Deep 3D Information Prediction and Understanding

SHANSHAN ZHAO

Supervisor: Prof. Dacheng Tao

A Thesis Submitted in Fulfilment of the Requirements for the Degree of
Doctor of Philosophy

Faculty of Engineering
The University of Sydney

2022

To my family.

Statement of Originality

This thesis is submitted to the School of Computer Science, Faculty of Engineering, the University of Sydney in fulfilment of the requirements for the degree of Doctor of Philosophy. I certify that the content of this thesis is my own work and this thesis has not been submitted for any degree or other purposes. I also certify that all the assistance received in preparing this thesis and sources have been acknowledged.

Shanshan Zhao
School of Computer Science
Faculty of Engineering
The University of Sydney

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my supervisor **Prof. Dacheng Tao**, who gave me an opportunity for pursing the Ph.D. degree in his team at The University of Sydney. During my three-year Ph.D. studies, Prof. Tao gives me patient and valuable guidance in every aspect of my research, which helps me improve and broaden my research vision. Moreover, his enthusiastic and positive working attitude also always encourages me to go ahead in my academic research. Without his supervision, I would not complete my Ph.D. research successfully.

I would like to express my sincere gratitude to **Dr. Mingming Gong** (The University of Melbourne), who provides me numerous constructive suggestions. He helps me analyze the motivation and logic behind the proposed methods, and improve the writing skills. I also would like to express my sincere appreciation to **Dr. Huan Fu**. The discussions about problem analysis, proposed solutions, and paper writing with him helped me a lot to complete my first research work.

I would like to thank my collaborators Dr. Tongliang Liu and Dr. Ang Li for their insightful discussions. I also would like to thank my colleagues: Dr. Chang Xu, Dr. Jiayan Qiu, Dr. Shaoli Huang, Dr. Chaoyue Wang, Dr. Baosheng Yu, Dr. Xiyu Yu, Dr. Zhe Chen, Dr. Dalu Guo, Dr. Liu Liu, Dr. Yuxuan Du, Dr. Erkun Yang, Dr. Jing Zhang, Dr. Qiuju Yang, Mr. Yu Cao, Mr. Lianbo Zhang, Mr. Zhuozhuo Tu, Mr. Cheng Wen, Mr. Zhen Wang, Mr. Zeyu Feng, Mr. Chenwei Ding, Mr. Jiaxian Guo, Mr. Xinqi Zhu, Mr. Youjian Zhang, Mr. Yufei Xu, Mr. Shuo Yang, Mr. Chang Li, Mr. Liang Ding, Mr. Qiming Zhang, Mr. Haoyu He, Mr. Kaining Zhang, Mr. Sen Zhang, Mr. Benteng Ma, Mr. Hao Guan, Mr. Lilei Wu, Ms. Qi Zheng, Ms. Haimei Zhao, Ms. Sihan

Ma, and Ms. Xiaofei Liu. Moreover, I also would like to thank my roommates Mr. Guangrui Li and Mr. Defa Ge for their company and supports. Living with them made my life in Sydney full of happiness, which is an interesting and memorable experience. I also would like to thank my friends Dr. Lei Bai, Dr. Wei Ji, and Dr. Yiming Wu, whom I often chat with and hear interesting things from.

Finally, I would like to pay the best regards and gratefulness to my family, especially to my parents. Their sincere encouragement, love, and infinite support helped me get through the difficult times in both life and study during my Ph.D. studies. I am extremely grateful to them.

Abstract

In comparison to 2D image data, 3D information is more closely related to the human visual perception and helps intelligent machines better understand the world. 3D information prediction and understanding, such as structure prediction and semantic analysis, play significant roles in 3D visual perception. Specific to the 3D structure, like depth data, although we can acquire it from various 3D sensors, there still have been tremendous attempts made to predict it from a single image, a video sequence, stereo data, or multi-modal data in machine learning frameworks. The main reason is that the 3D sensors are usually costly and the captured 3D data is generally sparse and noisy. Moreover, there are also numerous images in the website, of which we expect to obtain the depth map. Recent studies have demonstrated the superiority of deep neural networks, like deep convolutional neural networks (DCNNs), in relevant tasks. Despite the great success of deep learning, there are still many challenging issues to be solved. For example, although supervised deep learning has prompted the great performance improvements of the depth estimation model, the demand for amounts of ground truth depth data is hardly to satisfy in many scenarios. Therefore, unsupervised learning strategy is required for training 3D structure estimation model. In this thesis, we take a well-known specific task, *i.e.*, monocular depth estimation, as an example to study this problem. To reduce the demand for ground truth depth, we investigate the domain adaptation technique for learning depth model on synthetic data and explore the geometric information in real data to make the domain adaptation process aware of the geometric structure in real domain. Apart from the prediction from a single image or multiple images, we can also estimate the depth from multi-modal data, such as RGB

image data coupled with 3D laser scan data. To achieve this, some challenging issues need to be addressed. For example, since the 3D data is usually sparse and irregularly distributed, we are required to model the contextual information from the sparse data and fuse the multi-modal features. In this thesis, we examine the issues by studying the depth completion task. In specific, we propose to adopt graph propagation to capture the observed spatial contexts and introduce the symmetric gated fusion strategy to effectively combine the extracted multi-modal features.

Currently, various classical DCNNs have been proposed to process the 2D image data for various analyses, like semantic understanding. In contrast, for 3D point set, which is a significant 3D information representation, due to the sparsity and property to be unordered, instead of the conventional convolution, new operations which can model the local shape are required in order to understand the semantic contents. In this thesis, we select the point sets as the representation of 3D data, *i.e.,* 3D point cloud, and then design a basic operation for point cloud analysis. Previous works mainly consider the relation between each pair of adjacent points for feature aggregation but ignore the relation between edges, which encodes the local shape structure. To provide a remedy, we introduce a novel adaptive edge-to-edge interaction learning module. Besides, due to the diversity in configurations of the 3D laser scanners, the captured 3D data often varies from dataset to dataset in object size, density, and viewpoints. As a result, the domain generalization in 3D data analysis is also a critical problem. However, to our best knowledge, this problem is still under-explored. To provide a preliminary exploration into this issue, we also study domain generalization in 3D shape classification by proposing an entropy regularization term that measures the dependency between the learned features and class labels.

Through studying four specific tasks, this thesis focuses on several crucial issues in deep 3D information prediction and understanding, including model designing, multi-modal fusion, sparse data analysis, unsupervised learning, domain adaptation, and domain generalization, as introduced above.

Publication List

(1) **S. Zhao**, H. Fu, M. Gong, and D. Tao, "Geometry-Aware Symmetric Domain Adaptation for Monocular Depth Estimation", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9788-9798, 2019. **[Chapter 2]**

(2) **S. Zhao**, M. Gong, H. Fu, and D. Tao, "Adaptive Context-Aware Multi-Modal Network for Depth Completion", *IEEE Transaction on Image Processing*, vol. 30, pp. 5264-5276, 2021. **[Chapter 3]**

(3) **S. Zhao**, M. Gong, T. Liu, H. Fu, and D. Tao, "Domain Generalization via Entropy Regularization", *Advances in Neural Information Processing Systems*, pp. 16096-16107, 2020. **[Chapter 5]**

(4) **S. Zhao**, M. Gong, H. Fu, and D. Tao, "Adaptive Edge-to-Edge Interaction Learning for Point Cloud Analysis", *Under Review*, 2021. **[Chapter 4]**

(5) A. Li, **S. Zhao**, X. Ma, M. Gong, J. Qi, R. Zhang, D. Tao, and R. Kotagiri, "Short-term and Long-term Context Aggregation Network for Video Inpainting", *European Conference on Computer Vision (ECCV)*, pp. 728-743, 2020. (Spotlight)

CONTENTS

List of Figures

List of Tables

CHAPTER 1

Introduction

We live in a 3D world. We naturally understand the scene we are seeing, such as objects, scene structure, and object relation, through parsing the 3D information we directly capture from the 3D world. For an intelligent machine, it can also infer rich knowledge from the data containing 3D information, *e.g.,* geometric structure, which has been exploited in various scenarios, like autonomous driving (Chen *et al.*, 2017b), indoor navigation (Zhu *et al.*, 2019), robot manipulation (Mousavian *et al.*, 2019), augmented reality (Stekovic *et al.*, 2020), and virtual try-on (Pons-Moll *et al.*, 2017). We can obtain the 3D structure information using different sensors, such as Kinect, LiDAR, and RADAR. However, these sensors are usually costly and the captured 3D data is generally sparse and noisy. More importantly, there are numerous single images and video sequences in the website. Therefore, it is worth studying how to extract the 3D structure, such as depth information, from them, which can be further utilized to serve the downstream tasks, like object detection (You *et al.*, 2019), room layout estimation (Zhang *et al.*, 2020), and saliency detection (Zhang *et al.*, 2021b). Previous studies show that the human visual system can perceive specific 3D forms in single 2D contour images through associating 2D pictures with 3D structures (Sinha and Poggio, 1996). However, it is a challenging task for machine to capture the 3D information from single or multiple images. Traditional solutions to this target generally rely on handcrafted features, *e.g.,* HOG (Dalal and Triggs, 2005), SURF (Bay *et al.*, 2008), and SIFT (Lowe, 1999), and typical machine learning algorithms, like PGMs (Saxena *et al.*, 2006; Saxena *et al.*,

2009). Past several decades have witnessed the advancement gained by these conventional approaches for various computer vision tasks, including 3D structure estimation. However, designing suitable features for specific tasks itself is a tricky problem, and decoupling the feature engineering and model learning might cause that we cannot extract discriminative knowledge from the input data, like images. Luckily, in the deep learning era, these awkward issues have been alleviated greatly.

In 2012, a noteworthy work, AlexNet (Krizhevsky *et al.*, 2012), was proposed, which outperformed previous object recognition methods by a large margin. The presences of AlexNet and several classical networks (*e.g.,* VGGNet (Simonyan and Zisserman, 2015), GoogleNet (Szegedy *et al.*, 2015), and ResNet (He *et al.*, 2016)) following it have prompted the impressive development of DCNNs in the computer vision research community, especially for the 2D image understanding tasks. Despite being efficient for various computer vision tasks, these networks still have limitations in, such as, dense pixel prediction, sparse data representation, context modeling, and sequence modeling to name a few. To provide remedies, various strategies and modules are proposed to integrate with or replace the 2D convolution in those classical networks, such as, Atrous Spatial Pyramid Pooling (ASPP) (Chen *et al.*, 2017a), Sparse Convolution (Graham *et al.*, 2018), Deformable Convolution (Dai *et al.*, 2017b), Transformer / Attention (Vaswani *et al.*, 2017), and Convolutional LSTM (ConvLSTM) (Xingjian *et al.*, 2015).

The early study on deep learning for computer vision mainly focuses the 2D image analysis, including object detection (Girshick *et al.*, 2014), semantic segmentation (Long *et al.*, 2015), and 3D structure prediction from images (Eigen *et al.*, 2014). Relying on DCNNs, remarkable performance improvements have been achieved for 3D structure prediction, like depth estimation. In recent years, both academic and industrial circles attempt to make the autonomous driving and robotics existing our imagination become true. In these intelligent systems,

one key computer vision problem is estimating the 3D structure in a learning framework, while another is processing and understanding the data containing 3D information captured from the equipped sensors, like 3D laser scan data, or estimated using machine learning algorithms, like depth map. In comparison with 2D image analysis, the selection of deep networks for 3D information understanding relies more on the data format. For example, the RGB-D saliency detection (Peng *et al.*, 2014) and RGB-D segmentation (Wang and Neumann, 2018) tasks often take an RGB image and a depth map as input, which can be processed by the standard 2D convolution separately (Fu *et al.*, 2020) or jointly by the 3D convolution (Chu *et al.*, 2018). In comparison to the typical RGB-D saliency detection and RGB-D segmentation, in some scenarios where 3D laser scanners are available, we are often required to process the raw 3D information, like 3D point cloud, which is sparse and irregularly distributed, directly. Due to the sparsity and irregular characteristic, it is difficult to apply the standard 2D or 3D convolution into the raw 3D data. As a result, existing works often first select a specific representation for 3D data, such as voxels (Choy *et al.*, 2019), meshes (Hanocka *et al.*, 2019), and points (Qi *et al.*, 2017a), and then design or select suitable operations, like sparse convolutions (Choy *et al.*, 2019), mesh convolutions (Hanocka *et al.*, 2019), graph convolutions (Zhang and Rabbat, 2018), and multi-layer perceptron (Qi *et al.*, 2017b), to extract features for high-level semantic analysis.

Designing suitable networks for specific tasks using the domain knowledge, improving the robustness and generalization capabilities of deep models, and understanding the theory behind the deep learning are crucial issues for building future intelligent systems and are still ongoing. In this thesis, we study two problems in 3D vision, including 3D information prediction and understanding, and show our efforts to investigate the former two key issues in these two problems. Our research on the first problem could help the industries develop efficient 3D structure prediction deep networks by mining the specific domain knowledge,

while the study on the second could provide some solutions to the issues in real world applications, such as the lack of ground truth data and distribution inconsistency.

1.1 Depth Prediction and Point Cloud Analysis

There are various ways to obtain the 3D structure, which can be divided two main categories, *i.e.,* sensors and learning. In specific, a direct way to capture the structure is using sensors, such as Kinect for indoor and LiDAR for outdoor. However, these devices are usually costly and the captured 3D data is generally sparse and noisy. Moreover, in some scenarios, we expect to extract the structure information from images. To this end, lots of traditional machine algorithms (*e.g.,* HOG+PGMs) and deep learning algorithms have been proposed for relevant tasks. Since depth data is a significant 3D structure representation, a large number of previous works focus on the depth prediction task. Due to the strong capability of modeling the discriminative features, deep learning, especially DCNNs, has dominated the depth estimation community. For example, Eigen *et al.* (Eigen *et al.*, 2014) make the first attempt to apply DCNNs for depth estimation from a single image. In comparison with single images, stereo data, multi-modal data (*e.g.,* RGB image+sparse depth data), and video data contain more geometric information, which is helpful to promote the performance in depth estimation. In detail, deep stereo matching (or disparity estimation) models can be trained from a set of stereo images (Chang and Chen, 2018; Cheng *et al.*, 2020b). Moreover, exploiting the epipolar geometry, we can also learn a monocular depth estimation model from stereo data in a unsupervised learning framework (Garg *et al.*, 2016; Godard *et al.*, 2017). In this way, we can reduce the dependency on amounts of ground truth data. Structure-from-motion (SFM) is an essential computer vision problem, which aims at inferring the geometric structure from the motion information contained in a video sequence. In recent years, various unsupervised deep learning methods have been proposed to learn

the depth, camera pose, and optical flow jointly from videos (Yin and Shi, 2018; Ranjan *et al.*, 2019). In autonomous driving, as the RGB data and sparse 3D laser scan data are available simultaneously, how to recover a dense depth map from the multi-modal data, which is also called Depth Completion (Uhrig *et al.*, 2017), has attracted interests from researchers recently. Two key points for depth completion include sparse data processing and multi-modal data fusion, while most of previous methods only attempt to solve one of them (Jaritz *et al.*, 2018; Eldesokey *et al.*, 2019; Van Gansbeke *et al.*, 2019).

