and proposed open-vocabulary detection (OV-Det), which allows training with image-caption pairs before performing zero-shot detection. The recent success of CLIP [156] inspires the community to mitigate the modali ty gap with the guidance of pretrained Vision-Languages models (VLMs). However, adapting VLMs to OV-Det is non-trivial for two reasons, *i.e.*, semantic ambiguity and positional ambiguity as aforementioned. To solve them, one stream [234, 52, 136, 211] seeks to establish explicit region-text correspondence through pretraining on external sources from Internet. Instead, the other stream [59] attempts to improve the region-text affinity directly from different perspectives. ViLD [66] absorbs the knowledge from VLMs through region-wise distillation. Another line of works assigns pseudo labels to reliable proposals, *i.e.*, according to max-size proposals [237] or constrained clustering [55] More recent studies consider it as a matching problem in the semantic space, *i.e.*, OV-DETR [217] proposes conditional-matching where the CLIP-embeddings serve as conditions in query formulation, VLDet [117] clarifies the region-text correspondence by solving it as a set-matching problem, and F-VLM proposes a fusion strategy for the class prediction score that combines text and region features.

Chapter 3

Content-Consistent Matching for Domain Adaptive Semantic Segmentation

Following the foundational concepts introduced in Chapter 2, this chapter delves into the specific realm of 2D rigid scenes. This chapter will focus on domain adaptation in these scenes, identifying the unique challenges they present and building on the principles of generalization learning to address these challenges and develop viable solutions.

3.1 Introduction

Semantic segmentation [25, 126, 230, 118, 27, 79, 33] plays an important role in many real-world applications. However, in practice, an off-the-shelf segmentation model trained on one scenario (source) usually cannot generalize well to the new one (target). For example, for the self-driving task, we may collect data and train a segmentation model in one city, but such a model may fail to give accurate pixel-level predictions for the scenes of another unfamiliar city. As achieving massive in-domain pixel-level annotations are expensive and sometimes impossible, practitioners usually resort to domain adaptive training to achieve satisfactory results on the target.

Generally, a domain adaptive segmentation model is established on the labeled source images and the unlabeled target images. And the training tries to utilize the knowledge learned from the source and mitigate the domain shift. Previous methods usually achieved the adapted model through the adversarial training [186, 192, 129, 95, 46, 72, 24, 111] or by self-training [242, 243, 113, 224]. All those methods

Figure 3.1 : Examples of positive and negative source samples (Best viewed in color). Generally, positive source images (c) share similar layout with the target (b) while the negative source ones (a) do not. Intuitively, samples like (c) should be selected and samples like (a) should be excluded to help the adaptation. Moreover, the heatmap (d) indicates the pixel-wise similarities between target and positive source embeddings, in which red indicates higher similarity. It can be seen that the similarities vary a lot, even for the semantic-consistent pixels of the source image, which implies the pixels of positive source images should not be equally treated. Detailed information could be found in Sec. 3.2.2.

employed the entire source domain data throughout the training process, which neglects the fact that not all the source images could contribute to the improvement of adaptation performance, especially at certain training stages. We empirically find that there usually exist "negative" source images which may even harm the adaptation. As shown in Fig. 3.1, compared to the positive source images, the negative ones appear quite dissimilar to the target images. Moreover, from Fig. 3.1, we observe that for the visually similar pair of source (*i.e* positive source image) and target images, the pixel-wise similarities vary a lot spatially, which implies that the pixels of a source image also should not be treated equally.

In this paper, we propose Content-Consistent Matching (CCM) to match and select the effective source information actively to facilitate the adaptation process. To be specific, we perform *Semantic Layout Matching* to select the positive source samples and *Pixel-wise Similarity Matching* to emphasize effective pixels. For the Semantic Layout Matching, we propose a novel image representation that encodes the semantic layout information. Based on such semantic layout representation, we perform clustering on the target to discover the underlying patterns in the target domain. Then the source sample is selected as the positive one if it is close enough to these patterns. Moreover, we further select the positive source pixels to mitigate the negative transfer through proposed pixel-wise similarity matching. Similar to the matching strategy in the image level, pixel-wise similarity matching selects the source pixels that share similar feature distributions with the target samples.

As the target feature evolves during training, the same source sample may contribute differently to the adaptation process, *e.g.* a source sample could be negative before a certain stage but positive afterward. Thus we choose to iteratively update the matching results during training to enable more effective adaptation. Specifically, we perform the CCM along with a self-training paradigm, *i.e.* we alternatively update the representations through self-training and the source matching results through CCM. These two parts depend on each other and cooperate to mitigate the domain shift.

In a nutshell, our contributions are three-fold:

- We deal with the domain adaptive segmentation task from a new perspective, *i.e.* actively selecting positive source information for training to avoid negative transfer, which has not been investigated by previous methods.
- We propose Content-Consistent Matching (CCM), which consists of Semantic Layout Matching and Pixel-wise Similarity Matching, to select the positive

source samples and their positive pixels to facilitate the adaptation process.

- Experiments on two representative benchmarks (*i.e.* GTA5 → Cityscapes and SYNTHIA → Cityscapes) demonstrate that our method performs favorably against previous methods. Ablation studies also verify the effectiveness of the key components of our framework.

3.2 The Proposed Approach

We aim to train a segmentation network with parameters θ to give accurate pixel-level predictions $P(c|x, y; I^T, \theta)$ on the target T, where $c \in \{0, 1, \cdots, C-1\}$ denotes the underlying categories and $x \in \{0, 1, \cdots, W-1\}, y \in \{0, 1, \cdots, H-1\}$ are the horizontal and vertical coordinates of a pixel in a target image I^T, respectively. The segmentation network is trained with the combination of labeled source images D^S and unlabeled target images D^T. During training, we propose to use content-consistent matching (CCM) to select positive source samples and their effective pixels. Our CCM is performed upon the self-training paradigm, *i.e.* with selected positive source samples and their effective pixels, the network is trained with ground-truth source labels and pseudo target labels to update the feature representation, and based on the updated feature representation, the set of positive source samples and pixels are reconstructed.

3.2.1 Semantic Layout Matching

Semantic layout means how the categories are distributed spatially in an image (*i.e.* $P(x, y|c)$). It could be an important prior during the training of segmentation models. However, the semantic layout patterns may vary a lot across domains, leading to the domain shift and degenerating the generalization. For example, it is natural that part of the source domain images is captured from a distinct perspective compared to the target. Thus the semantic layout of these source images will be

quite different from the target. In this section, we propose using semantic layout matching to select the positive source samples to mitigate such domain shift.

Semantic Layout Matrix (SLM)

Directly using $P(x, y|c)$ to model the semantic layout would be inefficient due to its high dimension, and ineffective because it is not robust to the inaccurate target predictions.

Following the naive Bayes assumption, we propose to decouple $P(x, y|c)$ into the horizontal one $P(x|c)$ and vertical $P(y|c)$ one, *i.e.*,

$$P(x, y|c) \propto P(x|c)P(y|c). \tag{3.1}$$

Specifically, take the vertical distribution $P(y|c)$ for an example, $P(y|c)$ can be represented as

$$P(y|c) = \frac{P(c|y)P(y)}{P(c)} = \frac{\sum_x P(c|x, y)P(x)P(y)}{\sum_x \sum_y P(c|x, y)P(x)P(y)}, \tag{3.2}$$

Assuming $P(x)$ and $P(y)$ are the uniform distributions, *i.e.* $P(x = i) = \frac{1}{W}, i \in \{0, 1, \cdots, W-1\}$ and $P(y = j) = \frac{1}{H}, j \in \{0, 1, \cdots, H-1\}$, then

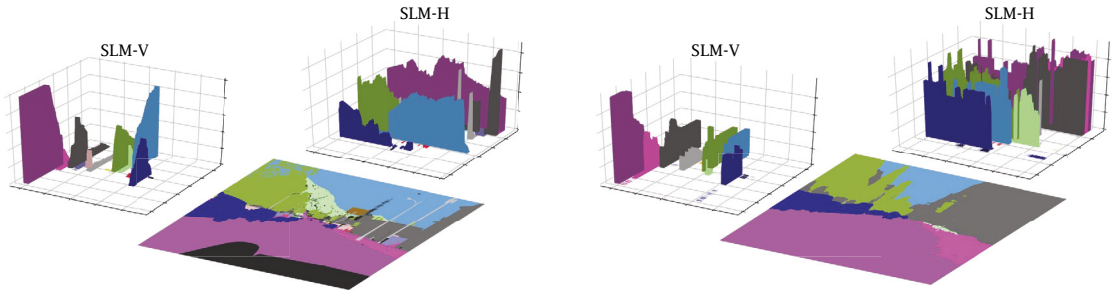
$$P(y|c) = \frac{\sum_x P(c|x, y)}{\sum_x \sum_y P(c|x, y)}. \tag{3.3}$$

For the source image, suppose its ground-truth label is c',

$$P(c|x, y; I^S) = \begin{cases} 1 & \text{if } c = c' \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

For the target images, as we don't know the ground-truth pixel-wise labels, we adopt the probability predictions $P(c|x, y; I^T, \theta)$ of current segmentation model with parameters θ to compute Eq. (3.3).

Following the general practice, the images (source and target) are customized as the same size during training, and the vertical semantic layout matrix $M_v \in \mathbb{R}^{C \times H}$



(a) Target Prediction and its SLM

(b) Positive Source Label and its SLM

Figure 3.2 : Illustration of SLM (Best viewed in color). Taking the class sky (annotated with sky blue) as an example, we explain how to use SLM to represent the spatial distribution. From SLM-V, we could know its vertical distribution: most of the pixels belonging to the sky are at the top. Through SLM-H, we could also found that the sky mostly lies in the middle and right of the images. (a) and (b) are pairs that share most similar spatial distributions.

can be expressed as (we omit the domain subscript for simplification)

$$M_v(\hat{c}, j) = \frac{\sum_x P(\hat{c}|x, y = j)}{\sum_x \sum_y P(\hat{c}|x, y)}. \tag{3.5}$$

Similarly, the horizontal semantic layout matrix $M_h \in \mathbb{R}^{C \times W}$ is

$$M_h(\hat{c}, i) = \frac{\sum_y P(\hat{c}|x = i, y)}{\sum_x \sum_y P(\hat{c}|x, y)}. \tag{3.6}$$

Finally, the semantic layout matrix $M \in \mathbb{R}^{C \times (H+W)}$ can be represented as

$$M = \left[M_h, M_v \right]. \tag{3.7}$$

Note that because the assumption in Eq. (3.1) may not be exactly satisfied in practice, we choose to concatenate the horizontal and vertical semantic layout matrix together rather than multiply them, which makes the training less dependent on the conditional independence assumption.

Matching and selection

Based on the proposed SLM, we can encode the semantic layout of each target image. We then adopt k-means clustering to discover the underlying K patterns of target SLMs. The source image, which is close enough to these patterns, can be viewed as a positive sample. Note that because we perform single-direction selection, clustering on the source images is not needed.

Specifically, we denote the centers of these K target clusters as $\hat{M}^{T,k}, k \in \{0, 1, \cdots, K-1\}$ and compute the similarity between the source sample and each of these cluster centers through

$$Sim(M^S, \hat{M}^{T,k}) = -\sum_c D_{KL}(\hat{M}_h^{T,k}(c,:)||M_h^S(c,:)) + D_{KL}(\hat{M}_v^{T,k}(c,:)||M_v^S(c,:)), \tag{3.8}$$

where the $D_{KL}(\cdot||\cdot)$ denotes the KL divergence. And the similarity score of a source image is

$$Score(M^S) = \frac{1}{K}\sum_k Sim(M^S, \hat{M}^{T,k}). \tag{3.9}$$

Based on the ranking of the above similarity scores among all the source samples, only the top-ranking source samples are selected for training. In our experiment, we set the selection proportion γ_{img} as a hyper-parameter to control the number of selected source images. We will discuss this further in our experiment part. Our proposed SLM is illustrated in Fig. 3.2.

Indeed, it is important to note that the layout distribution is one of the critical factors inducing the domain gap in semantic segmentation. When dealing with two domains that are closely correlated in terms of semantic layouts, the domain gap can often be attributed more to other elements, such as appearance and styles.

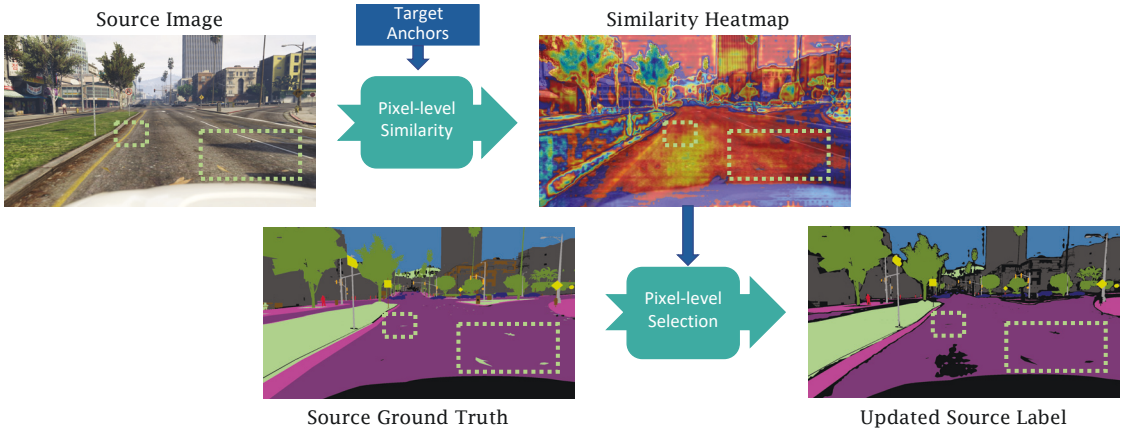


Figure 3.3 : Illustration of pixel-wise similarity matching. As marked by the green box, the leaves on the road are hardly spotted even by a human but annotated in the ground truth. Pixel-wise similarity matching excludes these pixels which may hinder the adaptation. The black area in the figure denotes the ignored pixels. Best viewed in color.

3.2.2 Pixel-wise Similarity Matching

For a source image, it is possible that partial regions or pixels are similar to the target, while others are not. That means the pixels in a source image should not be equally treated during the adaptation. Thus besides selecting positive source samples, we propose to select the positive source pixels that share similar characters with the target to mitigate the domain shift further. We name such pixel-level selection as pixel-wise similarity matching, as illustrated in Fig. 3.3.

For a target image, based on the network's outputs $P(c|x, y; I^T, \theta)$, we could assign a pseudo label to each pixel, *i.e.*

$$L^T(x, y) = \arg\max_c P(c|x, y; I^T, \theta). \tag{3.10}$$

Then the pixels are classified into C groups. The pixel with low confidence prediction $P(L^T(x, y)|x, y; I^T, \theta)$ is filtered (see Section 3.2.3 for more details). And

the average class distribution is calculated among each group

$$Q^T(c) = \frac{1}{|D^T|} \sum_i \frac{1}{|G^T_{i,c}|} \sum_{(\hat{x},\hat{y}) \in G^T_{i,c}} P_i(c|\hat{x}, \hat{y}; I^T_i, \theta), c \in \{0, 1, \cdots, C-1\}, \quad (3.11)$$

where $G^T_{i,c} = \{(\hat{x}, \hat{y}) | L^T_i(\hat{x}, \hat{y}) = c\}$, $Q^T(c) \in \mathbb{R}^C$, and the subscript i denotes the i-th target sample here, $|D^T|$ is the number of samples within target domain. By this way, it is expected that $Q^T(c)$ could describe the relationships between class c and all the other classes, based on the current predictions of the network.

From the network, we can also obtain the predictions for the source image, *i.e.* $P(c|x, y; I^S, \theta)$, thus it is natural to select such source pixels $\{(\tilde{x}, \tilde{y})\}$, where $P(c|\tilde{x}, \tilde{y}; I^S, \theta)$ matches well with Q^T. We adopt KL divergence to measure the distance between each pair of $P(c|x, y; I^S, \theta)$ and $Q^T(c)$. And the matching score for a source pixel at (x, y) with ground-truth label c is computed as

$$Score(x, y) = -D_{KL}(Q^T(c)||P(c|x, y; I^S, \theta)). \quad (3.12)$$

We rank the source pixels within the same ground-truth class according to their similarity score and select the top ranking pixels for each class. In our experiment, we select the same proportion of pixels γ_{pix} for each class.

3.2.3 Active Matching with Self-training

As the target feature evolves, the effect of the same source sample on the adaptation process may be different. In this paper, we choose to update the source matching results actively throughout the adaptation process. Notice that purely training with the source data may lead to the model biased towards the source distribution, we choose to employ our matching strategy along with the self-training paradigm.

To obtain a good initialization of target predictions, we start the self-training from the segmentation network trained on all the labeled source images D^S. Then

we alternatively update the network parameters θ and assign pseudo labels $L^T(x, y)$ on the target D^T according to Eq. (3.10).

Through pseudo labeling, the target pixels are grouped into C classes. For each class of pixels, we rank them according to the prediction confidences (*i.e.* $P(L^T(x, y)|x, y; I^T, \theta)$). Only the top ranking target pixels are selected for training, and the ratio of selection is set to r, which is shared among all the classes. To enable each target sample to have enough selected pixels, we also perform pixel ranking within each image. Then the top r pixels of a target image are also selected.The selected pixels $\{(\hat{x}, \hat{y})\}$ are assumed to have reliable pseudo labels.

The positive source samples $\hat{D}^S$ selected through our matching strategy, together with the pseudo-labeled target samples D^T, are adopted to train the network. And the network is trained with pixel-wise cross-entropy loss

$$\mathcal{L}_{ce} = \mathcal{L}_{ce}(\hat{D}^S; \theta) + \mathcal{L}_{ce}(D^T; \theta), \tag{3.13}$$

where

$$\mathcal{L}_{ce}(\hat{D}^S; \theta) = -\sum_i \sum_{(\tilde{x},\tilde{y})\in I_i^S} \log[P(L_i^S(\tilde{x}, \tilde{y})|\tilde{x}, \tilde{y}; I_i^S, \theta)], \tag{3.14}$$

$$\mathcal{L}_{ce}(D^T; \theta) = -\sum_i \sum_{(\hat{x},\hat{y})\in I_i^T} \log[P(L_i^T(\hat{x}, \hat{y})|\hat{x}, \hat{y}; I_i^T, \theta)]. \tag{3.15}$$

In Eq. (3.14) and Eq. (3.15), I_i^S and I_i^T are the i-th images in the $\hat{D}^S$ and D^T, respectively. Note that only the gradients coming from the positive source pixels $(\tilde{x}, \tilde{y}) \in I^S$ and target pixels $(\hat{x}, \hat{y})$ with reliable pseudo labels are back-propagated in each iteration.

Algorithm 1: Content-Consistent Matching

Input: parameters θ; source images D^S and labels L^S, target images D^T

Initialize θ with source trained segmentation model;

for *m=1 to M* **do**

  Update target pseudo labels L^T for each $I^T \in D^T$ and select target pixels $(\hat{x}, \hat{y})$ with reliable pseudo labels;

  Select positive source samples $\hat{D}^S$ and their positive pixels $(\tilde{x}, \tilde{y})$;

  for *n=1 to N* **do**

    1) forward and compute the $\mathcal{L}$ according to Eq. (3.18);

    2) back-propagating the gradients and updating θ;

  end

end

3.2.4 Objective

Additionally, we introduce entropy regularization to regularize the adaptation

$$\mathcal{L}_{ent}(D^S;\theta) = -\sum_i \sum_c \sum_{(x,y)\in I_i^S} P(c|x,y;I_i^S,\theta)\log[P(c|x,y;I_i^S,\theta)], \tag{3.16}$$

$$\mathcal{L}_{ent}(D^T;\theta) = -\sum_i \sum_c \sum_{(x,y)\in I_i^T} P(c|x,y;I_i^T,\theta)\log[P(c|x,y;I_i^T,\theta)], \tag{3.17}$$

And the entropy regularization is imposed on all the source and target images.

In total, the objective of our training procedure is

$$\mathcal{L} = \mathcal{L}_{ce}(\hat{D}^S;\theta) + \mathcal{L}_{ce}(D^T;\theta) + \lambda(\mathcal{L}_{ent}(D^S;\theta) + \mathcal{L}_{ent}(D^T;\theta)). \tag{3.18}$$

where λ is a constant indicating the strength of entropy regularization.

Our algorithm is summarized in Algorithm 1. Note that we update the target pseudo labels and perform source selection every N steps of network update. We perform selection and network update in such an asynchronous way because the

network update is a relatively slower process, and this way enables more efficient and effective training.

3.3 Experiments

3.3.1 Experimental Setup

Here we evaluate our methods on two popular transfer tasks, GTA5 [163] → Cityscapes [37] and SYNTHIA [166] → Cityscapes. For the source dataset, GTA5 contains 24996 images with resolution 1914 × 1052, and SYNTHIA contains 9400 images with resolution 1280 × 760. For the target, Cityscapes contains 2975 images for training and 500 images for validation with image resolution 2048 × 1024. Following the settings in [129, 186, 192], we train the model on the source dataset (GTA5 or SYNTHIA) and the training set of Cityscapes and report the result on the validation set of Cityscapes. We only transfer on the classes shared between the source domain and the target domain. For the evaluation metric, we evaluate our methods with mean Intersection over Union (mIoU).

Implementation Detail We start from DeepLabV2-Res101 [26, 69] with the backbone pretrained on the ImageNet [43]. Then we first finetune the whole network on the source data and use such a source-trained network to initialize the target (adaptation) model.

We choose to use Stochastic Gradient Descent (SGD) with momentum of 0.9 and weight decay of 5×10^{-4}. The learning rate decreases following the poly policy with power at 0.9. The initial learning rate is set to 7.5×10^{-5}. The M in Algorithm 1 is set to 6 and the N is set to 2 epochs, *i.e.* we train for 6 loops where each loop contains 2 epochs. For all the transfer tasks, the hyper-parameters γ_{img}, λ, r, and K are set to 0.4, 0.4, 0.1, and 10 respectively. The γ_{pix} is set to 0.9, 0.6 for GTA5 and SYNTHIA respectively.

Table 3.1 : Experiment results of GTA5 → Cityscapes. The "AT" and "ST" denote approaches established on adversarial training and self-training, respectively, while "AS" indicates methods utilizing both. We highlight the best in each column in **bold**.

GTA5 → CityScapes																					
	Meth.	road	side.	buil.	wall	fence	pole	light	sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	motor	bike	mIoU
Source Only	–	60.6	17.4	73.9	17.6	20.6	21.9	31.7	15.3	79.8	18.1	71.1	55.2	22.8	68.1	32.3	13.8	3.4	34.1	21.2	35.7
AdaptSeg[186]	AT	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
ADVENT[192]	AT	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
CLAN[129]	AT	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
DISE[23]	AT	91.5	47.5	82.5	31.3	25.6	33.0	33.7	25.8	82.7	28.8	82.7	62.4	30.8	85.2	27.7	34.5	6.4	25.2	24.4	45.4
SWD[95]	AT	92.0	46.4	82.4	24.8	24.0	35.1	33.4	34.2	83.6	30.4	80.9	56.9	21.9	82.0	24.4	28.7	6.1	25.0	33.6	44.5
SSF-DAN[46]	AT	90.3	38.9	81.7	24.8	22.9	30.5	37.0	21.2	84.8	38.8	76.9	58.8	30.7	85.7	30.6	38.1	5.9	28.3	36.9	45.4
MaxSquare[28]	–	89.4	43.0	82.1	30.5	21.3	30.3	34.7	24.0	85.3	39.4	78.2	63.0	22.9	84.6	36.4	43.0	5.5	**34.7**	33.5	46.4
MRNet[233]	–	90.5	35.0	84.6	34.3	24.0	36.8	**44.1**	42.7	84.5	33.6	82.5	**63.1**	34.4	**85.8**	32.9	38.2	2.0	27.1	41.8	48.3
PyCDA[113]	ST	90.5	36.3	84.4	32.4	**28.7**	34.6	36.4	31.5	**86.8**	37.9	78.5	62.3	21.5	85.6	27.9	34.8	18.0	22.9	**49.3**	47.4
CRST[243]	ST	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
CAG[224]	ST	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
SIM[196]	ST	90.1	44.7	**84.8**	34.3	**28.7**	31.6	35.0	37.6	84.7	43.3	85.3	57.0	31.5	83.8	42.6	48.5	1.9	30.4	39.0	49.2
BDL[233]	AS	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
Ours (CCM)	ST	**93.5**	**57.6**	84.6	**39.3**	24.1	25,2	35.0	17.3	85.0	40.6	**86.5**	58.7	28.7	**85.8**	**49.0**	**56.4**	5.4	31.9	43.2	**49.9**

For image preprocessing, we resize the shorter side of images to 720 and crop a patch with resolution 600 × 600 randomly. Besides, horizontal flip and random scale between 0.5 and 1.5 are introduced as data augmentation. For evaluation, images from Cityscapes are resized to 1024 × 512 as input and the mIoU is calculated on predictions upsampled to 2048 × 1024.

3.3.2 Comparison with Previous Methods

We evaluate our method on two unsupervised domain adaptation tasks: GTA5 → Cityscapes and SYNTHIA → Cityscapes. The results are presented in Table 3.1 and Table 3.2, respectively. In both tables, we use "AT" and "ST" to denote approaches established on adversarial training and self-training respectively, while "AS" in-

Table 3.2 : Experiment results of SYNTHIA → Cityscapes. The mIoU* denotes the mean IoU over classes without "*".

SYNTHIA → CityScapes																			
	Meth.	road	side.	buil.	wall*	fence*	pole*	light	sign	vege.	sky	pers.	rider	car	bus	motor	bike	mIoU	mIoU*
Source Only	–	47.1	23.3	75.6	7.1	0.1	23.9	5.1	9.2	74.0	73.5	51.1	20.9	39.1	17.7	18.4	34.0	34.5	40.1
AdaptSeg[186]	AT	84.3	42.7	77.5	–	–	–	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	–	46.7
ADVENT[192]	AT	85.6	42.2	79.7	–	–	–	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	–	48.0
CLAN[129]	AT	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
SSF-DAN[46]	AT	84.6	41.7	80.8	–	–	–	11.5	14.7	80.8	**85.3**	57.5	21.6	82.0	36.0	19.3	34.5	–	50.0
MaxSquare[28]	–	82.9	40.7	80.3	10.2	**0.8**	25.8	12.8	18.2	82.5	82.2	53.1	18.0	79.0	31.4	10.4	35.6	41.4	48.2
CAG[224]	ST	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	–
pyCDA[113]	ST	75.5	30.9	**83.3**	**20.8**	0.7	32.7	**27.3**	**33.5**	**84.7**	85.0	64.1	25.4	**85.0**	45.2	21.2	32.0	**46.7**	**53.3**
SIM[196]	ST	83.0	44.0	80.3	–	–	–	17.1	15.8	80.5	81.8	59.9	**33.1**	70.2	37.3	28.5	45.8	–	52.1
BDL[111]	AS	**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
ours (CCM)	ST	79.6	36.4	80.6	13.3	0.3	25.5	22.4	14.9	81.8	77.4	56.8	25.9	80.7	**45.27**	**29.9**	**52.0**	45.2	52.9

dicates methods utilizing both. All the models are based on DeepLabV2-Res101 backbone, except that pyCDA [113] is based on PSPNet [230] and CAG [224] is based on DeepLabV3+ [27]*. It can be seen that our method outperforms source only baseline with a large margin, which verifies the effectiveness of our approach.

For the task GTA5 → Cityscapes, we achieve 49.9% on mIoU, comparable to previous state-of-the-art method CAG [224] (50.2%). Notably, CAG adopts a more advanced backbone (DeeplabV3 vs. DeeplabV2) and a larger input resolution (*i.e.* 2200 × 1100), which naturally brings higher performance with higher model capabilities. Despite this, our method is sill on par with it. For the task from SYNTHIA → Cityscapes, to make a fair comparison, we report the mIoU on 13 classes (excluding "Wall", "Fence", and "Pole") and 16 classes. Our method achieves 52.9%

*https://github.com/RogerZhangzz/CAG_UDA/issues/6

Table 3.3 : Effect of different key components. The "CCM-SLM" stands for semantic layout matching, and "CCM-Fix" denotes source samples and pixels are only selected at the start of self-training. All the results are compared with our self-training baseline. Self-training means the network trained with cross-entropy loss and entropy regularization, without the source selection via CCM.

Module	GTA5→Cityscapes		SYNTHIA→Cityscapes	
	mIoU	Gain	mIoU	Gain
Self-training	48.1	-	41.2	-
CCM-Fix	48.9	+0.8	44.2	+3.0
CCM-SLM	48.8	+0.7	41.9	+0.7
CCM	49.9	+1.8	45.2	+4.0

and 45.2% mIoU on 13 classes and 16 classes respectively, both of which perform favorably against previous state-of-the-arts.