Different from depth estimation, which mainly takes 2D image or sequence as input, analyzing data containing 3D information, such as 3D point cloud and depth data, might involve multiple selections of input data format. As a result, in many cases, we are required to design novel operations for specific data format to analyze the input data instead of using the standard 2D and 3D convolutions. For instance, for typical RGB-D saliency task, we can directly exploit the 2D convolution (Peng *et al.*, 2014), while for point cloud analysis, an extensively studied problem, we might need to first represent the points in a suitable way and then develop new operations. In detail, in (Hanocka *et al.*, 2019), the mesh representation is adopted and then two novel operations, *i.e.,* mesh convolution and mesh pooling, are proposed to process the irregular triangular meshes. In comparison, voxelization is a more general operation for point cloud representation, and the voxelized points can be processed by the 3D convolution (Maturana and Scherer, 2015) directly or sparse convolution (Choy *et al.*, 2019) which is specially designed for the sparse data. However, due to the low resolution caused by voxelization, the voxel-based approaches might suffer from quantization loss of the structure. To alleviate this issue, another representation, *i.e.,* point sets, is widely adopted. In this way, different functions for associating the adjacent points to represent the local shape are proposed, such as EdgeConv (Wang *et al.*, 2018b), RSConv (Liu *et al.*, 2019d), and KPConv (Thomas *et al.*, 2019).

Whether for deep learning based depth prediction or point cloud understanding, one key issue is the performance drop caused by domain shifts, which occur when the testing target data is sampled from a different distribution to the source training data. According to the specific settings, two important research problems, *i.e.,* domain adaptation (Pan and Yang, 2009) and domain generalization (Blanchard *et al.*, 2011), are defined. In detail, in domain adaptation, we have access to one or multiple source domains with ground truth data and one target domain without ground truth data, while in domain generalization, we only have multiple source domains and no distribution information about the target. The differences between domain adaptation and generalization cause the different solutions to them. For example, in domain adaptation, since the target data is available, we can directly learn a mapping between the source and target domains (Saenko *et al.*, 2010; Sun and Saenko, 2016) or use the self-paced curriculum learning to generate pseudo-labels for model training (Zou *et al.*, 2018). In comparison, when we have no access to the target data, we can address domain generalization through learning domain-invariant features from the source domains (Li *et al.*, 2018d) or exploiting the data augmentation strategy (Xu *et al.*, 2021b). Specific to 3D structure prediction and point cloud data understanding, there are many works studying domain adaptation in depth prediction, 3D object detection, and semantic segmentation. For instance, Atapour *et al.* (Atapour-Abarghouei and Breckon, 2019b) train a monocular depth estimation on synthetic dataset, and use image style transfer task as domain adaptation technique to minimize the domain discrepancy between synthetic and real data. To cope with domain adaptation in stereo matching, Sakuma *et al.* (Sakuma and Konishi, 2021) propose an attention mechanism for the aggregation of features in the left and right views, which is incorporated into an image-to-image translation network for preserving the geometric structure during image translation. For 3D data analysis, the domain shifts often result from the geometric characteristics changes, like point cloud density, object scale, and distance of an object to the sensors. To deal with the issues, Zhang *et al.* (Zhang

et al., 2021a) propose to align the features at multiple scales with the distance information for 3D object detection, while Yi *et al.* (Yi *et al.*, 2021) design a sparse voxel completion network to address the domain gap caused by different 3D point sampling strategy for 3D semantic segmentation. Although domain adaptation techniques in 3D structure prediction and analysis have been studied well, to our best knowledge, domain generalization in these relevant tasks, especially point cloud analysis, is still under-examined.

As we introduced above, 3D information prediction and understanding involve various specific tasks and issues. In this thesis, we take two well-known problems, *i.e.,* depth prediction and point cloud analysis, to show our studies and efforts on deep 3D information prediction and understanding, respectively. In detail, through studying four specific tasks, including monocular depth estimation, depth completion, point cloud representation, and domain generalization, we aims at investigating several crucial issues, *i.e.,* 1) depth estimation from a single image with domain adaptation in a unsupervised learning framework; 2) dense depth recovery from a single RGB image and sparse depth data; 3) local shape representation for point cloud analysis; and 4) domain generalization in 3D shape classification. In the following, we briefly introduce these tasks and review some related works.

1.1.1 Monocular Depth Estimation

Monocular depth estimation is a straightforward way to predict 3D structure from a single image using (deep) learning algorithms. Relying on the powerful capability of modeling the contextual information, DCNNs have dominated the research community in monocular depth estimation. In specific, Eigen *et al.* (Eigen *et al.*, 2014) develop the first deep network for monocular depth estimation, which consists of two components, *i.e.,* a global coarse-scale subnetwork and a local fine-scale sub-network. The global one predicts the overall depth map structure, while the local one aligns the global representation with

local details, such as object and wall edges, to refine the coarse prediction. The whole model is trained on datasets containing labeled pairs of aligned RGB and depth images in an end-to-end fashion using a scale-invariant regression loss, which is adopted to address the scale ambiguous of objects. Following (Eigen *et al.*, 2014), various networks with the encoder-decoder structure have been proposed. For example, in (Laina *et al.*, 2016), the novel up-projection block containing a residual connection (He *et al.*, 2016) is introduced to replace the unpooling operation (Dosovitskiy *et al.*, 2015b) for increasing the spatial resolution of feature maps. In comparison with unpooling, up-projection is beneficial to yielding more accurate depth maps. Considering the continuous nature of the depth values, Liu *et al.* (Liu *et al.*, 2016a) explore the capacity of DCNNs and continuous conditional random field (CRF) jointly in an end-to-end deep network. To model the inherent ordinal correlation of depth values, Fu *et al.* (Fu *et al.*, 2018) consider depth estimation as an ordinal regression problem instead of a regression problem and propose an ordinary regression loss. The proposed model outperforms all previous methods. These methods yield high-performing depth maps, benefiting from the supervision of amounts of ground truth data. However, labelling ground truth depths is both costly and difficult. To reduce the demand of ground-truth training data, various unsupervised monocular depth estimation approaches (Godard *et al.*, 2017; Garg *et al.*, 2016) have been proposed. The main clue they exploit is the epipolar geometry constraint existing in the rectified stereo data. Such strategy only requires rectified stereo pairs without ground truth depths during training, which are easier to collect than the pair of RGB image and depth map. Similar clue can be also found in later works which train unsupervised monocular depth estimation network on video sequences (Zhan *et al.*, 2018; Yin and Shi, 2018).

Another solution to avoiding labelling the depth map for each image is using synthetic images with ground truth depth, which can be acquired from the virtual environment easily, like GTAV [1]. However, due to the domain shifts from synthetic data to real-world data, the model trained on synthetic data generally suffers performance drop on real data. To address this issue, domain adaptation techniques (Pan and Yang, 2009) are exploited through minimizing the domain gap. A typical solution to transferring knowledge from synthetic images with ground truth depth is exploiting the image-to-image translation technique to implement image style transfer and then feeding the transferred images into the depth estimation model (Atapour-Abarghouei and Breckon, 2019b; Zheng *et al.*, 2018). However, the image-to-image translation process often introduces undesirable distortions, which can degrade the performance of successive depth prediction, due to the lack of paired images during the training stage of image translation. To deal with this issue, in this thesis, we propose to explore the labels in the synthetic data and epipolar geometry in the real data jointly, which we prove that can be of benefit to better image style transfer and better depth prediction performance through conducting comprehensive experiments and ablations.

1.1.2 Depth Completion

In fact, estimating depth from a single image is always a very challenging task. Luckily, in some scenarios, we have access to multiple data sources, which can be exploited jointly for depth estimation. For instance, in an autonomous driving system, we can capture the RGB image using the camera, and the 3D laser scan data containing the depth and reflectance information using LiDAR. Projecting the laser scan onto a 2D image plane results in a 2D depth map, which contains sparse depth information. To make the depth data denser, we can integrate the

[1] https://github.com/aitorzip/DeepGTAV

sparse depth map with the dense RGB image data and then train a depth estimation model on them. Such task is called depth completion, which aims to recover a dense depth map from a pair of RGB image and aligned sparse depth map (Uhrig *et al.*, 2017). Considering that the depth data is sparse and irregularly distributed, some of previous works propose new convolution operations specific for sparse depth data processing. For example, Uhrig *et al.* (Uhrig *et al.*, 2017) propose a sparsity-invariant convolution, which, in comparison with the standard convolution, evaluates only pixels with depth values by exploiting binary validity masks. Instead of using the binary masks, Eldesokey *et al.* (Eldesokey *et al.*, 2019) propose an algebraically-constrained normalized convolution, where learned confidence maps with values ranging from 0 to 1 are used to normalize the feature maps. Another way to exploit the sparsity is propagating the depths directly. For example, Cheng *et al.* (Cheng *et al.*, 2018) propose to learn an affinity matrix for spatial depth propagation, while Park *et al.* (Park *et al.*, 2020), inspired by deformable convolution (Dai *et al.*, 2017b), propose to propagate the depth non-locally by learning the locations of the neighbours dynamically. Besides, there are some works which do not consider the sparsity specially and instead focus more on the multi-modal fusion or geometric constraints. For instance, Jaritz *et al.* (Jaritz *et al.*, 2018) study two fusion strategy, namely early fusion (concatenate the input maps) and late fusion (concatenate the intermediate feature maps), and the experiments show that the later one performs better. Gansbeke *et al.* (Van Gansbeke *et al.*, 2019) design two sub-networks for global and local information extraction and exploit the RGB image as guidance to help the local branch detect the noises in the LiDAR data. To utilize the 3D geometry information to regularize the depth completion, both Xu *et al.* (Xu *et al.*, 2019) and Qiu *et al.* (Qiu *et al.*, 2019) associate the surface normal information with the depth information within a sub-network.

Previous works mainly exploit the convolution with fixed-size kernel to process the sparse depth map, which cannot utilize the observed contextual information effectively. In addition, these works usually consider either sparsity or multi-modal fusion, while not both of them. In this thesis, we exploit the graph propagation strategy to capture the multi-modal contexts adaptively and further propose a symmetric gated fusion strategy for better multi-modal fusion.

1.1.3 Point Cloud Processing

Point cloud generally refers to a point set containing N unordered 3D points $\{(p_i, f_i)\}_{i=1}^{N}$, where f_i denotes the feature vector of point $p_i \in \mathbb{R}^3$. Point cloud processing aims to learn a local representation for each sampled point, which can then be used for various tasks, such as classification (Wu *et al.*, 2015), semantic segmentation (Armeni *et al.*, 2016), and point cloud completion (Tchapmi *et al.*, 2019). Due to the sparse, irregular and unordered structure of point cloud, it is difficult to directly apply the standard 2D and 3D convolutions, which have been widely applied on 2D image data, for point cloud processing. To tackle this issue, current works mainly design deep networks from two perspectives, namely generic operation and concrete task. In detail, some works aim to design a novel basic operation to represent the local shape effectively and the operation can be stacked into a deep network for classification and segmentation like the 2D convolution. In comparison, some other works do not develop new basic operations and instead they focus more on task-relevant learning strategy, such as the boundary information modeling for semantic segmentation (Gong *et al.*, 2021), candidate generation for instance segmentation (Jiang *et al.*, 2020b; Jiang *et al.*, 2020a), and self-supervised learning for unlabelled point cloud data (Sauder and Sievers, 2019). In this thesis, we follow the former route, *i.e.,* developing a new basic operation for local shape representation.

Although it is challenging to apply the standard convolution into the point cloud data directly, we can achieve it by voxelizing the points. The volumetric representation can then be fed into the conventional 3D CNNs (Maturana and Scherer, 2015). In fact, as the volumetric representation is still sparse, to reduce computational costs and memory requirements, sparse convolutions (Choy *et al.*, 2019) are proposed as the basic operation, where only the occupied voxels are calculated. To avoid voxelization, which causes low resolution and then the loss of the structure information, various works aiming at learning representations from the raw point cloud directly have been proposed. For this solution, the basic operation is required to be permutation-invariant, since the points are unordered. PointNet (Qi *et al.*, 2017a), the first attempt to this clue, utilize the MLP as the basic operation to process each point and then use the max-pooling operation to get the global representation. Following this work, lots of approaches are proposed to improve the basic local shape representation operation through, such as, considering the local structure (Qi *et al.*, 2017b), modeling the relation between adjacent points (Liu *et al.*, 2019d; Wang *et al.*, 2018b), exploiting robust sampling strategy (Yan *et al.*, 2020; Yang *et al.*, 2019a), adopting the attention mechanism (Wang *et al.*, 2019b), or introducing kernel points (Thomas *et al.*, 2019; Xu *et al.*, 2021a). A common operation in most of these works is modeling the point-to-point relation, which is used to associate the features of adjacent points. However, they often model the relation for each pair of adjacent points solely, which might make the learned representation for the edge lack the local structure information and thus not robust and discriminative. To alleviate this issue, we propose a novel module to model the edge-to-edge interaction, which can enhance the point-to-point relation and then improve the local structure representation. Experiments on several public datasets demonstrate the effectiveness of our methods.

1.1.4 Domain Adaptation and Generalization

An ideal situation for deep learning is the training data and the testing data come from the same distribution. However, this condition might not hold true in many cases, *i.e.,* the training and testing data are sampled from different distributions respectively. Due to dataset bias (Gretton *et al.*, 2009), a typical model trained on the training (source) data often fails to generalize well to the testing (target) data, and domain adaptation aims to address such an issue. We can achieve domain adaptation through learning a domain-invariant feature representation (Ganin and Lempitsky, 2015; Ganin *et al.*, 2016) or learning a mapping between the source and target domains (Gong *et al.*, 2012; Saenko *et al.*, 2010). A typical solution to learning domain-invariant representations is introducing a gradient reversal layer and minimizing the domain gap between source and target domains in an adversarial way (Ganin *et al.*, 2016). Domain mapping aims to transfer the data in one domain to a space where the source and target domains have similar distributions. For example, we can use image-to-image translation technique (Zhu *et al.*, 2017) to make the images in source domain have the same style to target domain. Domain shift is a common issue existing in computer vision. As a result, domain adaptation techniques have been studied for various tasks, such as segmentation (Li *et al.*, 2020b; Zhang *et al.*, 2017; ZHANG *et al.*, 2019a), object detection (Chen *et al.*, 2018; Khodabandeh *et al.*, 2019), and re-identification (Deng *et al.*, 2018; Bak *et al.*, 2018). In this thesis, we study the domain adaptation problem in a well-known 3D structure prediction task, *i.e.,* monocular depth estimation, by exploring the geometric structure of natural images.

In comparison with domain adaptation, domain generalization is more challenging, due to the lack of target data. In detail, in the setting of domain generalization, we only have multiple source domains available but have no access to the target domain, and the model trained on the source domains is required to generalize well to the target. Since multiple source domains are available, a

classic solution for domain generalization is learning a domain-invariant feature representation across source domains (Li *et al.*, 2018d; Muandet *et al.*, 2013). For example, Li *et al.* (Li *et al.*, 2018d) exploit the adversarial learning to minimize the domain gap across the source domains for each category. Recently, another effective solution, *i.e.,* data augmentation, has been widely studied in many works (Zhou *et al.*, 2020a; Xu *et al.*, 2021b; Zhou *et al.*, 2020b). For instance, Xu *et al.* (Xu *et al.*, 2021b) propose a Fourier-based data augmentation strategy, motivated by the property of the Fourier transformation that the phase component of Fourier spectrum contains the high-level semantic information and the amplitude component contains the low-level information. By using the MixUp strategy (Zhang *et al.*, 2018a) to perturb the amplitude information in the original images, new images are generated, which are then used to train the model together with the original data. 2D object classification task is commonly used to evaluate the generalization capability in the domain generalization literature, while recently some works have studied the domain generalization in more complex tasks, such as semantic segmentation (Yue *et al.*, 2019; Choi *et al.*, 2021) and re-identification (Zhao *et al.*, 2021). Domain shifts also exist in 3D data, like point cloud data, and domain adaption on 3D point cloud has been studied in many works (Yi *et al.*, 2021; Achituve *et al.*, 2021). However, to the best of my knowledge, there is no work studying the typical domain generalization problem for point cloud tasks, like shape classification. To provide an initial exploration to this problem, in this thesis, we study the domain generalization problem in 3D shape classification. Considering the dependency between the learned features and the category, we propose an entropy-regularization approach, which ensures the conditional invariance of learned features and then improves the domain generalization capabilities. We validate the effectiveness of our method on both 3D and 2D object classification datasets.