Specifically, despite its simplicity, CCM outperforms previous state-of-the-art adversarial-training (denote as "AT") based method "SSF-DAN" [46] by +4.5% and +2.9% on GTA5 → Cityscapes and SYNTHIA → Cityscapes, respectively. Compared with methods established on self-training, CCM achieves comparable or even better results. For example, our method is on par with pyCDA [113], *i.e.* 49.9% (ours) vs. 47.4% (pyCDA) on GTA5 → Cityscapes and 45.2%/52.9% (mIoU and mIoU* of ours) vs. 46.7%/53.3% (mIoU and mIoU* of pyCDA) on SYNTHIA → Cityscapes. It is worth noting that pyCDA adopts AdaBN [110] to enhance its adaptation performance, which additionally calibrates the low-level statistics of features between two domains. This technique can also be employed in our framework to improve the performance further. Also, our method mainly focuses on selecting positive source information to mitigate the domain shift and help the

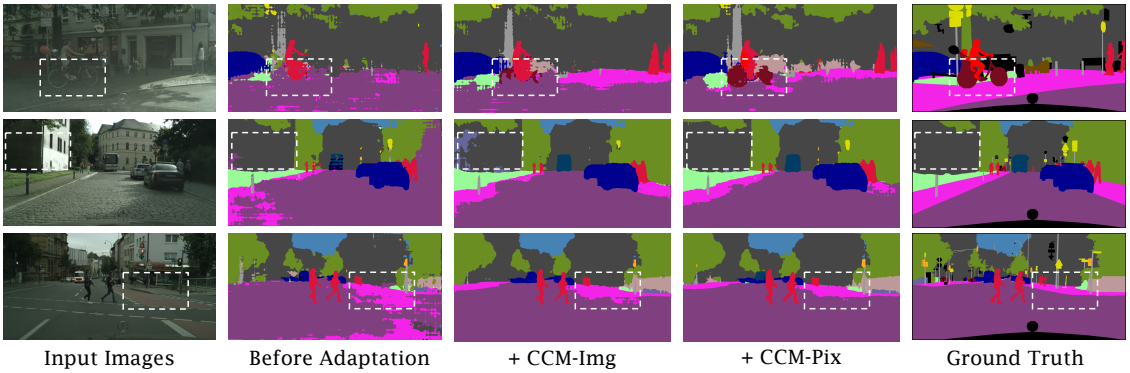


Figure 3.4 : Visualization of the segmentation results (GTA5 → Cityscapes). Pay attention to the dashed box to see the effect of different modules.

adaptation, which is also complementary to these methods and can be combined with them to boost the adaptation performance.

3.3.3 Ablation Studies

Effect of different key components

We verify the effect of each key component in our framework in Table 3.3. It can be seen that compared to the source-only results, self-training improves the adaptation performance apparently. Despite such a strong baseline, "CCM-SLM", which selects positive source samples through proposed SLM, improves beyond self-training by +0.7% on both tasks. Further, through combining SLM with pixel similarity matching, "CCM" improves beyond self-training by +1.8% and +4.0% for the GTA5 → Cityscapes and SYNTHIA → Cityscapes, respectively. These noticeable performance gains verify the effectiveness of the proposed matching and selection strategy.

Fig. 3.4 gives an intuitive illustration about the effect of CCM. It can be seen that through adaptation with SLM, the pixel-level predictions have been largely improved. Further, pixel-wise similarity matching enables the adapted model to

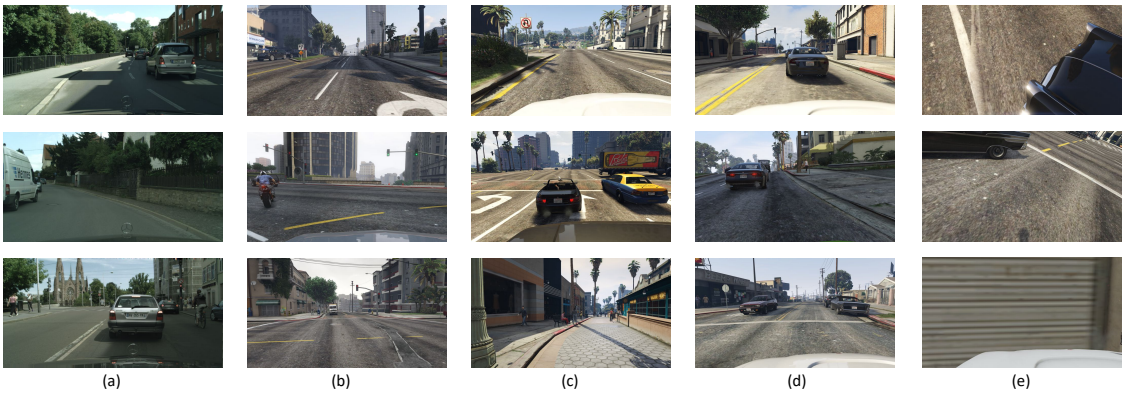


Figure 3.5 : Examples retrieved by semantic layout matrix(SLM). In each row, (b-e) are source images retrieved by target sample (a), where (b-d) are positive samples and (e) are negative ones.

learn more details about the object and thus leads to more accurate predictions.

Compared with the "CCM-Fix" which only selects positive source samples and their positive pixels at the start of self-training and adopts them throughout the adaptation, actively update the positive source set (denoted as "CCM") achieves noticeable improvement (*i.e.* +1.0% for both tasks). This is because source samples may contribute differently to the adaptation at different training stages and the matching results should be updated as the target predictions evolve, The results imply that self-training and CCM could benefit each other and cooperate to mitigate the domain shift.

Visualization of semantic layout matching results

In Fig. 3.5, we show the source images retrieved by individual target images via semantic layout matching at the final training stage. In each row, (c-e) are source images retrieved by the target sample (a) via SLM, in which (b-d) are top positive samples and (e) are negative ones with the lowest matching scores.

It can be seen that similar layout is shared among the target samples and the

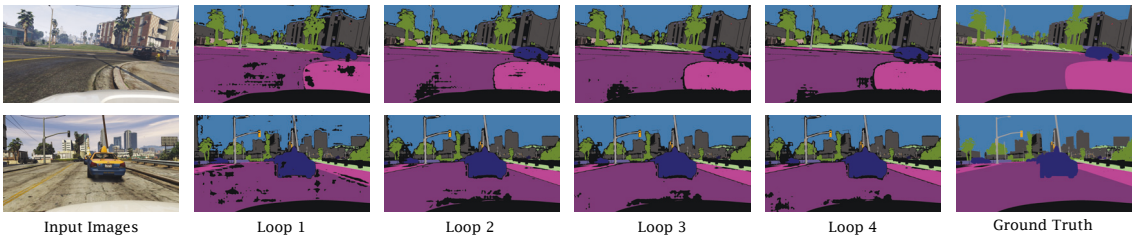


Figure 3.6 : Visualization of selected source pixels at different training stages. As the training goes on, the ignored source pixels become more and more concentrated on the object boundary. The black area denotes the ignored pixels during training. The results are based on task GTA5 → Cityscapes.

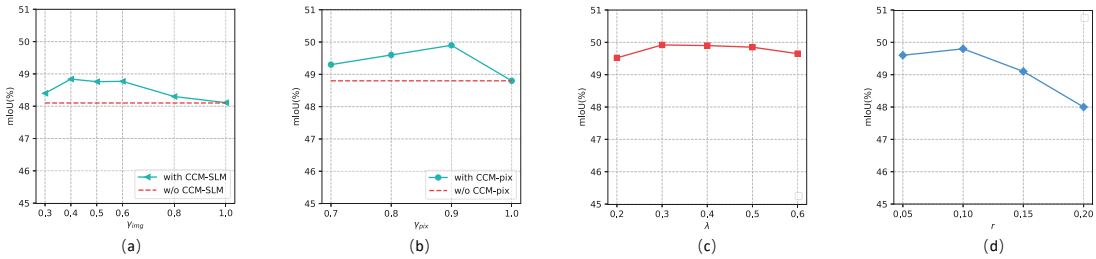


Figure 3.7 : (a): Performance with/without CCM-SLM under different γ_{img}. (b): Performance with/without CCM-Pix under different γ_{pix}. (c): Sensitivity analysis of λ. (d): Sensitivity analysis of r. The results shown are based on the task GTA5 → Cityscapes. The trend on another task is similar.

retrieved source samples. For example, all positive source samples in the first row have trees on the left. Moreover, it is also obvious that negative source samples on the right-most column (e) have totally different layout. Additionally, the retrieved source samples remain reasonable variations in appearances, which will also benefit the generalization of adapted model. All of these results give an intuitive illustration why semantic layout matching can help reduce the domain shift and improve the generalization ability.

Table 3.4 : Sensitivity to K

K	5	10	15
mIoU(%)	49.8	49.9	49.7

Effect of pixel-wise similarity matching

Fig. 3.6 demonstrates the selected source pixels through pixel-wise similarity matching during the training. It can be seen that as the training goes on, the ignored pixels become more and more concentrated on the object boundary, which is reasonable and implies that the adaptation keeps improving. Moreover, at the early stage, we notice that the ignored pixels are ambiguous ones that are hard to distinguish, *e.g.* the pixels of the cracks on the road. The pixel selection through pixel-wise similarity matching enables the model to learn in a curriculum way to an extent.

Sensitivity to the hyper-parameters

We investigate the sensitivity of our method to the hyper-parameters γ_{img}, γ_{pix}, λ, r, and show the results in Fig. 3.7. From Fig. 3.7 (a), it can be seen that trained with SLM, with the increase of γ_{img}, the mIoU firstly increases then decreases, illustrating a bell shape curve. The mIoU decreases when γ_{img} is above a certain threshold, indicating that there exist negative samples harming the adaptation and it is necessary to perform source sample selection to exclude such negative samples. With SLM, the optimal mIoU achieved is 0.7% higher than that trained without SLM. The trend of sensitivity to γ_{pix} is similar to that of γ_{img}. Our method achieves consistent improvement over baselines (the red lines) within a wide range of γ_{img} and γ_{pix}.

As illustrated in Fig. 3.7 (c), entropy regularization provides consistent improve-

ment within a wide range of λ. From Fig. 3.7 (d), we observe that our method is also robust to r within a wide range. When r is above a certain threshold, the performance drops because more inaccurate target data is involved in training. Besides, we analyze the sensitivity of our model to K and report the results in Table 3.4, which further verifies the robustness of our approach.

3.4 Conclusion and Discussion

In this chapter, we propose using Content-Consistent Matching (CCM), which consists of Semantic Layout Matching and Pixel-wise Similarity Matching, to match and select positive source data to facilitate the adaptive training of the segmentation model. Our matching strategy is performed from both the image-level and the pixel-level, *i.e.* semantic layout matching selects the positive source samples, and pixel-wise similarity matching emphasizes the effective source pixels. Experiment results on two representative benchmarks demonstrate that our method performs favorably against previous state-of-the-arts.

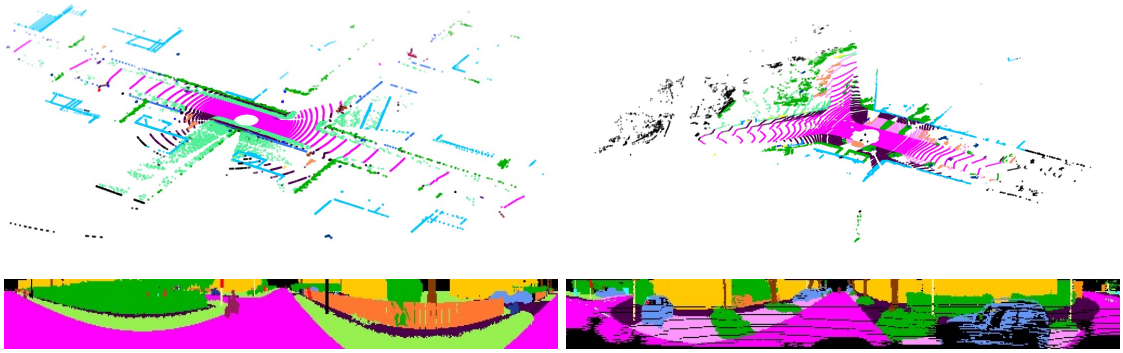
Chapter 4

Adversarially Masking Synthetic to Mimic Real: Adaptive Noise Injection for Point Cloud Segmentation Adaptation

Expanding on the previous exploration of 2D scenes, this chapter shifts the focus to domain adaptation in 3D rigid scenes. By exploring the added complexity and depth of 3D structures, this chapter aims to further our understanding of how domain adaptation principles can be effectively applied in varying structural contexts.

4.1 Introduction

Recently, point cloud semantic segmentation task attracts increasing attention because of its important role in various real-world applications, *e.g.*, autonomous driving, augmented reality, *etc.* Despite remarkable progress [74, 153, 114, 212, 216, 150, 35], most algorithms are designed for the fully-supervised setting, where massive annotated data is available. In the real world, it is costly and time-consuming to annotate large amounts of data, especially for labeling each point in the segmentation task. Synthetic data is easy to obtain and its label can be automatically generated, which largely reduces the human effort of annotating data. However, it is usually infeasible to directly apply networks trained on synthetic data to real-world data due to the apparent domain gap between them. In this chapter, we consider the synthetic-to-real domain adaptation [145, 152, 49, 175, 44, 202, 104, 155] for point cloud segmentation. Specifically, we aim to utilize the fully-annotated synthetic point clouds (source domain) and unlabeled point clouds collected from imperfect real-world sensors (target domain) to train a network to support the segmentation



LiDAR scan from SyncLiDAR

LiDAR scan form SemKITTI

Figure 4.1 : Comparison between a synthetic LiDAR scan (upper) and a real scan (lower). Both original point clouds and projected LiDAR images are given. Black points denote noise and other colors denote points from different classes. Compared with synthetic data which is integral and clean, point clouds collected from the real world typically contain unexpected and irregular noise which may impede the adaptation.

of real-world point clouds (target domain).

Domain adaptation solutions aim to discover and mitigate the domain shift from source to target domain. Through comparing the synthetic and real-world point clouds, we observe that the domain shift can be largely attributed to the unexpected and irregular noise existing in the target domain data. As with [197], we consider "noise" to be the missing points of certain instances/objects, where all pixel channels are zero. Such noise may be caused by various factors such as non-reflective surfaces (*e.g.*, glass). As shown in Fig. 4.1, the synthetic point cloud is integral and clean, but the real one contains large amounts of noisy points. A model trained on clean source data may find it hard to understand the scene context under the distraction of noises and thus cannot achieve satisfactory segmentation results on target point clouds.

Previous domain adaptation methods [210, 62, 63, 23, 77, 76, 145, 113, 73, 103]

(*e.g.*, adversarial training), which have been proven effective in the 2D visual tasks, can be applied to this 3D segmentation setting. For example, SqueezeSegV2 [198] employs geodesic correlation alignment [138] to align the point-wise feature distributions of two domains. However, without explicitly modeling and dealing with the noise, these methods bring quite weak benefits to the adaptation performance. Recently, several works attempt to deal with the target noise to mitigate the domain gap. Rochan et al. [165] randomly select target noise masks and apply the selected mask to source samples. Wu et al. [197] compute one dataset-level mask and apply it to all source samples. Zhao et al. [231] use CycleGAN [238] to perform noise inpainting which is then used to learn synthetic noise generation module. The issues of these previous works are two-fold: 1) they cannot adaptively determine the injected noises according to the context of source samples; 2) the generated mask cannot be guaranteed to reduce the domain shift. Thus, these methods may achieve sub-optimal results.

In this chapter, we aim to mitigate the domain shift caused by the target noise by learning to adaptively mask the source points during the adaptation procedure. To reach this goal, we need to deal with two problems: 1) how to learn a spatial mask that can be adaptively determined according to the specific context of a source sample, and 2) how to guarantee the learned masks help narrow the domain gap. To solve the first problem, we design a learnable masking module named "Adaptive Spatial Masking (ASM)" module, which takes source Cartesian coordinates and features as input, to generate point-wise source masks. We incorporate Gumbel-Softmax operation into the masking module so that it can generate binary masks and be trained end-to-end via gradient back-propagation. To solve the second problem, we incorporate adversarial training into the masking module learning process. Specifically, during training, we add an additional domain discriminator on top of the feature extractor. By encouraging features from two domains (features of

masked source samples and those of normal target samples) to be indistinguishable, the masking module is able to learn to generate masks mimicking the pattern of target noise and narrow the domain gap. Note that these two designs cooperate with each other to better align features across domains and improve the adaptation performance.

In a nutshell, our contributions can be summarized as:

- We notice that the pattern of target noise is unexpected and irregular. Thus, we propose to model the target noise in a learnable way. Previous works, which do not explicitly model the target noise or ignore such characteristics, are less effective.
- We propose to adversarially mask source samples to mimic the target noise patterns. In detail, we design a novel learnable masking module and incorporate adversarial training. Both components cooperate with each other to promote the adaptation.
- Experiments on two synthetic-to-real adaptation benchmarks, *i.e.*SynLiDAR $\rightarrow$ SemKITTI and SynLiDAR $\rightarrow$ nuScenes, demonstrate that our method can effectively improve the adaptation performance.

4.2 The Proposed Approach

In the following sections, we first provide necessary preliminaries (§ 4.2.1) for domain adaptive point cloud segmentation. Second, we introduce the Adversarial Masking method (§ 4.2.2). Finally, we give the training objectives for the adaptation process (§ 4.2.3).

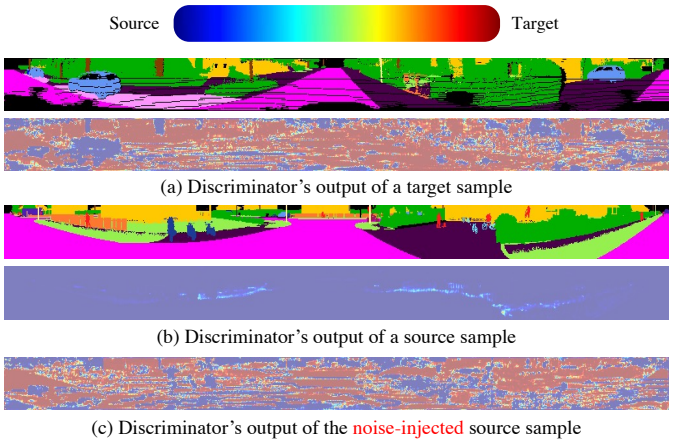


Figure 4.2 : Heatmap of discriminator's output, where color red indicates the target domain and blue denotes the source. As shown in (a) and (b), source samples and target ones are well identified by the domain discriminator. However, in (c), injecting noise extracted from a random target sample to the source sample can easily fool the discriminator.

4.2.1 Preliminaries

In domain adaptive point cloud segmentation, we are provided with annotated source scans $\mathcal{S} = \{(\boldsymbol{P}_i^s, \boldsymbol{M}_i^s)\}_{i=1}^{N^s}$ and unlabeled target scans $\mathcal{T} = \{(\boldsymbol{P}_i^t)\}_{i=1}^{N^t}$, where $\boldsymbol{P}_i \in \mathbb{R}^{n_i \times 4}$ denotes the set of points with coordinates (x, y, z) and intensity, $\boldsymbol{M}_i \in \mathbb{R}^{n_i}$ denotes the ground-truth annotation for the point cloud, and n_i is the number of points in the i-th scan. For more efficient processing, we employ spherical projection to transform each raw point cloud $\boldsymbol{P}$ into 2D image $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 5}$, and the labels are transformed to $\boldsymbol{Y} \in \mathbb{R}^{H \times W}$ accordingly. The details are presented below. We aim to train a segmentation model on $\mathcal{S}$ and $\mathcal{T}$ to make accurate predictions on target points.

Spherical Projection. For more efficient processing, we transform the sparse point clouds into 2D images with spherical projection like [197, 198]. Specifically, for a point with coordinate (x, y, z), we project it into a 2D LiDAR image with

coordinates (p, q):

$$\begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(1 - \arctan2^*(y, x)/\pi) \cdot W \\ (1 - (\arcsin(z \cdot r^{-1}) + f_{up}) \cdot f^{-1}) \cdot H \end{bmatrix}, \tag{4.1}$$

where $r = \sqrt{x^2 + y^2 + z^2}$ is the range of this point. $f = f_{up} + f_{down}$ is the vertical field-of-view of the LiDAR sensor. For each projected point with coordinate (p, q), we concatenate its Cartesian coordinates (x, y, z), range (r), and intensity, then obtain the projected LiDAR image $\boldsymbol{I}$ with the shape $H \times W \times 5$. The intensity channel models the strength of LiDAR beams. As such, raw point clouds with sparse and unordered structures are transformed into 2D images, so that 2D convolutions can be applied directly. The main reason we adopt the spherical projection is to align with previous works in this task [197, 198]. Our proposed method is also applicable for voxel-based pipelines, which is substantiated in Table 4.3 later in the text.

Domain Adversarial Training (DAT). Domain adversarial training [187, 72, 58] has been proven effective in aligning the feature distributions across domains. During training, an additional domain discriminator is introduced to classify the features into different domains. Through adversarial training, the network is encouraged to generate features that are indistinguishable across domains. Consequently, domain-invariant features can be learned, hence benefiting the adaptation performance.

Specifically, let D denotes the discriminator, the above min-max game can be formed as (the original GAN loss format [65]):

$$\begin{aligned} \min_G \max_D V_{GAN}(G, D) = \mathbb{E}_{I^t \sim \mathcal{T}}[\log(D(G(\boldsymbol{I}^t)))]) \\ + \mathbb{E}_{I^s \sim \mathcal{S}}[\log(1 - D(G(\boldsymbol{I}^s)))]), \end{aligned} \tag{4.2}$$

where G denotes the feature extractor of the model.

*We use the arctan2 function in the Numpy library (www.numpy.org).

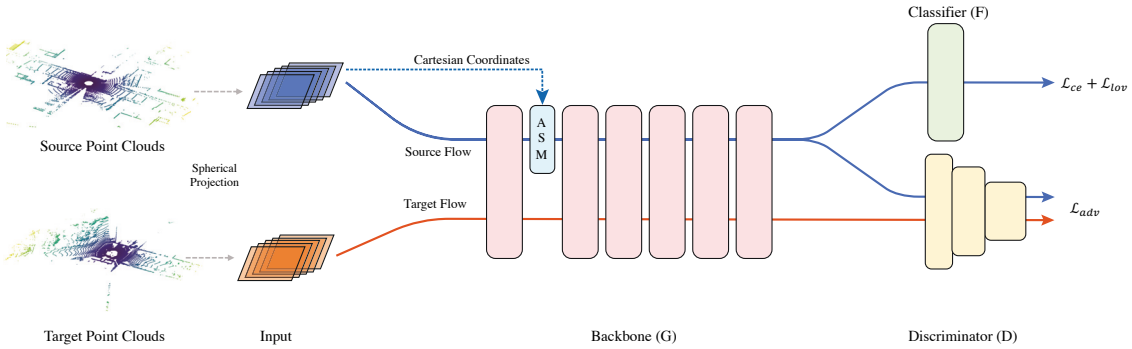


Figure 4.3 : Illustration of the Adversarial Masking Framework. In this chapter, we aim to mitigate the domain gap induced by target noise via masking source samples to mimic the target patterns. First, we propose a module, named Adaptive Spatial Masking (ASM), which can learn to mask the source points. Then we train the ASM-equipped model in an adversarial way. The two key components of our framework (*i.e.*. ASM and adversarial training) collaboratively contribute to the final adaptation performance. Specifically, adversarial training encourages ASM to mimic target noises, while ASM eases the adversarial training to better align features across domains.

4.2.2 Adversarial Masking

Our framework is illustrated in Fig. 4.3. Following the general practice in domain adaptation, the network is shared across source and target domain data. The network consists of a backbone (G) to extract features and a task classifier (F) to distinguish the samples into different categories. During training, we insert our designed masking module (ASM) into the backbone to mask source points and attach an additional discriminator (D) on top of the backbone to assign domain labels to features from both domains. On the one hand, the domain discriminator is trained to differentiate masked source samples and target samples. On the other hand, the ASM is encouraged to learn to mask source points to mimic the target noise

patterns, and features are trained to be domain-invariant to confuse the domain discriminator. As a result, the adversarial training and the masking module work collaboratively to narrow the domain discrepancy.

Target Noise Hinders Adversarial Training. As adversarial training has been proven effective in 2D visual domain adaptation tasks, *e.g.*, image classification, semantic segmentation, it is natural to see if adversarial training can help learn domain-invariant features in this 3D synthetic-to-real adaptation scenario. As shown in Fig. 4.2, we observe an interesting phenomenon that the discriminator converges quickly and can easily differentiate most of target points from the source ones. In contrast, injecting noise (from a random target sample) to source samples helps alleviate such an issue, *i.e.*, the features of many source points can confuse the domain discriminator in terms of their domain labels. We assume that in plain adversarial training, target noise may serve as a shortcut for the discriminator to classify samples into different domains. Only with adversarial training, it is hard to discover the noise patterns of the target and meanwhile align feature distributions across domains. Thus, we propose to explicitly model the noise patterns of the target and adaptively inject noise into source samples in order to ease the conventional adversarial training.

Adaptive Spatial Masking (ASM) Module. As discussed above, the way of randomly injecting target noise to source samples can alleviate the issue of adversarial training to an extent. However, such a way may not be an optimal choice because it cannot adaptively determine the distribution of injected noise according to the specific context of each source sample. Moreover, the pattern of injected noise may be irregular, and the copy-and-paste way of injecting noise may not accurately capture those irregular patterns.

Thus, we design a learnable module to perform spatial masking on source sam-

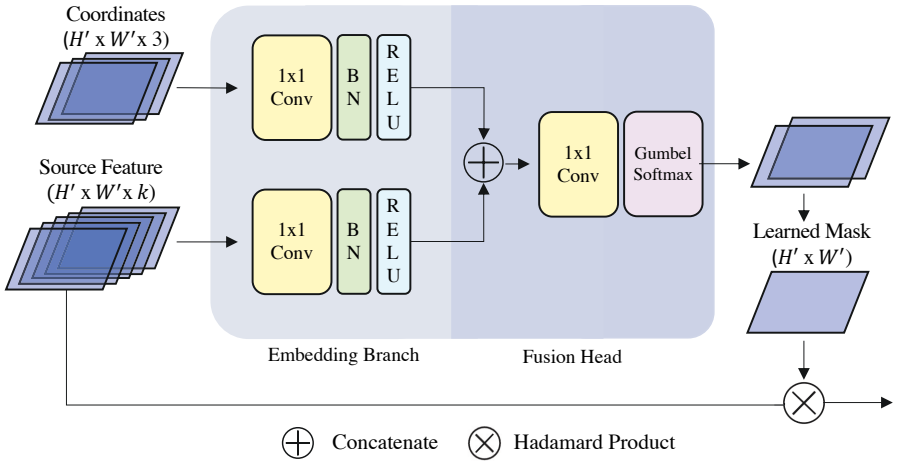


Figure 4.4 : Illustration of Adaptive Spatial Masking (ASM). The proposed ASM takes source Cartesian coordinates and source features as input, and outputs two differentiable binary maps which divide points into two groups, *i.e.*, preserved and ignored. Then we impose the first mask on original source features.

ples. We name our module as Adaptive Spatial Masking (ASM) module. To be concrete, we present the diagram of ASM in Fig. 4.4. The module consists of two embedding branches and one fusion head that all employ 1×1 convolutions. First, the two branches take source Cartesian coordinates $\boldsymbol{O}^s \in \mathbb{R}^{H' \times W' \times 3}$ (*i.e.*. the x, y, z channels of projected LiDAR image but downsampled to $H' \times W'$ to match the size of feature map) and source features $\boldsymbol{E}^s \in \mathbb{R}^{H' \times W' \times k}$ as inputs respectively. Then, the embedded features from two branches are fused via a fusion head to generate the desired source mask. Finally, we calculate the element-wise product between the learned mask and original source features. Then the masked source feature is forwarded through the remaining layers for predictions.

Note that we attach a Gumbel-Softmax [82] layer to the end of the fusion head. The output of Gumbel-Softmax has two channels, each of which has spatial size $H' \times W'$. We use the first channel to indicate which points should be preserved and the second channel to indicate which points should be ignored. Then, the

first channel is utilized to mask the source features with the element-wise product. Note that Gumbel-Softmax enables us to apply binary masks during the forward process, while supporting gradient back-propagation to update the parameters of our network. In contrast, the plain softmax layer cannot actually zero out source points, and thus leads to inferior results. We will show an empirical comparison of these designs in Sec. 4.3.3.

As shown in Fig. 4.3, we insert ASM (*i.e.*, denoted with color blue) after a specific shallow layer to inject noises to source samples. Note that we do not directly insert ASM after the input of the projected LiDAR image. It is because that we empirically find the shallow features are also useful to learn a better source mask, while the input only contains the coordinate information. In our implementation, we place ASM after the first convolution block (*i.e.*, `conv-bn-relu`). Note that ASM is only applied to source samples during the training process. So we simply remove ASM module for the inference on target samples.

Adversarial Masking. With the proposed masking module, the model can inject noise to source samples by zeroing out shallow features of partial source points. However, we cannot guarantee that the generated mask can mimic the patterns of target noise and thus mitigating the domain gap. To solve this, we integrate the ASM-equipped model with an adversarial training paradigm. Specifically, with ASM, the corresponding discrimination and generation loss of adversarial training are

$$\min_{\theta_D} \mathcal{L}_{dis} = \mathbb{E}_{I^t \sim \mathcal{T}}[(1 - D(G(\boldsymbol{I}^t)))^2] + \mathbb{E}_{I^s \sim \mathcal{S}}[(D(\bar{G}(\boldsymbol{I}^s)))^2],$$

$$\min_{\theta_G, \theta_{ASM}} \mathcal{L}_{gen} = \mathbb{E}_{I^s \sim \mathcal{S}}[(1 - D(\bar{G}(\boldsymbol{I}^s)))^2], \tag{4.3}$$

where $\bar{G}$ is the backbone with ASM module inserted and $\theta_{ASM}, \theta_G, \theta_D$ denote the parameters of ASM module, backbone and discriminator respectively. Note that different from the loss format in Eq. 4.2, we choose LSGAN [132] in our implementation

for its better stability.