1.2 Outline

In the first chapter, we first present the problem of deep 3D information prediction and understanding, then introduce four specific tasks, including monocular depth estimation, depth completion, point cloud analysis, and domain generalization. Through studying these tasks, we investigate several crucial issues, such as multi-modal fusion, unsupervised learning, and model generalization, in deep 3D information prediction and understanding. The reminder of this thesis is organized as five chapters, and the outline is as follows:

- **Chapter 2** This chapter studies the domain adaptation technique and unsupervised learning in 3D structure prediction by examining the well-known monocular depth estimation task. We study how to explore the geometric structure of natural images to improve the performance of image-to-image translation and depth estimation model.

- **Chapter 3** This chapter studies the sparse data representation and multi-modal fusion in 3D structure prediction by investigating the depth completion task. We present how to exploit graph propagation strategy to capture rich contextual information for the sparse data, and introduce an effective strategy for multi-modal fusion.

- **Chapter 4** This chapter studies how to design novel operations for sparse data representation in understanding data containing 3D information by exploring the relation learning in point cloud analysis. We propose a novel edge-to-edge interactive learning module to enhance the point-to-point relation for point cloud processing.

- **Chapter 5** This chapter studies the challenging domain generalization problem in processing data containing 3D information. We investigate the naive adversarial training for domain-invariant representations, and propose an entropy regularization approach to guarantee the conditional invariance of learned features. The method is evaluated on both 3D and 2D object classification datasets.

- **Chapter 6** This chapter concludes our thesis and suggests some future research possibilities.

1.3 Contributions

The main contributions of the thesis are summarized as follows:

- In Chapter 2, we propose an end-to-end domain adaptation framework for unsupervised monocular depth estimation. We explore the labels in the synthetic data and epipolar geometry in the real data jointly to preserve the geometric structure during image translation. By conducting experiments, we show that training the monocular depth estimator using ground truth depth in the synthetic domain coupled with the epipolar geometry in the real domain can boost the performance. We demonstrate the effectiveness of our method on KITTI dataset (Menze and Geiger, 2015) and the generalization performance on Make3D dataset (Saxena *et al.*, 2009).

- In Chapter 3, we introduce the proposed co-attention guided graph propagation for depth completion, which is adaptive to the sparsity patterns of sparse depth input and thus enables the unobserved pixels to capture useful observed contextual information more effectively. To fuse the multi-modal contextual information better, we further present

the symmetric gated fusion strategy, which can learn the heterogeneity of the two modalities adaptively. We demonstrate the effectiveness of our model on two benchmarks, *i.e.,* KITTI Depth Completion dataset (Geiger *et al*., 2012a) and NYU-v2 dataset (Silberman *et al*., 2012).

- In Chapter 4, we propose an adaptive edge-to-edge interaction learning module for point cloud analysis, *i.e.,* AE^{2}IL, which is able to enhance the learned point-to-point relation and makes it more aware of the local structure. We further extend the AE^{2}IL to a symmetric version, namely SymAE^{2}IL, for better capturing the local shape information. Then, we exploit the proposed modules to design models for point cloud classification and segmentation. We conduct experiments on several public point cloud datasets, and the results show that our methods outperform previous approaches and achieve state-of-the-art performance.

- In Chapter 5, we first revisit the typical domain-invariant feature representation learning methods for domain generalization, and then argue that the naive adversarial training can only guarantee the invariant marginal distribution across source domains. To improve the domain generalization capability, we propose an entropy regularization term to ensure the conditional invariance. Together with the adversarial training on the marginal distribution, our method achieves better generalization capabilities in both 3D shape classification task and 2D object recognition task.

CHAPTER 2

Geometry-Aware Symmetric Domain Adaptation for Monocular Depth Estimation

---

Supervised depth estimation has achieved high accuracy due to the advanced deep network architectures. Since the groundtruth depth labels are hard to obtain, recent methods try to learn depth estimation networks in an unsupervised way by exploring unsupervised cues, which are effective but less reliable than true labels. An emerging way to resolve this dilemma is to transfer knowledge from synthetic images with ground truth depth via domain adaptation techniques. However, these approaches overlook specific geometric structure of the natural images in the target domain (i.e., real data), which is important for high-performing depth prediction. Motivated by the observation, we propose a geometry-aware symmetric domain adaptation framework (GASDA) to explore the labels in the synthetic data and epipolar geometry in the real data jointly. Moreover, by training two image style translators and depth estimators symmetrically in an end-to-end network, our model achieves better image style transfer and generates high-quality depth maps. The experimental results demonstrate the effectiveness of our proposed method and comparable performance against the state-of-the-art.

2.1 Introduction

Monocular depth estimation (Saxena *et al.*, 2006; Saxena *et al.*, 2009; Eigen *et al.*, 2014; Ladicky *et al.*, 2014) has been an active research area in the field

of computer vision. Recent years have witnessed the great strides in this task, especially after DCNNs were exploited to estimate depth from a single image successfully (Eigen *et al.*, 2014). Until now, there have been lots of follow-up works (Liu *et al.*, 2016b; Laina *et al.*, 2016; Eigen and Fergus, 2015; Li *et al.*, 2015; Xu *et al.*, 2017; Wang *et al.*, 2015; Fu *et al.*, 2018) improving or extending this work. However, since the proposed deep models are trained in a fully supervised fashion, they require a large amount of data with ground truth depth, which is expensive to acquire in practice. To address this issue, unsupervised monocular depth estimation has been proposed (Godard *et al.*, 2017; Zhan *et al.*, 2018; Garg *et al.*, 2016; Xie *et al.*, 2016), using geometry-based cues and without the need of image-depth pairs during training. Unfortunately, this kind of method tends to be vulnerable to illumination change, occlusion and blurring and so on. Compared to real-world data, synthetic data is much easier to obtain the depth map. As a result, some works propose to exploit synthetic data for visual tasks (Lai *et al.*, 2017; Long *et al.*, 2013; Dosovitskiy *et al.*, 2015a). However, due to domain shift from synthetic to real, the model trained on synthetic data often fails to perform well on real data. To deal with this issue, domain adaptation techniques are utilized to reduce the discrepancy between datasets/domains [1] (Atapour-Abarghouei and Breckon, 2018; Chen *et al.*, 2018; Long *et al.*, 2013).

Existing works (Atapour-Abarghouei and Breckon, 2018; Kundu *et al.*, 2018; Zheng *et al.*, 2018) using synthetic data via domain adaptation have achieved impressive performance for monocular depth estimation. These approaches typically perform domain adaptation either based on synthetic-to-realistic translation or inversely. However, due to the lack of paired images, the image translation function usually introduces undesirable distortions in addition to the style change. The distorted image structures significantly degrade the performance of

[1] We will use *domain* and *dataset* interchangeably for the same meaning in most cases of this chapter.

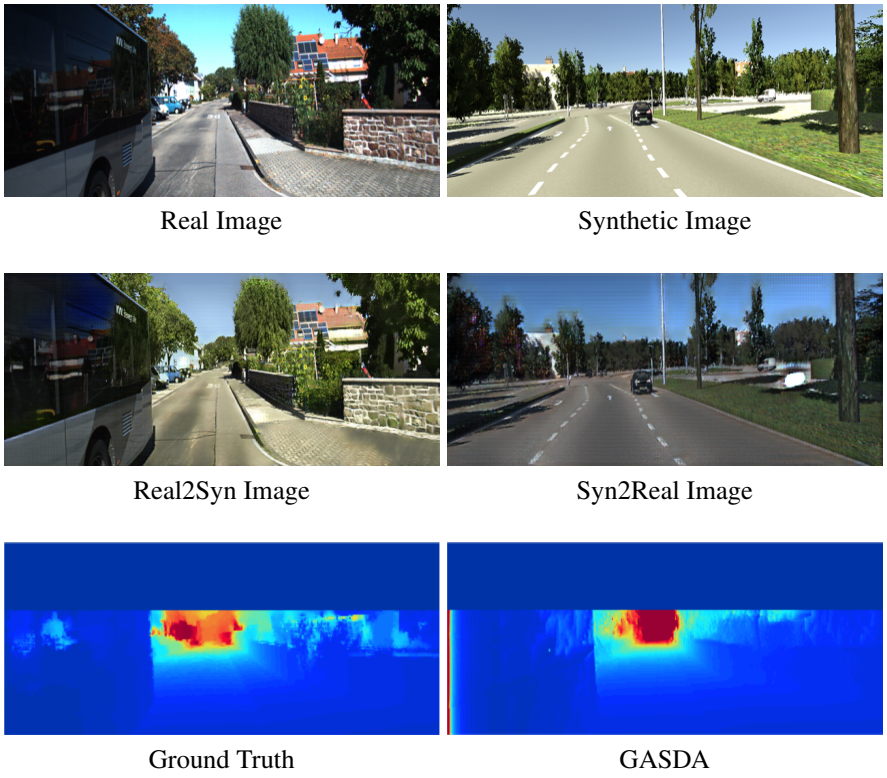


Figure 2.1. Estimated Depth by GASDA. Top to bottom: input real image in the target domain (KITTI dataset (Menze and Geiger, 2015)) and synthetic image for training (vKITTI dataset (Gaidon *et al.*, 2016)), intermediate generated images in our approach, ground truth depth map and estimated depth map using proposed GASDA.

successive depth prediction. Fortunately, the unsupervised cues in the real images, for example, stereo pairs, produces additional constraints on the possible depth predictions. Therefore, it is essential to simultaneously explore both synthetic and real images and the corresponding depth cues for generating higher-quality depth maps.

Motivated by the above analysis, we propose a **Geometry-Aware Symmetric Domain Adaptation Network (GASDA)** for unsupervised monocular depth estimation. This framework consists of two main parts, namely symmetric style translation and monocular depth estimation. Inspired by CycleGAN (Zhu *et al.*, 2017), our GASDA employs both synthetic-to-realistic and realistic-to-synthetic

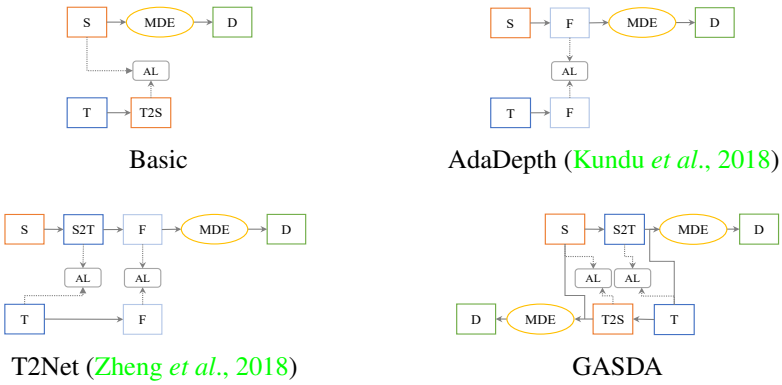


Figure 2.2. Different frameworks for monocular depth estimation using domain adaptation. First row to second row: basic pipeline, approach proposed in (Kundu *et al.*, 2018), (Zheng *et al.*, 2018) and this work, respectively. S, T, F, S2T (T2S) and D represent the synthetic data, real data, extracted feature, generated data, and estimated depth. AL and MDE mean adversarial loss and monocular depth estimation, respectively. Compared with existing methods, our approach utilizes real stereo data and takes into account synthetic-to-real as well as real-to-synthetic during translation.

translations coupled with a geometry consistency loss based on the epipolar geomery of the real stereo images. Our network is learned by groundtruth labels from the synthetic domain as well as the epipolar geometry of the real domain. Additionally, the learning process in the real and synthetic domains can be regularized by enforcing consistency on the depth predictions. By training the style translation and depth prediction networks in an end-to-end fashion, our model is able to translate images without distorting the geometric and semantic content, and thus achieves better depth prediction performance.

2.2 Related Work

Monocular Depth Estimation has been intensively studied over the past decade due to its crucial role in 3D scene understanding. Typical approaches sought the solution by exploiting probabilistic graphical models (*e.g.*, MRFs) (Saxena *et al.*, 2009; Saxena *et al.*, 2006; Liu *et al.*, 2010), and non-parametric techniques (Liu *et al.*, 2014; Karsch *et al.*, 2014; Liu *et al.*, 2011). However, these

methods showed some limitations in performance and efficiency because of the employment of hand-crafted features and the low inference speed.

Recent studies demonstrated that high-performing depth estimators can be obtained relying on DCNNs (Eigen *et al.*, 2014; Liu *et al.*, 2016b; He *et al.*, 2018b; Xu *et al.*, 2018a; Repala and Dubey, 2018; Qi *et al.*, 2018; Cao *et al.*, 2016; Laina *et al.*, 2016; Roy and Todorovic, 2016; Chen *et al.*, 2016). Eigen *et al.* (Eigen *et al.*, 2014) developed the first end-to-end deep model for depth estimation, which consists of a coarse-scale network and a fine-scale network. To exploit the relationships among image features, Liu *et al.* (Liu *et al.*, 2016b) proposed to integrate continuous CRFs with DCNNs at super-pixel level. While previous works considered depth estimation as a regression task, Fu *et al.* (Fu *et al.*, 2018) solved depth estimation in the discrete paradigm by proposing an ordinal regression loss to encourage the ordinal competition among depth values.

A weakness of supervised depth estimation is the heavy requirement of annotated training images. To mitigate the issue, several notable attempts have investigated depth estimation in an unsupervised manner by means of stereo correspondence. Xie *et al.* (Xie *et al.*, 2016) proposed the Deep3D network for 2D-to-3D conversion by minimizing the pixel-wise reconstruction error. This work motivated the development of subsequent unsupervised depth estimation networks (Garg *et al.*, 2016; Godard *et al.*, 2017; Yin and Shi, 2018; Zhou *et al.*, 2017). In specific, Garg *et al.* (Garg *et al.*, 2016) showed that unsupervised depth estimation could be recast as an image reconstruction problem according to the epipolar geometry. Following Garg *et al.* (Garg *et al.*, 2016), several later works improved the structure by exploiting left-right consistency (Godard *et al.*, 2017), learning depth in a semi-supervised way (Kuznietsov *et al.*, 2017), and introducing temporal photometric constraints (Zhan *et al.*, 2018).

Domain Adaptation (Pan *et al.*, 2010) aims to address the problem that the model trained on one dataset fails to generalize to another due to *dataset bias* (Torralba and Efros, 2011). In this community, previous works either learn the domain-invariant representations on a feature space (Ganin and Lempitsky, 2015; Ganin *et al.*, 2016; Long *et al.*, 2013; Ajakan *et al.*, 2014; Gong *et al.*, 2016; Gong *et al.*, 2018; Li *et al.*, 2018d) or learn a mapping between the source and target domains at feature or pixel level (Saenko *et al.*, 2010; Sun and Saenko, 2016; Gong *et al.*, 2012; Zhang *et al.*, 2013). For example, Long *et al.* (Long *et al.*, 2013) aligned feature distribution across the source and target domains by minimizing a Maximum Mean Discrepancy (MMD) (Gretton *et al.*, 2012). Tzeng *et al.* (Tzeng *et al.*, 2014) proposed to minimize MMD and the classification error jointly in a DCNN framework. Sun *et al.* (Sun and Saenko, 2016) proposed to match the mean and covariance of the two domain's deep features using the Correlation Alignment (CORAL) loss (Sun *et al.*, 2016).