4.2.3 Training Objective

Overall, the model is optimized with three objectives, *i.e.*, cross-entropy loss ($\mathcal{L}_{ce}$), Lovasz-Softmax loss [6] ($\mathcal{L}_{lov}$), and the adversarial training loss ($\mathcal{L}_{gen}$, $\mathcal{L}_{dis}$):

$$\min_{\theta_G,\theta_{ASM},\theta_F} \mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{lov} + \lambda \mathcal{L}_{gen} \tag{4.4}$$

$$\min_{\theta_D} \mathcal{L}_{dis} \tag{4.5}$$

where θ_F denotes the parameters of the classifier and the cross-entropy loss is calculated as:

$$\mathcal{L}_{ce} = -\sum_{h,w}^{H,W} \log[F(\bar{G}(I^s)(h,w,Y_{h,w}))]. \tag{4.6}$$

The segmentation model and the domain discriminator are updated alternatively with objective Eq. 4.4 and Eq. 4.5, respectively. Note that $Y_{h,w}$ is the label for the pixel at position (h,w) of a projected LiDAR image.

4.3 Experiments

4.3.1 Experimental Setup

Datasets. In this chapter, we perform experiments on two synthetic-to-real benchmarks, *i.e.*, SynLiDAR→ SemKITTI and SynLiDAR→ nuScenes.

SynLiDAR [203] is a synthetic LiDAR dataset for point cloud segmentation, which is collected from a simulated driving scene environment. This dataset collects 198,396 scans with 19482 M points, covering various scenes on Unreal Engine 4 platform. This dataset provides point-wise annotations for 32 classes that are in line with SemKITTI.

SemKITTI [4] is a large-scale point cloud dataset for point cloud segmentation. This dataset is collected from a Velodyne HDL-64E LiDAR and contains 22

Table 4.1 : Experiments results of SynLiDAR [203] → SemKITTI [4] with SqueezeSegV3-21 [205] as the backbone.

Methods	Type	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	road	parking	sidewalk	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mIoU
Source Only	–	4.2	12.8	6.7	0.9	3.7	9.6	14.3	1.4	0.3	23.2	47.6	5.0	54.3	21.4	27.0	23.2	3.1	$15.2_{\pm 0.3}$
AdaptSeg [186]	2D	0.6	5.9	3.4	1.9	5.4	8.2	13.6	60.0	1.7	37.8	36.4	3.7	31.8	17.3	40.1	20.9	4.4	$17.2_{\pm 0.3}$
ADVENT [192]	2D	16.6	9.3	8.0	2.1	3.8	8.4	16.5	46.7	3.0	31.4	34.7	6.8	44.5	20.4	28.9	20.0	4.0	$17.9_{\pm 0.3}$
CBST [243]	2D	7.6	9.0	4.8	1.2	1.1	10.5	11.1	8.8	1.9	20.0	48.5	**8.9**	35.8	20.3	25.2	28.6	4.9	$14.6_{\pm 0.5}$
PLCA [89]	2D	7.8	7.1	4.0	3.1	3.8	**14.3**	18.1	44.4	**10.2**	19.8	44.9	4.6	43.9	12.6	21.2	26.3	3.6	$17.0_{\pm 0.3}$
SqueezeSegV1 [197]	3D	**43.7**	7.4	5.1	**4.6**	**5.8**	6.5	13.9	36.7	3.1	31.0	37.4	7.4	28.9	**26.2**	29.9	27.1	5.0	$17.4_{\pm 0.3}$
SqueezeSegV2 [198]	3D	20.4	8.0	4.0	2.4	5.1	8.0	17.3	59.4	3.2	34.9	39.4	8.4	41.3	21.9	32.2	29.0	4.7	$20.0_{\pm 0.4}$
ePointDA [231]	3D	18.8	8.0	6.1	2.6	3.6	6.4	14.6	50.8	9.4	32.7	33.3	8.2	27.6	22.4	30.2	26.9	**7.2**	$18.2_{\pm 0.6}$
LiDAR-Net [85]	3D	22.3	7.2	10.3	2.0	2.3	8.1	18.8	43.2	5.6	33.5	32.6	4.9	29.8	26.5	22.5	21.4	5.4	$17.4_{\pm 0.4}$
CoSMix [172]	3D	15.1	4.3	3.2	1.0	1.4	5.4	5.5	47.6	2.9	32.9	54.1	7.8	58.1	24.8	**41.0**	**32.1**	3.0	$20.0_{\pm 0.6}$
RandMask+ADV	3D	42.0	10.0	**13.0**	2.4	4.7	8.2	**20.7**	30.5	3.7	29.2	37.7	5.0	30.7	22.9	26.7	24.7	4.1	$18.6_{\pm 0.3}$
FreqMask+ADV	3D	22.9	9.3	5.3	2.2	3.0	5.4	13.9	50.6	3.8	31.9	38.1	6.5	32.0	25.4	38.1	25.5	4.6	$18.7_{\pm 0.3}$
SpatialDropout+ADV	3D	30.1	7.3	3.2	3.1	5.1	10.3	11.6	44.3	3.2	30.4	40.1	7.7	30.8	22.3	27.9	23.9	3.9	$18.0_{\pm 0.2}$
Ours	3D	19.7	**13.8**	9.7	2.1	4.1	8.0	8.2	**64.5**	8.0	**36.0**	**54.6**	6.7	**58.0**	24.7	35.8	29.1	4.2	$\mathbf{22.8}_{\pm 0.3}$
Oracle	–	91.9	25.5	42.0	42.7	26.1	32.9	54.7	94.2	42.8	82.0	80.8	39.9	84.5	48.9	72.2	54.0	28.8	$55.5_{\pm 0.2}$

sequences with 41000 frames. This dataset contains annotations for 25 categories. Following [241, 205], we choose sequences 00-10 for training except sequence 08 for validation.

nuScenes-lidarseg [13] is another LiDAR dataset that collected from real world. This dataset is collected from a different LiDAR sensor, *i.e.*, a 32-beam LiDAR with FOV of $[-30°, 10°]$. Following its guideline, 850 scenes are chosen for training and the other 150 scenes for validation.

For SynLiDAR → SemKITTI and SynLiDAR → nuScenes, part of the labels are merged to match across domains.

Evaluation. Following common practice [241, 74, 206], we adopt mean intersection over union (*i.e.*, mIoU) as the evaluation metric, which is averaged over all

Table 4.2 : Experiments results of SynLiDAR [203] → nuScenes [13] with SalsaNext [38] as the backbone.

Methods	Type	bicyle	bus	car	other-vehicle	motorcycle	pedestrian	truck	road	other-ground	sidewalk	terrain	manmade	vegetation	mIoU
Source Only	–	0.4	1.0	3.7	0.2	2.1	16.0	17.0	24.2	0.1	14.9	8.6	36.0	25.5	$11.5_{\pm 0.3}$
AdaptSeg [186]	2D	0.7	0.5	13.2	0.7	2.2	17.0	13.8	44.3	0.4	18.3	10.6	39.2	**27.5**	$14.5_{\pm 0.3}$
ADVENT [192]	2D	0.5	1.6	11.4	0.7	2.1	18.0	18.3	43.2	0.7	19.2	10.3	40.2	27.2	$14.8_{\pm 0.3}$
CBST [243]	2D	0.4	0.2	8.2	0.4	1.3	5.5	13.4	38.2	0.7	15.0	10.4	26.0	30.8	$11.6_{\pm 0.5}$
PLCA [89]	2D	0.5	8.5	12.3	1.4	1.1	19.3	16.6	33.9	3.6	16.6	4.8	33.1	22.2	$13.4_{\pm 0.3}$
SqueezeSegV1 [197]	3D	0.7	5.7	19.7	0.8	2.9	17.9	**19.0**	33.8	0.3	14.9	**15.1**	26.5	23.5	$13.9_{\pm 0.4}$
SqueezeSegV2 [198]	3D	0.7	2.5	10.6	0.6	**4.1**	19.5	14.6	33.4	0.2	17.3	6.2	43.5	23.7	$13.6_{\pm 0.3}$
ePointDA [231]	3D	0.4	3.0	9.8	0.6	3.0	18.5	13.8	31.9	0.3	15.0	8.2	35.6	23.4	$12.6_{\pm 0.3}$
LiDAR-Net [85]	3D	0.6	4.0	10.1	0.8	2.7	18.8	13.3	34.1	0.5	14.9	8.8	38.7	20.1	$12.9_{\pm 0.3}$
CoSMix [172]	3D	0.3	2.6	0.5	0.4	0.6	7.1	1.7	**60.1**	**14.3**	11.9	8.6	33.4	18.1	$12.3_{\pm 0.4}$
RandMask+ADV	3D	0.5	6.0	17.4	1.6	2.6	18.9	15.0	33.1	0.5	15.7	8.0	38.9	27.1	$14.3_{\pm 0.3}$
FreqMask+ADV	3D	0.5	6.7	19.0	1.1	3.1	**22.3**	14.1	31.1	0.4	16.6	7.3	40.0	26.5	$14.5_{\pm 0.3}$
SpatialDropout+ADV	3D	0.5	**8.6**	19.9	1.6	1.7	13.9	16.5	50.7	3.6	16.9	8.9	30.6	20.3	$14.9_{\pm 0.3}$
Ours	3D	**0.9**	1.2	**26.9**	**2.2**	2.6	17.4	18.2	57.4	0.8	**21.8**	7.6	**43.9**	20.1	$\mathbf{17.0}_{\pm 0.3}$
Oracle	–	25.3	71.8	85.1	34.3	44.0	65.3	63.2	95.3	69.3	70.7	71.1	81.2	73.9	$65.4_{\pm 0.3}$

classes. Note that we report the averaged results over 3 random runs. No post-processing is applied, *e.g.*, conditional random field.

Implementation. For the segmentation task, we choose two representative backbones, *i.e.*, SqueezeSegV3-21 [205] and SalsaNext [38]. The ASM employs the Straight Through variant of Gumbel-Softmax [82]. The model is optimized using momentum SGD with momentum of 0.9 and weight decay 1×10^{-4}. Warmup is applied for the first epoch to linearly increase the learning rate to the base learning rate. Then learning rate decays exponentially. The base learning rate is set to 4×10^{-3} and 2×10^{-3} for SynLiDAR→SemKITTI and SynLiDAR→nuScenes, respectively. The discriminator is optimized using Adam optimizer with learning rate of 1×10^{-3}. The batch size is set to 24 and the model is optimized for 50 epochs totally. The output channel of the embedding branch in ASM is set to 32. The λ in Eq. 4.4 is set to 0.001.

4.3.2 Comparisons with Previous Methods

We compare our method with previous 2D and 3D domain adaptation methods. The "2D" and "3D" indicate the settings that the methods are originally designed for. For 2D methods, we re-implement representative methods, *i.e.*, CBST [243], PLCA [89], AdaptSeg [186], and ADVENT [192]. As for 3D adaptation techniques, we re-implement SqueezeSegV1 [197] SqueezeSegV2[198], ePointDA [231], LiDAR-Net [85], and CoSMix [172] with the identical backbone. Besides, we also present the results with three variants of source masks, *i.e.*, "SpatialDropout" that randomly drops the source points spatially, "RandMask" randomly selects masks from the target samples, and "FreqMask" where points are randomly dropped according to the point-wise frequency map of target noise over the dataset. We use "ADV" to denote the adversarial training paradigm and use "Oracle" to denote the full supervision baseline. For a fair comparison, all presented results use the same supervision on source samples, *i.e.*, $\mathcal{L}_{ce} + \mathcal{L}_{lov}$.

In Table 4.1 and Table 4.2, we present the results on SynLiDAR $\rightarrow$ SemKITTI and SynLiDAR $\rightarrow$ nuScenes, respectively. First, compared with the source-only baseline, our method achieves apparent improvements, *i.e.*, +7.6% mIoU absolute gain on SemKITTI and +5.5% mIoU on nuScenes, which justifies the necessity of performing adaptation. Second, compared with 2D techniques, our method still holds its superiority, *e.g.*, our method outperforms AdaptSeg by 5.6% and 2.5% mIoU on SynLiDAR $\rightarrow$ SemKITTI and SyncLiDAR $\rightarrow$ nuScenes respectively. Especially, we notice that CBST shows inferior performance, which may be because of the low quality of pseudo labels resulting from the large gap between source and target point clouds. Third, compared with 3D solutions, our method also attains superior results, *e.g.*. on SemKITTI, we achieve 2.8% and 4.6% absolute gain compared to SqueezeSegV2 and ePointDA, respectively. Finally, even with adversarial training, various non-learnable masking strategies (RandMask, SpatialDropout,

Table 4.3 : Experiments results of SynLiDAR [203] → SemKITTI [4] with Minkowski-Unet (voxel-based pipeline).

Method	car	bi.cle	mt.cle	truck	oth-v.	pers.	b.clst	m.clst	road	park.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traff.	mIoU
ADDA [188]	52.5	4.5	11.9	0.3	3.9	9.4	27.9	0.5	52.8	4.9	27.4	0.0	61.0	17.0	57.4	34.5	42.9	23.2	4.5	23.0
Ent-Min [192]	58.3	5.1	14.3	0.3	1.8	14.3	**44.5**	0.5	50.4	4.3	34.8	0.0	48.3	19.7	67.5	34.8	**52.0**	33.0	6.1	25.8
ST [243]	62.0	5.0	12.4	1.3	9.2	16.7	44.2	0.4	53.0	2.5	28.4	0.0	57.1	18.7	69.8	**35.0**	48.7	32.5	6.9	26.5
PCT [203]	53.4	5.4	7.4	0.8	10.9	12.0	43.2	0.3	50.8	3.7	29.4	0.0	48.0	10.4	68.2	33.1	40.0	29.5	6.9	23.9
ST-PCT [203]	70.8	**7.3**	13.1	1.9	8.4	12.6	44.0	0.6	56.4	4.5	31.8	0.0	**66.7**	**23.7**	**73.3**	34.6	48.4	**39.4**	11.7	28.9
CosMix [172]	75.1	6.8	29.4	**27.1**	**11.1**	**22.1**	25.0	24.7	**79.3**	14.9	46.7	**0.1**	53.4	13.0	67.7	31.4	32.1	37.9	**13.4**	32.2
Ours	**75.8**	**7.3**	**34.6**	26.8	10.8	21.3	40.3	**25.1**	60.4	**18.3**	**48.1**	**0.1**	58.4	14.3	72.3	33.3	40.2	36.6	8.2	**33.3**

FreqMask) fail to achieve competitive results against ours. This is because these masking strategies cannot be adaptively adjusted according to the different contexts, and adversarial training is not able to impact the imposed source masks as they are not learnable. Moreover, we can also observe an apparent gap to the full supervision training (Oracle), indicating there is still a long way to go in minimizing the domain gap in point cloud semantic segmentation.

In Table 4.3, we run experiments using the voxel-based pipelines, following the configuration of CoSMix [172], which employs the voxel-based pipeline. Our method surpasses CoSMix by 1.1 % mIoU, validating a better adaptation performance on both benchmarks. This also validates that the proposed method is also applicable to the voxel-based pipeline.

4.3.3 Ablation Studies

Effect of Adaptive Spatial Masking (ASM). First, in Fig. 4.2, we show qualitatively that injecting noise can ease the adversarial training. And the quantitative comparison in Table 4.1 and 4.2 with other masking strategies (SpatialDropout, RandMask, FreqMask) also verifies that ASM derives better masks for easing the

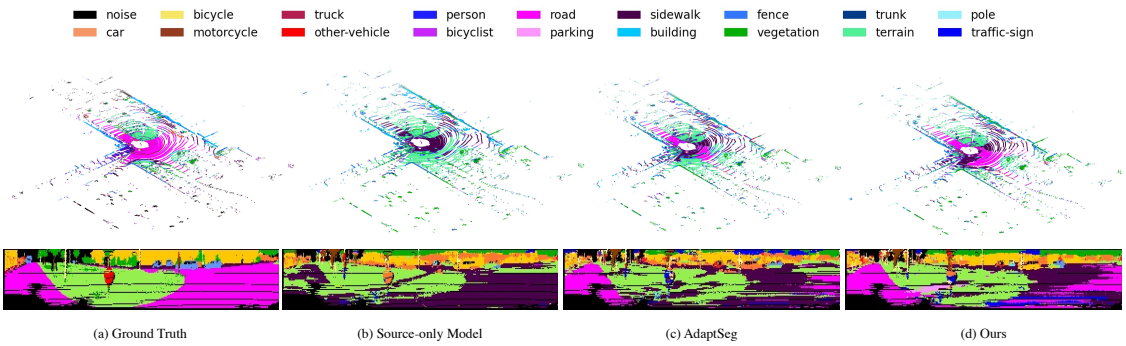


Figure 4.5 : Visualization of Segmentation Results (SynLiDAR → SemKITTI). We compare our method (d) with (a) ground truth, (b) source-only, and (c) AdaptSeg [186]. We present visualizations of both raw points (the first row) and projected point clouds (the second row). We show representative crops of projected 2D images due to the space limit.

adaptation.

Effect of different branches of masking module. As discussed in Sec. 4.2.2, we use two embedding branches in the proposed ASM module. We evaluate the contribution of the two branches in Table 4.4 (a), where branch e receives source feature as input and branch o receives source Cartesian coordinates as input. It can be seen that removing either of them leads to an obvious drop in mIoU compared to the result using both branches. This verifies that both branches contribute to generating more effective masks.

Effect of Gumbel-Softmax. To evaluate the contribution of Gumbel-Softmax, we compare the results with training using plain Softmax which generates soft masks (*i.e.*, each mask value is within $[0, 1]$) for both forward and backward processes. As shown in Table 4.4 (b), using plain Softmax results in an obvious drop of mIoU, *i.e.*, -3.2% mIoU. This is because plain Softmax cannot actually zero out source points, rendering it hard to mimic the target noise patterns to mitigate the domain shift.

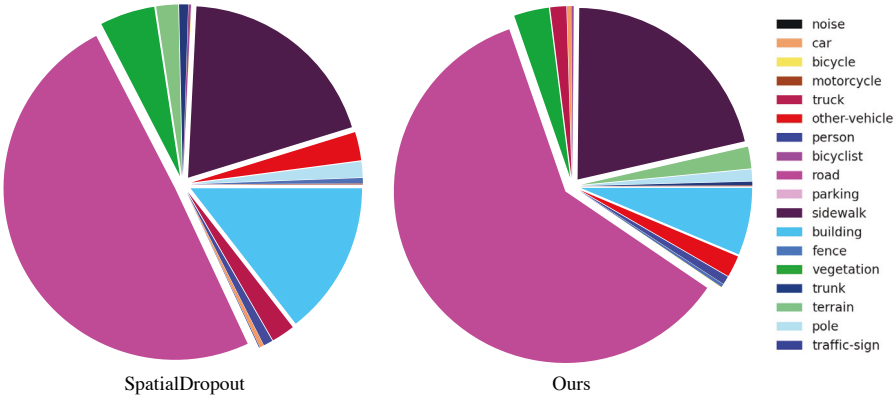


Figure 4.6 : Statistics of ignored source points (SynLiDAR → SemKITTI). Compared with performing masking randomly, our method exhibits a different preference toward different classes. For example, contrary to SpatialDropout, fewer points from class "Building" are ignored and more points from "Road" are dropped.

Effect of different masking layers. In Table 4.4 (c), we compare the results of inserting ASM at different layers of the network, including input (*i.e.*, masking the projected LiDAR image), ours (*i.e.*, after the first conv. layer), middle (between the encoder and the decoder), and end of the backbone. From the table, we observe that inserting ASM at the shallower layer can achieve better results, which avoids features being affected by domain shift from the early stage. Compared with the result of inserting ASM directly after the input, ours achieves better results, verifying the important role of exploiting shallow feature information in learning better masks. Besides, inserting ASM at the end of the backbone is worse but not that far from "Ours". This is because masking at the end of the backbone also introduces noises to the discriminator and the classifier.

Optimization strategy with ASM . We investigate the optimization with the ASM module and present it in Table 4.4 (d). Only adversarially updating ASM leads to inferior results than updating both (*i.e.*, θ_G and θ_{ASM} in Eq.(4.4)). This

Table 4.4 : Ablation studies on Adaptive Spatial Masking. Experiments are conducted on SynLiDAR [203] → SemKITTI [4].

Module	Modification	mIoU
(a) Two Branches	Branch e (embedding only)	21.7
	Branch o (coordinate only)	22.0
	Both Branch	22.8
(b) Mask Type	Plain Softmax (soft)	19.6
	Gumbel-Softmax (binary)	22.8
(c) Masking Layer	Input	22.0
	Ours	22.8
	Middle	21.7
	End of the backbone	21.6
(d) Update Strategy	$\mathcal{L}_{gen}$ optimizes θ_{ASM} only	21.6
	$\mathcal{L}_{gen}$ optimizes θ_{ASM} and θ_G	22.8

shows that besides adversarially updating ASM, adversarially updating features also contributes to a better adaptation.

Analysis of masked samples To better understand the masking module, in Fig. 4.6, we present the class distribution of points that are ignored and compare with random dropout using a similar ignore ratio. Compared with random dropout, our result exhibits a different pattern/distribution, *e.g.*, our method ignores more points of class "Road" but fewer points of class "Building" and "Vegetation". However, our method outperforms it with a large margin, *i.e.*, +4.8% on SemKITTI and +2.1% on nuScenes. This indicates that our method can derive more reasonable noise distributions for mitigating the domain gap.

Sensitivity to hyper parameters. In Table 4.5, we present the sensitivity of our method to λ on both datasets. The performance of our method first increases

Table 4.5 : Sensitivity Analysis of λ (coefficient of $\mathcal{L}_{gen}$).

Transfer	5×10^{-4}	1×10^{-3}	5×10^{-3}
SynLiDAR $\rightarrow$ SemKITTI	21.9	22.8	21.6
SynLiDAR $\rightarrow$ nuScenes	16.6	17.0	16.3

and then decreases a little bit with the increase of λ from 5×10^{-4} to 5×10^{-3}. The bell shape of change verifies the regularization effect of adversarial training on the adaptation performance. Note that, within a wide range of choices of λ, our method consistently outperforms previous solutions by a large margin, which further verifies the effectiveness of our design.

Visualization In Fig. 4.5, we present the visualization of segmentation results on SynLiDAR→SemKITTI. From these figures, we observe that our method attains obvious improvement against source-only baseline and previous approach, which is in line with the superior results of our method shown in Table 4.1 and 4.2.

4.4 Conclusion

In this chapter, we aim to mitigate the domain gap caused by target noises in synthetic-to-real point cloud segmentation adaptation. To this end, we propose Adversarial Masking, where a masking module is designed to derive learnable masks and the adversarial training paradigm encourages the masking module to mimic injecting target noises to source samples. The adversarial training and the masking module cooperate with each other to promote domain-invariant feature learning. Extensive experiments are conducted to prove the effectiveness of the proposed method.

Chapter 5

Decompose to Generalize: Species-Generalized Animal Pose Estimation

Diverging from the rigid structures studied in the prior chapters, this chapter explores domain generalization with diverse non-rigid structures. This chapter will examine the distinct obstacles presented by non-rigid structures and seek to develop methodologies that derive generalizable representations, building a comprehensive understanding of adaptability across different visual structures.

5.1 Introduction

Animal pose estimation [17, 96, 139, 133] aims to identify and localize the anatomical joints* of animal bodies, and has received increasing attention for its wide application, *i.e.*, biology, zoology, and aquaculture. A critical challenge in realistic animal pose estimation is the cross-species problem, *i.e.*, using the already-learned pose estimator for novel species. Specifically, it is infeasible to collect and annotate all the animal species, because the animal kingdom is a vast group of millions of different species. Under this background, cross-species generalization is of great value for realistic applications.

This chapter tackles the cross-species animal pose estimation from the domain-generalization (DG) viewpoint (*i.e.*, a species is a respective domain) and reveals a unique factor for this cross-species generalization, *i.e.*, the relation between different

*Joint here refers to the keypoints over the bodies, including articulation points where two or more bones meet, and facial landmarks, *e.g.*, eyes and noses.

Figure 5.1 : Two reasons that bring domain gap to the joint relation. 1) structural discrepancy: the part lengths (*i.e.* the distances between different joints) may vary for different species (left most); 2) the visual similarities between some different joints are inconsistent for different species, *e.g.*, the visual similarities between faces and other body parts are different for tiger, fox, and the cow.

joints. The joint relation in our view can be visual (*e.g.*, the color relation between neighboring joints), structural (*e.g.*, the nose is under the eyes) and many more. Our focus on the joint relation is different from the popular concern in general domain generalization [5, 8, 9, 41], which mainly considers the distribution shift between the source and the unseen target domain(s). The generic DG methods usually learn domain-invariant representations [140, 60, 105, 109, 12, 209, 64], enhance the generalizability through meta-learning [45, 2, 98, 100] or data augmentation [235, 173, 21, 191]. While these methods are potential for cross-species generalization as well, the joint relation is a unique viewpoint that has never been explored under other DG scenarios.

The importance of joint relation is two-fold: 1) on the one hand, some joint relation is consistent across all the species and is beneficial. With consistent relation, two joints may mutually confirm each other, *e.g.*, the *eye* helps confirm the *nose* and vice versa, because they are consistently close in all species. 2) on the other hand, some joint relation is inconsistent for different species due to species variation and is thus harmful for generalization, *e.g.*, the length of non-rigid body parts such as legs. Such inconsistent relation makes the already-learned mutual confirmation become a

severe distraction rather than any benefit. We note that the latter (negative) impact has more or less been recognized by some earlier literature [17], while the former (positive) impact was neglected. In contrast to [17], we argue that both two factors are important and should be considered in combination.

With these two insights, we propose a Decompose-to-Generalize (D-Gen) pose estimation method to break the inconsistent relations while preserving the consistent ones. Specifically, D-Gen first decomposes the body joints into several joint concepts. The decomposition facilitates that each individual concept contains multiple closely-related joints and that the joints in different concepts are far away or prone to inconsistent relations. Given these joint concepts, D-Gen promotes the interaction between intra-concept joints and meanwhile suppresses the interaction between inter-concept joints. The approach for interaction promotion / suppression is very simple: D-Gen splits the top layers of the backbone network into several pose-estimation branches, each one of which is responsible for a corresponding joint concept. Intuitively, the joints in different branches have less interaction, compared to the joints in the same branch. Consequently, D-Gen suppresses the distraction from inconsistent joint relation and yet preserves the beneficial mutual confirmation of consistent joint relation, thus improving cross-species generalization.

We explore three strategies for joint decomposition, *i.e.*, heuristic, geometric and the attention-based manner. The geometric manner clusters the joints based on their geometric distances. The attention-based manner uses the attention mechanism to learn the affinity between joint features and uses the affinity matrix for decomposition. Experimental results show that all these three strategies substantially improve cross-species generalization, validating the effectiveness of our joint decomposition. Another interesting observation is that the attention-based strategy surpasses the other two strategies, indicating that attention-based concepts are better than the concept derived from human intuition (*i.e.*, heuristic) and the pure

geometric relation. Since the attention-based approach combines deep feature and geometric priors, its superiority against the geometric manner suggests that there are multiple forms of joint relation beyond the structural relation.

5.2 The Proposed Approach

5.2.1 Overview

Preliminaries. From the perspective of domain generalization (DG), each animal species in pose estimation is within an independent domain. Let us assume the training set $\mathcal{D}$ contains M species (domains), *i.e.*, $\mathcal{D} = \{D_1, D_2, ...D_M\}$, forming a joint sample space $\mathcal{X} \times \mathcal{Y}$ ($\mathcal{X}$ is the image space and $\mathcal{Y}$ is the corresponding label space). Specifically, the i-th species (domain) $\mathcal{D}_i$ contains N_i samples, *i.e.*, $D_i = \{(x_j^{(i)}, y_j^{(i)})\}_j^{N_i}$. The goal of cross-species pose estimation is to train a pose estimator on $\mathcal{D} = \{D_1, D_2, ...D_M\}$ and then directly apply it to a novel unseen species $D_u (u \notin \{1, 2, ..., M\})$, which requires good cross-domain generalizability. The trained deep model consists of a feature extractor F and an estimator head G, which are parameterized by ϕ_F and ϕ_G, respectively.

The framework of the proposed D-Gen is illustrated in Fig. 5.2. The key motivation is that some joint relations are consistent across all the species and are thus beneficial for cross-species generalization, while some other joint relations are inconsistent and harmful. Therefore, D-Gen seeks to break the inconsistent relations while preserving the consistent ones. To this end, D-Gen consists of two stages, *i.e.*, joints decomposition (Section 5.2.2) and network split (Section 5.2.3). In the joint decomposition stage, D-Gen divides the body joints into several joint concepts so that each concept contains multiple closely-related joints. Section 5.2.2 introduces three different decomposition strategies, *i.e.*, heuristic, geometric and attention-based decomposition. Afterward, Section 5.2.3 proposes a very simple network split method that splits the top layers of the backbone network into several concept-

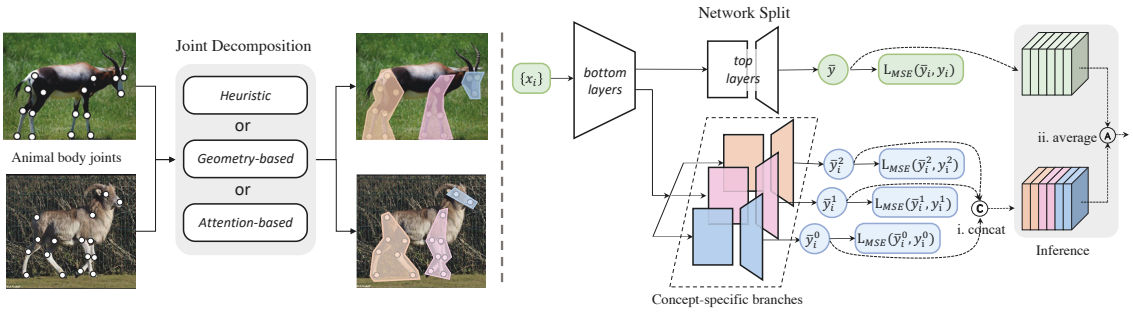


Figure 5.2 : Overview of "Decompose to Generalize" (D-Gen) scheme. D-Gen consists of two stages, *i.e.*, joints decomposition (left) and the sub-sequential network split (right). **1)** In the joints decomposition stage, D-Gen leverages different strategies (*e.g.*, heuristic, geometry-based or attention-based) to divide the body joints into several joint concepts, so that each concept contains closely-related joints (Section 5.2.2). **2)** Given the decomposed joint concepts, D-Gen correspondingly splits the top layers of the baseline network into multiple concept-specific branches (Section 5.2.3). This network split suppresses the interaction between inter-concept joints and yet preserves the interaction within each concept. **3)** During inference, D-Gen concatenates the predictions of all the concept-specific branches (step i) and averages them with the baseline prediction (step ii).

specific branches, corresponding to the decomposed joint concepts. Importantly, Sec. 5.2.3 also investigates the mechanism through gradients analysis and reveals that the network split suppresses the gradient conflict among inconsistently-related joints (*i.e.*, different concepts).