Coming to domain adaptation for depth estimation, Atapour *et al.* (Atapour-Abarghouei and Breckon, 2018) developed a two-stage framework. In specific, they first learned a translator to stylize the natural images so as to make them indistinguishable with the synthetic images, and then trained a depth estimation network using the original synthetic images in a supervised manner. Kundu *et al.* (Kundu *et al.*, 2018) proposed a content congruent regularization method to tackle the model collapse issue caused by domain adaptation in high dimensional feature space. Recently, Zheng *et al.* (Zheng *et al.*, 2018) developed an end-to-end adaptation network, *i.e.* $\mathrm{T}^2\mathrm{Net}$, where the translation network and the depth estimation network are optimized jointly so that they can improve each other. However, these works overlooked the geometric structure of the natural images from the target domain, which has been demonstrated significant for depth estimation (Godard *et al.*, 2017; Garg *et al.*, 2016). Motivated by the observation, we propose a novel geometry-aware symmetric domain adaptation network, *i.e.*, GASDA, by exploiting the epipolar geometry of the stereo

images. The differences between GASDA and previous depth adaptation approaches (Kundu *et al.*, 2018; Zheng *et al.*, 2018) are shown in Figure 2.2.

2.3 Method

2.3.1 Method Overview

Given a set of N synthetic image-depth pairs $\{(x_s^i, y_s^i)\}_{i=1}^N$ (*i.e.*, source domain X_s), our goal here is to learn a monocular depth estimation model which can accurately predict depth for natural images contained in X_t (*i.e.*, target domain). It is difficult to guarantee the model generalize well to the real data (Atapour-Abarghouei and Breckon, 2018; Zheng *et al.*, 2018) due to the domain shift. We thus provide a remedy by exploiting the epipolar geometry between stereo images and developing a geometry-aware symmetric domain adaptation network (GASDA). Our GASDA consists of two main parts like existing works, including the style transfer network and the monocular depth estimation network.

Specifically, unlike (Atapour-Abarghouei and Breckon, 2018; Zheng *et al.*, 2018; Kundu *et al.*, 2018), we consider both synthetic-to-real (Zheng *et al.*, 2018) and real-to-synthetic translations (Atapour-Abarghouei and Breckon, 2018; Kundu *et al.*, 2018). As a result, we can train two depth estimators F_s and F_t on the original synthetic data (X_s) and the generated realistic data ($G_{s2t}(X_s)$) using the generator G_{s2t} in supervised manners, respectively. These two models are complementary, since F_s has clean training set X_s but dirty test set $G_{t2s}(X_t)$ generated by the generator G_{t2s} with noises, such as distortion and blurs, caused by unsatisfied translation, and vise verse for F_t. Nevertheless, because the depth information is rather relevant to specific scene geometry which might be different between source and target domains, the models trained on X_s or $G_{s2t}(X_s)$ still could fail to perform well on $G_{t2s}(X_t)$ or X_t. To provide a solution, we exploit the epipolar geometry of real stereo pairs $\{(x_{t_l}^i, x_{t_r}^i)\}_{i=1}^M$ ($x_{t_l}^i$ and $x_{t_r}^i$ represent

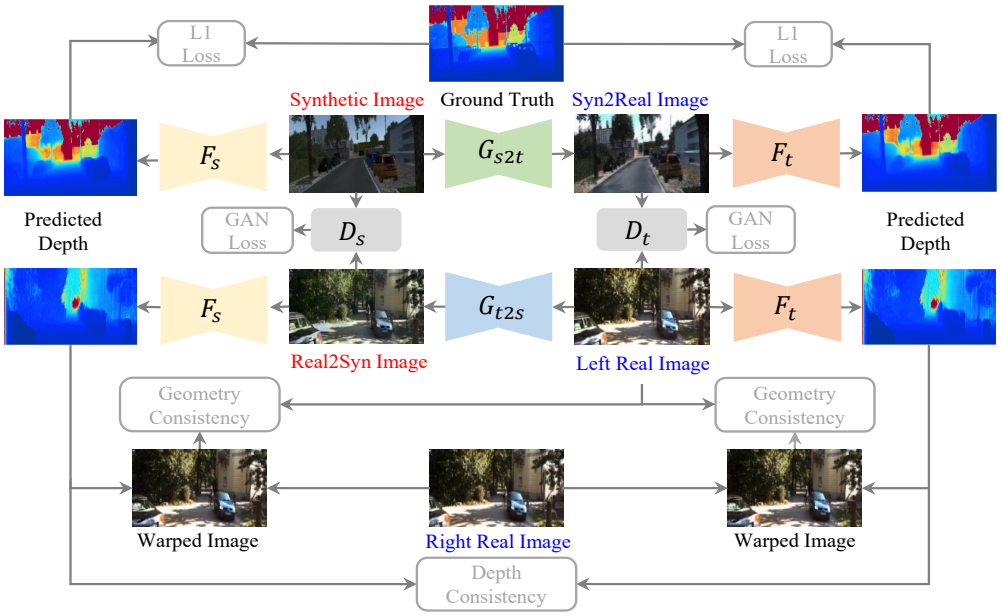


Figure 2.3. The proposed framework. It consists of two main parts: image style translation and monocular depth estimation. i) Style translation network, incorporating two generators (*i.e.*, G_{s2t} and G_{t2s}) and two discriminators (*i.e.*, D_t and D_s), is based on CycleGAN. ii) Monocular depth estimation network contains two complementary subnetworks (*i.e.*, F_s and F_t). We omit the side outputs, for brevity. More details can be found in Section 2.3, Section 2.4.1.

the left and right image respectively[2]) during training to encourage F_t and F_s to capture the relevant geometric structure of target/real data. In addition, we introduce an additional depth consistency loss to enforce the predictions from F_t and F_s are consistent in local regions. The overall framework of GASDA is illustrated in Figure 2.3. For simplicity, we will omit the superscript i in most cases.

2.3.2 GASDA

Bidirectional Style Transfer Loss Our goal here is to learn the bidirectional translators G_{s2t} and G_{t2s} to bridge the gap between the source domain (synthetic) X_s and the target domain (real) X_t. Specifically, taking G_{s2t} as an example, we expect the $G_{s2t}(x_s)$ to be indistinguishable from real images in X_t. We

[2]We will omit the subscript l of t_l for the left image in most cases of this chapter.

thus employ a discriminator D_t, and train G_{s2t} and D_t in an adversarial fashion by performing a minimax game following (Goodfellow *et al.*, 2014). The adversarial losses are expressed as:

$$\begin{aligned}\mathcal{L}_{gan}(G_{s2t}, D_t, X_t, X_s) =& \mathbb{E}_{x_t \sim X_t}[D_t(x_t) - 1] + \\ & \mathbb{E}_{x_s \sim X_s}[D_t(G_{s2t}(x_s))], \\ \mathcal{L}_{gan}(G_{t2s}, D_s, X_t, X_s) =& \mathbb{E}_{x_s \sim X_s}[D_s(x_s) - 1] + \\ & \mathbb{E}_{x_t \sim X_t}[D_s(G_{t2s}(x_t))].\end{aligned} \tag{2.1}$$

Unluckily, the vanilla GANs suffer from mode collapse. To provide a remedy and ensure the input images and the output images paired up in a meaningful way, we utilize the cycle-consistency loss (Zhu *et al.*, 2017). Specifically, when feeding an image x_s to G_{s2t} and G_{t2s} orderly, the output should be a reconstruction of x_s, and vice versa for x_t, *i.e.* $G_{t2s}(G_{s2t}(x_s)) \approx x_s$ and $G_{s2t}(G_{t2s}(x_t)) \approx x_t$. The cycle consistency loss has the form as:

$$\begin{aligned}\mathcal{L}_{cyc}(G_{t2s}, G_{s2t}) = & \mathbb{E}_{x_s \sim X_s}[||G_{t2s}(G_{s2t}(x_s)) - x_s||_1] \\ & + \mathbb{E}_{x_t \sim X_t}[||G_{s2t}(G_{t2s}(x_t)) - x_t||_1].\end{aligned} \tag{2.2}$$

Apart from the adversarial loss and cycle consistency loss, we also employ an identity mapping loss (Taigman *et al.*, 2016) to encourage the generators to preserve geometric content. The identity mapping loss is given by:

$$\begin{aligned}\mathcal{L}_{idt}(G_{t2s}, G_{s2t}, X_s, X_t) = & \mathbb{E}_{x_s \sim X_s}[||G_{t2s}(x_s) - x_s||_1] \\ & + \mathbb{E}_{x_t \sim X_t}[||G_{s2t}(x_t) - x_t||_1].\end{aligned} \tag{2.3}$$

The full objective for the bidirectional style transfer is as follow:

$$\begin{aligned}\mathcal{L}_{trans}(G_{t2s}, G_{s2t}, D_t, D_s) = &\mathcal{L}_{gan}(G_{s2t}, D_t, X_t, X_s) \\ &+ \mathcal{L}_{gan}(G_{t2s}, D_s, X_t, X_s) \\ &+ \lambda_1 \mathcal{L}_{cyc}(G_{t2s}, G_{s2t}) \\ &+ \lambda_2 \mathcal{L}_{idt}(G_{t2s}, G_{s2t}, X_t, X_s)\end{aligned} \tag{2.4}$$

where λ_1 and λ_2 are the trade-off parameters.

Depth Estimation Loss We can now render the synthetic images to the "style" of the target domain (KITTI), and then capture a new dataset $X_{s2t} = G_{s2t}(X_s)$. We train a depth estimation network F_t on X_{s2t} in a supervised manner using the provided ground truth depth in the synthetic domain X_s. Here, we minimize the ℓ_1 distance between the predicted depth $\tilde{y}_{ts}$ and ground truth depth y_s:

$$\mathcal{L}_{tde}(F_t, G_{s2t}) = ||y_s - \tilde{y}_{ts}||. \tag{2.5}$$

In addition to F_t, we also train a complementary depth estimator F_s on X_s directly with the ℓ_1 loss:

$$\mathcal{L}_{sde}(F_s) = ||y_s - \tilde{y}_{ss}|| \tag{2.6}$$

where $\tilde{y}_{ss} = F_s(x_s)$ is the output of F_s. Both the F_s and F_t are important backbones to alleviate the issue of geometry and semantic inconsistency coupled with the subsequent losses. The full depth estimation loss is expressed as:

$$\mathcal{L}_{de}(F_t, F_s, G_{s2t}) = \mathcal{L}_{sde}(F_s) + \mathcal{L}_{tde}(F_t, G_{s2t}). \tag{2.7}$$

Geometry Consistency Loss Combining the components above, we have already formulated a naive depth adversarial adaptation framework. However, the G_{s2t} and G_{t2s} are usually imperfect, which would make the predictions $\tilde{y}_{st} = F_s(G_{t2s}(x_t))$ and $\tilde{y}_{tt} = F_t(x_t)$ unsatisfied. Besides, previous depth adaptation approaches overlook the specific physical geometric structure which may

vary from scenes/datasets. Our main objective is to accurately estimate depth for real scenes, so we consider the geometric structure of the target data in the training phase. To this end, we present a geometric constraint on F_t and F_s by exploiting the epipolar geometry of real stereo images and unsupervised cues. Specifically, we generate an inverse warped image from the right image using the predicted depth, to reconstruct the left. We thus combine an ℓ_1 with single scale SSIM (Wang *et al.*, 2004) term as the geometry consistency loss to align the stereo images:

$$\begin{aligned}
\mathcal{L}_{tgc}(F_t) &= \eta\frac{1-SSIM(x_t, x_{tt}^{'})}{2} + \mu||x_t - x_{tt}^{'}||, \\
\mathcal{L}_{sgc}(F_s, G_{t2s}) &= \eta\frac{1-SSIM(x_t, x_{st}^{'})}{2} + \mu||x_t - x_{st}^{'}||, \\
\mathcal{L}_{gc}(F_t, F_s, G_{t2s}) &= \mathcal{L}_{tgc}(F_t) + \mathcal{L}_{sgc}(F_s, G_{t2s})
\end{aligned} \tag{2.8}$$

where $\mathcal{L}_{gc}$ represents the full geometry consistency loss, $\mathcal{L}_{tgc}$ and $\mathcal{L}_{sgc}$ denote the geometry consistency loss of F_t and F_s respectively. $x_{tt}^{'}$ ($x_{st}^{'}$) is the inverse warp of x_{t_r} using bilinear sampling (Jaderberg *et al.*, 2015) based on the estimated depth map y_{tt} (y_{st}), the baseline distance between the cameras and the camera focal length (Godard *et al.*, 2017). In our experiments, η is set to be 0.85, and μ is 0.15.

Depth Smoothness Loss To encourage depths to be consistent in local homogeneous regions, we exploit an edge-aware depth smoothness loss:

$$\mathcal{L}_{ds}(F_t, F_s, G_{t2s}) = e^{-\nabla x_t}||\nabla\tilde{y}_{tt}|| + e^{-\nabla x_t}||\nabla\tilde{y}_{st}|| \tag{2.9}$$

where ∇ is the first derivative along spatial directions. We only apply the smoothness loss to X_t and X_{t2s} (real data), since X_s and X_{s2t} (synthetic data) have full supervision.

Depth Consistency Loss We find that the predictions for x_t, *i.e.*, $F_t(x_t)$ and $F_s(G_{t2s}(x_t))$, show inconsistency in many regions, which is in contrast to our intuition. One of the possible reason is that G_{t2s} might fail to translate x_t with details. To enforce such coherence, we introduce an ℓ_1 depth consistency loss

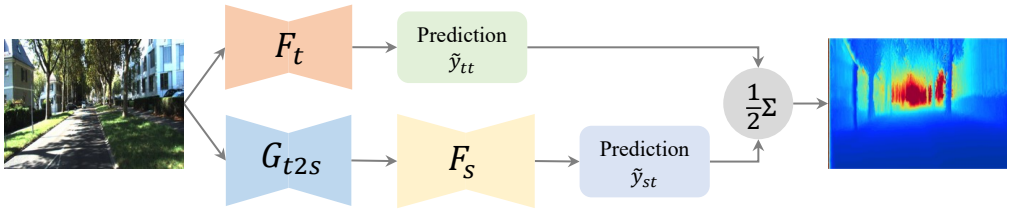


Figure 2.4. Inference Phase (Section 2.3.3).

with respect to $\tilde{y}_{tt}$ and $\tilde{y}_{st}$ as follows:

$$\mathcal{L}_{dc}(F_t, F_s, G_{t2s}) = ||\tilde{y}_{tt} - \tilde{y}_{st}||. \tag{2.10}$$

Full Objective Our final loss function has the form as:

$$\begin{aligned}
&\mathcal{L}(G_{s2t}, G_{t2s}, D_t, D_s, F_t, F_s) \\
&= \mathcal{L}_{trans}(G_{s2t}, G_{t2s}, D_t, D_s) + \gamma_1 \mathcal{L}_{de}(F_t, F_s, G_{s2t}) \\
&+ \gamma_2 \mathcal{L}_{gc}(F_t, F_s, G_{t2s}) + \gamma_3 \mathcal{L}_{dc}(F_t, F_s, G_{t2s}) \\
&+ \gamma_4 \mathcal{L}_{ds}(F_t, F_s, G_{t2s})
\end{aligned} \tag{2.11}$$

where $\gamma_n (n \in \{1, 2, 3, 4\})$ are trade-off factors. We optimize this objective function in an end-to-end deep network.