5.2.2 Joints Decomposition

We employ three different concept decomposition strategies, *i.e.*, the heuristic, geometry-based, and the attention-based strategy. The attention-based decomposition is learned based on the baseline network (*i.e.*, we insert an attention module between the feature extractor and the estimation head in the baseline, as in

Fig. 5.3). The heuristic decomposition is drawn from the anatomy knowledge, and the geometry-based decomposition is learned through the geometric distance. Empirically, we find the latter two decomposition strategies are indeed inferior to the attention-based strategy. However, we think the corresponding exploration is valuable for showing that joint decomposition can bring general benefit to cross-species generalization (regardless of the decomposition strategy).

Attention-based Decomposition.

We first learn a pixel-to-concept attention Fig. 5.3 (a) and then use the attention results to infer the decomposition (Fig. 5.3 (b)).

Learning pixel-to-concept attention. As shown in Fig. 5.3 (a), we append an attention module after the feature extractor to discover the correlation between different joint features. The attention module takes the feature maps $Z \in \mathbb{R}^{hw \times d}$ (h and w are the spatial size of the feature map with d channels) as its input. Meanwhile, we pre-define that there are k concepts (k is a hyper-parameter) and correspondingly provide k concept embeddings, *i.e.*, $E \in \mathbb{R}^{k \times d}$ for learning pixel-to-concept attention. Given the feature maps Z and the concept embeddings E, the attention module uses three linear projection to get the `Query`, `Key` and `Value`, respectively, which is formulated as:

$$\texttt{Query} = ZW_q, \texttt{Key} = EW_k, \texttt{Value} = EW_v, \tag{5.1}$$

where $W_q/W_k/W_v \in \mathbb{R}^{d \times d_l}$ are linear layers that project the inputs into the identical low dimension space. We note that the concept embeddings E and the three linear projections are all learnable through the attention mechanism.

The pixel-to-concept attention is defined as:

$$Z^* = Z + W_m(softmax(\frac{ZW_q(EW_k)}{\sqrt{d_h}})(EW_v)), \tag{5.2}$$

where W_m is a linear layer that maps the dimension of features back to d, and the residual connection is added for training stability.

The above attention makes each pixel-level features absorb information from the shared set of concept embeddings. The resulting feature maps Z^* are then fed into the estimation head. We note that while our intention is to use such pixel-to-concept attention as the clues for joint decomposition, the attention actually has another positive side-effect: it already promotes interaction among closely-related joints in an implicit manner. It is because the closely-related joints are likely to absorb information from the same concept embedding(s), thus gaining implicit interactions.

Decomposition by comparing pixel-wise features to concept embedding. After the model finishes learning the pixel-to-concept attention, we use the already-learned k concept embeddings $E \in \mathbb{R}^{k \times d}$ to infer the decomposition, as illustrated in Fig. 5.3 (b). Specifically, given an image and the predicted position of each visible joint, we extract the corresponding joint features from the feature maps Z. Let $\hat{z}_(i, m)$ denotes the extracted feature of the m-th joint from the i-th image. According to the cosine similarity between $\hat{z}_(i, m)$ and the concept embeddings $E = \{e_j\}_{j=1}^{k}$, the concept-of-interest of the m-th joint in the i-th image is inferred by:

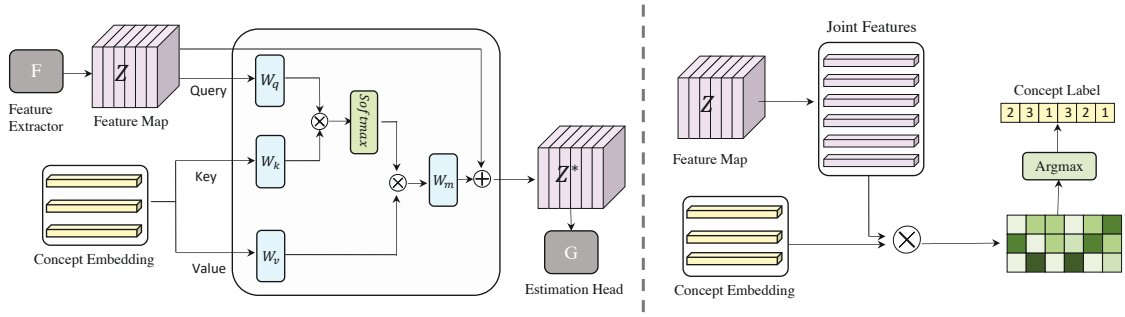
$$C_{i,m} = \arg\max_{j} \cos < \hat{z}_{i,m}, e_j >, \quad j = 1, 2...k, \tag{5.3}$$

where the $<, >$ denotes the operation of getting the angle of two vectors.

Finally, the concept-of-interest for the m-th joint is voted from all the training samples.

Heuristic and Geometry-based Decomposition.

In addition to the attention-based decomposition, we explore two alternatives, *i.e.*, the heuristic and geometry-based decomposition.



(a) The proposed attention module

(b) Concept generation with the learned attention

Figure 5.3 : Attention-based decomposition. Left: Illustration of the proposed attention mechanism. The attention module takes the feature map as query, and a learnable concept embedding as the key for learning the affinity matrix, which is leveraged to promote the inter-joints interactions. Right: Illustration of the concept generation. For the learned feature map, we first extract the joint features based on the predicted joint location. Then with a simple nearest neighbor search with concept embedding, we could obtain the concept label for each joint. The final division of concepts is obtained from the voting from all training images. $\bigotimes$ denotes the matrix multiplication and $\bigoplus$ denotes the element-wise sum.

Heuristic decomposition. From the perspective of human knowledge, the joints could be simply divided according to anatomy. Therefore, the heuristic strategy draws the knowledge from the human prior, and decomposes the body joints into three parts, *i.e.*, head, hind leg, and foreleg. Intuitively, the joints within each single part have relatively consistent mutual relations. That being said, experimental results (Sec. 5.3.2) show that the improvement achieved by heuristic decomposition is relatively small, indicating that relying on intuition is a sub-optimal choice.

Geometry-based decomposition. considers the distances between every pair of joints and can be viewed as a pure-structural strategy. The joint distances are derived from the ground-truth joint position and are used to construct an affinity matrix of the joints. Given the affinity matrix, we employ spectral clustering [142,

176] to cluster all the joints into multiple concepts. Some examples of the geometry-based concepts are shown in Fig. 5.6.

5.2.3 Network Split

Given the joint concepts from a decomposition strategy, D-Gen correspondingly splits the top layers of the baseline network into multiple concept-specific branches, as illustrated in Fig. 5.2 (the right part). Specifically, the top layers refer to the last l_{top} residual blocks in the backbone (ResNet), while the bottom layers are the rest. Here, we analyze the rationale for this simple network split through gradient analysis.

To be concrete, we use gradient conflict [215] to estimate the inter-joint relationships during the optimization. Assuming σ_{ij} is the angle between the gradient of joint i and joint j, the conflict rate is defined as the proportion of gradients holding different directions, *i.e.*, $cos\sigma_{ij} < 0$. As shown in Fig. 5.4, the joints from the same concept exhibit higher consistency in the gradient space (*i.e.*, lower conflict rate) while joints from different concepts show a higher possibility to conflict with each other, thus hampering the optimization. Therefore, an intuitive approach to circumvent the gradient conflict is to isolate the optimization of severely-conflict joints (*i.e.*, different concepts) into different network branches, resulting in the proposed network split.

The overall training objective is formulated as:

$$\mathcal{L} = \mathcal{L}_{MSE}(\phi_G(\phi_F(x)), y) + \sum_{i}^{k} \mathcal{L}_{MSE}(\hat{\phi}_G^i(\phi_F(x)), y^i). \tag{5.4}$$

where $\hat{\phi}_G^i$ denotes the feature extractor with the i th concept branch and y^i denotes the ground truth for the i th concept. We use $\mathcal{L}_{MSE}()$ to denote the mean squared error.

During inference, we first concatenate the output of concept-specific branches

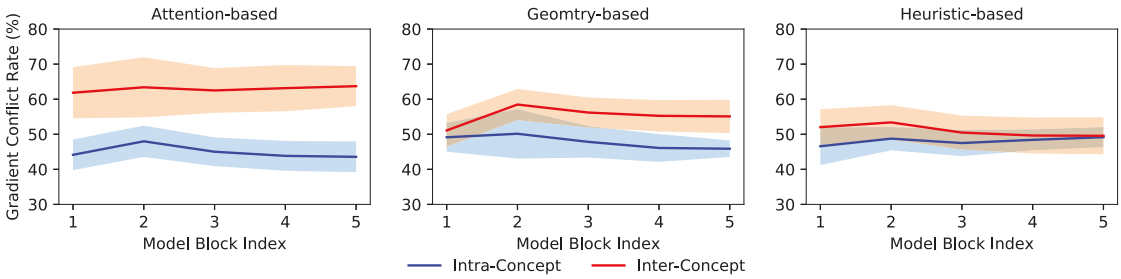


Figure 5.4 : Gradient conflict rate for three decomposition strategies. Generally, the gradient conflict among intra-concept joints is lower than the conflict among inter-concept joints. Moreover, compared with the other two strategies, the attention-based decomposition achieves lower conflict for intra-concept joints, indicating better decomposition. An effective approach to circumvent the gradient conflict is to isolate the optimization of severely-conflicted joints into different branches.

and then average them with the baseline branch to obtain the final prediction. Notably, before the concatenation, we first sort them according to their original joint index as the joints are not sequentially divided.

Discussion: The improvement is NOT mainly due to model ensemble. A side effect of network split is the increase of model size. More specifically, D-Gen splits the top layers of the baseline network and thus can be viewed of the ensemble of multiple concept-specific branches. However, we note that model ensemble (or increasing the model size) is not the major reason for our improvement. An explicit evidence is that: if we replace the joints decomposition strategy with random strategy (*i.e.*, randomly dividing the joints into several concepts, as detailed in Sec. 5.3.2), the resulting D-Gen only gains very slight improvement, *i.e.*, +1.5% on Animal Pose Dataset (Table 5.3). In contrast, using attention-based joint decomposition, D-Gen obtains +6.5 % improvement. This observation confirms that model ensemble is only a trivial reason for the improvement of D-Gen.

5.3 Experiments

5.3.1 Experimental Setup

Dataset. We evaluate our method on two large-scale animal datasets. *AP-10K* [214] is a large-scale benchmark for mammal animal pose estimation, and contains 23 animal families and 54 species following the taxonomic rank. It annotates 10,015 images and provides annotations with **17 keypoints**. *Animal Pose Dataset* [17] collects and annotates 5 species, *i.e.*, cat, dog, sheep, cow, and horse, from 3000+ images, where both bounding boxes and keypoints annotated (**20 keypoints**) are provided. *Animal Kingdom* [143] is another dataset with a higher diversity that includes mammals, fishes, birds, amphibians, and reptiles with **23 keypoint** annotations.

Training. The training process consists of two stages, i.e., the first stage derives the concept division with the learned concept embeddings, and the second stage optimizes with the concept-specific branches. To be specific, the first stage takes 60 epochs and the second takes 150 epochs. As for our one-stage solutions, we optimize them for 210 epochs. For the attention-based decomposition, the attention module is activated for both stages.

Evaluation. Following [180, 34], we evaluate the pose similarities with OKS:

$$\text{OKS} = \frac{\sum_i \exp(-d_i^2/2s^2k_i^2)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}. \tag{5.5}$$

Here d_i denotes the Euclidean distance between a detected keypoint and its corresponding ground truth, v_i is the visibility flag of the ground truth, s is the object scale, and k_i is a per-keypoint constant that controls falloff. The reported average precision is the average of AP scores at OKS = (0.50, 0.55, ..., 0.90, 0.95.)

Data augmentation. In training, we employ the following data augmentation: random rotation ([−30°, 30°]), random scale ([0.75, 1.5]), random translation ([−40,

40]) to crop an input image patch of size 256×256 and random flip. In inference, only resize and normalization is performed.

Evaluation. Follow [180], the evaluation metric is based on Object Keypoint Similarity (OKS). We report the mean average precision (AP) at OKS=0.50, 0.55,...0.90, 0.95. The mean value and standard error over three *random* runs are reported in all main results.

Implementation details. We start from ResNet50 [69] with the backbone pretrained on ImageNet [43]. The batch size is set to 64, and the learning rate for the first stage and the second is set to 5×10^{-4} and 5×10^{-5}, respectively. We optimize the model with Adam for 210 epochs, where the learned rate decrease ($\times 10^{-1}$) at 170 and 200, respectively. The size of the input image is 256×256 and the heatmap is with size 64×64. The number of the concept-specific blocks is set to 2, and k is set to 3 for all transfer tasks.

5.3.2 Comparison with Previous Methods.

We evaluate our approach under two domain generalization (DG) scenarios, *i.e.*, intra-family DG and inter-family DG, where the latter involves larger domain gaps regarding both the visual and structural knowledge. Following the common practice, we perform domain generalization with the leave-one-out setting, *i.e.*, selecting one species / family as the target domain and the rest as the source domains. Since these two datasets are relatively new and only a few results have been reported, we re-implement representative DG methods, *i.e.*, Fish [177], SWAD [22], MixStyle [235], and GradSur [131] for comparison and we adopt ERM (empirical risk minimization that trains on all source domains), Oracle (training on target domains), and Oracle-Full (train on both source and target domains) as baselines. Moreover, we use a random decomposition in addition to the three discussed decomposition strategies to show that model ensemble (through random-decomposed branches) does not promise

Table 5.1 : Intra-family DG results (AP) on AP-10K [214] with the leave-one-out manner. K.C.=King Cheetah, S.L.=Snow Leopard, and A.S. = Argali Sheep. Num.= number of training samples for this species.

						Previous SOTA				The proposed D-Gen			
Family	Target	Num.	ERM	Oracle	Oracle-F	MixStyle	GradSur	SWAD	Fish	Random	Heuristic	Geometry	Attention
Felidae	Bobcat	151	56.4	58.4	87.1	58.1	62.0	55.1	60.4	57.9	58.8	58.2	**63.2**
	Cat	307	30.2	54.8	79.7	40.0	28.1	27.0	34.1	35.3	37.8	35.1	**41.0**
	Cheetah	148	58.6	58.7	86.3	59.0	57.9	54.1	58.5	52.2	56.0	53.1	**60.1**
	Jaguar	187	61.8	58.4	93.2	62.1	60.9	**62.5**	58.7	56.8	54.8	56.2	53.7
	K. C	22	53.1	30.6	89.8	58.6	62.0	63.6	59.0	64.6	62.9	**67.9**	62.1
	Leopard	142	57.8	51.1	88.1	60.4	57.7	**64.3**	60.4	57.7	57.1	58.2	55.4
	Lion	177	40.3	45.3	84.9	**45.7**	34.7	40.9	41.4	40.3	36.6	37.4	38.8
	Panther.	106	41.5	55.4	86.2	45.5	40.7	38.6	37.2	46.2	43.9	44.3	**48.4**
	S.L.	73	65.4	54.8	95.4	69.0	**73.7**	67.3	61.0	61.2	62.0	63.9	64.9
	Tiger	144	58.9	68.9	87.6	57.1	59.3	47.4	57.8	56.2	**64.2**	62.3	62.4
	Average	–	52.4 ± 0.6	53.6 ± 0.3	87.8	53.1 ± 0.9	53.7 ± 0.5	52.1 ± 0.6	53.9 ± 1.2	52.9 ± 1.0	53.4 ± 0.9	53.7 ± 0.3	**55.0 ± 0.5**
Ursidae	Black Bear	39	30.0	43.2	85.2	37.7	37.5	30.6	24.8	26.8	36.1	**41.1**	40.7
	Brown Bear	171	15.5	42.2	83.1	14.9	17.6	24.1	23.7	15.8	**28.0**	20.4	24.8
	Panda	164	9.0	46.9	79.0	13.9	7.5	11.1	17.5	13.8	13.2	12.7	**14.5**
	Polar Bear	156	10.6	47.1	85.1	9.1	11.5	13.3	17.6	9.1	**21.2**	8.3	18.7
	Average	–	18.8 ± 0.4	44.9 ± 0.2	83.1	18.9 ± 0.3	18.6 ± 0.5	19.8 ± 0.4	20.4 ± 0.5	18.8 ± 0.9	21.0 ± 0.6	20.6 ± 0.4	**24.7 ± 0.4**
Bovidae	Antelope	298	51.3	63.1	88.4	52.6	53.4	52.4	56.2	56.0	53.4	53.9	**60.9**
	A.S	268	69.7	70.1	95.6	71.3	74.1	67.6	75.1	72.6	73.6	74.3	**77.3**
	Bilson	208	42.8	51.6	91.8	49.8	43.6	46.2	47.7	46.1	48.8	45.8	**49.8**
	Buffalo	228	61.0	71.6	91.4	57.5	64.8	63.7	62.3	60.0	61.1	62.9	**68.1**
	Cow	228	46.6	43.9	82.1	50.7	46.1	49.7	46.9	48.9	49.7	48.8	**51.8**
	Sheep	355	41.7	57.4	85.0	41.0	42.1	**45.1**	41.6	40.6	41.0	41.2	40.1
	Average	–	52.2 ± 0.6	59.6 ± 0.3	89.1	53.8 ± 0.6	54.1 ± 0.4	53.9 ± 0.5	55.0 ± 0.4	53.8 ± 0.8	54.6 ± 0.3	54.5 ± 0.4	**57.9 ± 0.4**

improvement.

Intra-family DG. For intra-family domain generalization, we evaluate our approach on the three largest families in AP-10K, *i.e.*, Bovidae, Canidae, and Ursidae. The results are summarized in Table 5.1, from which we draw three observations.

1) Comparing "Previous SOTA" against the "ERM" baseline, we find the improvement achieved by previous state-of-the-art DG methods is very trivial (if there is any). We infer it is because these methods mainly focus on the style gap of the source and target domain, and are not capable to tackle the distribution shift in terms of structural and visual relations.

2) Comparing the proposed D-Gen against the "ERM" baseline, we observe that all the three decomposition strategies (heuristic, geometry-based, and attention-based) achieve consistent improvement. For example, within the Bovidae family,

these three methods surpass the ERM baseline by +2.3%, +3.2%, and +5.2%, respectively. This observation validates the effectiveness of D-Gen.

3) Comparing the three decomposition strategies against each other and the additional "Random" strategy, we observe that the attention-based strategy is the best among all. It indicates that the deep network is capable to discover better joint concepts than human intuition and pure-geometric knowledge. Another important observation is that the "Random" strategy barely achieves any improvement over the baseline. It indicates that D-Gen barely benefits from the model ensemble, because the random strategy already takes the advantage of model ensemble.

4) Under circumstances with fewer labeled samples, domain generalization can serve as a powerful baseline for pose estimation. For example, our method surpasses the Oracle with 4.8 % on the species Bobcat from the family Felidae, where only 150 images are provided with annotations.

5) In certain situations, the attention-based solution yields results that are inferior to others. This disparity can be attributed to the fact that, in several scenarios, domain discrepancies may arise from factors beyond mere geometry and appearance. For instance, in the case of Ursidae species, which share similar skeletal structures, geometrical relationships play a minimal role in contributing to the domain gap. Consequently, this leads to the suboptimal performance of both geometry-based and attention-based solutions. However, notwithstanding these challenges, our method demonstrates noticeably superior performance overall.

Inter-family DG. We further evaluate the proposed D-Gen under the inter-family DG scenario. Without loss of generality, we select several animal families with relatively large diversity for each dataset (Table 5.2 for AP-10K and Table 5.3 for Animal Pose Dataset). We observe our method maintains its superiority against previous methods, which is consistent with the observation under the intra-family

Table 5.2 : Inter-family DG results (AP) on AP-10K [214] with the leave-one-out manner. Cerc.=Cercopithecidae.

					Previous SOTA				The proposed D-Gen			
Target	Num.	ERM	Oracle	Oracle-F	MixStyle	GradSur	SWAD	Fish	Random	Heuristic	Geometry	Attention
Bovidae	1467	48.2	64.8	90.3	49.6	47.0	48.8	49.3	48.6	46.0	49.9	**52.1**
Felidae	1457	39.7	64.4	87.3	43.1	34.2	42.4	39.8	43.0	42.8	41.9	**47.7**
Canidae	1130	50.2	65.9	88.8	48.2	46.8	53.6	50.6	53.3	49.7	53.9	**54.8**
Ursidae	530	30.3	68.8	83.7	31.3	**34.5**	31.4	32.3	31.5	30.7	28.4	31.1
Cerc.	623	19.9	67.9	84.2	20.3	19.1	21.8	24.4	24.7	**40.8**	23.0	27.2
Equidae	482	37.8	64.2	82.6	37.9	34.2	40.5	38.8	39.7	40.6	**42.8**	39.0
Hominidae	345	39.6	66.9	83.1	38.4	32.4	38.7	39.7	38.4	40.0	**46.1**	45.6
Average	–	37.7 ± 0.4	66.1 ± 0.2	85.7	38.4 ± 0.4	35.5 ± 0.5	39.8 ±0.6	39.3 ± 0.5	39.6 ± 0.9	40.0 ± 0.7	40.9 ± 0.7	**42.9 ± 0.6**

Table 5.3 : Results (AP) on Animal Pose Dataset [17] with the leave-one-out manner.

				Previous SOTA				The proposed D-Gen			
Target	Num.	ERM	Oracle	MixStyle	GradSur	SWAD	Fish	Random	Heuristic	Geometry	Attention
Cow	1214	22.6	36.8	25.0	24.4	18.8	27.8	26.7	**29.9**	25.7	29.6
Sheep	980	21.6	33.0	23.2	22.0	21.6	21.2	20.4	26.1	25.4	**28.4**
Horse	651	22.7	37.5	27.8	26.2	26.1	25.4	28.5	31.7	**33.6**	31.8
Cat	614	26.3	30.7	27.8	24.8	25.1	26.3	24.3	23.9	23.2	**28.1**
Dog	502	21.2	31.1	25.1	28.2	22.2	26.2	**24.4**	23.7	25.0	26.4
Average	–	23.4 ± 0.4	33.8 ± 0.3	25.8 ± 0.6	23.9 ± 0.4	22.7 ± 0.5	25.4 ± 0.4	24.9 ± 0.8	27.1 ± 0.4	26.6 ± 0.5	**28.9 ± 0.4**

scenario. Moreover, we observe that under the inter-family scenario, the achieved results are lower than the intra-family scenario. It confirms our intuition that inter-family generalization is more challenging due to larger cross-species distribution shifts.

5.3.3 Ablation studies

In this section, we conduct ablation studies under two scenarios, *i.e.*, intra-family DG (within the Bovidae family, in particular) and the inter-family DG, to further investigate the proposed method.

Different number of concepts (k). In Fig. 5.5 (a)(b), we report the results with varying k to evaluate its influence. With k increasing, these decomposition solutions show different tendencies, *i.e.*, the attention-based variant only fluctuates

Table 5.4 : Results (PCK@0.05) on Animal Kingdom Dataset [143] with the leave-one-out manner. AM=Amphibians.

Family	Mammals	AM	Reptiles	Birds	Fishes	Avg.
ERM	11.5	15.0	11.4	13.2	12.4	12.7
Oracle	37.3	69.1	66.5	52.0	45.5	54.1
SWAD	13.4	18.3	15.5	17.4	12.6	15.4
Fish	16.8	17.9	17.7	18.1	16.6	17.4
Heur.	16.6	16.9	18.1	18.8	15.1	17.1
Geom.	17.1	18.5	18.9	19.1	16.1	17.9
Att.	18.3	19.0	19.4	19.9	18.8	**19.1**

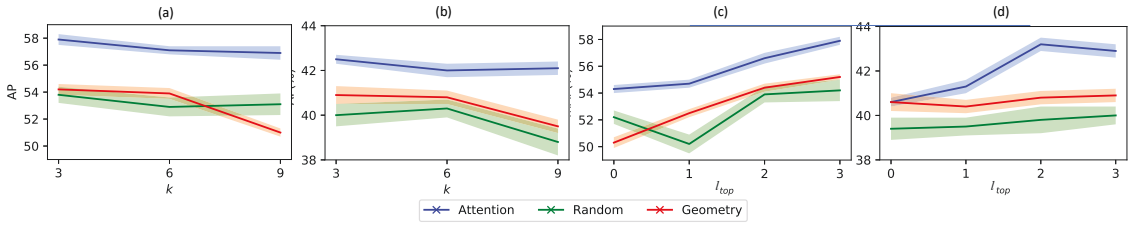


Figure 5.5 : The impact of k (the number of concepts) and l_{top} (the number of concept-specific blocks) with error bars. Both the intra-family (a and c) and the inter-family scenarios (b and d) are presented.

in a small range, while the other two show degraded performance. Such a phenomenon verifies that the attention-based solution is not sensitive to the selection of k. Further investigation reveals that, as the k increased, the concept embedding still derives concept groups with a stable number, *i.e.*, from 3 to 5. The reason is that, when k is large, only part of the concept embeddings attend and modulate the inter-joint interactions while others not.

How does l_{top} (number of concept-specific blocks) influence the generalization results? In Fig. 5.5 (c)(d), we present the sensitivity analysis to the number of concept-specific blocks (l_{top}). As we could observe, with increasing l_{top}, the performance improves constantly, justifying the necessity of the network split

Figure 5.6 : Concept visualization on AP-10K, where different colors denote different mined concepts.

pipeline. Especially, even with the shared backbone, our method still achieves very competitive results, *i.e.*, 40.6 for inter-family DG, proving its superiority against previous methods. In contrast, the other two solutions only achieve very limited improvement with the increasing l_{top}. The reason is that, as illustrated in Fig. 5.4, the joints of their mined concepts still exhibit high conflict with each other, which is the network split cannot mitigate.

Visualization of the learned concepts. Fig. 5.6 compares three joint decomposition strategies through visualization. For a clear comparison, we adopt three concepts (painted in red, blue, and green color) for all the strategies. We observe that the attention-based decomposition tends to group joints from comparably distant positions into a concept, benefiting from the learning of long-range dependencies with the attention module. In contrast, the heuristic and geometry-based solutions mainly focus on the local structures and tend to associate joints with their neighbors.

Ablation of the pixel-to-concept attention mechanism. We note that the pixel-to-concept attention mechanism actually has two impacts on the attention-based decomposition, *i.e.*, 1) it learns clues for joint decomposition, and 2) it incurs interaction between joints and concept proxies, while the other two decomposition strategies do NOT have such effect. Therefore, although Table 1 and Table 2 in the manuscript show that the attention-based decomposition achieves the largest improvement among all the four strategies, we are not clear what is the exact reason for its superiority. We answer this question by adding the same attention module into the network under the other three decomposition strategies (without changing their joint decomposition results). The results are summarized in Table 5.5 (intra-family) and Table 5.6 (inter-family), from which we draw two observations as below:

First, we observe that adding the attention module for joint interaction does not promise improvement. For example, it slightly compromises the random and the heuristic decomposition (53.8→53.0 and 54.6→52.4) under the intra-family DG scenario. Second, for geometry-based decomposition, adding the attention module leads to a slight improvement, *i.e.*, + 0.6 for geometry-based solution in Table 5.5. However, it is worth noting that, even with the attention, it still cannot compete with the attention-based decomposition. Combining these two observations, we conclude that the superiority of the attention-based solution should not be mainly ascribed to the pixel-to-concept interactions but to better joint concepts.

What if treat all joints as one concept or treat each joint as individual concepts? In Table 5.7, we complement the results when manually set $k = 1$ and $k = 17$, which treats all joints one concept or treat each joint as one concept. Despite the network split, the manual concept division barely improves the performance over the ERM baseline, which again justifies the effectiveness of the proposed D-Gen paradigm. These results also validate that simply increasing the model capacity cannot promise improvement in the generalization.

Table 5.5 : Ablation studies on attention module for intra-family domain generalization (family Bovidae). Experiments here are conducted on AP-10K [214] with the leave-one-out manner. A.S. = Argali Sheep.