2.3.3 Inference

In the inference phase, we aim to predict the depth map for a given image in real domain (*e.g.* KITTI dataset (Menze and Geiger, 2015)) using the resultant models. In fact, there are two paths acquiring predicted depth maps: $x_t \rightarrow F_t(x_t) \rightarrow \tilde{y}_{tt}$ and $x_t \rightarrow G_{t2s}(x_t) \rightarrow x_{t2s} \rightarrow F_s(x_{t2s}) \rightarrow \tilde{y}_{st}$, as shown in Figure 2.4, and the final prediction is the average of $\tilde{y}_{tt}$ and $\tilde{y}_{st}$:

$$\tilde{y}_t = \frac{1}{2}(\tilde{y}_{tt} + \tilde{y}_{st}). \tag{2.12}$$

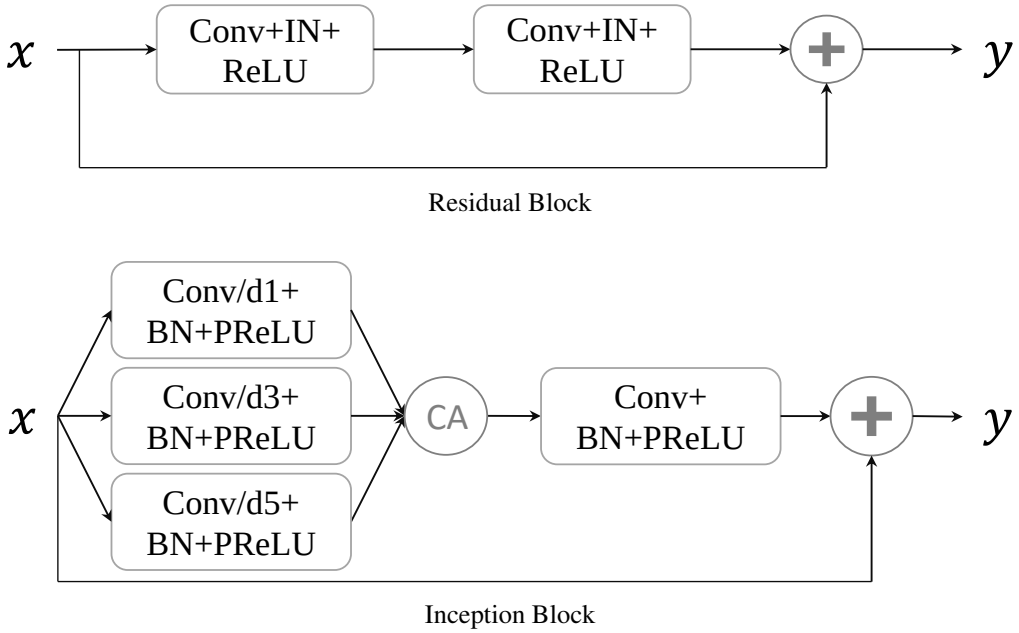


Figure 2.5. All the convolution operations in each block are with the same feature channels, kernel size and stride size, as presented in Table 2.2 and Table 2.1. Conv/dn denotes the n-dilated convolution operation (Yu and Koltun, 2015), and CA represents the concatenation operation.

2.4 Experiments

In this section, we first present the details about our network architecture and the learning strategy. Then, we perform GASDA on one of the largest dataset in the context of autonomous driving, *i.e.*, KITTI dataset (Menze and Geiger, 2015). We also demonstrate the generalization capabilities of our model to other real-world scenes contained in Make3D (Saxena *et al.*, 2009). Finally, we conduct various ablations to analyze GASDA.

2.4.1 Implementation Details

Network Architecture Our proposed framework consists of six sub-networks, which can be divided into three groups: G_{s2t} and G_{t2s} for image style translation, D_t and D_s for discrimination, F_t and F_s for monocular depth estimation.

Depth Estimator F_t/F_s					
Incex	Layer	Input	FC	KS	SS
1	Conv+BN+PReLU	Image	64	7	1
2	Pooling	1	64	2	2
3	Conv+BN+PReLU	2	128	3	1
4	Conv+BN+PReLU	3	128	3	1
5	Pooling	4	128	2	2
6	Conv+BN+PReLU	5	256	3	1
7	Conv+BN+PReLU	6	256	3	1
8	Pooling	7	256	2	2
9	Conv+BN+PReLU	8	256	3	1
10	Conv+BN+PReLU	9	256	3	1
11	Pooling	10	256	2	2
12	Conv+BN+PReLU	11	512	3	1
13	Conv+BN+PReLU	12	512	3	1
14	Pooling	13	512	2	2
15	IPBlock	14	512	3	1
16	IPBlock	15	512	3	1
17	IPBlock	16	512	3	1
18	Conv+BN+PReLU	17	512	3	1
19	DeConv+BN+PReLU	18	256	3	2
20	CA+Conv+BN+PReLU	19,8	512	3	1
21	DeConv+BN+PReLU	20	128	3	2
22	CA+Conv+Tanh	19,8	1	3	1
23	Upsample ($\times 2$)	22	1	-	-
24	CA+Conv+BN+PReLU	21,5,23	256	3	1
25	DeConv+BN+PReLU	24	64	3	2
26	CA+Conv+Tanh	21,5,23	1	3	1
27	Upsample ($\times 2$)	26	1	-	-
28	CA+Conv+BN+PReLU	25,2,27	128	3	1
29	DeConv+BN+PReLU	28	32	3	2
30	CA+Conv+Tanh	25,2,27	1	3	1
31	Upsample ($\times 2$)	30	1	-	-
32	CA+Conv+Tanh	29,31	1	3	1

Table 2.1. The depth estimators employed in our experiment. CA: concatenation. BN: batch normalization (Ioffe and Szegedy, 2015). PReLU: parametric rectified linear unit (He *et al.*, 2015). FC, KS and SS refer to the feature channel, kernel size, and stride size, respectively. IPBlock, denoting the inception block, is showed in Figure 2.5.

The detailed configurations of image translator, discriminator and depth estimator are shown in Table 2.2 and Table 2.1. The networks in each group share the identical network architecture but are with different parameters. Specifically,

Image Translator G_{s2t}/G_{t2s}			
Layer	Feature Channel	Kernel Size	Stride Size
Conv+IN+ReLU	64	7	1
Conv+IN+ReLU	128	3	2
Conv+IN+ReLU	256	3	2
ResBlock	256	3	1
ResBlock	256	3	1
ResBlock	256	3	1
ResBlock	256	3	1
ResBlock	256	3	1
ResBlock	256	3	1
ResBlock	256	3	1
ResBlock	256	3	1
ResBlock	256	3	1
Deconv+IN+ReLU	128	3	2
Deconv+IN+ReLU	64	3	2
Conv+Tanh	3	7	1
Discriminator D_s/D_t			
Layer	Feature Channel	Kernel Size	Stride Size
Conv+LReLU	64	4	2
Conv+IN+LReLU	128	4	2
Conv+IN+LReLU	256	4	2
Conv+IN+LReLU	512	4	1
Conv	512	4	1

Table 2.2. The generators and discriminators for image style translation employed in our experiment. IN: instance normalization (Ulyanov *et al.*, 2017). LReLU: LeakyReLU (Maas *et al.*, 2013) respectively. ResBlock, referring to the residual block, is showed in Figure 2.5.

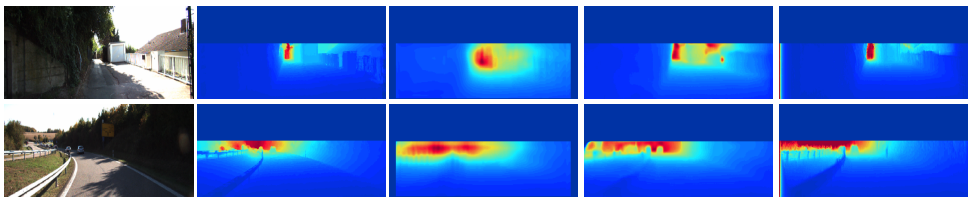


Figure 2.6. Qualitative comparison of our results against methods proposed by Eigen *et al.* (Eigen *et al.*, 2014) and Zheng *et al.* (Zheng *et al.*, 2018) on KITTI. Ground truth has been interpolated for visualization. To facilitate comparison, we mask out the top regions, where ground truth depth is not available. Our approach preserves more details and yields high-quality depth maps.

Method	Sup.	Dataset	Cap.	Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
				Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
(Eigen *et al.*, 2014)	Yes	K	$80m$	0.203	1.548	6.307	0.282	0.702	0.890	0.958
(Liu *et al.*, 2016b)	Yes	K	$80m$	0.202	1.614	6.523	0.275	0.678	0.895	0.965
(Zhou *et al.*, 2017)	No	K	$80m$	0.208	1.768	6.856	0.283	0.678	0.885	0.957
(Zhou *et al.*, 2017)	No	K+CS	$80m$	0.198	1.836	6.565	0.275	0.718	0.901	0.960
(Kuznietsov *et al.*, 2017)	Semi	K	$80m$	0.113	0.741	4.621	0.189	0.862	0.960	0.986
(Godard *et al.*, 2017)	No	K	$80m$	0.148	1.344	5.927	0.247	0.803	0.922	0.964
All synthetic(baseline1)	No	S	$80m$	0.253	2.303	6.953	0.328	0.635	0.856	0.937
All real(baseline2)	No	K	$80m$	0.158	1.151	5.285	0.238	0.811	0.934	0.970
(Kundu *et al.*, 2018)	No	K+S(DA)	$80m$	0.214	1.932	7.157	0.295	0.665	0.882	0.950
(Kundu *et al.*, 2018)	Semi	K+S(DA)	$80m$	0.167	1.257	5.578	0.237	0.771	0.922	0.971
GASDA	No	K+S(DA)	$80m$	**0.149**	**1.003**	**4.995**	**0.227**	**0.824**	**0.941**	**0.973**
(Kuznietsov *et al.*, 2017)	Yes	K	$50m$	0.117	0.597	3.531	0.183	0.861	0.964	0.989
(Garg *et al.*, 2016)	No	K	$50m$	0.169	1.080	5.104	0.273	0.740	0.904	0.962
(Godard *et al.*, 2017)	No	K	$50m$	0.140	0.976	4.471	0.232	0.818	0.931	0.969
All synthetic(baseline1)	No	S	$50m$	0.244	1.771	5.354	0.313	0.647	0.866	0.943
All real(baseline2)	No	K	$50m$	0.151	0.856	4.043	0.227	0.824	0.940	0.973
(Kundu *et al.*, 2018)	No	K+S(DA)	$50m$	0.203	1.734	6.251	0.284	0.687	0.899	0.958
(Kundu *et al.*, 2018)	Semi	K+S(DA)	$50m$	0.162	1.041	4.344	0.225	0.784	0.930	0.974
(Zheng *et al.*, 2018)	No	K+S(DA)	$50m$	0.168	1.199	4.674	0.243	0.772	0.912	0.966
GASDA	No	K+S(DA)	$50m$	**0.143**	**0.756**	**3.846**	**0.217**	**0.836**	**0.946**	**0.976**

Table 2.3. Results on KITTI dataset using the test split suggested in (Eigen *et al.*, 2014). For the training data, K represents KITTI dataset, CS is CityScapes dataset (Cordts *et al.*, 2016), and S is vKITTI dataset. Sup. refers to Supervised. Methods, which apply domain adaptation techniques, are marked by the gray.

Method	Dataset	Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
(Godard *et al.*, 2017)	K	0.124	1.388	6.125	0.217	0.841	0.936	0.975
(Godard *et al.*, 2017)	K+CS	0.104	1.070	5.417	0.188	0.875	0.956	0.983
(Atapour-Abarghouei and Breckon, 2018)	K+S*	**0.101**	1.048	5.308	0.184	**0.903**	**0.988**	**0.992**
GASDA	K+S	0.106	**0.987**	**5.215**	**0.176**	0.885	0.963	0.986

Table 2.4. Results on 200 training images of KITTI stereo 2015 benchmark. S* is captured from GTA5, and more similar to real data than vKITTI. Our approach yields lower errors than state-of-the-art approaches, and achieve competitive accuracy compared with (Atapour-Abarghouei and Breckon, 2018).

we employ generators (G_{s2t} and G_{t2s}) and discriminators (D_s and D_t) provided by CycleGAN (Zhu *et al.*, 2017). For monocular depth estimators F_t and F_s, we utilize the standard encoder-decoder structures with skip-connections and side outputs as (Zheng *et al.*, 2018).

Datasets The target domain is KITTI (Menze and Geiger, 2015), which is a real-world computer vision benchmark consisting of $42,382$ rectified stereo pairs in the resolution about 375×1242. In our experiments, the ground truth depth maps provided by KITTI are only for evaluation purpose. The source domain is

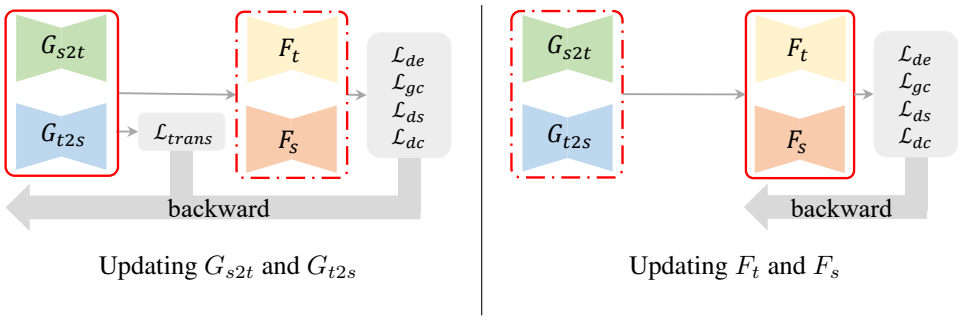


Figure 2.7. Iteratively updating stage. We learn our model by iteratively updating image style translators and depth estimators, *i.e.*, freezing the module with dashed box while updating the one with solid line box. See main text for details. We omit D_t and D_s for brevity.

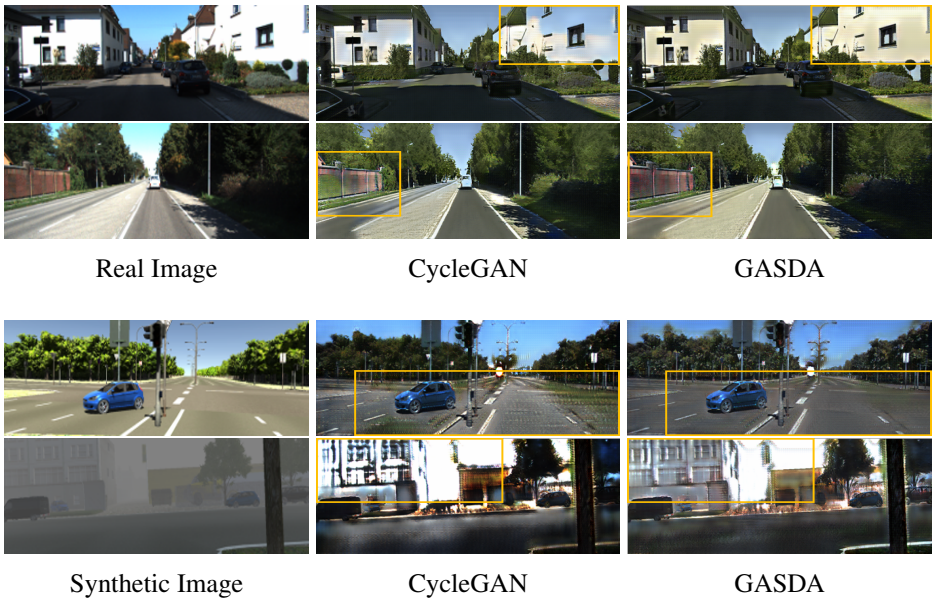


Figure 2.8. Qualitative image style translation results of our approach and CycleGAN. First row: real-to-synthetic translation; Second row: synthetic-to-real translation. Our method can preserve geometric and semantic content better for both synthetic-to-real translation and the inverse one. Note that, the translation result is a by-product of GASDA. The improvement is marked by the yellow box.