Decomposition	With attention	Target Domain						Average
		Antelope	A.S.	Bilson	Buffalo	Cow	Sheep	
Random		56.0	72.6	46.1	60.0	48.9	40.6	53.8
	✓	52.6 (↓)	67.7 (↓)	51.1 (↑)	60.0 (−)	43.7 (↓)	42.8 (↑)	53.0 (↓)
Heuristic		53.4	73.6	48.8	61.1	49.7	41.0	54.6
	✓	49.6 (↓)	69.8 (↓)	48.4 (↓)	58.9 (↓)	47.8 (↓)	39.7 (↓)	52.4(↓)
Geometry		53.9	74.3	45.8	62.9	48.8	41.2	54.5
	✓	55.4 (↑)	70.3 (↓)	47.0 (↑)	63.8 (↑)	48.8 (−)	44.8 (↑)	55.1 (↑)
Attention	✓	60.9	77.3	49.8	68.1	51.8	40.1	57.9

Comparison on more datasets. In Table 5.10, we present the generalization results to Horse-C dataset [133]. Following their evaluation protocol, we validate the effectiveness of the proposed method on this benchmark and report the results, and again attains superior results.

Analysis on joint feature extraction In Table 5.11, we testify the sensitivity to the feature extraction process, i.e., localizing the joint features with ground truth or the prediction. The negligible difference verifies that our method is robust to the choice on clues for feature extraction.

Analysis on model capacity In Table 5.9, we present the analysis of the model capacity, *i.e.*, FLOPs and inference speed. And in Table 5.8, we compare the time for each training iteration . Apparently, our method does not require longer time even with two stages when compared with some previous SOTAs. This is because some of them employ extra inner steps during the optimization. For example, Fish [177] requires to perform n inner steps to aggregate the gradients of n domains independently.

Table 5.6 : Ablation studies on attention module for inter-family domain generalization. Experiments here are conducted on AP-10K [214] with the leave-one-out manner. Cerc.=Cercopithecidae.

Decomposition	With attention	Target Domain							Average
		Bovidae	Felidae	Canidae	Ursidae	Cerc.	Equidae	Hominidae	
Random		48.6	43.0	53.3	31.5	24.7	39.7	38.4	39.6
	✓	49.3 (↑)	42.0 (↓)	51.2 (↓)	32.0 (↑)	25.6 (↑)	40.6 (↑)	37.7 (↓)	39.8 (↓)
Heuristic		46.0	42.8	49.7	30.7	40.8	40.6	40.0	40.0
	✓	46.6 (↑)	43.9 (↑)	49.2 (↓)	32.8 (↑)	29.7 (↓)	42.2 (↑)	41.3 (↑)	40.8 (↑)
Geometry		49.9	41.9	53.9	28.4	23.0	42.8	46.1	40.9
	✓	50.8 (↑)	42.6 (↑)	54.8 (↑)	36.4 (↑)	25.2 (↑)	39.9 (↓)	42.6 (↓)	41.8 (↑)
Attenion	✓	52.1	47.7	54.8	31.1	27.2	39.0	45.6	42.9

Evaluation of uncertainty. In Table 5.12, we repeat our experiments on two more data splits with three different random seeds. Under such variation, we re-evaluate the uncertainty in two scenarios, i.e., intra-family DG within Bovidae and inter-family DG on AP-10K. The results show consistent uncertainty with the main paper on both scenarios.

More qualitative comparisons between different joint decomposition strategies. In Fig. 5.7, we provide more visualizations of the mined concepts with different decomposition strategies, and we make the following observations:

1) Despite being intuitively reasonable, heuristic decomposition may induce ambiguity between some hard-to-distinguish joints. For example, the left and right forelegs are assigned to the same concept, while distinguishing them within the same concept might not be easy.

2) For the geometry-based solution, we notice that it tends to focus on and identify closely-related structures in the geometric space, *i.e.*, the forelegs and the hind legs.

3) In Fig. 5.7 (c), we find that the left eye and the right eye are assigned to

Table 5.7 : DG results (AP) on AP-10K [214] under varying k (number of concepts) with D-Gen paradigm or manual setting.

	D-Gen				Manual	
	Decom.	3	6	9	1	17
Intra-DG	Attention-based	57.9	57.1	56.9	52.8	51.9
	Geometry-based	54.2	53.9	51.0		
	Random	53.8	52.9	53.1		
Inter-DG	Attention-based	42.5	42.0	42.1	38.0	38.2
	Geometry-based	40.9	40.8	39.5		
	Random	39.6	40.3	38.8		

Table 5.8 : Comparison on the training time of each iteration.

Method	ERM	MisStyle	GradSur	SWAD	Fish	Ours
Time of iteration (ms)	250	251	310	290	2246	280

different concepts. This result may seem counter-intuitive at first glance but is actually reasonable from a closer look. It is because the left and right eyes are very small and hard to distinguish. Therefore, associating the left and right eyes with some different easy-to-distinguish joints helps to discriminate them, as well. On the contrary, both the heuristic and geometry-based strategies assign the left and right eyes to the same concept and are actually inferior.

Qualitative comparisons on pixel-to-concept attention. To better understand the pixel-to-concept attention, we present the visualization of the learned attention maps in Fig. 5.8. As we could observer, each concept embedding apparently favors specific regions, and such preference is consistent over different species. Such a phenomenon verifies that the proposed attention mechanism indeed encourages the inter-joint relationships that can benefit the generalization.

Qualitative comparison with previous solutions on pose estimation

Table 5.9 : Analysis on the model capacity.

Model configuration	FLOPs (G)	Inference speed(MS)
Original Network (ResNet-50)	21.0	9.2
with network split	24.8	12.4

Table 5.10 : PCK@0.3(%) for out-of-distribution generalization to Horse-C [133] (FF=front foot; HF = Hind foot; HH = Hind Hock).

Method	Nose	Eye	Shoulder	Wither	Elbow	NearFF	OffFF	Hip	NearHH	NearHF	OffHF	Average
[133]	68.2	73.6	85.4	85.8	88.1	72.6	70.2	89.2	85.7	77.0	74.1	79.1
D-Gen (attention)	67.9	76.4	86.1	83.8	88.3	79.6	74.3	90.1	86.6	79.3	76.6	80.9

results. In Fig. 5.9 and Fig. 5.10, we present the qualitative results on intra-family and inter-family DG, respectively, and compare with previous solutions. Compared with the previous solution, the proposed D-Gen paradigm can deliver more accurate estimations, proving the effectiveness of the proposed method. Concretely, the larger variation in appearance and shape renders previous solutions less effective, especially on the joints from non-rigid parts, *e.g.*, legs. In contrast, our method, especially the attention-based variant, shows stronger robustness to them and delivers more accurate estimations on both intra-DG and inter-DG scenarios.

Table 5.11 : Results with different clues for joint feature extraction. Experiments here are under the inter-family DG scenario on AP-10K [214]. Cerc.=Cercopithecidae.

Extraction clue	Target Domain							Average
	Bovidae	Felidae	Canidae	Ursidae	Cerc.	Equidae	Hominidae	
Ground truth	53.1	47.2	52.3	35.9	28.7	47.5	46.6	43.2
Prediction	52.1	47.7	54.8	31.1	27.2	39.0	45.6	42.9

Table 5.12 : More random runs for uncertainty evaluation. Experiments here are conducted on AP-10K [214] in the leave-one-out manner. The intra-family DG is performed on the family Bovidae.

Transfer	Data Split 1				Data Split 2			
	Random run 1	Random run 2	Random run 3	Average	Random run 1	Random run	2 Random run 3	Average
Intra-Family DG	57.91	57.13	58.03	57.69 ± 0.49	57.47	58.34	57.32	57.81 ±0.47
Inter-Family DG	43.13	42.14	42.63	42.63 ± 0.50	42.92	43.31	41.98	42.73 ± 0.68

(a) Heuristic decomposition

(b) Geometry-based decomposition

(c) Attention-based decomposition

Figure 5.7 : Concept visualization on AP-10K [214], where different mined concepts are in different colors. A noticeable observation is that the attention-based strategy assigns the left and right eyes to different concepts. This decomposition result seems counter-intuitive but is actually reasonable: it associates the hard-to-distinguish left and right eyes with different easy-to-distinguish joints, so that the latter joints provide clues for distinguishing the left and right eyes.

Figure 5.8 : Attention weight visualization on AP-10K [214], where the bottom two rows correspond to the heatmap of different concepts. With a variety of animal species, *i.e.*, cow, wolf, sheep, and dog, the concept embedding can effectively attend and associate specific keypoints, which further justifies the effectiveness of the proposed approach.

Figure 5.9 : Pose estimation result under the intra-family DG on AP-10K [214]. Experiments here follow the leave-one-out protocol on Family Bovidae and with the Antelope as the target domain. Compared with solutions, our method demonstrates stronger capability on joint localization and identification, especially on joints from non-rigid part, *e.g.*, legs.

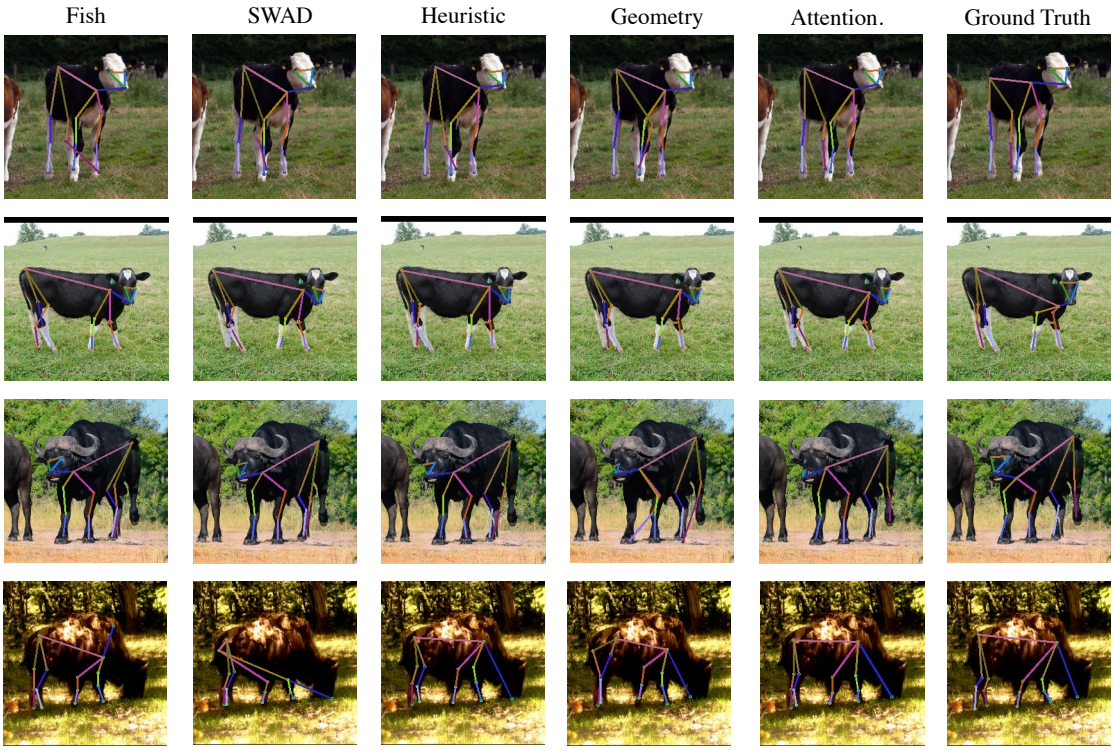


Figure 5.10 : Pose estimation result under the inter-family DG on AP-10K [214]. Experiments here follow the leave-one-out protocol and the target domain is Bovidae. Under a larger gap between species, our method maintains its superiority against previous solutions.

5.4 Conclusion

In this chapter, we propose a "decompose to generalize" (D-Gen) scheme for cross-species animal pose estimation. In contrast to generic domain generalization methods, D-Gen focuses on the joint relation for improving the cross-species generalization. Specifically, D-Gen decomposes the body joints into several joint concepts, so that the joints in a single concept have relatively consistent relations. Based on the joint concepts, D-Gen splits the feature extractor into multiple concept-specific branches. This simple network split suppresses the inconsistent interactions between inter-concept joints and yet maintains the consistent interactions between intra-concept joints. Experimental results validate the effectiveness of D-Gen. We hope our work on cross-species pose estimation can provides a new viewpoint for understanding the domain generalization problem.

Chapter 6

Domain Consensus Clustering for Universal Domain Adaptation

Venturing into new territories, Chapters 6 and 7 shift the focus towards the investigation of generalizability with novel categories. In this chapter, the discussion centers around a novel clustering algorithm, which is proposed to effectively differentiate between known and novel categories in various scenarios of category shift.

6.1 Introduction

Deep convolutional neural networks have achieved significant progress in many fields, such as image classification [178, 75], semantic segmentation [26, 27], *etc.* However, as a data-driven technique, the severe reliance on annotated in-domain data greatly limits its application to cross-domain tasks. As a feasible solution, unsupervised domain adaptation (UDA) [146] tries to solve this by transferring the knowledge from an annotated domain to an unlabeled domain, and has achieved significant progress in multiple tasks [99, 102, 87, 89, 108, 122]. Despite UDA's achievement, most UDA solutions assume that two domains share identical label set, which is hard to satisfy in real-world scenarios.

In light of this, several works considering the unaligned label set have been proposed: open set domain adaptation, partial domain adaptation, and universal domain adaptation. Open set domain adaptation (OSDA) [171] assumes the target domain possesses private classes that are unknown to the source domain. Analogously, partial domain adaptation (PDA) [18] describes a setting where only the

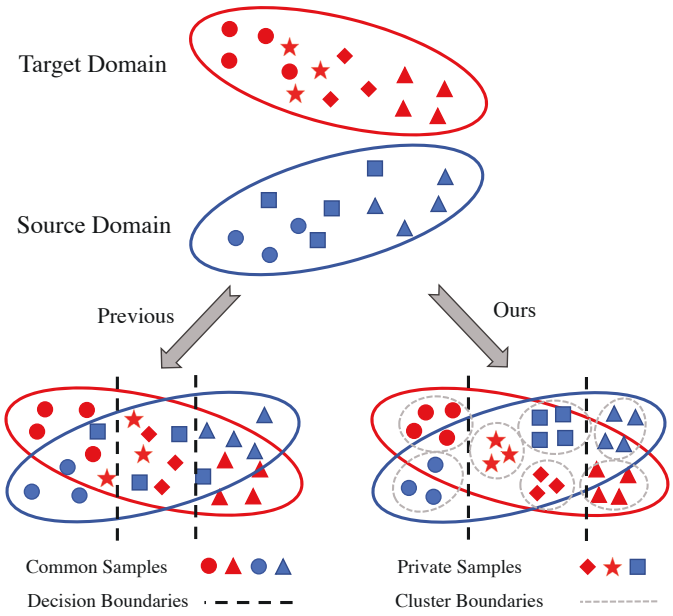


Figure 6.1 : A comparison between previous methods and ours. Previous methods simply treat private samples as one general class and ignore its intrinsic data structure. Our approach aims to better exploit the diverse distribution of private samples via forming discriminative clusters on both common samples and private samples.

source domain holds private classes. However, both OSDA and PDA still require prior knowledge where the private classes lie in. As a result, they are limited to one scenario and fail to generalize to other scenarios. For example, an OSDA solution would fail in the PDA scenario as it only seeks private samples in the target domain. To solve this, [213] takes a step further to propose a more general yet practical setting, universal domain adaptation (UniDA), which allows both domains to own private classes.

The main challenge of transferring over unaligned label space is how to effectively separate common samples from private samples in both domains. To achieve this goal, many efforts have been devoted to performing common sample discovery from different perspectives, such as designing new criteria [18, 213, 56, 169] or introducing extra discriminator [222, 19, 125, 32]. However, previous practices mainly

focus on identifying common samples but treat private samples as a whole, *i.e.*, *unknown* class (Bottom left in Fig. 6.1). Despite making progress, the *intrinsic* structure (*i.e.*, the variations within each semantic class and the relationships between different semantic classes) of the private samples is not fully exploited. As the private samples in nature belong to distinct semantic classes, treating them as one general class is arguably sub-optimal, which further induces lower compactness and less discriminative target representations.

In this chapter, we aim to better exploit the intrinsic structure of the target domain via mining both common classes and individual private classes. We propose Domain Consensus Clustering (DCC), which utilizes the domain consensus knowledge to form discriminative clusters on both common samples and private samples (Bottom right in Fig. 6.1). Specifically, we mine the domain consensus knowledge from two aspects, *i.e.*, semantic-level and sample-level, and integrate them into two consecutive steps. Firstly, we leverage Cycle-Consistent Matching (CCM) to mine the semantic consensus among cluster centers so that we could identify common clusters from both domains. If two cluster centers reach *consensus*, *i.e.*, both centers act as the other's nearest center simultaneously, this pair will be regarded as common clusters. Secondly, we propose a metric, domain consensus score, to acquire cross-domain classification agreements between identified common clusters. Concretely, domain consensus score is defined as the proportion of samples that hold corresponding cluster label across domains. Intuitively, more samples reach consensus, the distribution shift between matched clusters is smaller. Therefore, domain consensus score could be regarded as a constraint that ensures the precise matching of CCM. Moreover, domain consensus score also offers a necessary guidance that determines the number of target clusters, and encourages the samples to be grouped into clusters of both common and private classes. Finally, for those common clusters with high domain consensus scores, we exploit a class-aware align-

ment technique on them to mitigate the distribution shift. As for those centers that fail to find their *consensus* counterparts, we also enhance their cluster-based consistency. To be specific, we employ a prototypical regularizer to encourage samples to approach their attached cluster centers. In this way, those samples belonging to different private categories will be encouraged to be distinguishable from each other, which also contributes to learning better representations.

Our contribution can be summarized as: 1) We tackle the UniDA problem from a new perspective, *i.e.*, differentiating private samples into different clusters instead of treating them as whole. 2) We propose Domain Consensus Clustering (DCC), which mines domain consensus knowledge from two levels, *i.e.*, semantic-level and sample-level, and guides the target clustering in the absence of prior knowledge. 3) Extensive experiments on four benchmarks verify the superior performance of proposed method compared with previous works.

6.2 The Proposed Approach

In universal domain adaptation, we are provided with annotated source samples $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{n^s}$, and unlabeled target samples $\mathcal{D}^t = \{(\boldsymbol{x}_i^t)\}_{i=1}^{n^t}$. Since the label set may not be identical, we use C^s, C^t to represent label set for two domains accordingly. Then we denote $C = C^s \cap C^t$ as the common label set. We aim to train a model on $\mathcal{D}^s$ and $\mathcal{D}^t$ to classify target samples into $|C| + 1$ classes, where private samples are grouped into one *unknown* class.

The model consists of two modules: (1) feature extractor f_ϕ that maps the input images into vector representation: $v = f_\phi(\boldsymbol{x})$, and (2) classifier g_ϕ that assigns each feature representation v into one of C^s classes: $p = g_\phi(v)$. For samples from two domains, we group them into clusters, respectively. The cluster assignment of source samples is based on the ground truth and the source center is the mean embedding of source samples within one specific class. For the c-th source cluster $\mathcal{D}_c^s = \{\boldsymbol{x}_i^s\}_{i=1}^{n_c^s}$,

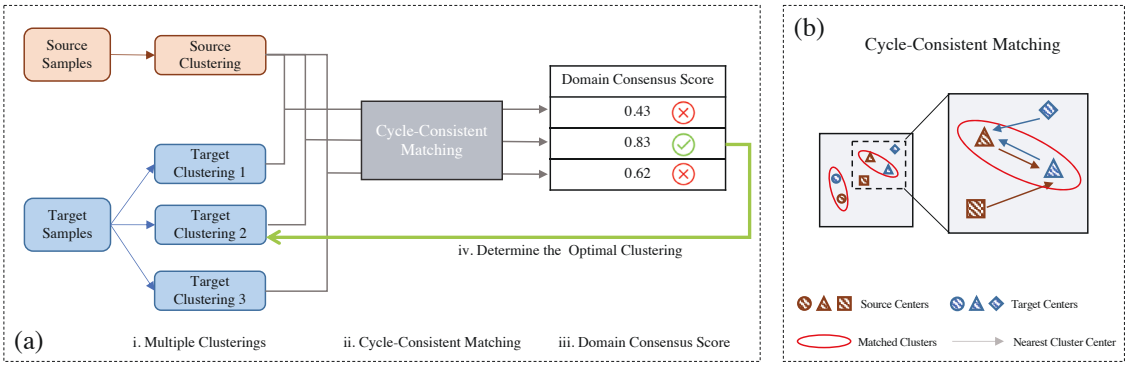


Figure 6.2 : **(a)** Illustration of Domain Consensus Clustering (DCC). i) As the number of target classes is not given, we aim to select the optimal target clustering from multiple candidates. ii) For obtained target clusters, we leverage cycle-consistent matching (CCM) to identify clusters representing common classes from both domains. iii) Then we utilize domain consensus score to estimate the degree of agreement between matched clusters. iv) Finally, based on the domain consensus score, we could determine the optimal target clustering. **(b)** Illustration of Cycle-Consistent Matching. If two clusters from different domains act as the other's nearest neighbor, samples from the two clusters are identified as common samples that share the same semantic labels.

its cluster center is:

$$\mu_c^s = \frac{1}{n_c^s} \sum_{\boldsymbol{x}_i^s \in \mathcal{D}_c^s} \frac{f_\phi(\boldsymbol{x}_i^s)}{||f_\phi(\boldsymbol{x}_i^s)||}. \tag{6.1}$$

As for target samples, we adopt K-means to group them into K clusters and obtain corresponding centers $\{\mu_1^t, ..., \mu_K^t\}$.

In this chapter, we aim to utilize the domain consensus knowledge to guide the target clustering, which exploits the intrinsic structure of the target representations. Specifically, we mine the domain consensus knowledge from two levels. Firstly, the semantic-level consensus among cluster centers is utilized to identify cycle-consistent clusters as common classes (§ 6.2.1). Secondly, we design a novel metric named

"domain consensus score" to utilize the sample-level consensus to specify the number of target clusters (§ 6.2.2). Finally, we discuss the cluster optimization and objectives in § 6.2.3. The overview of our approach is presented in Fig. 6.2.

6.2.1 Cycle-Consistent Matching

The main challenge of universal domain adaptation is how to separate common samples from private samples. Unlike previous works [213, 56] that perform sample-level identification on the common samples, this chapter aims to mine both common classes and individual private classes simultaneously with discriminative clusters. Now a question naturally arises: *how to associate common clusters that represent the same semantic classes from both domains?* To achieve this, we propose Cycle-Consistent Matching (CCM) to link clusters from the same common classes through mining semantic-level consensus.

As illustrated in Fig. 6.2 (b), for each cluster center, we search for its nearest cluster center in the other domain. If two clusters reach *consensus*, *i.e.*, both act as the other's nearest center simultaneously, such a pair of clusters is recognized as common clusters. The intuition here is simple: cluster centers from the same class usually lie close enough to be associated compared to the clusters representing private classes. Further, to ensure this assumption, we utilize the sample-level consensus to promote the effectiveness of CCM, which is detailed in the next section.

6.2.2 Domain Consensus Score

Enabled by CCM, we could identify common samples from both domains. Nevertheless, another problem is not yet solved: *how to determine the number of target clusters without knowing the exact number of underlying target classes?* To solve this, one plausible solution is to adopt existing clustering evaluation criteria [167, 42, 16] to estimate the number of clusters. However, these techniques

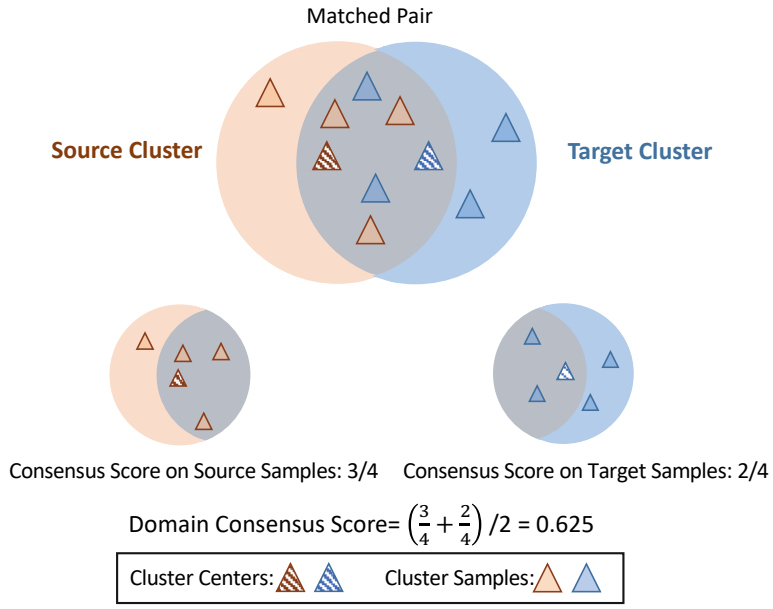


Figure 6.3 : Illustration of Domain Consensus Score. For each sample from matched clusters, we search for its nearest cluster center in the other domain. Then domain consensus score is calculated as the proportion of samples that reach consensus, *i.e.*, the labels of their nearest cluster centers in the other domain match with those achieved by CCM.

are designed for the single-domain scenario and cannot directly take cross-domain knowledge into consideration. Hence, we propose a metric, domain consensus score, which utilizes the sample-level consensus to determine the number of target clusters, thus forming discriminative clusters.

As shown in Fig. 6.3, for each sample from paired clusters, we search for its nearest cluster in the other domain, and then determine if it reaches *consensus*, *i.e.*, this sample holds corresponding cluster label across domains. Through collecting samples that reach consensus, the agreement for this pair of clusters can be evaluated.

Concretely, given a pair of matched clusters $\{v_i^s\}_{i=1}^m$ and $\{v_i^t\}_{i=1}^n$ with correspond-

ing centers μ_c^s and μ_k^t, we aim to measure the sample-level consensus from two views, *i.e.*, the source view and the target one. To obtain consensus score on source view, for each source sample, we calculate its similarities with all target cluster centers $\{\mu_1^t, ..., \mu_K^t\}$:

$$r_{i,k}^s = Sim(v_i^s, \mu_k^t), k \in \{1, ..., K\}, \tag{6.2}$$

where $Sim(\cdot)$ denotes the cosine similarity, *i.e.*, $Sim(\boldsymbol{a}, \boldsymbol{b}) = \frac{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}$. Then the consensus score could be formulated as the proportion of samples that reach consensus:

$$\mathcal{S}_{(c,k)}^s = \frac{\sum_{i=1}^m \mathbb{1}\{\arg\max_k(r_{i,k}^s) = k\}}{m}, \tag{6.3}$$

where $\mathbb{1}\{\arg\max_k(r_{i,k}^s) = k\}$ is a indicator to judge if v_i^s holds corresponding cluster index (k) across domains.

Analogously, we could obtain the consensus score on target samples $\mathcal{S}_{(c,k)}^t$. Then we average the score of two views to obtain the consensus score of this matched pair, *i.e.*, $\mathcal{S}_{(c,k)} = \frac{\mathcal{S}_{(c,k)}^s + \mathcal{S}_{(c,k)}^t}{2}$. Finally, we calculate the mean of consensus scores of all matched pairs of clusters.

To specify the number of target clusters K, we perform multiple clusterings with different K and then we determine the optimal one according to the domain consensus score. Concretely, for different instantiations of K which are equally spaced, we compute the consensus score for each one, and the instantiation of K with the highest score is chosen for subsequent clustering.

Empirically, we find DCC tends to separate samples from one class into multiple clusters at the beginning, which is also known as over-clustering. The reason is that to achieve a higher consensus score, more accurate matching between clusters is preferred. Consequently, at the beginning, DCC prefers small clusters with "easy" samples (*i.e.*less impacted by the domain shift), which may make the number of clusters larger than the underlying number of target classes. As the adaptation

proceeds, the number of clusters tends to decrease and converges to a certain number after a period of training.

6.2.3 Cluster Optimization and Objectives

In this section, we first introduce the clustering update strategy. Then we enumerate objectives, *i.e.*, prototypical regularizer, contrastive domain discrepancy. Finally, we present the overall objective and the weight of each item.

Alternate Update. To avoid the accumulation of inaccurate labels, we optimize the model and update the clustering alternatively. Ideally, we expect to specify the number of clusters K with only one search, but it is impossible due to the large domain gap at the initial stage. Hence, DCC specifies the K based on domain consensus score for each update of the clustering. Empirically we find that: 1) in each round of searching, the domain consensus scores exhibit a bell curve as K increases. 2) K converges to a specific value after several initial rounds of searching, *i.e.*after early stages of training. Motivated by these observations, we adopt two stopping criteria to improve the efficiency of searching, *i.e.*, stop the searching once the consensus score drops a certain number of times continuously, and fix the K once it holds a certain value for a certain number of rounds.

Prototypical Regularizer. To enhance the discriminability of target clusters, we impose a prototypical regularizer on target samples. Specifically, let $M = [\mu_1^t, \mu_2^t, ..., \mu_K^t]$ denotes the prototype bank that stores all L2-normalized target cluster centers and these prototypes is updated iteratively during training. Then the regularizer could be formulated as:

$$\mathcal{L}_{reg} = -\sum_{i=1}^{n_t}\sum_{k=1}^{K} \hat{y}_{i,k}^t \log \hat{p}_{(i,k)}, \tag{6.4}$$

where $\hat{y}_i^t$ is the one-hot cluster label, and

$$\hat{p}_{(i,k)} = \frac{exp({v_i}^T \mu_k^t/\tau)}{\sum_{k=1}^{K} exp({v_i}^T \mu_k^t/\tau)}. \tag{6.5}$$

Here v_i is L2-normalized feature vector of target samples. τ is a temperature parameter that controls the concentration of the distribution [71], and we set it to 0.1 empirically.

Contrastive Domain Discrepancy (CDD). Since the identified common samples are grouped into clusters, we leverage CDD [88, 87] to facilitate the alignment over identified common samples in a class-aware style. We impose $\mathcal{L}_{cdd}$ to minimize the intra-class discrepancies and enlarge the inter-class gap. Consequently, the enhanced discriminability, in turn, enables DCC to perform more accurate clustering.

Overall Objective. The model is jointly optimized with three terms, *i.e.*, cross-entropy loss on source samples $\mathcal{L}_{ce}$, domain alignment loss $\mathcal{L}_{cdd}$, and the regularizer $\mathcal{L}_{reg}$:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{cdd} + \gamma \mathcal{L}_{reg}, \tag{6.6}$$

$$\mathcal{L}_{ce} = -\sum_{i=1}^{n_s} \sum_{c=1}^{|C_s|} \hat{y}_{i,c}^{s} \log(\sigma(g_\phi(f_\phi(\boldsymbol{x}_i^s)))), \tag{6.7}$$

where σ denotes the softmax function, and $\hat{y}_i^s$ is the one-hot encoding of source label. λ is set to 0.1 for all datasets.

As mentioned, the target clustering usually converges to the optimal one after several rounds of searching, so simply applying a constant weight on $\mathcal{L}_{reg}$ may hinder the convergence as it promotes the inter-cluster separation. Therefore, we apply a ramp-up function on γ, *i.e.*, $\gamma = e^{-\omega \times \frac{i}{N}}$, where i and N denote current and global iteration, and $\omega = 3.0$. Such an incremental weight allows the size of clusters to grow in the earlier stage while preventing them from absorbing extra private samples after getting saturated. As two tunable parameters (*i.e.*, τ and γ), we choose to fix the temperature but change the weights during training. The rationale stems from the fact that while the weight accounts for the global distributions, the temperature solely adjusts the concentration of distributions within a small batch. Tuning the

Table 6.1 : Results (%) on **Office-31** for UniDA (ResNet-50).

UniDA	A→W		D→W		W→D		A→D		D→A		W→A		Avg	
	Acc.	HM	Acc.	HM	Acc.	HM	Acc.	HM	Acc.	HM	Acc.	HM	Acc.	HM
DANN [58]	80.65	48.82	80.94	52.73	88.07	54.87	82.67	50.18	74.82	47.69	83.54	49.33	81.78	50.60
RTN [128]	85.70	50.21	87.80	54.68	88.91	55.24	82.69	50.18	74.64	47.65	83.26	49.28	83.83	51.21
IWAN [222]	85.25	50.13	90.09	54.06	90.00	55.44	84.27	50.64	84.22	49.65	86.25	49.79	86.68	51.62
PADA [18]	85.37	49.65	79.26	52.62	90.91	55.60	81.68	50.00	55.32	42.87	82.61	49.17	79.19	49.98
ATI [148]	79.38	48.58	92.60	55.01	90.08	55.45	84.40	50.48	78.85	48.48	81.57	48.98	84.48	51.16
OSBP [171]	66.13	50.23	73.57	55.53	85.62	57.20	72.92	51.14	47.35	49.75	60.48	50.16	67.68	52.34
UAN [213]	85.62	58.61	94.77	70.62	97.99	71.42	86.50	59.68	85.45	60.11	85.12	60.34	89.24	63.46
CMU [56]	86.86	67.33	**95.72**	**79.32**	**98.01**	80.42	89.11	68.11	88.35	**71.42**	88.61	72.23	91.11	73.14
Ours	**91.66**	**78.54**	94.52	79.29	96.20	**88.58**	**93.70**	**88.50**	**90.43**	70.18	**91.97**	**75.87**	**93.08**	**80.16**

Table 6.2 : HM (%) on **Office-Home** for UniDA (ResNet-50).

UniDA	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
DANN [58]	42.36	48.02	48.87	45.48	46.47	48.37	45.75	42.55	48.70	47.61	42.67	47.40	46.19
RTN [128]	38.41	44.65	45.70	42.64	44.06	45.48	42.56	36.79	45.50	44.56	39.79	44.53	42.89
IWAN [222]	40.54	46.96	47.78	44.97	45.06	47.59	45.81	41.43	47.55	46.29	42.49	46.54	45.25
PADA [18]	34.13	41.89	44.08	40.56	41.52	43.96	37.04	32.64	44.17	43.06	35.84	43.35	40.19
ATI [148]	39.88	45.77	46.63	44.13	44.39	46.63	44.73	41.20	46.59	45.05	41.78	45.45	44.35
OSBP [171]	39.59	45.09	46.17	45.70	45.24	46.75	45.26	40.54	45.75	45.08	41.64	46.90	44.48
UAN [213]	51.64	51.70	54.30	61.74	57.63	61.86	50.38	47.62	61.46	62.87	52.61	65.19	56.58
CMU [56]	56.02	**56.93**	**59.15**	66.95	64.27	67.82	54.72	51.09	66.39	68.24	57.89	69.73	61.60
Ours	**57.97**	54.05	58.01	**74.64**	**70.62**	**77.52**	**64.34**	**73.60**	**74.94**	**80.96**	**75.12**	**80.38**	**70.18**

latter can be more challenging and may lead to unstable training.

Inference. At the inference stage, we do not perform any clustering. With the prototypes $M = [\mu_1^t, \mu_2^t, ..., \mu_K^t]$, we can assign each sample a label the same as the nearest prototype. In this way, common samples can be naturally separated from private ones in the target domain.

Table 6.3 : The division on label set, *i.e.*, Common Class (C) / Source-Private Class ($\hat{C}_s$) / Target Private Class ($\hat{C}_t$).

Dataset	Class Split ($\|C\|/\|\hat{C}_s\|/\|\hat{C}_t\|$)		
	PDA	OSDA	UniDA
Office-31	10 / 21 / 0	10 / 0 / 11	10 / 10 / 11
OfficeHome	25 / 40 / 0	25 / 0 / 40	10 / 5 / 50
VisDA	–	6 / 0 / 6	6 / 3 / 3
DomainNet	–	–	150 / 50 / 145

6.3 Experiments

6.3.1 Experimental Setup

Besides the setting [213] where private classes exist in both domains (UniDA), we also validate our approach on other two sub-cases, *i.e.*partial domain adaptation (PDA) and open set domain adaptation (OSDA).

Dataset. We conduct experiments on four datasets. **Office-31** [168] consists of 4652 images from three domains: DSLR (**D**), Amazon (**A**), and Webcam (**W**). **Office-Home** [190] is a more challenging dataset, which consists of 15500 images from 65 categories. It is made up of 4 domains: Artistic images (**Ar**), Clip-Art images (**Cl**), Product images (**Pr**), and Real-World images (**Rw**). **VisDA** [152], is a large-scale dataset, where the source domain contains 15K synthetic images and the target domain consists of 5K images from the real world. **DomainNet** [151] is the largest domain adaptation dataset with about 0.6 million images. Like [56], we conduct experiments on three subsets from it, *i.e.*, Painting (**P**), Real (**R**), and Sketch (**S**).

Following existing works [148, 171, 18, 213], we separate the label set into three parts: common classes C, source-private classes $\hat{C}_s$ and target-private classes $\hat{C}_t$.

The separation of four datasets is described in Table 6.3. The classes are separated according to their alphabetical order.

Evaluation. For all experiments, we report the averaged results of three runs. In OSDA and UniDA, target-private classes are grouped into a single *unknown* class, and we report two metrics, *i.e.*, **Acc.** and **HM**, where the former is the mean of per-class accuracy over common classes and *unknown* class, and the latter is the harmonic mean on accuracy of common samples and private ones like [56, 11]. In VisDA under OSDA, we present **OS** and **OS*** results as previous works [171, 125], where **OS** is same as **Acc.** and **OS*** only calculates the mean accuracy on common classes. In PDA, we report the mean of per-class accuracy over common classes.

Implementation details. Our implementation is based on PyTorch. We start from ResNet-50 [69] with the backbone pretrained on ImageNet [43]. The classifier consists of two fully-connected layers, which follows the previous design [213, 56, 171, 18]. For a fair comparison, we adopt VGGNet [178] as backbone for OSDA task on VisDA.

We optimize the model using Nesterov momentum SGD with momentum of 0.9 and weight decay of 5×10^{-4}. The learning rate decays with the factor of $(1+\alpha\frac{i}{N})^{-\beta}$, where i and N denote current iteration and global iteration, and we set $\alpha = 10$ and $\beta = 0.75$. The batch size is set to 36. The initial learning rate is set to 1×10^{-4} for Office-31 and VisDA, and 1×10^{-3} for Office-Home and DomainNet.

6.3.2 Comparison with Previous Methods

We compare our method with previous state-of-the-arts in three sub-cases of universal domain adaptation, *i.e.*, OSDA, PDA, and UniDA. For OSDA and PDA, we compare our method to the universal domain adaptation methods, without knowing the prior that private classes exist only in source domain (*i.e.*PDA) or only in target domain (*i.e.*OSDA). Also, we compare our method to the baselines tailed for OSDA

Table 6.4 : Results (%) on **VisDA** for OSDA (VGGNet) and UniDA (ResNet-50). *: variants of OSVM using MMD and DANN.

Method	OSDA		Method	UniDA	
	OS	OS*		Acc.	HM
OSVM [81]	52.5	54.9	RTN [128]	53.92	26.02
MMD+OSVM*	54.4	56.0	IWAN [222]	58.72	27.64
DANN+OSVM*	55.5	57.8	ATI [148]	54.81	26.34
ATI-λ [148]	59.9	59.0	OSBP [171]	30.26	27.31
OSBP [171]	62.9	59.2	UAN [213]	60.83	30.47
STA [125]	66.8	63.9	USFDA [92]	63.92	–
Inheritune [92]	68.1	64.7	CMU [56]	61.42	34.64
Ours	**68.8**	**68.0**	*Ours*	**64.20**	**43.02**

and PDA settings, by taking the prior of each setting into consideration.

UniDA Setting. In the most challenging setting, *i.e.*UniDA, our approach achieves new state-of-the-arts. Table 6.1 shows the results on Office-31. The proposed method surpasses all compared methods in terms of both accuracy and HM. Especially, with respect to HM, our method outperforms previous state-of-art method CMU [56] by 7%, which shows our method strikes a better balance between the identification of common and private samples. Office-Home (Table 6.2) is a more challenging dataset where the number of private classes is much more than common classes (55 vs. 10). Under this extreme scenario, our method demonstrates a stronger capability on the common-private separation (9% improvement in terms of HM), which benefits from the higher compactness of private samples. We also test DCC on VisDA and present the results in Table 6.4. Notably, with higher accuracy, our method shows +9% improvement compared to CMU [56] in terms of HM, im-

Table 6.5 : HM (%) on **Office** and **Office-Home** under the OSDA scenario (ResNet-50). The reported numbers for previous OSDA methods are cited from [11]. We use 'U' and 'O' to denote methods designed for UniDA setting and OSDA setting, respectively.

Method	Type	Office							Office-Home												
		A2W	A2D	D2W	W2D	D2A	W2A	Avg	Ar2Cl	Ar2Pr	Ar2Rw	Cl2Ar	Cl2Pr	Cl2Rw	Pr2Ar	Pr2Cl	Pr2Rw	Rw2Ar	Rw2Cl	Rw2Pr	Avg
STA_{max} [125]	O	75.9	75.0	69.8	75.2	73.2	66.1	72.5	55.8	54.0	68.3	57.4	60.4	66.8	61.9	53.2	69.5	67.1	54.5	64.5	61.1
OSBP [171]	O	82.7	82.4	**97.2**	91.1	75.1	73.7	83.7	55.1	65.2	72.9	**64.3**	64.7	**70.6**	**63.2**	53.2	73.9	66.7	54.5	72.3	64.7
ROS [11]	O	82.1	82.4	96.0	**99.7**	77.9	77.2	85.9	**60.1**	**69.3**	76.5	58.9	65.2	68.6	60.6	56.3	**74.4**	**68.8**	60.4	**75.7**	**66.2**
Ours	O	**87.1**	**85.5**	91.2	87.1	**85.5**	**84.4**	**86.8**	52.9	67.4	**80.6**	49.8	**66.6**	67.0	59.5	52.8	64.0	56.0	**76.9**	62.7	64.2
UAN [213]	U	46.8	38.9	68.8	53.0	**68.0**	54.9	55.1	0.0	0.0	0.2	0.0	0.2	0.2	0.0	0.0	0.2	0.2	0.0	0.1	0.1
Ours	U	**54.8**	**58.3**	**89.4**	**80.9**	67.2	**85.3**	**72.6**	**56.1**	**67.5**	**66.7**	**49.6**	**66.5**	**64.0**	**55.8**	**53.0**	**70.5**	**61.6**	**57.2**	**71.9**	**61.7**

plying a higher capacity on identifying private samples. In Table 6.7, we present the results on a large scale dataset DomainNet. DCC yields consistent improvement, verifying its effectiveness on large-scale dataset.

OSDA and PDA setting. In Table 6.5, Table 6.6, and Table 6.4, we present the results under PDA and OSDA scenarios. We use 'P' and 'O' to denote the methods specifically designed for PDA and OSDA accordingly, and use 'U' to denote the UniDA methods. As shown in the tables, our approach performs favorably against previous methods in different scenarios, *i.e.*with and without using the prior knowledge. Particularly, our method without using the prior ('U') yields even better result compared to competitive methods tailed for the PDA setting. For example, on Office-Home, our method ('U') achieves 70.9% average accuracy, which outperforms PADA [18] (62.1%) and ETN [19] (70.5%), demonstrating that our method can effectively separate common samples from private ones.

Table 6.6 : Accuracy (%) on **Office** and **Office-Home** under the PDA scenario (ResNet-50). We use 'U' and 'P' to denote methods designed for UniDA setting and PDA setting, respectively.

Method	Type	Office							Office-Home												
		A2W	A2D	D2W	W2D	D2A	W2A	Avg	Ar2Cl	Ar2Pr	Ar2Rw	Cl2Ar	Cl2Pr	Cl2Rw	Pr2Ar	Pr2Cl	Pr2Rw	Rw2Ar	Rw2Cl	Rw2Pr	Avg
IWAN [222]	P	90.5	89.2	95.6	99.3	94.3	99.4	94.7	53.9	54.5	78.1	61.3	48.0	63.3	54.2	52.0	81.3	76.5	56.8	82.9	63.6
PADA [18]	P	82.2	86.5	92.7	99.3	95.4	**100.0**	92.7	52.0	67.00	78.7	52.2	53.8	59.1	52.6	43.2	78.8	73.7	56.6	77.1	62.1
ETN [19]	P	94.5	95.0	100.0	100.0	**96.2**	94.6	96.7	**59.2**	77.0	79.5	62.9	65.7	75.0	68.3	**55.4**	84.4	75.7	57.7	84.5	70.5
RTNet [32]	P	96.2	**97.6**	100.0	100.0	92.3	95.4	96.9	63.2	80.1	80.7	66.7	69.3	77.2	71.6	53.9	**84.6**	**77.4**	57.9	85.5	72.3
Ours	P	**99.7**	96.1	**100.0**	**100.0**	95.3	96.3	**97.9**	59.0	**84.4**	**83.4**	**67.8**	**72.7**	**79.8**	**68.4**	53.2	83.7	75.8	**59.0**	**88.3**	**73.0**
UAN [213]	U	76.8	79.7	93.4	98.3	82.7	83.7	85.8	24.5	35.0	41.5	34.7	32.3	32.7	32.7	21.1	43.0	39.7	26.6	46.0	34.2
Ours	U	**97.6**	**87.3**	**100.0**	**100.0**	**96.6**	**96.3**	**96.3s**	**54.2**	**47.5**	**57.5**	**83.8**	**71.6**	**86.2**	**63.7**	**65.0**	**75.2**	**85.5**	**78.2**	**82.6**	**70.9**

6.3.3 Ablation Studies

Effect of Cycle-Consistence Matching (CCM). To show how CCM can effectively identify common classes, we vary the number of common classes ($|C|$) and observe the identified common classes under different values of K. In Fig. 6.4 (a), as K increases, the number of matched clusters tends to converge to a value which is quite near to $|C|$.

Effect of Domain Consensus Score. To better understand domain consensus score, we conduct a series of experiments to reveal its mechanism.

First, we decompose the domain consensus score into two parts, *i.e.*, $\mathcal{S}^s$ and $\mathcal{S}^t$; $\mathcal{S}^s$ is the consensus score of source samples while the other denotes score of target samples. As shown in Fig. 6.4 (b), as the K increases, $\mathcal{S}^s$ and $\mathcal{S}^t$ show opposite trend, *i.e.*, $\mathcal{S}^s$ increases but $\mathcal{S}^t$ decreases. As the K increases, target samples are divided into more and smaller target clusters. Therefore, smaller target samples could better match source clusters, which causes the increase of $\mathcal{S}^t$. On the other hand, as more target clusters form, source clusters are more easily distracted by nearby target clusters, which explains the drop of $\mathcal{S}^s$.

Table 6.7 : HM (%) on **DomainNet** for UniDA (ResNet-50).

Method	P→R	R→P	P→S	S→P	R→S	S→R	Avg
DANN [58]	31.18	29.33	27.84	27.84	27.77	30.84	29.13
RTN [128]	32.27	30.29	28.71	28.71	28.63	31.90	30.08
IWAN [222]	35.38	33.02	31.15	31.15	31.06	34.94	32.78
PADA [18]	28.92	27.32	26.03	26.03	25.97	28.62	27.15
ATI [148]	32.59	30.57	28.96	28.96	28.89	32.21	30.36
OSBP [171]	33.60	33.03	30.55	30.53	30.61	33.65	32.00
UAN [213]	41.85	43.59	39.06	38.95	38.73	43.69	40.98
CMU [56]	50.78	**52.16**	**45.12**	44.82	**45.64**	50.97	48.25
Ours	**56.90**	50.25	43.66	**44.92**	43.31	**56.15**	**49.20**

Second, in Fig. 6.4 (c), we visualize the evolution of domain consensus score as training progresses. As expected, the domain consensus score saturates after the early rounds, which indicates that our method can find the optimal number of clusters quickly. Moreover, this also implies that the searching is only necessary at the early stage.

Third, we compare domain consensus score with previous general metrics to determine the number of clusters (*i.e.*, calinski harabasz score [16], davies bouldin score [42], and silhouette score [167]), and present the results in Fig. 6.5. Our metric obviously outperforms previous ones, proving the benefits of taking the distribution shift into account.

Effect of Domain Consensus Clustering. Fig. 6.4 (d) shows the evolution of K during training under three scenarios. In theses experiments, we do not employ the proposed stop criteria (see Section 6.2.3). As shown in these figures, the number of clusters converges to the optimal value after several initial searches, which is consistent with the convergence of consensus score (Fig. 6.4 (c)). This indicates that the searching of K is only necessary in the early stage of training, which justifies

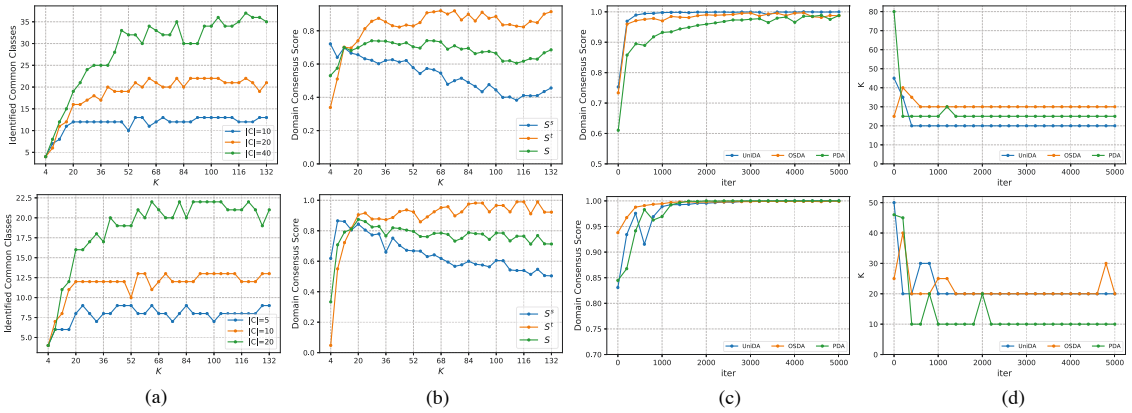


Figure 6.4 : Ablation Analysis (Best viewed in color). (a) Number of identified common classes w.r.t. K under varying $|C|$. (b) Decomposed consensus score w.r.t. K. (c) The evolution of consensus score as training progresses. (d) The evolution of K as training progresses. The first row is extracted from **Ar→Rw** of Office-Home and the second row is from **A→D** of Office-31.

Table 6.8 : Effect of $\mathcal{L}_{cdd}$ and $\mathcal{L}_{reg}$.

	$\mathcal{L}_{cdd} + \mathcal{L}_{reg}$	$\mathcal{L}_{cdd}$	$\mathcal{L}_{reg}$
Office-Home	**70.18**	61.48	62.85
DomainNet (P→R)	**56.90**	54.16	53.55

the proposed stop criteria.

Effect of $\mathcal{L}_{cdd}$ and $\mathcal{L}_{reg}$. To evaluate the contribution of $\mathcal{L}_{cdd}$ and $\mathcal{L}_{reg}$, we train the model with each component solely and present the results in Table 6.8, which verifies the contribution of each term.

Sensitivity to Hyper-parameters. To show the sensitivity of our method to the hyper-parameter λ, we conduct experiments on Office-31 under UniDA setting, and present the results in Fig. 6.6 (a). Within a wide range of λ (0.1-0.3), the performance only varies in a small range, showing that our method is robust to different

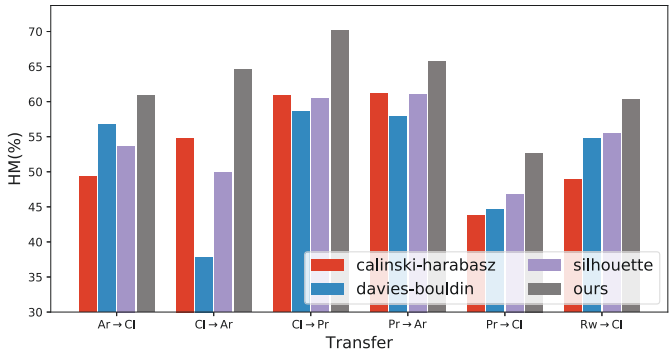


Figure 6.5 : Performance comparison on cluster evaluation metrics.

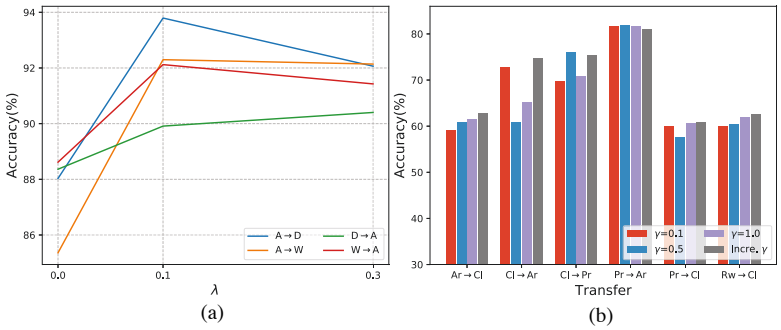


Figure 6.6 : (a) Sensitivity to λ on Office31. (b) Comparison between constant γ (*i.e.*, 0.1, 0.5, 1.0) and dynamic γ ('Incre. γ') (Office-Home). All experiments are conducted under UniDA setting.

choices of λ. Also, we compare our way of progressively increasing γ (denoted as 'Incre. γ') with using various constant values on Office-Home under UniDA setting. As shown in Fig. 6.6 (b), 'Incre. γ' achieves better results for most of the tasks, which verifies the effectiveness of this design.

6.4 Conclusion

In this chapter, we propose Domain Consensus Clustering (DCC), which performs adaptation over unaligned label space via encouraging discriminative target clusters. To be specific, DCC exploits domain consensus knowledge from two lev-

els, *i.e.*, semantic-level and sample-level, to identify private samples and guide the target clustering. Experiments on four benchmarks show superior performance of proposed methods, compared to previous state-of-the-arts.

Chapter 7

Decouple to Contrast: Orthogonalized Ambiguity Reduction for Open-Vocabulary Object Detection

Building on the innovative approaches of Chapter 6, Chapter 7 takes a step further to explore the potential of visual-text correspondence. By leveraging this correspondence, this chapter aims to identify and classify novel categories, thus broadening the spectrum of our understanding of category generalization.

7.1 Introduction

Object detection [161, 20, 240, 185], as a fundamental vision task, aims to localize objects in images and classify their semantic classes. Despite significant progress, most detectors are still confined to the closed-set training vocabularies, *e.g.*, 80 classes for COCO [121] and 1203 for LVIS [67] , which are insufficient for real-world application. Enriching the detection vocabulary with manual annotation is notoriously expensive. Therefore, Open-Vocabulary Detection (OV-Det) [218, 47, 159] is proposed to expand the vocabulary autonomously, with the help of text descriptions.

The key point of OV-Det is to establish the correspondence between the text description and local image region that refer to the same object-of-interest. The recent success of the large Vision-Language Model (VLM) has brought great inspiration to OV-Det, because VLM shares a similar objective, *i.e.*, it learns visual-semantic correspondences between text description and holistic image. Therefore, many recent works [217, 237, 117, 93] leverage the pretrained visual-semantic correspondences in

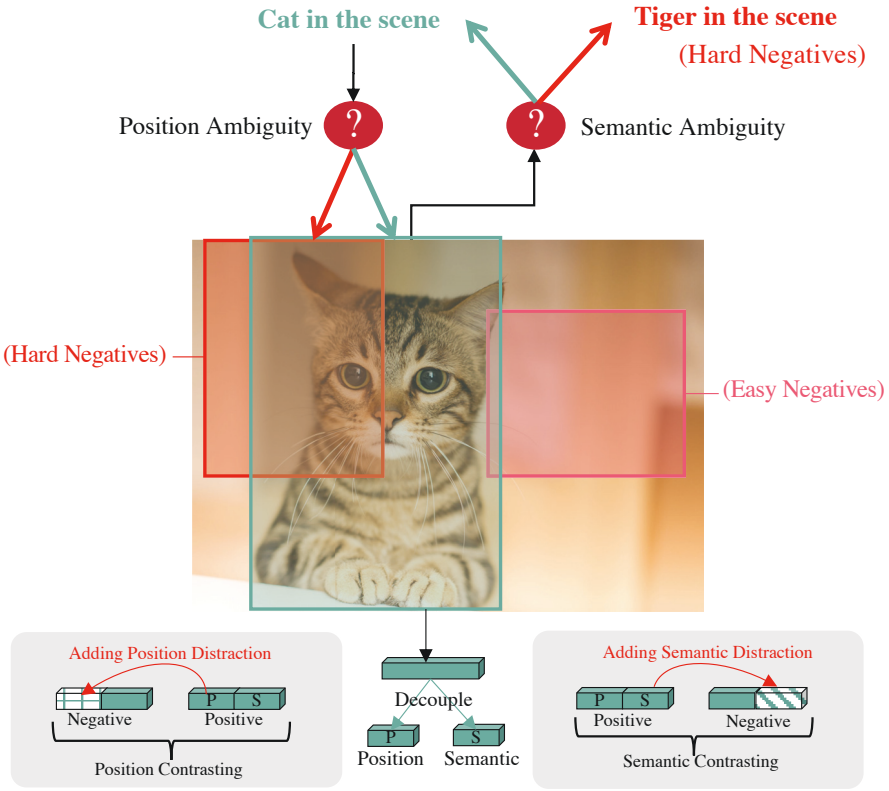


Figure 7.1 : We view the semantic and positional ambiguities in OV-Det from a unified contrastive learning viewpoint, *i.e.*, they both incur negative samples. We note that a negative sample with "semantic / positional" offset is more difficult than negative samples with "semantic + positional" offsets. Therefore, the proposed DeCo decouples them and contrasts a positive sample with the hard negative samples (semantic / positional distraction) orthogonally.

VLMs, *e.g.*, CLIP [156], for cross-modality association.

There are two well-recognized challenges for OV-Det to utilize the VLMs, *i.e.*, the semantic and localization ambiguities . 1) *Semantic Ambiguity:* Since the VLMs are employed in an off-the-shelf manner, the domain gap between pretraining and detection data makes their visual-semantic correspondences unfit for the object-of-interest and thus incurs semantic ambiguity. 2) *Localization Ambiguity:* Since the VLMs are trained with holistic image (rather than local patch) and usually apply

random-crop augmentation for position robustness, their visual-semantic correspondences are not sensitive to the positional offset. Previous literature can be divided into two streams *w.r.t.* these two challenges. One stream attempts to mitigate the semantic gap between proposals and text embeddings, through direct region-wise distillation [66], novel fusion strategy on prediction score [93], CLIP-conditioned proposal generation [217], or novel pseudo-labeling assignment [55]. The other stream seeks to mine and establish fine-grained region-text alignment via large-scale pre-training [136, 234, 211].

This chapter jointly considers these two challenges from a unified contrastive learning viewpoint and proposes a "Decouple-to-Contrast" (DeCo) paradigm. Fig. 7.1 illustrates our key idea: no matter which type of ambiguity happens, the detector may confuse a positive sample with a negative one. Therefore, contrasting these negative samples (blurred with either semantic or positional offset) against positive samples can help eliminate the ambiguities. Based on this unified contrastive learning viewpoint, another important insight is: although these two ambiguities usually occur simultaneously, decoupling them as two independent factors for negative samples is beneficial. It is because compared with the negative samples with "semantic + positional" offsets, a negative sample with only "semantic / positional" offset is more similar to the positive sample and is thus more difficult. Combining these two insights, we reach our idea of decoupling the semantic / positional ambiguity and eliminating them in the unified contrastive framework.