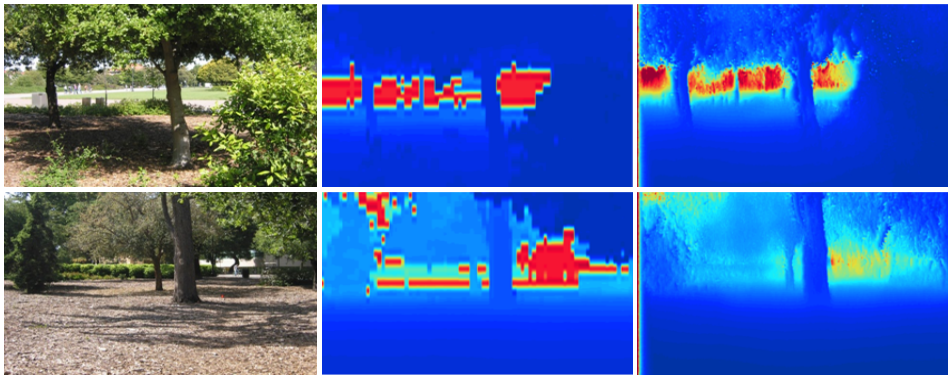
Virtual KITTI (vKITTI) (Gaidon *et al.*, 2016), which contains 50 photo-realistic synthetic videos with $21,260$ image-depth pairs of size 375×1242. Additionally, in order to study the generalization performance of our approach, we also apply the trained model to Make3D dataset (Saxena *et al.*, 2009). Since Make3D does

Method	Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Domain Adaptation							
SYN	0.253	2.303	6.953	0.328	0.635	0.856	0.937
SYN2REAL	0.229	2.094	6.530	0.294	0.691	0.886	0.951
SYN2REAL-E2E	**0.220**	**1.969**	**6.377**	**0.284**	**0.703**	**0.895**	**0.956**
Geometry Consistency							
REAL	0.158	1.151	5.285	0.238	0.811	0.934	0.970
SYN-GC	0.156	1.123	5.255	0.235	0.814	0.937	0.971
SYN2REAL-GC	0.153	**1.112**	**5.213**	0.233	0.819	0.938	0.972
SYN2REAL-GC-E2E	**0.152**	1.130	5.227	**0.231**	**0.821**	**0.939**	**0.972**
Symmetric Domain Adaptation							
REAL2SYN-SYN-GC-E2E	0.160	1.226	5.412	0.240	0.806	0.933	0.969
GASDA-w/oDC	0.151	1.098	5.136	0.230	0.822	0.940	0.972
GASDA-F_t	0.150	1.014	5.041	0.228	**0.824**	**0.941**	**0.973**
GASDA-F_s	0.156	1.087	5.157	0.235	0.813	0.936	0.971
GASDA	**0.149**	**1.003**	**4.995**	**0.227**	**0.824**	**0.941**	**0.973**

Table 2.5. Quantitative results for ablation study on KITTI dataset using the test split suggested in (Eigen *et al.*, 2014). SYN, REAL, REAL2SYN, and SYN2REAL represent the model trained on X_s, X_t, $G_{t2s}(X_t)$, and $G_{s2t}(X_s)$; E2E represents the end-to-end training; GC and DC denote the geometry consistency and depth consistency, respectively; GASDA-F_t (F_s) represents the output of F_t (F_s) in GASDA.

not offer stereo images, we directly evaluate our model on the test split without training or further fine-tuning.

Training Details We implement GASDA in *PyTorch*. We train our model in a two-stage manner, *i.e.*, a warming up stage and end-to-end iteratively updating stage. In the warming up stage, we first optimize the style transfer networks for 10 epochs with the momentum of $\beta_1 = 0.5$, $\beta_2 = 0.999$, and the initial learning rate of $\alpha = 0.0002$ using the ADAM solver (Kingma and Ba, 2014b). Then we train F_t on $\{X_t, G_{s2t}(X_s)\}$, and F_s on $\{X_s, G_{t2s}(X_t)\}$ for around 20 epochs by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\alpha = 0.0001$. To make style translators generate high-quality images, so as to improve the subsequent depth estimators, we fine-tune the network in an end-to-end iteratively updating fashion as shown in Figure 2.7. In specific, we optimize G_{s2t} and G_{t2s} with the supervision of F_t and F_s for m epochs, and then train F_s and F_t for n epochs. We set $m = 3$ and $n = 7$ in our experiments, and repeat this process until the network converges (around 40 epochs). In this stage, we employ the same momentum and solver as the first stage with the learning rates of $2e-6$ and $1e-5$ for the two respectively.



Input Image Ground Truth GASDA

Figure 2.9. Qualitative results on Make3D dataset. Left to right: input image, ground truth depth, and our result.

The trade-off factors are set to $\lambda_1 = 10$, $\lambda_2 = 30$, $\gamma_1 = 50$, $\gamma_2 = 50$ and $\gamma_3 = 50$ and $\gamma_4 = 0.5$. In the training phase, we down-sample all the images to 192×640, and increase the training set size using some common data augmentation strategies, including random horizontal flipping, rotation with the degrees of $[-5°, 5°]$, and brightness adjustment.

2.4.2 KITTI Dataset

We test our models on the 697 images extracted from 29 scenes, and use all the $23,488$ images contained in other 32 scenes for training $(22,600)$ and validation (888) (Eigen *et al.*, 2014; Godard *et al.*, 2017). To make a comparison with previous works, we evaluate our results in the regions with the ground truth depth less than $80m$ or $50m$ using standard error and accuracy metrics (Godard *et al.*, 2017; Zheng *et al.*, 2018). Note that, the maximum depth value in vKITTI is $655.35m$ instead of $80m$ in KITTI, but unlike (Zheng *et al.*, 2018), we do not clip the depth maps of vKITTI to $80m$ during training. In Table 2.3, we report the benchmark scores on the Eigen split (Eigen *et al.*, 2014) where the training sets are only KITTI and vKITTI. GASDA obtains a convincible improvement over previous state-of-the-art methods. Specifically, we make the

Method	Trained*	Error Metrics (lower, better)		
		Abs Rel	Sq Rel	RMSE
(Karsch *et al.*, 2014)	Yes	0.398	4.723	7.801
(Laina *et al.*, 2016)	Yes	0.198	1.665	5.461
(Kundu *et al.*, 2018)	Yes	0.452	5.71	9.559
(Godard *et al.*, 2017)	No	0.505	10.172	10.936
(Kundu *et al.*, 2018)	No	0.647	12.341	11.567
(Atapour-Abarghouei and Breckon, 2018)	No	0.423	9.343	**9.002**
GASDA	No	**0.403**	**6.709**	10.424

Table 2.6. Results on 134 test images of Make3D. Trained* indicates whether the model is trained on Make3D or not. Errors are computed for depths less than $70m$ in a central image crop (Godard *et al.*, 2017). It can be observed that our approach is comparable with those trained on Make3D.

comparisons with two baselines, *i.e.,* All synthetic (baseline1, trained on labeled synthetic data) and All real (baseline2, trained on real stereo pairs), and the latest domain adaptation methods (Zheng *et al.*, 2018; Kundu *et al.*, 2018) and (semi-)supervised/unsupervised methods (Eigen *et al.*, 2014; Liu *et al.*, 2016b; Kuznietsov *et al.*, 2017; Garg *et al.*, 2016; Godard *et al.*, 2017; Zhou *et al.*, 2017). The significant improvements in all the metrics demonstrate the superiority of our method. Note that, GASDA yields higher scores than (Kundu *et al.*, 2018) which employs additional ground truth depth maps for natural images contained in KITTI. GASDA cannot outperform (Atapour-Abarghouei and Breckon, 2018) in the Eigen split. The main reason is that the synthetic images employed in (Atapour-Abarghouei and Breckon, 2018) are captured from GTA5 [3], and the domain shift between GTA5 and KITTI is not that significant than the one between vKITTI and KITTI. In addition, the training set size in (Atapour-Abarghouei and Breckon, 2018) is about three times than ours. However, GASDA performs competitively on the official KITTI stereo 2015 dataset (Geiger *et al.*, 2012b) and Make3D compared with (Atapour-Abarghouei and Breckon, 2018), as reported in Table 2.4 and Table 2.6. Apart from quantitative results, we also show some example outputs in Figure 2.6. Our approach preserves more details, and is able to recover depth information of small objects, such as the distant cars and rails, and generate clear boundaries.

[3] https://github.com/aitorzip/DeepGTAV.

2.4.3 Make3D Dataset

To discuss the generalization capabilities of GASDA, we evaluate our approach on Make3D dataset (Saxena *et al.*, 2009) quantitatively and qualitatively. We do not train or further fine-tune our model using the images provide by Make3D. As shown in Table 2.6 and Figure 2.9, although the domain shift between Make3D and KITTI is large, our model still performs well. Compared with state-of-the-art models (Kundu *et al.*, 2018; Karsch *et al.*, 2014; Laina *et al.*, 2016) trained on Make3D in a supervised manner and others using domain adaptation (Kundu *et al.*, 2018; Atapour-Abarghouei and Breckon, 2018), GASDA obtains impressive performance.

2.4.4 Ablation Study

Here, we conduct a series of ablations to analyze our approach. Quantitative results are shown in Table 2.5, and some sampled results for style transfer are shown in Figure 2.8.

Domain Adaptation We first demonstrate the effectiveness of domain adaptation by comparing two simple models, *i.e.* SYN (F_s trained on X_s) and SYN2REAL (F_t trained on $G_{s2t}(X_s)$). As shown in Table 2.5, SYN cannot capture satisfied scores on KITTI due to the domain shift. After the translation, the domain shift is reduced which means that the synthetic data distribution is relative closer to real data distribution. Thus, SYN2REAL is able to generalize better to real images. Further, we train the style translators (G_{s2t} and G_{t2s}) and the depth estimation network (F_t) in an end-to-end fashion (SYN2REAL-E2E), which guides to a further improvement as compared to SYN2REAL. As a conclusion, the depth estimation network can improve the style transfer by providing a pixel-wise semantic constraint to the translation networks. Moreover, we can also observe the improvement in Figure 2.8 by comparing the translation results of original CycleGAN (Zhu *et al.*, 2017) with ours.

Geometry Consistency We then study the significance of the geometric constraint coming from stereo images based on the epipolar geometry. In specific, we employ the stereo images provided by KITTI when optimizing F_t in SYN2REAL-E2E. We enforce the geometry consistency between the stereo images as a constraint as stated in Eq. 2.8. The model SYN2REAL-GC-E2E outperforms SYN2REAL-E2E by a large margin, which demonstrates that the geometry consistency constraint can significantly improve standard domain adaptation frameworks. On the other hand, the comparisons among SYN2REAL-GC, SYN-GC (trained on real data and synthetic data without domain adaptation) and REAL (F_t trained on real stereo images without extra data) can show the significance of synthetic data with ground truth depth and domain adaptation.

Symmetric Domain Adaptation In contrast to previous works, we expect to fully take advantage of the bidirectional style translators G_{s2t} and G_{t2s}. Thus, we learn REAL2SYN-SYN-GC-E2E whose network architecture is symmetrical to the aforementioned SYN2REAL-GC-E2E. We jointly optimized the two coupled with a depth consistency loss. As shown in Table 2.5, GASDA is superior than GASDA-w/oDC which demonstrates the effectiveness of the depth consistency loss. In addition, the comparisons (GASDA-F_t *v.s.* SYN2ERAL-GC-E2E and GASDA-F_s *v.s.* REAL2SYN-GC-E2E) show that the two can benefit each other in the jointly training.

2.4.5 More Qualitative Results

Lastly, we show other qualitative results on the KITTI Eigen Split (Eigen *et al.*, 2014) (Figure 2.10) and CityScapes dataset (Cordts *et al.*, 2016) (Figure 2.11).

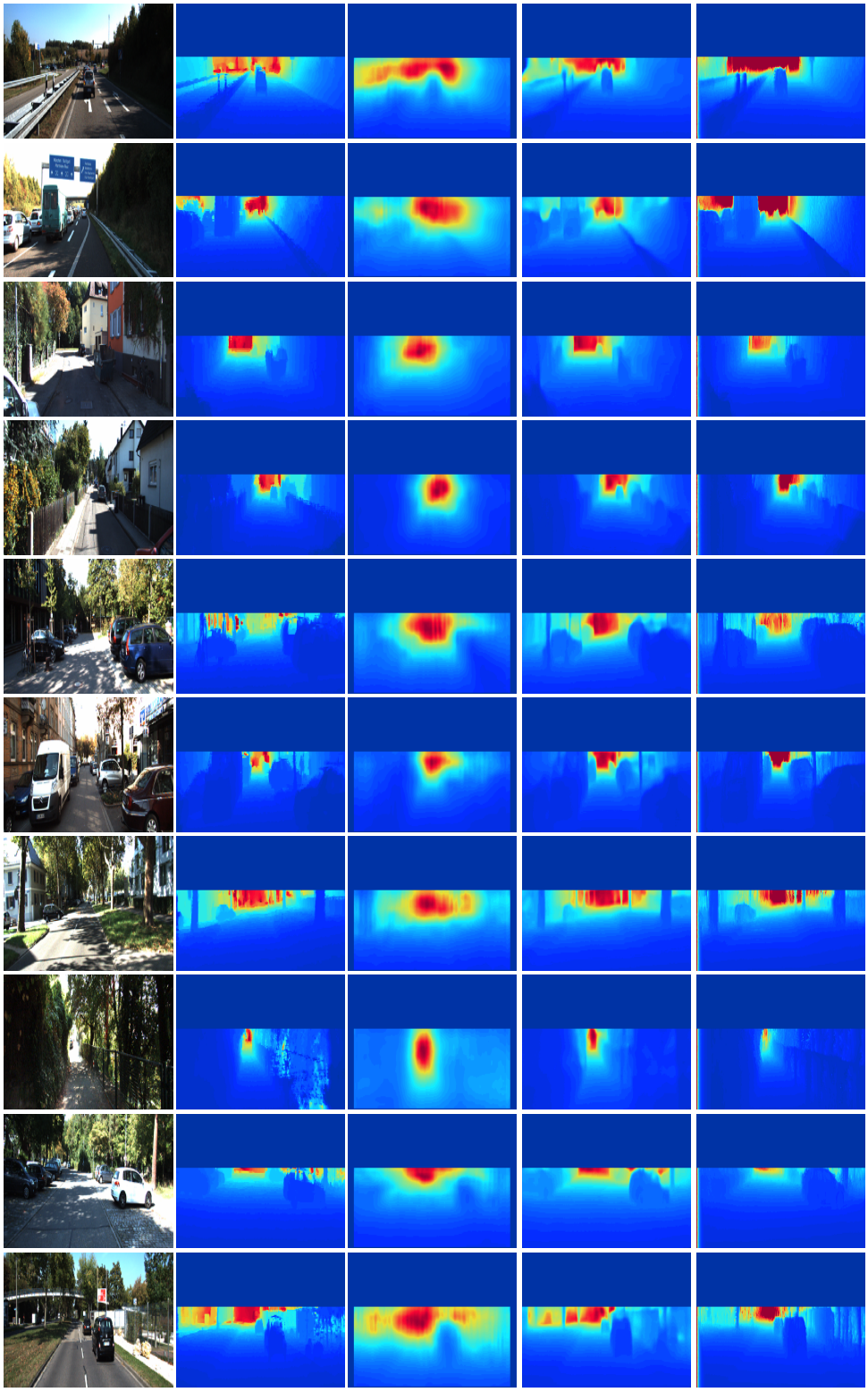


Figure 2.10. Qualitative comparisons of our results with methods proposed by Eigen *et al.* (Eigen *et al.*, 2014) and Zheng *et al.* (Zheng *et al.*, 2018) on the KITTI Eigen Split. The model is trained on KITTI using the split of Eigen *et al.* (Eigen *et al.*, 2014).