We implement DeCo under the popular DETR-style pipeline [20, 240]. We think the DETR-style pipeline is a good choice for our requirement of decoupling ambiguities, because recent DETR-style detectors [123, 221] explicitly disentangle their object queries into content (semantic) and positional embedding. More concretely, the object queries in DETR-style detectors may be viewed as initial proposals to be refined through the decoder. Given an initial object query, DeCo injects the decou-

pled semantic / positional distraction into the content / positional embedding in an orthogonal manner: injecting distraction to one embedding while maintaining the other embedding unchanged. Then DeCo trains the decoder to discriminate those hard negative queries through contrastive learning.

To be more specific, we introduce two parallel branches of contrastive learning. The first branch contrasts the semantic-negative queries with the positive query. Given an object query, we find some semantic neighbors from the text embedding space and then add these semantic neighbors to the content embedding of the query. The output states of these semantic negative queries are compared with the positive query through contrastive loss (InfoNCE [144], in particular). In analogy to the semantic branch, the second branch adds positional offsets onto the positional embedding and uses them as negative queries. During inference, DeCo removes these two branches and resumes the conventional DETR-style pipeline, thereby having a negligible increase in computational overhead. Extensive experiments and ablation analysis verify the effectiveness of the proposed method. For example, on COCO, DeCo achieves 32.3 AP on novel classes, setting the new state-of-the-art.

The contribution of this chapter can be summarized as:

- We propose to jointly mitigate the semantic and positional ambiguities for open-vocabulary detection (OV-Det) from a unified contrastive learning viewpoint.
- We decouple these two ambiguities into two independent factors for generating negative samples, because a negative sample with only "semantic / positional" offset is more difficult than negative samples with "semantic + positional" offsets.
- Extensive experiments are conducted on OV-Det benchmarks, and the results show that our method achieves the new state-of-the-art.

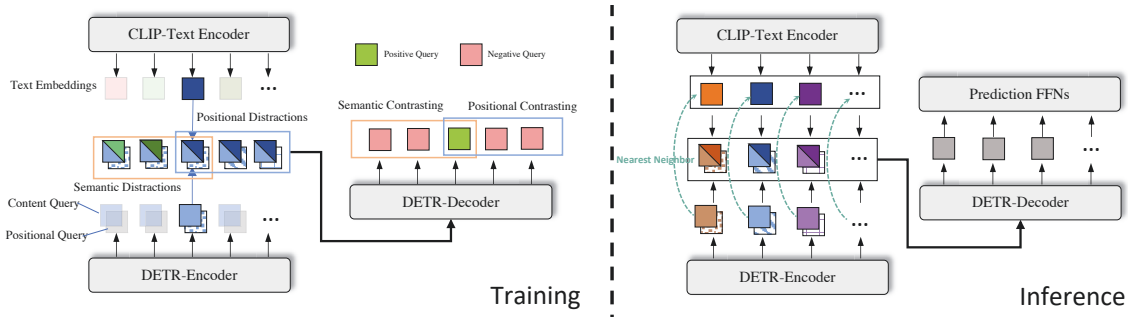


Figure 7.2 : Schematic of the proposed "Decouple to Contrast" (DeCo). During training (left), DeCo injects semantic / position distractions to the content query / positional query in an orthogonal manner, *i.e.*fixing the semantic part and disturbing the position part and vice versa. The detailed method for injecting the semantic and position distractions are illustrated in Section 7.2.3 (Fig. 7.3) and Section 7.2.4 (Fig. 7.4), respectively. After feeding them into the decoder, we impose contrastive learning schemes on them accordingly, thus mitigating these ambiguities in an orthogonalized manner. During inference (right), we remove these two contrasting branches. Given a query, we find its nearest CLIP text embedding, fuse its semantic part with the matched CLIP embedding, and then feed it into the decoder to make a prediction.

7.2 Methodology

7.2.1 Preliminaries

Problem Setup. Given an image $I \in \mathbb{R}^{H\times W\times 3}$, the detector is required to detect objects in it with their coordinates of bounding boxes and semantic classes. Conventional detection pipeline assumes the training set and the test set hold the identical label space, *i.e.*, $C_{train} = C_{test}$. Instead, Open-Vocabulary Detection (OV-Det) considers a more realistic scenario that the test set contains novel objects that do not occur during training, *i.e.*, $C_{train} = C_{base}$ but $C_{test} = C_{base} \bigcup C_{novel}$. The main objective for OV-Det is to generalize to both base classes and novel ones with

the aid of text descriptions. Following [234, 211], we use the text encoder inherited from CLIP [156] and encode class names with prompt templates to obtain the text embeddings for each class, which are denoted as $\{t_1, t_2, \cdots, t_c\}$.

General DETR pipeline. DETR [20] formulates object detection as a set predictions problem, which consists of a backbone, encoder-decoder transformer, and prediction Feed-forward Networks (FFNs). Given an image, DETR first employs a backbone and DETR-encoder to transform it into a sequence of feature tokens, *i.e.*, $\boldsymbol{X} = \{x_1, x_2, \cdots, x_d\}$. Second, DETR sends these feature tokens and a set of queries $\{q_i\}_{i=1}^{n}$ into the transformer decoder. Within the decoder, these queries interact with the feature tokens through cross-attention and gradually get refined. Third, DETR employs prediction FFNs to derive predictions $\boldsymbol{P} = \{p_1, p_2, \cdots, p_n\}$ with taking the refined queries as the input. Finally, DETR performs bipartite matching between predictions and the ground truth. The matching process associates predictions to the ground-truth objects by minimizing their matching cost, which considers both positional overlap and semantic consistency. After matching, DETR assigns corresponding labels to these predictions and imposes supervision accordingly.

As investigated in Conditional DETR and DAB-DETR, queries in DETR [20] consist of two elements, the positional embedding and content embedding, which are referred as positional query and content query in this chapter.

In the following sections, we first provide an overview of the proposed framework and explain the core motivation behind DeCo (Sec 7.2.2). Then we elaborate on how to reduce ambiguity regarding semantics and positions in Sec. 7.2.3 and Sec. 7.2.4, respectively. Finally, we detail the training objectives and inference process in Sec. 7.2.5.

7.2.2 Overview: Decouple to Contrast

As depicted in Fig. 7.2, DeCo first decouples each query into the content query and the positional query, and then reduces their ambiguities through contrastive learning in an orthogonal manner. The orthogonal manner, in our definition, is to fix one (out of two) query and to disturb the other one. Specifically, for the positional contrasting, DeCo introduces distractions onto the positional query while fixing its content part, and then leverages contrastive learning to reduce the position ambiguity. Similarly, for the semantic contrasting, DeCo introduces distractions onto the content query while maintaining its position, and then leverages another contrastive learning to eliminate the semantic ambiguity. The rationale here is: compared with samples that have two ambiguities ("semantic + positional"), samples with only one distraction (semantic or positional) is more challenging, *e.g.*, high objectiveness but ambiguous semantic meaning in the text latent space. According to the good practice in contrastive learning [164, 86], these hard negative samples are more informative for eliminating ambiguities. Therefore, we contrast the positive query with those hard negatives, and thus efficiently align the detector according to the text latent space.

7.2.3 Semantic Contrasting

The inherent gap between vision and language leads to semantic mismatches between query features and their corresponding text descriptions. Although CLIP has helped alleviate this gap to some extent, it still has a considerable gap with specialized detectors. In response to the semantic ambiguity, DeCo calibrates the semantic feature space between queries and CLIP in a contrastive learning manner.

Fig. 7.3 provides a detailed illustration of the semantic contrasting scheme. For each object query $q = [q^s, q^p]$ (q^s and q^p are the semantic and positional part accordingly), we compare its semantic part against all the CLIP text embeddings and

find its top-k nearest neighbors $\{t_i\}$ $(i = 1, 2, \cdots, k)$. In practice, this comparison is implemented through a linear classification head consisting of all the CLIP text embeddings (as in [237]). After comparison, we select the top-k neighbors $\{t_i\}$ and inject them into the semantic part of the object query. The injection is implemented by vector addition, *i.e.*, adding each neighbor onto the semantic query $[q^s + t_i, q^p]$.

If k is large enough, these top-k neighbors will cover the ground-truth class of the object query. Injecting the ground-truth CLIP embedding into the semantic query improves the object query and thus derives a positive query, while the other $(k-1)$ CLIP embeddings (from the top-k) incur semantic distractions and derive $(k-1)$ negative queries. In practice, we set $k = 3$, resulting in 1 positive query and 2 negative queries. These k queries are then fed into the decoder, which is formulated as:

$$q'_i = h_{dec}([q^s + t_i, q^p], \boldsymbol{X}), i = 1, 2, \cdots, k, \tag{7.1}$$

where t_i is the i-th nearest neighbor for query q and $\boldsymbol{X}$ is the feature tokens from the encoder. h_{dec} denotes the decoder, and q'_i is a corresponding output from the decoder.

Eqn. 7.3 depicts only a single query q and its k decoder outputs for easy understanding. Since there are actually multiple (*e.g.*, 300) queries selected from the encoder output, we have $300 \times k$ decoder outputs in total. Given all these outputs, we utilize bipartite matching to associate them to the ground truth. If an output q'_j is associated with a ground-truth object, we select it as a positive sample for semantic contrasting. Meanwhile, we select all the $(k-1)$ outputs from the same query as its negative samples. Specifically, we use the popular InfoNCE [144] loss to contrast the positive q'_j and the other $(k-1)$ negative outputs, which is formulated

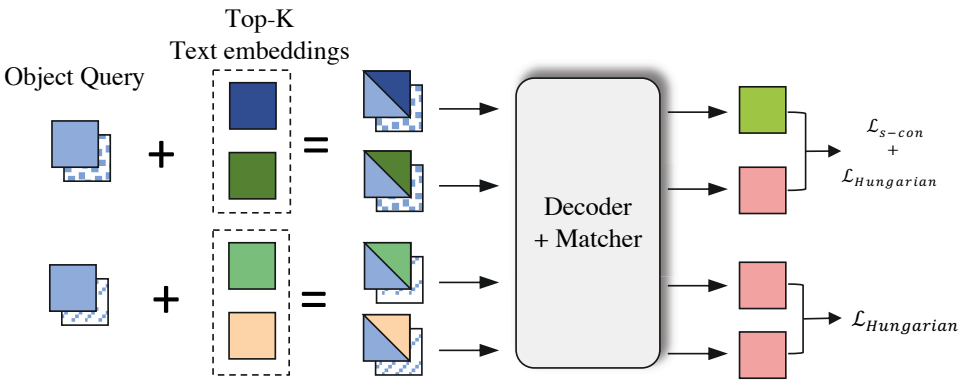


Figure 7.3 : Illustration of Semantic Contrasting. For each object query, we search its k-nearest neighbors of class text embeddings and inject them to form a contrastive group. After feeding each group into the decoder and the matcher, we identify the queries successfully matched with the ground truth (green). Then we impose contrastive loss on groups have matched queries, to clarify their semantic ambiguities in the text latent space.

as:

$$\mathcal{L}_{s-con} = -log\frac{exp((q'_j \odot q'_j)/\tau)}{\sum_{i=1}^{k} exp((q'_i \odot q'_j)/\tau)}, \tag{7.2}$$

where τ is the temperature and is set to 0.02 empirically, $\odot$ denotes the cosine similarity.

We note that during bipartite matching, many (most) queries have NULL positive outputs, *i.e.*, none of its k outputs is matched to the ground-truth object. For those queries, we do not enforce any contrastive learning, and simply ascribe them into the "non-object" class in the optimization of DETR.

7.2.4 Positional Contrasting

The position ambiguity between VLMs and the specialized object detector arises from their discrepancy in object recognition. CLIP prioritizes semantic matching

in holistic images while neglecting the positional variation inside images, *i.e.*, confirming the presence of an object without determining its location. In contrast, the detector is expected to precisely capture the positions of objects and identify their semantic classes. In light of this, this chapter develops a simple positional contrasting paradigm to reduce the position ambiguities in an end-to-end manner.

Fig. 7.4, illustrates the positional contrasting branch. It disturbs the position part of each query while suppressing its semantic ambiguity. Specifically, given an object query, we first suppress its semantic ambiguity by merging ground-truth semantics, *i.e.*, adding the CLIP text embedding of the ground-truth object into its content part. Afterward, we disturb its position embedding to generate a positive query (through slight positional jitter) and a negative query (through relatively large positional distractions). The details are as below:

• Slight positional jitter for a positive query. Augmenting the positive query with slight jitter is a good practice from DN-DETR [101]. For a bounding box with coordinates (x, y, w, h), we jitter its position into $(x+\rho\varrho w, y+\rho\varrho h, w+\rho\varrho w, h+\rho\varrho h)$, where ρ controls the noise scale and ϱ is a random number drawn from $(-0.5, 0.5)$.

• Large positional offset for generating a negative query. To generate a positional negative query, we disturb the original object query with large positional offsets. We design two alternative approaches, *i.e.*, random noising as in DINO [221] and positional shuffling. The random noising is similar to the manner of augmenting a positive query, except that the noise scale is much larger, *i.e.*, $[\rho, 2\rho]$. The positional shuffling randomly exchanges the position of different queries within the same mini-batch. Empirical studies show that these two approaches are both effective, and the latter performs better (Sec. 7.3.3). We postulate the reason is: the offset approach mainly draw attention to foreground-background ambiguity, which barely benefits discriminating different classes. In contrast, the shuffling is performed among fore-

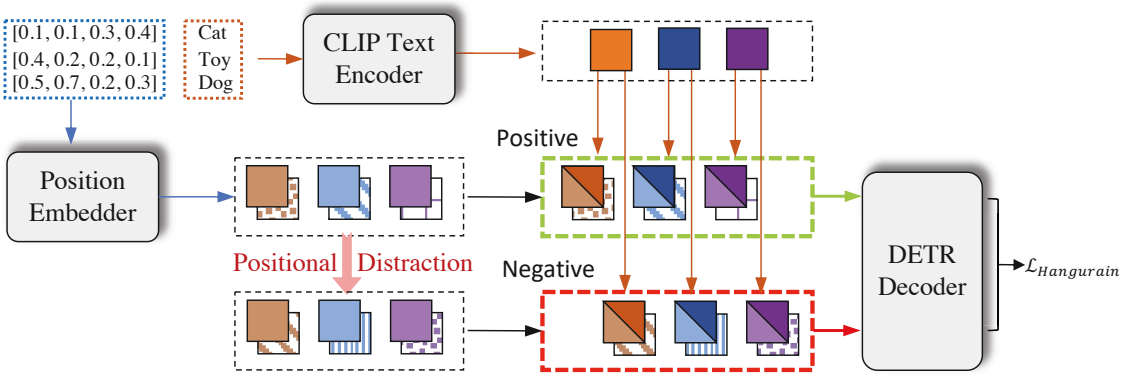


Figure 7.4 : Illustration of Positional Contrasting. Given a ground-truth object, we retrieve its class name and position, and then encode them with CLIP text encoder and position embedder separately. With their combinations as positive queries, we add position distractions on them to form negative queries, and feed them into the decoder together. Finally, positive queries are encouraged to match with the actual position and semantics while the others are excluded as "non-object" in the matching loss.

ground regions, therefore focusing on discriminating different foreground regions. According to common sense in contrastive learning, contrasting different fine-grained classes (within the foreground) against each other helps generalization towards novel classes, which is the keynote of OV-Det.

Given the positional positive and negative queries, DeCo contrasts them by applying different supervisions. Specifically, positive queries are urged to perform classification and position regression while the negatives are excluded into the "non-object" class. In particular, the above contrasting paradigm can be integrated the $\mathcal{L}_{Hungarian}$ of [240] seamlessly, *i.e.*, $\mathcal{L}_{box}$ and $\mathcal{L}_{focal}$ on positives for box regression and classification while also the $\mathcal{L}_{focal}$ excludes the negatives into the "non-object" class.

Table 7.1 : COCO [121] Open-vocabulary Object Detection Benchmark. Here we use CLIP[Caption] / CLIP[prompt] / CLIP[image] to denote the supervision encoded with CLIP [156]from different sources, *i.e.*, captions, prompts, and image proposals. We use "Bboxs" to denote the bounding box annotation. $\mathcal{V}_{Base}$ and $\mathcal{V}_{Novel}$ denote samples from the base and novel classes. MRCNN / FRCNN denote Mask-RCNN [68] / Faster-RCNN [161], and FPN is feature pyramid networks [119]

Method	Architecture	# Proposals (Train / Val)	Supervision on $\mathcal{V}_{Base}$	Supervision on $\mathcal{V}_{Novel}$	Resolution	Results		
						AP^b	AP^b_{novel}	AP^b_{base}
OVR-CNN [218] [CVPR21]	R50c4+FRCNN	2000/1000	CLIP[Captions] + Bboxs	CLIP[Captions]	800	46.0	22.8	39.9
ViLD [66] [ICLR22]	R50+FPN+MRCNN	1000/1000	CLIP[Prompts][Images] + Bboxs	CLIP[Prompts]	1024	51.3	27.6	59.5
RegionClip [234] [CVPR22]	R50c4+MRCNN	2000/1000	CLIP[Prompts] + Bboxs	CLIP[Prompts]	800	42.7	14.2	–
RegionClip [234] [CVPR22]	R50c4+MRCNN	2000/1000	CLIP[Captions] + Bboxs	CLIP[Captions]	800	47.5	26.8	–
Detic [237] [ECCV22]	Res50+F-RCNN	1000/1000	CLIP[Prompts] + Bboxs	CLIP[Prompts]	800	44.7	24.1	–
PromptDet [52] [ECCV22]	R50-FPN+MRCNN	1000/1000	CLIP[Prompts] + Bboxs	CLIP[Prompts]	640	50.6	26.6	–
OV-DETR [217] [ECCV22]	R50+DeformDETR	4500 / 19500	CLIP[Prompts][Images] + Bboxs	CLIP[Prompts]	800	52.7	29.4	61.0
F-VLM [93] [ICLR23]	R50 + FPN +MRCNN	1000/1000	CLIP[Prompts] + Bboxs	CLIP[Prompts]	1024	39.6	28.0	–
VLDet [117] [ICLR23]	R50c4+FRCNN	1000/1000	CLIP[Captions] + Bboxs	CLIP[Captions]	800	44.6	30.0	–
VLDet [117] [ICLR23]	R50c4+FRCNN	1000/1000	CLIP[Prompts] + Bboxs	CLIP[Prompts]	800	42.8	28.2	–
Ours	R50+DeformDETR	1000 / 300	CLIP[Prompts] + Bboxs	CLIP[Prompts]	800	53.5	30.1	62.6
Ours	R50+DeformDETR	1000 / 300	CLIP[Captions] + Bboxs	CLIP[Captions]	800	55.0	32.3	64.0

7.2.5 Training Objectives and Inference.

The overall training objectives are as follows:

$$\mathcal{L} = \mathcal{L}_{Hungarian} + \lambda_s \mathcal{L}_{s-con}. \tag{7.3}$$

We choose $\mathcal{L}_{Hungarian}$ following Defromable DETR [240], which consists of a classification loss ($\mathcal{L}_{focal}$ [120]), a $\mathcal{L}_1$ regression loss, and a GIoU loss [162].

During training, the semantic contrasting branch and the positional contrasting branch work in a parallel manner. Especially, we apply attention masks for self-attention modules in the decoder like [101, 84], thereby preventing their interactions and ground truth leakage.

Inference. As shown in Fig. 7.2 (Right), both branches are removed during the

Table 7.2 : Transfer Detection Benchmark. The generalization results on the test set of PASCAL-VOC [48] and the validation set of COCO, where the model is trained on LVIS but replaces its classifier with corresponding text embeddings.

Method	Pascal VOC		COCO		
	AP^b_{50}	AP^b_{75}	AP^b	AP^b_{50}	AP^b_{75}
ViLD-text [66] [ICLR22]	40.5	31.6	28.8	43.4	31.4
ViLD [66] [ICLR22]	72.2	56.7	36.6	55.6	39.8
OV-DETR [217] [ECCV22]	76.1	59.3	38.1	58.4	41.4
F-VLM [93][ICLR23]	–	–	32.5	53.1	34.6
Ours	**77.1**	**60.4**	**39.3**	**59.0**	**43.1**

inference. For each selected query, we attach it with its nearest text embedding and fuse them as element-wise sum, then feed them into the decoder and the prediction FFNs for predictions. Therefore, the proposed designs actually brings negligible computation overhead for inference.

7.3 Experiments

7.3.1 Experimental Setup

Dataset. Following common practice [218, 66], We evaluate the proposed method on two standard open-vocabulary detection benchmarks based on COCO [121] and LVIS [67], which is denoted as OV-COCO and OV-LVIS afterward.

LVIS [67] provides ample annotations (*i.e.*, masks and bounding boxes) for 100K images that span over 1203 classes. The classes can be divided into three groups, *i.e.*, frequent, common, and rare based on their proportion in the dataset. In line

with ViLD [66], the 337 rare classes are excluded from the training set and evaluated as novel classes during the evaluation. And the frequent and common classes are treated as base classes (866 in total).

COCO [121] is a widely-adopted object detection dataset, which provides annotations of 80 categories with 118K images. Following the zero-shot split in [3, 218], after removing 15 classes according to the WordNet hierarchy, 48 classes are chosen as base classes and the other 17 are novel classes. The training / validation sets follow the official setup but only images containing at least one common object are included in the training set.

Besides, following [218, 117, 237], we adopt COCO-Caption [29] and CC3M [174] to offer captions for COCO and LVIS, respectively.

Implementation Details. The detector setup follows OV-DETR [217], *i.e.*, ResNet50 [69] backbone and Deformable DETR [240]. As examined in OV-DETR, this model setup is generally in line with the capacity of Mask-RCNN, guaranteeing a fair starting point for comparison. The model is trained for 50 epochs with the batchsize of 16, where the model is optimized with the AdamW optimizer. Like ViLD [66], we adopt the open-source CLIP model based on ViT-B/32 for extracting text embeddings, and obtain the text embeddings of classes with prompt templates. The segmentation results are obtained following the training recipe from OV-DETR [217], where an additional masking head is attached for mask prediction. ρ is set to 0.2, k is set to 3, and λ_{s-con} is 2.0 for all experiments.

Evaluation. Following common practice, on OV-LVIS, we report the mask mAP for novel and overall classes, respectively. For OV-COCO, we report the box AP50 (at IoU of 0.5) for the base and overall classes accordingly. AP^m denotes the mask AP and AP^b denotes the box AP.

Table 7.3 : LVIS [67] Open-vocabulary Object Detection Benchmark. We use CLIP[Caption] / CLIP[prompt] / CLIP[image] to denote the supervision encoded with CLIP [156] from captions, prompts, and image proposals, respectively. "Bboxs" and "Masks" denote the bounding box and mask annotation. $\mathcal{V}_{Base}$ and $\mathcal{V}_{Novel}$ denote samples from the base and novel classes.

Method	Architecture	# Proposals (Train / Val)	Supervision on $\mathcal{V}_{Base}$	Supervision on $\mathcal{V}_{Novel}$	Resolution	Results	
						AP^m_{novel}	AP^m
ViLD [66] [ICLR22]	R50+FPN+MRCNN	1000/1000	CLIP[Prompts][Images] + Bboxs + Masks	CLIP[Prompts]	1024	16.6	25.5
RegionClip [234] [CVPR22]	R50c4+MRCNN	2000/1000	CLIP[Captions] + Bboxs+ Masks	CLIP[Captions]	800	17.1	28.2
Detic [237] [ECCV22]	R50+CenterNet2	1000/1000	CLIP[Prompts] + Bboxs+ Masks	CLIP[Prompts]	1024	17.8	26.8
OV-DETR [217] [ECCV22]	R50+DeformDetr	4500/390K	CLIP[Prompts][Images] + Bboxs+ Masks	CLIP[Prompts]	800	17.4	26.6
F-VLM [93] [ICLR23]	R50+FPN+MRCNN	1000/1000	CLIP[Prompts] + Bboxs + Masks	CLIP[Prompts]	1024	18.6	24.2
VLDet [117] [ICLR23]	R50c4 + CenterNet2	1000/1000	CLIP[Captions] + Bboxs+ Masks	CLIP[Captions]	640	20.4	30.4
VLDet [117] [ICLR23]	R50c4 + CenterNet2	1000/1000	CLIP[Prompts] + Bboxs+ Masks	CLIP[Prompts]	640	18.9	31.2
Ours	R50+DeformDETR	1000/300	CLIP[Prompts] + Bboxs + Masks	CLIP[Prompts]	800	19.6	32.1
Ours	R50+DeformDETR	1000/300	CLIP[Prompts] + Bboxs + Masks	CLIP[Prompts]	1024	20.2	32.4
Ours	R50+DeformDETR	1000/300	CLIP[Captions] + Bboxs + Masks	CLIP[Captions]	800	21.3	32.8

7.3.2 Comparison with Previous Methods.

OV-COCO and OV-LVIS benchmark. Table 7.1 and Table 7.3 show the comparison on OV-COCO and OV-LVIS accordingly, from which we draw three observations as below:

1) Compared with previous methods, our DeCo shows consistent superiority against them on both datasets, *e.g.*, + 2.1 and + 1.0 AP_{novel} on OV-COCO and OV-LVIS than F-VLM, respectively.

2) Regarding the number of employed proposals / queries, our DeCo is efficient. We note that the proposals (in Faster-RCNN decoders) and the queries (in DETR-style detectors) have a similar role because they both can be viewed as coarse candidates to be refined. Generally, using more proposals / queries improves the detection at the cost of more computation. Our DeCo uses relatively fewer queries while achieving higher mAP, and is thus efficient comparably.

Table 7.4 : Contribution of the proposed branches on OV-COCO and OV-LVIS. w.r.t.Semantic Contrasting (SemCon) and Positional Contrasting (PosCon).

SemCon	PosCon	OV-COCO		OV-LVIS	
		AP^b	AP^b_{novel}	AP^m	AP^m_{novel}
✓	✓	53.5	30.1	32.1	19.6
	✓	51.4	28.2	30.9	17.1
✓		52.0	29.1	31.5	17.6
		39.2	2.1	22.4	0.0

2) Comparing DeCo under two different settings, *i.e.*, using prompts (32.2 AP^m) and using captions (33.3 AP^m), we find that DeCo benefits from using more detailed text descriptions (the caption). This is another evidence that DeCo makes good use of the CLIP knowledge.

Transfer Detection Benchmark. Following [66, 217], we also explore the potential of DeCO as a generalizable detectors with text descriptions. For fair comparison, we adopt the same settings: we train the detector on LVIS and transfer it to Pascal VOC and COCO by replacing the text embeddings of class vocabularies. As shown in Table 7.2, the proposed DeCo presents good generalization. It surpasses the previous methods by a clear margin, *e.g.*, + 1.0 AP_{50} than OV-DETR on Pascal VOC.

7.3.3 Ablation Studies and Analysis.

In this section, we investigate the major components and characteristics of DeCo through comprehensive ablations.

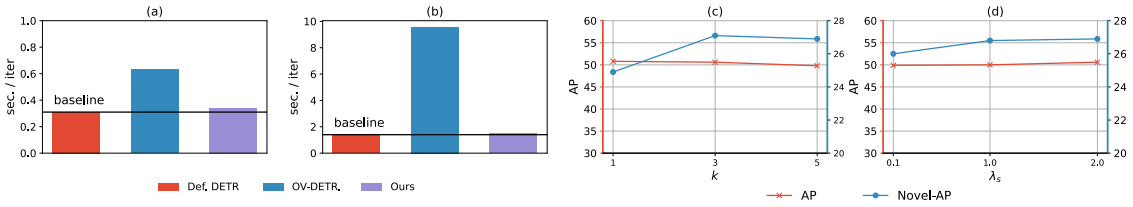


Figure 7.5 : (a)(b) Comparison of inference speed on COCO (a) and LVIS (b). (c) Comparison under different numbers of semantic neighbors, *i.e.*, k. (d) Sensitivity analysis to λ_s, the coefficient for the semantic contrastive loss. Note experiments report here adopt the 1 $\times$ schedule for training efficiency.

Contribution of semantic / positional contrasting. In Table 7.4, we gradually remove the semantic and position contrasting branches. It is observed that adding the semantic / positional contrasting both bring significant improvement (over the baseline in the last line). Moreover, combining these two branches brings further improvement, *e.g.*, "semantic contrasting + positional contrasting" is higher than semantic contrasting / positional contrasting by +1.0 / +1.9 AP_{Novel} on COCO and + 2.0 / +2.5 AP_{Novel} on LVIS. Therefore, we infer that these two components are both effective, and jointly achieve complementary benefit.

Analysis on the contrastive learning details. For a deep understanding of the two contrasting branches, we replace or modify some detailed designs in Table 7.5.

First, in Line #1, we enforce the semantic contrastive loss $\mathcal{L}_{s-con}$ onto all query groups, instead of onto the groups that have ground-truth matches. It leads to an obvious drop, *e.g.*, 2.8 AP_{Novel} on OV-COCO. It is because aligning text embeddings with semantic-confusing queries (*i.e.*, partial or multiple objects) can be harmful to the calibration between CLIP and the detector. Then in Line #2, we change the semantic neighbors to k random text embeddings instead of k-nearest ones. It leads to -1.6 AP_{Novel} decrease and verifies our intuition that the hard negatives are more

Table 7.5 : Ablation analysis on the semantic contrasting (SemCon) and positional contrasting (PosCon). Results here are obtained with the 1 × training schedule.