2.5 Conclusion

In this chapter, we present an unsupervised monocular depth estimation framework GASDA, which trains the monocular depth estimation model using the labelled synthetic data coupled with the epipolar geometry of real stereo data in a unified and symmetric deep learning network. Our main motivation is learning a depth estimation model from synthetic image-depth pairs in a supervised fashion, and at the same time taking into account the specific scene geometry information of the target data. Moreover, to alleviate the issues caused by domain shift, we reduce the domain discrepancy using the bidirectional image style transfer. Finally, we implement image translation and depth estimation in an end-to-end network so that then can improve each other. Experiments on KITTI and Make3D datasets show GASDA is able to generate desirable results quantitatively and qualitatively, and generalize well to unseen datasets.

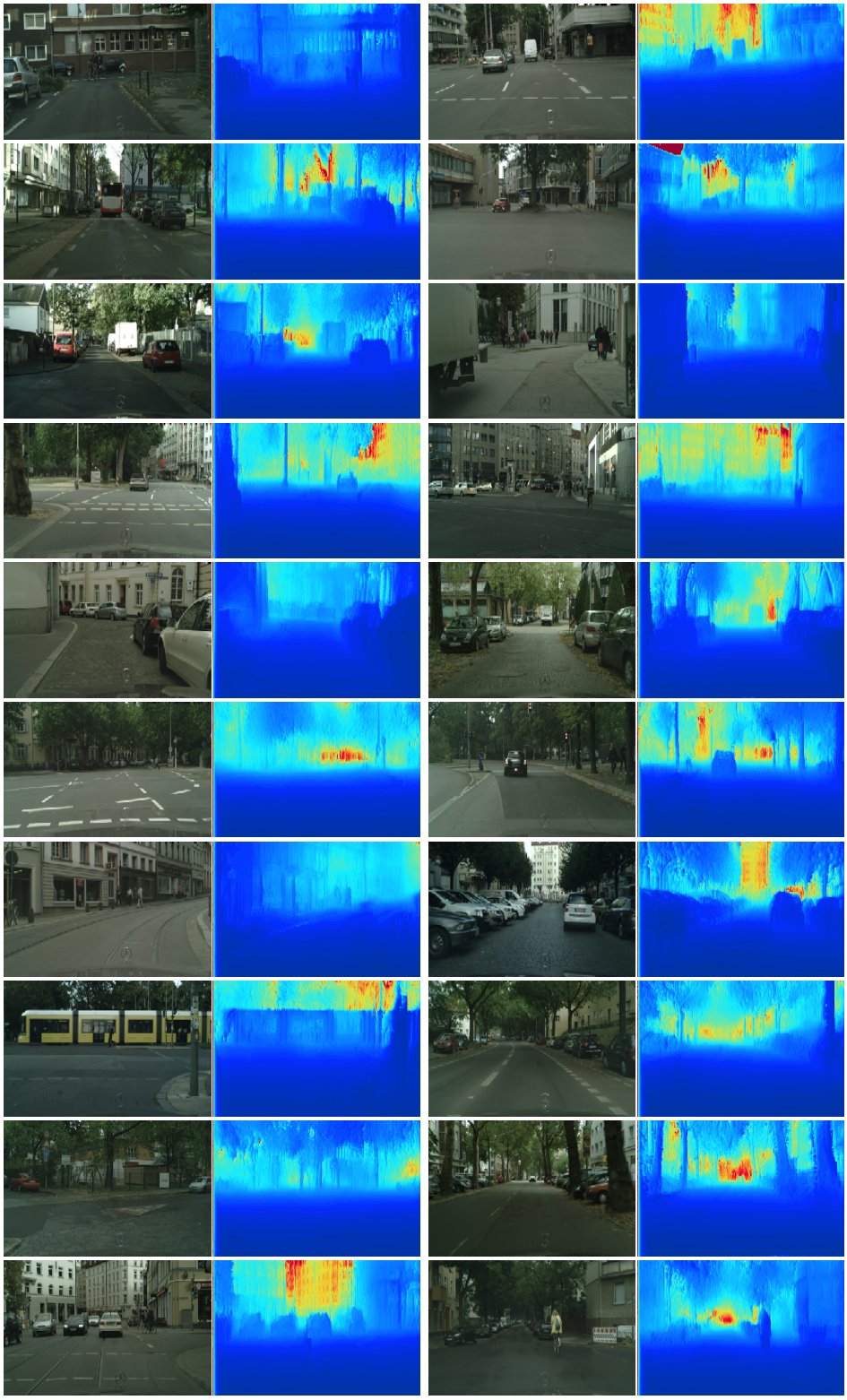


Figure 2.11. Qualitative results on CityScapes dataset. The model is trained on KITTI using the split of Eigen *et al.* (Eigen *et al.*, 2014) without further fine-tuning.

CHAPTER 3

Adaptive Context-Aware Multi-Modal Network for Depth Completion

---

Taking advantage of domain adaptation techniques, last chapter explores 3D structure information prediction from a single image, *i.e.,* monocular depth estimation. In this chapter, we study depth completion, which aims to recover a dense depth map from the sparse depth data and the corresponding single RGB image, *i.e.,* multi-modal data. The observed pixels provide the significant guidance for the recovery of the unobserved pixels' depth. However, due to the sparsity of the depth data, the standard convolution operation, exploited by most of existing methods, is not effective to model the observed contexts with depth values. To address this issue, we propose to adopt the graph propagation to capture the observed spatial contexts. Specifically, we first construct multiple graphs at different scales from observed pixels. Since the graph structure varies from sample to sample, we then apply the attention mechanism on the propagation, which encourages the network to model the contextual information adaptively. Furthermore, considering the mutli-modality of input data, we exploit the graph propagation on the two modalities respectively to extract multi-modal representations. Finally, we introduce the symmetric gated fusion strategy to exploit the extracted multi-modal features effectively. The proposed strategy preserves the original information for one modality and also absorbs complementary information from the other through learning the adaptive gating weights. Our model, named Adaptive Context-Aware Multi-Modal Network (ACMNet), achieves the

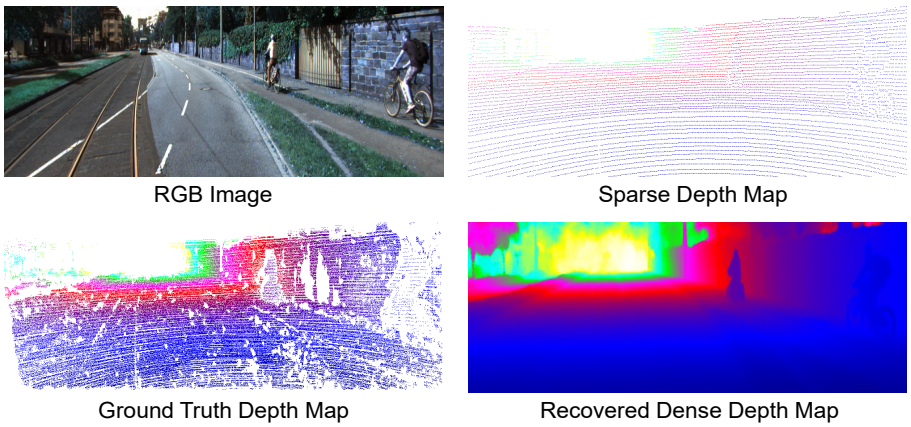


Figure 3.1. Depth Completion from LiDAR Data and RGB Image by ACMNet. Top: RGB image and sparse LiDAR data; Bottom: ground truth depth map and dense depth map obtained by our approach.

state-of-the-art performance on two benchmarks, *i.e.*, KITTI and NYU-v2, and at the same time has fewer parameters than latest models.

3.1 Introduction

Depth information is crucial for 3D vision tasks, *e.g.*, 6D object pose estimation (Wang *et al.*, 2019a), 3D object detection (Xu *et al.*, 2018b), and human pose estimation (Moon *et al.*, 2018). To complete these tasks, various depth sensors such as LiDAR have been invented to acquire depth information. However, current depth sensors are not able to obtain dense maps for outdoor scenes, which are essential in various applications, especially autonomous driving. Therefore, depth completion from sparse depth maps[1] and RGB images has attracted intensive attention. Depth completion is a challenging problem because the depth values obtained by sensors are highly sparse and irregularly spaced. For example, in the KITTI dataset (Geiger *et al.*, 2012a), there are only 5.9% pixels with depth information obtained by the Velodyne HDL-64e

[1]The sparse depth map is generated by projecting the LiDAR data to the image plane, and the value in locations without depth information is 0.

(64 layers) LiDAR in the whole image space, as shown in Figure 3.1. Traditional methods (Ferstl *et al.*, 2013; Herrera *et al.*, 2013; Schneider *et al.*, 2016) rely on handcrafted features and global constraints on the output depth values, which are inaccurate. Recent studies (Zhang and Funkhouser, 2018; Uhrig *et al.*, 2017; Ma and Karaman, 2018; Jaritz *et al.*, 2018; Imran *et al.*, 2019; Atapour-Abarghouei and Breckon, 2019b; Cheng *et al.*, 2019; Chen *et al.*, 2019; Zhong *et al.*, 2019; Eldesokey *et al.*, 2020; Lu *et al.*, 2020) have demonstrated great advantages of deep Convolutional Neural Networks (CNNs) on depth completion. By extending the convolutional operation with sparsity-invariance (Uhrig *et al.*, 2017; Huang *et al.*, 2020; Eldesokey *et al.*, 2019) or introducing more geometric information (Qiu *et al.*, 2019; Xu *et al.*, 2019), these deep methods can achieve way better performance than traditional methods.

In spite of the encouraging progress, existing depth completion methods suffer from a significant issue, which limits the depth completion performance. Specifically, the conventional convolutional operation applies kernels with regular structure (*e.g.,* 3×3) at all locations, which ignores the fact that the observed depth values are irregularly distributed in a sparse depth map and associates limited observed contexts for the unobserved, as shown in Figure 3.2. Thus, CNN-based methods are not adaptive to the pattern of observed spatial contextual information in a sparse depth map, resulting in a sub-optimal prediction of depth in unobserved locations.

To address this issue and further boost depth completion accuracy, we propose an Adaptive Context-Aware Multi-Modal Network (ACMNet, shown in Figure 3.3). Firstly, inspired by recent works on point cloud analysis (Wang *et al.*, 2018b), we model the observed contextual information adaptively by applying attention based graph propagation within multiple graphs constructed from observed pixels. Based on the efficient graph propagation, the model can associate the spatial context with observed depth values and then enhance the features of the unobserved pixels. To illustrate this, we provide a simple example in

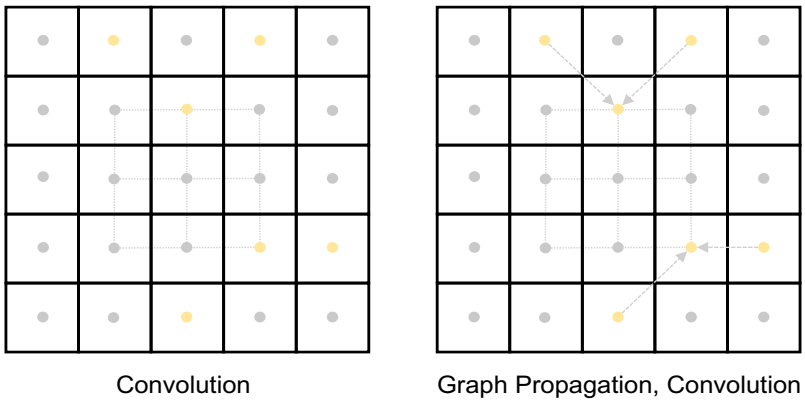


Figure 3.2. Illustration of convolution and graph propagation. Left: convolution (3×3 kernel); Right: graph propagation (2-nearest neighbours) and convolution (3×3 kernel). The observed pixels are marked by the yellow, while the unobserved are marked by the gray.

Figure 3.2. Compared to the sole convolutional operations, the proposed graph propagation (followed by a convolution) can make the unobserved pixels capture more related observed contextual information.

Furthermore, since we have multi-modality data, we need to reconsider the novel graph propagation in a multi-modal setting. Firstly, to better learn the relationship between observed pixels (nodes), we use the co-attention mechanism (Lu *et al.*, 2016) to propagate the multi-modal information of observed pixels in a symmetric structure. This step is conducted in the encoder to extract multi-scale and multi-modal features. However, this mechanism does not consider the fusion of multi-modal contextual information. A simple way to fuse the multi-modal data is by applying the simple concatenation or element-wise summation operation on the extracted feature maps, which was used by most of the existing works, *e.g.,* (Jaritz *et al.*, 2018; Qiu *et al.*, 2019). However, this type of fusion strategy cannot fully explore the heterogeneity of the two modalities. To address the issue, we further present the symmetric gated fusion strategy to combine the depth and RGB information in the decoder. In specific, the presented fusion strategy consists of two branches. One branch

focuses on fusing the RGB information as supplementary into the depth information through learning an adaptive gating function, and the other one does the opposite. Therefore, each branch can maintain its own information and benefit from supplementary information from the other. Benefiting from the adaptive co-attention guided graph propagation and symmetric gated multi-modal feature fusion, our ACMNet is able to generate high-quality dense depth maps.

3.2 Related Work

Depth Completion. Traditional approaches solve the depth completion problem by formulating the task as an energy function optimization problem (Ferstl *et al.*, 2013; Barron and Poole, 2016; Herrera *et al.*, 2013; Schneider *et al.*, 2016). However, these works showed some limitations in performance due to the employment of hand-crafted features.

Currently, CNNs have been a dominant solution for depth completion (Qiu *et al.*, 2019; Chodosh *et al.*, 2018; Cheng *et al.*, 2018; Tang *et al.*, 2019; Van Gansbeke *et al.*, 2019; Ma *et al.*, 2019; Yang and Soatto, 2018; Huang *et al.*, 2020; Atapour-Abarghouei and Breckon, 2019a; Cheng *et al.*, 2020a; Eldesokey *et al.*, 2020; Lu *et al.*, 2020; Liao *et al.*, 2017; Li *et al.*, 2020a), outperforming traditional methods by a wide margin. In specific, to learn representations of the irregular and sparse LiDAR data, Uhrig *et al.* (Uhrig *et al.*, 2017) proposed the sparsity-invariant convolutional operation. Following this work, some variants of the sparse convolution are introduced (Eldesokey *et al.*, 2019; Huang *et al.*, 2020; Eldesokey *et al.*, 2020). In the case of additional RGB data, Jaritz *et al.* (Jaritz *et al.*, 2018) showed that the late fusion strategy outperformed the early fusion. Ma *et al.* (Ma *et al.*, 2019) utilized self-supervised learning on sparse LiDAR data coupled with the stereo image pair to mitigate the need for ground truth dense depth. Yang *et al.* (Yang *et al.*, 2019b) exploited the Conditional Prior Network (Yang and Soatto, 2018) to learn a depth prior on

synthetic images. Additionally, there are also a bunch of works (Zhang and Funkhouser, 2018; Van Gansbeke *et al.*, 2019; Xu *et al.*, 2019) exploring other cues. For example, Zhang *et al.* (Zhang and Funkhouser, 2018) trained a network to predict local surface normals for indoor scene depth completion, and later an extension for outdoor scenes was introduced in their latest work (Qiu *et al.*, 2019). Similarly, Xu *et al.* (Xu *et al.*, 2019) also explored the surface normal information to improve the performance by introducing a diffusion module. Cheng *et al.* (Cheng *et al.*, 2018; Cheng *et al.*, 2020a) proposed to learn affinities between adjacent pixels for the spatial propagation of the depth information. Following the two works, a recent work (Park *et al.*, 2020) improved the propagation strategy through concentrating on the non-local neighbors and introducing a learnable affinity normalization. Inspired by the guided image filtering, Tang *et al.* (Tang *et al.*, 2019) designed a guided convolution module, which generates dynamic spatially-variant kernels using the image features, to extract the depth image features. In comparison, a recent work (Xiong *et al.*, 2020) proposed to dynamically learn the filter by applying the Graph Neural Network (GNN) (Zhou *et al.*, 2018) on the graph constructed from the predicted dense depth map. In a nutshell, existing works mainly exploit the standard convolutional operation or its variations to extract the contextual information. In specific, they first model the representation for the RGB and sparse depth data separately, and then fuse them together in a single path. In contrast to these approaches, out work applies the graph propagation strategy (CGPM) on the observed points so that the unobserved pixels can capture more useful observed contextual information. In addition, taking advantage of the proposed symmetric gated fusion strategy, our method can make better use of the multi-modal information (SGFM). Finally, it is worth pointing out that although the latest work (Xiong *et al.*, 2020) also exploits the graph models, there are many differences between it and ours. For example, it aims to consider the neighborhood relationship of the points in the 3D space through constructing a 3D graph from the dense depth map, which is obtained using a deep model. To arrive at this,

it applies the dynamic kernel, which is learned through using a typical GNN model on the constructed graph, on the dense features at $1/8$ of original scale. In comparison, in this chapter we study the propagation of the contexts with observed depth values at multiple scales in a multi-modal setting to enhance the features of the unobserved pixels.