	Modification	OV-COCO	
		AP	AP_{novel}
#0	DeCo (1 × schedule)	50.6	27.1
#1	SemCon: Apply $\mathcal{L}_{s-con}$ on all queries	47.8	24.3
#2	SemCon: Random semantic neighbors	48.4	25.5
#3	PosCon (Position Jittering)	49.2	26.2

useful for suppressing the semantic ambiguity.

Second, in Line #3 of Table 7.5, we present the results of applying large positional offset for negative queries in positional contrasting. Compared with Line #0 (which uses the position shuffling for positional contrasting), it decreases -1.4 AP and -0.9 AP_{Novel}. Therefore, we recommend the shuffling manner for generating positional negative queries. The reason is that positional offset encourages foreground-background discrimination, which benefits little for the region-text affinity. In contrast, the positional shuffle can foster such affinity through contrastive learning between text embeddings and foreground queries.

Analysis on inference speed. In Fig. 7.5 (a)(b), we compare the inference speed of three DETR-style detectors, *i.e.*, the Deformable DETR [240] baseline, OV-DETR [217] and our DeCo on COCO and LVIS respectively. It is observed that OV-DETR incurs significant latency over the baseline, especially when the number of classes is large (on LVIS). This is because OV-DETR needs to combine each query with text embeddings from all the potential classes, therefore increasing the computational cost within the decoder by $N\times$ (N is the total class number). In contrast, our DeCo only brings a very slight increase in computational overhead,

because it uses the default 300 queries.

Hyper-parameter analysis. Here we analyze the sensitivity to hyper-parameters. Experiments here use the 1× schedule for efficiency. First, in Fig. 7.5 (c), we vary k, *i.e.*, the number of semantic neighbors, and compare their results. We can observe that performance improves as k increases initially but get saturated latterly. The reason is that $k = 3$ essentially includes meaningful hard negative samples while too large may bring some easy negative samples, which contribute less to the reduction of semantic ambiguity. Second, in Fig. 7.5 (d), we present the sensitivity analysis to λ_s, the coefficient in Eqn. 7.3. With a wide range of λ_s (0.1-2.0), the performance suffers very limited fluctuations but yields consistent improvements, showing our method is not sensitive to the selection of λ_s.

Visualizations. In Fig. 7.6, we visualize the affinity between encoder features and several query formulas. As we can observe, the CLIP embeddings show apparent position ambiguities, *i.e.*, weak affinity with the corresponding regions. Instead, the proposed fusion strategy derives more explicit and reasonable region-query correspondences, further validating the effectiveness of the proposed method.

7.4 Conclusion

In this chapter, we propose "Decouple to Contrast" (DeCo), a novel OV-Det method that jointly mitigates the semantic and position ambiguities from the unified contrastive learning viewpoint. Specifically, DeCo decouples these two ambiguities into two orthogonal factors for generating negative samples, because a negative sample with only "semantic / positional" offset is more challenging than negative ones with "semantic + positional" offsets. Extensive experiments are conducted on OV-Det benchmarks, and the results show that DeCo achieves the new state of the art.

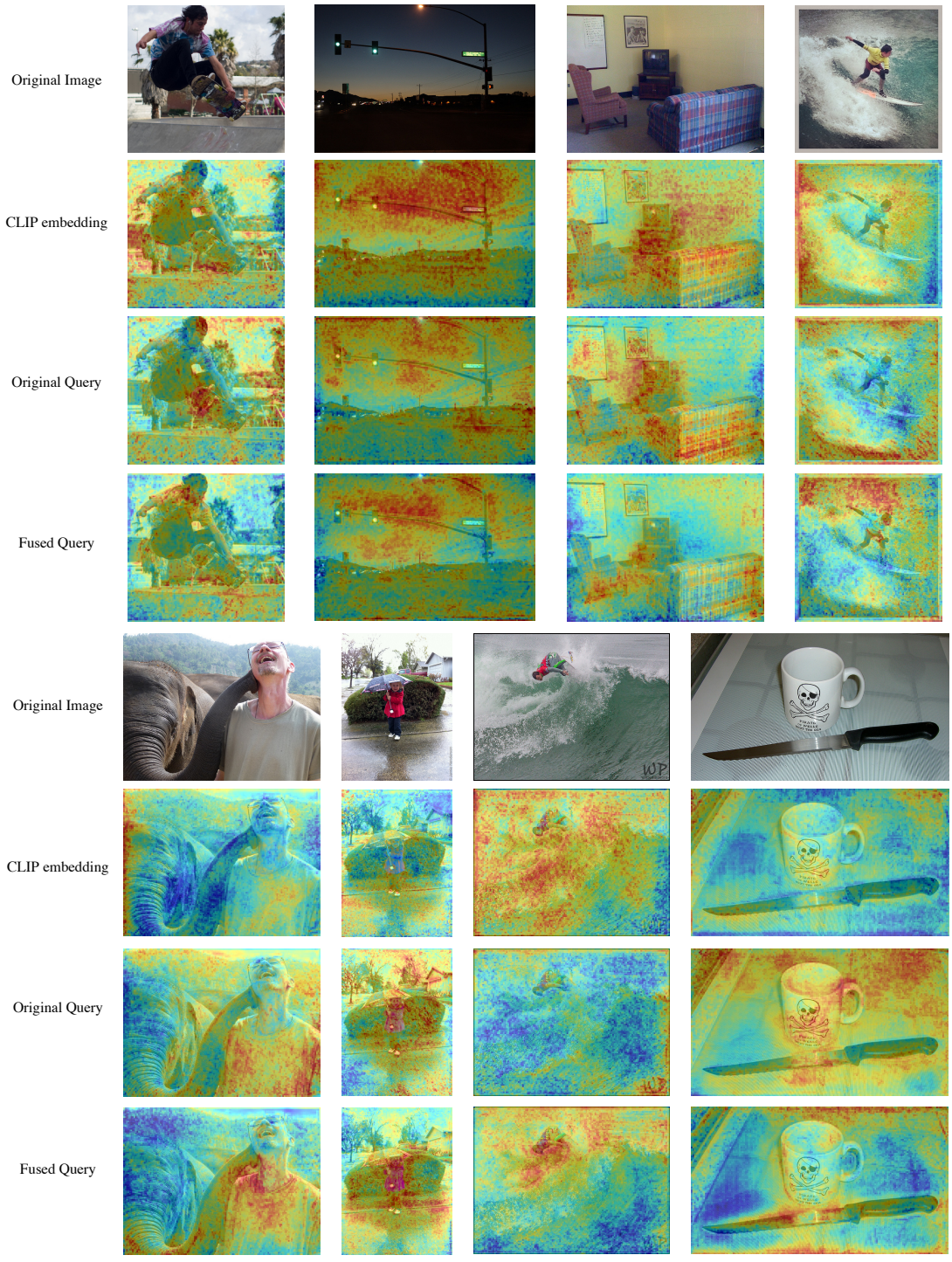


Figure 7.6 : Visualization of the affinity between encoder features and several variants of queries, *i.e.*, default queries, corresponding CLIP embeddings, and their fusions with DeCo. Results here are drawn from experiments on COCO.

Chapter 8

Conclusion and Future Works

Concluding my exploration, this thesis has navigated the intricate landscape of generalization within computer vision, a pivotal domain in the era of dynamic and evolving data environments. I embarked on a journey to unravel the two fundamental facets of generalizability: adapting to novel structures and accommodating novel categories.

My exploration in generalization with unseen visual structures uncovered the significance of structural knowledge extraction from diverse visual scenarios, encompassing 2D and 3D rigid scenes, as well as non-rigid structures. Through meticulous investigation, I unearthed the obstacles that hinder generalization, ranging from layout distribution disparities to disruptive dropout noises and variations in inter-joint relationships. The developed strategies, including layout matching, adversarial masking, and the "decompose to generalize" paradigm, provided effective solutions to enhance adaptability and generalizability across these diverse scenarios.

Furthermore, my pursuit of generalizing to novel categories addressed the pivotal questions of distinguishing and classifying previously unseen object categories. I tackled the "category shift" problem by introducing a clustering pipeline that effectively separates known from unknown categories based on cross-domain consensus knowledge. In parallel, I leveraged cross-modality insights from Vision-Language Models (VLMs) to discern distinctions between known and novel categories through discriminative mappings in the latent text space. The "decouple to contrast" methodology emerged as a key innovation to mitigate ambiguities between

visual and text representations.

Overall, this thesis has made substantial contributions to the field of computer vision by providing effective techniques and methodologies for enhancing generalizability across diverse environments and data distributions, encompassing various structural forms and dynamic scenarios. Furthermore, my approaches for generalizing to novel object categories have been rigorously tested and validated, offering promising prospects for real-world applications where adapting to unforeseen challenges is paramount. As the landscape of computer vision continues to evolve, the insights and strategies presented in this thesis pave the way for more robust and adaptable vision models in dynamic environments.

While this research has significantly improved generalizability, several exciting avenues for future exploration remain:

Generalizing Foundation Models to Novel Distributions. Despite the promising generalizability demonstrated by foundation models, they can still be vulnerable when confronted with extreme or rare scenarios. I aim to develop methodologies to identify and handle the Out-of-Distribution (OOD) scenarios, thereby further enhancing the generalizability of foundation models with the complex worlds.

Generalizing to Novel Categories with Multi-Modal Knowledge. Inspired by advancements in Vision-Language Models, my research aims to leverage multi-modal knowledge to enhance open-world generalizability. Besides discovering novel categories, open-world generalization requires a more comprehensive understanding of the unseen world, encompassing aspects such as human-object interactions, attribute recognition, and more. The abundance of language cues offers significant potential in this endeavor. By combining language and vision effectively, I aim to advance generalizability and adaptability with complex visual scenes by mining fine-grained details and concepts.

In conclusion, my Ph.D. research has primarily centered around addressing the generalization problem within a diverse array of visual structures. However, recent advancements in foundational models and generative models have reached an impressive pinnacle that surpassed my initial expectations. Sadly, these models have showcased remarkable generalizability, rendering my previous research somewhat obsolete. Nonetheless, I firmly believe that there remains a long run ahead for vision foundation models to attain the level of generalizability exemplified by ChatGPT in Natural Language Processing. This is because the realm of computer vision involves intricate and dynamic low-level properties as well as complex high-level concepts that cannot be comprehensively captured solely through the expansion of training datasets and annotations.

Bibliography

[1] Zeynep Akata, Mateusz Malinowski, Mario Fritz, and Bernt Schiele. Multi-cue zero-shot learning with strong supervision. In *CVPR*, pages 59–68, 2016.

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018.

[3] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, 2018.

[4] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *ICCV*, 2019.

[5] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[6] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018.

[7] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2846–2854, 2016.

[8] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *Journal of Machine Learning Research*, 2021.

[9] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems*, 2011.

[10] Guillem Brasó, Nikita Kister, and Laura Leal-Taixé. The center of attention: Center-keypoint grouping via attention for multi-person pose estimation. *ICCV*, 2021.

[11] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation. In *ECCV*. Springer, 2020.

[12] Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. Exploiting domain-specific features to enhance domain generalization. In *Advances in Neural Information Processing Systems*, volume 34, pages 21189–21201, 2021.

[13] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CVPR*, 2020.

[14] Yuanhao Cai, Zhicheng Wang, Zhengxiong Luo, Binyi Yin, Angang Du, Haoqian Wang, Xiangyu Zhang, Xinyu Zhou, Erjin Zhou, and Jian Sun. Learning delicate local representations for multi-person pose estimation. In *European Conference on Computer Vision*, pages 455–472. Springer, 2020.

[15] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018.

[16] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 1974.

[17] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*,

2019.

[18] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *ECCV*, 2018.

[19] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *CVPR*, 2019.

[20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[21] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019.

[22] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[23] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *IEEE CVPR*, June 2019.

[24] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *IEEE CVPR*, June 2019.

[25] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[26] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and

Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017.

[27] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, September 2018.

[28] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *IEEE ICCV*, October 2019.

[29] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[30] Yang Chen, Yu Wang, Yingwei Pan, Ting Yao, Xinmei Tian, and Tao Mei. A style and semantic memory mechanism for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9164–9173, October 2021.

[31] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *IEEE CVPR*, June 2019.

[32] Zhihong Chen, Chao Chen, Zhaowei Cheng, Boyuan Jiang, Ke Fang, and Xinyu Jin. Selective transfer with reinforced transfer network for partial domain adaptation. In *CVPR*, 2020.

[33] Bowen Cheng, Liang-Chieh Chen, Yunchao Wei, Yukun Zhu, Zilong Huang, Jinjun Xiong, Thomas S Huang, Wen-Mei Hwu, and Honghui Shi. Spgnet: Semantic prediction guidance for scene parsing. In *IEEE ICCV*, pages 5218–5228, 2019.

[34] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[35] Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park, and In So Kweon. Pointmixer: Mlp-mixer for point cloud understanding. *ECCV*, 2022.

[36] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *IEEE ICCV*, October 2019.

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

[38] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving, 2020.

[39] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, June 2020.

[40] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*, 2016.

[41] Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 129–136. JMLR Workshop and Conference Proceedings, 2010.

[42] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE*

TPAMI, 1979.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[44] Runyu Ding, Jihan Yang, Li Jiang, and Xiaojuan Qi. Doda: Data-oriented sim-to-real domain adaptation for 3d semantic segmentation. In *ECCV*, 2022.

[45] Qi Dou, Daniel C. Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[46] Liang Du, Jingang Tan, Hongye Yang, Jianfeng Feng, Xiangyang Xue, Qibao Zheng, Xiaoqing Ye, and Xiaolin Zhang. Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation. In *IEEE ICCV*, October 2019.

[47] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, 2022.

[48] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010.

[49] Hehe Fan, Xiaojun Chang, Wanyue Zhang, Yi Cheng, Ying Sun, and Mohan Kankanhalli. Self-supervised global-local structure modeling for point cloud domain adaptation with reliable voted pseudo labels. In *CVPR*, 2022.

[50] Siqi Fan, Qiulei Dong, Fenghua Zhu, Yisheng Lv, Peijun Ye, and Fei-Yue Wang. Scf-net: Learning spatial contextual features for large-scale point cloud segmentation. In *CVPR*, 2021.

[51] Z. Fang, J. Lu, F. Liu, J. Xuan, and G. Zhang. Open set domain adaptation: Theoretical bound and algorithm. *TNNLS*, 2020.

[52] Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. Promptdet: Towards open-vocabulary detection using uncurated images. In *ECCV*, 2022.

[53] Qianyu Feng, Guoliang Kang, Hehe Fan, and Yi Yang. Attract or distract: Exploit the margin of open set. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7990–7999, 2019.

[54] Wenkai Xu Feng Liu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J. Sutherland. Learning deep kernels for non-parametric two-sample tests. In *ICML*, 2020.

[55] Vladimir Fomenko, Ismail Elezi, Deva Ramanan, Laura Leal-Taix'e, and Aljoša Ošep. Learning to discover and detect objects. In *NeurIPS*, 2022.

[56] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *ECCV*, 2020.

[57] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, ICML'15, page 1180–1189. JMLR.org, 2015.

[58] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

[59] Mingfei Gao, Chen Xing, Juan Carlos Niebles, Junnan Li, Ran Xu, Wenhao Liu, and Caiming Xiong. Open vocabulary object detection with pseudo bounding-box labels. *arXiv preprint arXiv:2111.09452*, 2021.

[60] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015.

[61] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich fea-

ture hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[62] Rui Gong, Yuhua Chen, Danda Pani Paudel, Yawei Li, Ajad Chhatkuli, Wen Li, Dengxin Dai, and Luc Van Gool. Cluster, split, fuse, and update: Meta-learning for open compound domain adaptive semantic segmentation. In *CVPR*, 2021.

[63] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *IEEE CVPR*, June 2019.

[64] Yunye Gong, Xiao Lin, Yi Yao, Thomas G Dietterich, Ajay Divakaran, and Melinda Gervasio. Confidence calibration for domain generalization under covariate shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8958–8967, 2021.

[65] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[66] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary detection via vision and language knowledge distillation. *ICLR*, 2022.

[67] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[68] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[70] Tong He, Chunhua Shen, and Anton van den Hengel. DyCo3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *CVPR*, 2021.

[71] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[72] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[73] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022.

[74] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE TPAMI*, 2021.

[75] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[76] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Cross-view regularization for domain adaptive panoptic segmentation. In *CVPR*, 2021.

[77] Jiaxing Huang, Shijian Lu, Dayan Guan, and Xiaobing Zhang. Contextual-relation consistent domain adaptation for semantic segmentation. In *ECCV*, 2020.

[78] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[79] Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, and Thomas Huang. Ccnet: Criss-cross attention for semantic segmentation. *TPAMI*, 2020.

[80] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *Advances in Neural Information Processing Systems*, volume 34, pages 2427–2440, 2021.

[81] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *ECCV*, 2014.

[82] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *ICLR*, 2017.

[83] Zhong Ji, Yanwei Fu, Jichang Guo, Yanwei Pang, Zhongfei Mark Zhang, et al. Stacked semantics-guided attention model for fine-grained zero-shot learning. *NeurIPS*, 31, 2018.

[84] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. *arXiv preprint arXiv:2207.13080*, 2022.

[85] Peng Jiang and Srikanth Saripalli. Lidarnet: A boundary-aware domain adaptation model for point cloud semantic segmentation. *ICRA*, 2021.

[86] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020.

[87] Guoliang Kang, Lu Jiang, Yunchao Wei, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for single-and multi-source domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[88] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, pages 4893–4902, 2019.

[89] Guoliang Kang, Yunchao Wei, Yi Yang, Yueting Zhuang, and Alexander G Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. In *NeurIPS*, 2020.

[90] Guoliang Kang, Liang Zheng, Yan Yan, and Yi Yang. Deep adversarial at-

tention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. In *ECCV*, pages 401–416, September 2018.

[91] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11977–11986, 2019.

[92] Jogendra Nath Kundu, Naveen Venkat, Rahul M V, and R. Venkatesh Babu. Universal source-free domain adaptation. In *CVPR*, 2020.

[93] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. Open-vocabulary object detection upon frozen vision and language models. In *ICLR*, 2023.

[94] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019.

[95] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *IEEE CVPR*, June 2019.

[96] Chen Li and Gim Hee Lee. Coarse-to-fine animal pose and shape estimation. In *Advances in Neural Information Processing Systems*, volume 34, pages 11757–11768, 2021.

[97] Chen Li and Gim Hee Lee. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[98] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence*, 2018.

[99] Dongxu Li, Xin Yu, Chenchen Xu, Lars Petersson, and Hongdong Li. Trans-

ferring cross-domain knowledge for video sign language recognition. In *CVPR*, pages 6205–6214, 2020.

[100] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. Episodic training for domain generalization. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[101] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, 2022.

[102] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *European Conference on Computer Vision*, pages 440–456. Springer, 2020.

[103] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[104] Guangrui Li, Yifan Sun, Zongxin Yang, and Yi Yang. Decompose to generalize: Species-generalized animal pose estimation. In *The Eleventh International Conference on Learning Representations*, 2023.

[105] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018.

[106] Liulei Li, Tianfei Zhou, Wenguan Wang, Jianwu Li, and Yi Yang. Deep hierarchical semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1246–1257, 2022.

[107] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M. Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*

(ICCV), pages 8886–8895, October 2021.

[108] Peike Li, Yunchao Wei, and Yi Yang. Consistent structural relation learning for zero-shot segmentationg. In *NeurIPS*, 2020.

[109] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018.

[110] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognit.*, 80:109–117, 2018.

[111] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *IEEE CVPR*, June 2019.

[112] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. Zero-shot object detection with textual descriptions. In *AAAI*, 2019.

[113] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *IEEE ICCV*, October 2019.

[114] Hanxue Liang, Hehe Fan, Zhiwen Fan, Yi Wang, Tianlong Chen, Yu Cheng, and Zhangyang Wang. Point cloud domain adaptation via masked local 3d structure prediction. *ECCV*, 2022.

[115] Jian Liang, Yunbo Wang, Dapeng Hu, Ran He, and Jiashi Feng. A balanced and uncertainty-aware approach for partial domain adaptation. In *ECCV*, 2020.

[116] Zhidong Liang, Zehan Zhang, Ming Zhang, Xian Zhao, and Shiliang Pu. Rangeioudet: Range image based real-time 3d object detector optimized by

intersection over union. In *CVPR*, 2021.

[117] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai. Learning object-language alignments for open-vocabulary object detection. In *ICLR*, 2023.

[118] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE CVPR*, July 2017.

[119] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[120] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[121] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[122] Feng Liu, Guangquan Zhang, and Jie Lu. Heterogeneous domain adaptation: An unsupervised approach. *IEEE TNNLS*, 2020.

[123] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *ICLR*, 2022.

[124] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[125] Xiaofeng Liu, Site Li, Lingsheng Kong, Wanqing Xie, Ping Jia, Jane You, and B.V.K. Kumar. Feature-level frankenstein: Eliminating variations for

discriminative recognition. In *IEEE CVPR*, June 2019.

[126] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[127] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, page 97–105, 2015.

[128] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*, 2016.

[129] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *IEEE CVPR*, 2019.

[130] Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. In *NeurIPS*, 2017.

[131] Lucas Mansilla, Rodrigo Echeveste, Diego H. Milone, and Enzo Ferrante. Domain generalization via gradient surgery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6630–6638, October 2021.

[132] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.

[133] Alexander Mathis, Thomas Biasi, Steffen Schneider, Mert Yuksekgonul, Byron Rogers, Matthias Bethge, and Mackenzie W. Mathis. Pretraining boosts out-of-domain robustness for pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1859–1868, January 2021.

[134] Jiaxu Miao, Yunchao Wei, Yu Wu, Chen Liang, Guangrui Li, and Yi Yang. Vspw: A large-scale dataset for video scene parsing in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[135] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *IROS*, 2019.

[136] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Anurag Arnab Aravindh Mahendran, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. *ECCV*, 2022.

[137] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Posefix: Model-agnostic general human pose refinement network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7773–7781, 2019.

[138] Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. *ICLR*, 2018.

[139] Jiteng Mu, Weichao Qiu, Gregory D. Hager, and Alan L. Yuille. Learning from synthetic animals. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[140] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

[141] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[142] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis

and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[143] Xun Long Ng, Kian Eng Ong, Qichen Zheng, Yun Ni, Si Yong Yeo, and Jun Liu. Animal kingdom: A large and diverse dataset for animal behavior understanding. In *CVPR*, 2022.

[144] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[145] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *CVPR*, 2020.

[146] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010.

[147] Yingwei Pan, Ting Yao, Yehao Li, Chong-Wah Ngo, and Tao Mei. Exploring category-agnostic clusters for open-set domain adaptation. In *CVPR*, 2020.

[148] P. Panareda Busto, A. Iqbal, and J. Gall. Open set domain adaptation for image and action recognition. *IEEE TPAMI*, 2020.

[149] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017.

[150] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *CVPR*, 2022.

[151] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

[152] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang,

and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

[153] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, 2017.

[154] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017.

[155] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *NeurIPS*, 2019.

[156] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[157] Shafin Rahman, Salman Khan, and Nick Barnes. Improved visual-semantic alignment for zero-shot object detection. In *AAAI*, 2020.

[158] Shafin Rahman, Salman H Khan, and Fatih Porikli. Zero-shot object detection: Joint recognition and localization of novel concepts. *IJCV*, 2020.

[159] Hanoona Rasheed, Muhammad Maaz, Muhammad Uzair Khattak, Salman Khan, and Fahad Shahbaz Khan. Bridging the gap between object and image-level representations for open-vocabulary detection. In *36th Conference on Neural Information Processing Systems (NIPS)*, 2022.

[160] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[161] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[162] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. *CVPR*, 2019.

[163] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[164] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021.

[165] Mrigank Rochan, Shubhra Aich, Eduardo R Corral-Soto, Amir Nabatchian, and Bingbing Liu. Unsupervised domain adaptation in lidar semantic segmentation with self-supervision and gated adapters. *ICRA*, 2022.

[166] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE CVPR*, June 2016.

[167] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 1987.

[168] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[169] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self-supervision. In *NeurIPS*, 2020.

[170] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *ICML*, pages 2988–2997. JMLR. org, 2017.

[171] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada.

Open set domain adaptation by backpropagation. In *ECCV*, 2018.

[172] Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Elisa Ricci, and Fabio Poiesi. Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. *ECCV*, 2022.

[173] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018.

[174] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[175] Yuefan Shen, Yanchao Yang, Mi Yan, He Wang, Youyi Zheng, and Leonidas J. Guibas. Domain adaptation on point clouds via geometry-aware implicits. In *CVPR*, 2022.

[176] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[177] Yuge Shi, Jeffrey Seely, Philip H. S. Torr, N. Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *ICLR*, 2022.

[178] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[179] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, pages 443–450. Springer, 2016.

[180] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution

representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[181] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *CVPR*, 2021.

[182] Haotian* Tang, Zhijian* Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, 2020.

[183] Wei Tang and Ying Wu. Does learning specific features for related parts help human pose estimation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[184] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *ICCV*, 2019.

[185] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *CVPR*, 2019.

[186] 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 CVPR*, 2018.

[187] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE CVPR*, July 2017.

[188] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[189] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*,

abs/1412.3474, 2014.

[190] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation, 2017.

[191] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

[192] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Mathieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *IEEE CVPR*, 2019.

[193] Dongkai Wang, Shiliang Zhang, and Gang Hua. Robust pose estimation in crowded scenes with direct pose-level inference. In *Advances in Neural Information Processing Systems*, volume 34, pages 6278–6289, 2021.

[194] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019.

[195] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM TOG*, 2019.

[196] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-mei Hwu, Thomas S. Huang, and Honghui Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *IEEE CVPR*, pages 12635–12644, June 2020.

[197] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convo-

lutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, 2018.

[198] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, 2019.

[199] Guile Wu and Shaogang Gong. Collaborative optimization and aggregation for decentralized domain generalization and adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6484–6493, October 2021.

[200] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *IEEE CVPR*, June 2019.

[201] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. *CVPR*, 2019.

[202] Zhonghua Wu, Yicheng Wu, Guosheng Lin, Jianfei Cai, and Chen Qian. Dual adaptive transformations for weakly supervised point cloud segmentation. *ECCV*, 2022.

[203] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Synlidar: Learning from synthetic lidar sequential point cloud for semantic segmentation. *AAAI*, 2022.

[204] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In Jennifer Dy and Andreas Krause, editors, *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 5423–5432, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[205] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer,

and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *ECCV*, 2020.

[206] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *CVPR*, 2021.

[207] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. *CVPR*, 2020.

[208] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR*, June 2020.

[209] Fu-En Yang, Yuan-Chia Cheng, Zu-Yun Shiau, and Yu-Chiang Frank Wang. Adversarial teacher-student representation learning for domain generalization. In *Advances in Neural Information Processing Systems*, volume 34, pages 19448–19460, 2021.

[210] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020.

[211] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. DetCLIP: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. In *NeurIPS*, 2022.

[212] Li Yi, Boqing Gong, and Thomas Funkhouser. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In *CVPR*, 2021.

[213] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *CVPR*, 2019.

[214] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. Ap-

10k: A benchmark for animal pose estimation in the wild. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[215] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[216] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. *CVPR*, 2022.

[217] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. In *ECCV*, 2022.

[218] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, 2021.

[219] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7093–7102, 2020.

[220] Guojun Zhang, Han Zhao, Yaoliang Yu, and Pascal Poupart. Quantifying and improving transferability in domain generalization. In *Advances in Neural Information Processing Systems*, volume 34, pages 10957–10970, 2021.

[221] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Harry Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2023.

[222] J. Zhang, Z. Ding, W. Li, and P. Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *CVPR*, 2018.

[223] Marvin Mengxin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta,

Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group shift. In *NeurIPS*, 2021.

[224] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In *NeuralPS*, pages 433–443, 2019.

[225] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *IEEE CVPR*, June 2018.

[226] Yang Zhang, Philip David, Hassan Foroosh, and Boqing Gong. A curriculum domain adaptation approach to the semantic segmentation of urban scenes. *IEEE TPAMI*, pages 1–1, 2019.

[227] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *IEEE ICCV*, page 6, Oct 2017.

[228] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *CVPR*, June 2020.

[229] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *ICCV*, 2019.

[230] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE CVPR*, pages 2881–2890, 2017.

[231] Sicheng Zhao, Yezhen Wang, Bo Li, Bichen Wu, Yang Gao, Pengfei Xu, Trevor Darrell, and Kurt Keutzer. epointda: An end-to-end simulation-to-real domain adaptation framework for lidar point cloud segmentation. In *AAAI*, 2021.

[232] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *arXiv preprint*

arXiv:2003.03773, 2020.

[233] Zhedong Zheng and Yi Yang. Unsupervised scene adaptation with memory regularization in vivo. *IJCAI*, 2020.

[234] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *CVPR*, 2022.

[235] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021.

[236] Tianfei Zhou, Liulei Li, Xueyi Li, Chun-Mei Feng, Jianwu Li, and Ling Shao. Group-wise learning for weakly supervised semantic segmentation. *IEEE Transactions on Image Processing*, 31:799–811, 2021.

[237] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022.

[238] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *IEEE ICCV*, 2017.

[239] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Don't even look once: Synthesizing features for zero-shot detection. In *CVPR*, 2020.

[240] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2020.

[241] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*, 2021.

[242] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised

domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pages 289–305, 2018.

[243] Yang Zou, Zhiding Yu, Xiaofeng Liu, B.V.K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *IEEE ICCV*, October 2019.