Monocular Depth Estimation. From approaches based on probabilistic graphical models (*e.g.*, MRFs) with hand-crafted features (Saxena *et al.*, 2006; Saxena *et al.*, 2009) to the deep learning-based (Fu *et al.*, 2018; Liu *et al.*, 2016b; Godard *et al.*, 2017; Eigen *et al.*, 2014; Zhou *et al.*, 2017; Garg *et al.*, 2016; He *et al.*, 2018a; Kim *et al.*, 2018; Cao *et al.*, 2018; Wang *et al.*, 2020; Yang *et al.*, 2020a; Zhang *et al.*, 2018b), the improvement of performance for monocular depth estimation has been pushed forward. Eigen *et al.* (Eigen *et al.*, 2014) were the first to develop deep models for depth estimation. Following their work, a lot of supervised approaches (Laina *et al.*, 2016; Liu *et al.*, 2016a; Fu *et al.*, 2018; Eigen and Fergus, 2015) have been proposed. However, these methods rely on large quantities of ground truth depth data, which is hard to acquire. To address this issue, Garg *et al.* (Garg *et al.*, 2016) and Godard *et al.* (Godard *et al.*, 2017) proposed to predict depth maps from stereo pair images by exploring unsupervised cues, while some recent works tried to utilize synthetic data (Atapour-Abarghouei and Breckon, 2018; Zhao *et al.*, 2019b; Zheng *et al.*, 2018; Nath Kundu *et al.*, 2018; PNVR *et al.*, 2020) based on the domain adaptation technique (Pan and Yang, 2009).

Graph-based Models. Conventional deep learning modules, such as CNNs, do not perform well on graphs. To model the graph data efficiently, Graph Models have been applied on various computer vision tasks (Shi *et al.*, 2019; Ji *et al.*, 2020; Wu *et al.*, 2020), such as action recognition (Shi *et al.*, 2019; Yan *et al.*, 2018), point cloud analysis (Wang *et al.*, 2019b; Wang *et al.*, 2018b), few-shot image classification (Garcia and Bruna, 2017; Kim *et al.*, 2019), and person re-identification (Wu *et al.*, 2020). Graph Models are able to learn the

representation of each target node by propagating its neighborhood information in a data-driven way and thus associate the contextual information. In this work, we design an attention-based graph propagation module and then extend it to the co-attention guided graph propagation for multi-modal data, which is capable of learning an efficient multi-modal representation for the input data through encouraging the adaptive contextual interactions.

Multi-modal Information Fusion. Multi-modal information fusion has been studied in various computer vision tasks, such as visual question answering (Ben-Younes *et al.*, 2017), video action recognition (Simonyan and Zisserman, 2014), 3D object detection (Yoo *et al.*, 2020), and many more. A simple approach to fuse the multi-modal data is applying concatenation or summation operation into the input data or extracted feature maps (Jaritz *et al.*, 2018; Simonyan and Zisserman, 2014). However, for a specific task, different modalities often provide different information, and therefore, the naive fusion strategy might fail to combine them effectively. To address this issue, some works, *e.g.,* (Yoo *et al.*, 2020; Hori *et al.*, 2018), proposed to exploit the attention mechanism to improve the performance. As for depth completion, current works mainly employed the naive fusion strategy. In fact, both naive strategy and attention based approaches fuse the multi-modal features in a single way, which is not enough to extract complementary information and then limits the performance. In contrast, we present the symmetric gated fusion strategy consisting of two fusion paths, each of which only focuses on one modal and extracts useful information adaptively from the other.

3.3 Our Approach

3.3.1 Problem Formulation

Our goal is to recover a dense depth map from the observed sparse depth data and a single RGB image. Mathematically, given a set of paired RGB image and sparse depth map $\{(\mathbf{X}_I, \mathbf{X}_S)_i\}_{i=0}^{N-1}$, we expect to learn a mapping function $f(\cdot)$ that satisfies $\mathbf{Y} = f(\mathbf{X}_S, \mathbf{X}_I)$, where $\mathbf{X}_S \in \mathbb{R}^{H\times W}$, $\mathbf{X}_I \in \mathbb{R}^{3\times H\times W}$, and $\mathbf{Y} \in \mathbb{R}^{H\times W}$ represent the sparse depth map, the RGB image, and the ground truth depth map, respectively. To achieve this target, we develop a high-performing depth completion network (ACMNet) building on two novel modules, including a co-attention guided graph propagation module (CGPM) and a symmetric gated fusion module (SGFM), as shown in Figure 3.3. In specific, we first employ a series of CGPMs to effectively extract contextual information from $\mathbf{X}_S$ and $\mathbf{X}_I$. Then we exploit SGFMs to learn the complementarity between contextual representations from multi-modalities. In the following, we will present our network architecture and the proposed modules in detail.

3.3.2 Network Architecture

Our overall network architecture follows a two-stream encoder-decoder fashion as previously (Van Gansbeke *et al.*, 2019; Ma *et al.*, 2019; Jaritz *et al.*, 2018; Atapour-Abarghouei and Breckon, 2019b), but with the improvement by integrating the novel CGPM and SGFM. We show the whole framework in Figure 3.3, and briefly explain the encoder and the decoder right here.

Encoder. The encoder targets learning discriminative multi-scale features from both the sparse depth and the RGB image. While researchers reached a consensus that standard convolutional operations can perform well in the image data, how to extract rich information from observed spatial contexts is still an open problem due to the extreme sparsity (Uhrig *et al.*, 2017; Eldesokey *et al.*, 2019;

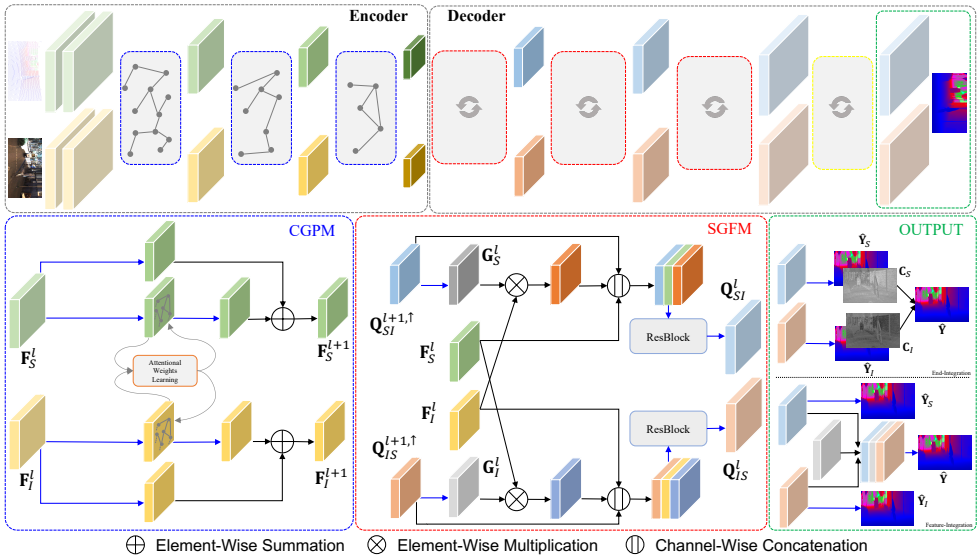


Figure 3.3. The proposed ACMNet in this chapter. Left upper part: Encoder; Right upper part: Decoder. In encoding stage, we extract multi-scale multi-modal features using a stack of CGPMs (Marked by blue dotted box, Sec. 3.3.3), and the adaptive attentional weights are learnt from spatial locations, depth features and RGB features. In decoding stage, we fuse the multi-modal features progressively by exploiting the SGFMs, represented by red dotted boxes (Sec. 3.3.4). Lastly, final output is calculated from the dense maps and confidence maps produced by the two branches of the decoder or predicted using the intermediate fused features maps, shown in the green dotted box (Sec. 3.3.5). Note that, the yellow dotted box denotes that there is no ResBlock behind the initial fusion (see Sec. 3.3.4) in the SGFM. Blue arrow: convolution; Gray arrow: graph propagation; Black arrow: summation/multiplication/concatenation.

Huang *et al.*, 2020; Jaritz *et al.*, 2018; Van Gansbeke *et al.*, 2019; Tang *et al.*, 2019; Xu *et al.*, 2019). In this chapter, we show that the proposed CGPM has the potential to capture the related contextual information from the observed pixels with various patterns in an adaptive manner through learning dynamic weights of the relationship between adjacent nodes in the constructed graph. Specifically, our encoder consists of two conventional convolutional layers followed by a stack of CGPMs. The encoded features at each scale $\{\mathbf{F}_S^l\}_{l=1}^L$ and $\{\mathbf{F}_I^l\}_{l=1}^L$ can be computed as:

$$(\mathbf{F}_S^l, \mathbf{F}_I^l) = f_e^l(\mathbf{F}_S^{l-l}, \mathbf{F}_I^{l-1}), \mathbf{F}_S^l, \mathbf{F}_I^l \in \mathbb{R}^{C^l \times \frac{H}{2^l} \times \frac{W}{2^l}}, \tag{3.1}$$

where $l = 1, 2, ..., L$, and f_e^l denotes the CGPM at level l, and $\mathbf{F}_S^0$ and $\mathbf{F}_I^0$ are the outputs of the beginning convolutional layers.

Decoder. The decoder aims to predict depth values of unobserved pixels in $\mathbf{X}_S$ given multi-scale and multi-modal features generated by the encoder mentioned above. To this end, one of the commonly studied problems is how to take full advantage of multi-modal representations. A straightforward idea is to directly concatenate or sum features progressively at different scales (Jaritz *et al.*, 2018; Qiu *et al.*, 2019). However, as analyzed before, these naive fusion strategies fail to model the complementary information between multiple modalities satisfyingly. To alleviate the issue, we propose an adaptive symmetric gated fusion strategy to fuse the multi-modal contextual representations in a parallel structure. In specific, we design two parallel branches in the decoder, *i.e.,* the depth and image branches. The depth branch preserves discriminative information of the sparse depth modality and meanwhile adaptively captures comprehensive information from the image model through learning dynamic gating weights, and vise versa for the image branch. The overall decoder architecture is described as follows.

As shown in Figure 3.3, at the beginning of the decoder, we feed $\mathbf{F}_S^L$ coupled with $\mathbf{F}_I^L$ into the first SGFM to generate the fused feature $\mathbf{Q}_{SI}^L$ and $\mathbf{Q}_{SI}^{L,\uparrow}$, which is acquired by up-sampling $\mathbf{Q}_{SI}^L$ through one deconvolutional layer. At the following levels l from $L-1$ to 0, $\mathbf{Q}_{SI}^{l+1,\uparrow}$, $\mathbf{F}_S^l$ and $\mathbf{F}_I^l$ are fed into the SGFM at level l together. Similarly, we can obtain the intermediate features $\mathbf{Q}_{IS}^l$ in the image branch. The procedure can be expressed as:

$$\begin{aligned}(\mathbf{Q}_{SI}^L, \mathbf{Q}_{IS}^L, \mathbf{Q}_{SI}^{L,\uparrow}, \mathbf{Q}_{IS}^{L,\uparrow}) &= f_d^L(\mathbf{F}_S^L, \mathbf{F}_I^L), \\ (\mathbf{Q}_{SI}^l, \mathbf{Q}_{IS}^l, \mathbf{Q}_{SI}^{l,\uparrow}, \mathbf{Q}_{IS}^{l,\uparrow}) &= f_d^l(\mathbf{Q}_{SI}^{l+1,\uparrow}, \mathbf{Q}_{IS}^{l+1,\uparrow}, \mathbf{F}_S^l, \mathbf{F}_I^l),\end{aligned} \tag{3.2}$$

where $l = L-1, L-2, ..., 0$, and f_d^l represents the SGFM.

Finally, we present two methods, *i.e.,* end-integration and feature-integration, to combine the two branches to obtain the final recovered dense depth map, which will be described in detail in Sec. 3.3.5.

3.3.3 Co-Attention Guided Graph Propagation (CGPM)

The proposed CGPM is composed of a residual connection and a co-attention guided graph propagation module. First, we introduce the basic graph propagation module, which is employed in CGPM. In specific[2], given the spatial position set $P = \{p_0, p_1, ..., p_{n-1}\}$ of n pixels with observed depth values, we define a graph $G(V, E)$, where V is the vertex (or node) set corresponding to P, and $E \subseteq |V| \times |V|$ is the edge set. For a vertex i, we connect it to the k nearest neighbours according to the spatial locations. Note that, we build an individual graph for the CGPM at each scale. Thus, to obtain a specific P^l at level l, which is in lower resolution, we generate $\mathbf{X}_S^l$ by applying max-pooling based down-sampling operation on $\mathbf{X}_S^{l-1}$. The graph's construction process can be found in Figure 3.4. In the following, we first introduce the basic attention guided graph propagation component at level l by taking the image stream as an example, then present the full CGPM.

Given the graph G constructed from P^l and the input feature maps $\mathbf{F}_I^{l-1}$, we expect to learn discriminative $\mathbf{F}_I^l$ by both adaptively encoding the contextual information of scenes and exploiting guidance for unobserved pixels from observed pixels. Specifically, we exploit two efficient stages, *i.e.,* adaptive feature propagation within observed pixels and feature enhancement of unobserved pixels.

At the first stage, we employ one standard convolutional layer to extract $\mathbf{F}'_I$ from $\mathbf{F}_I^{l-1}$, and denote $\mathbf{F}'_{Io} \in \mathbb{R}^{n \times C}$ as the feature vectors of all the nodes in G. Then,

[2]In the following part, we deprecate the scale indexes l to simplify our presentation in some cases.

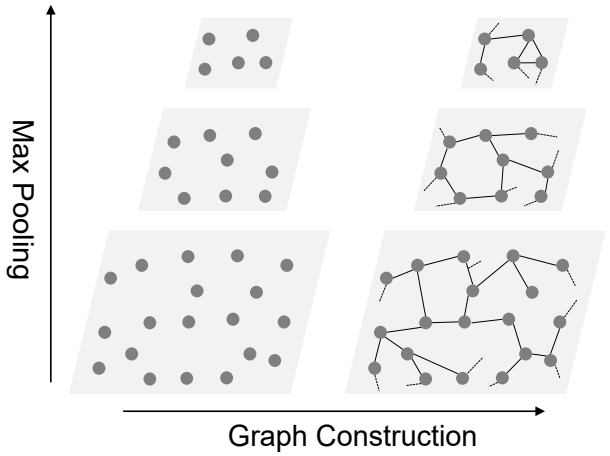


Figure 3.4. Graph Construction. At each scale, we use k (*e.g.,* k is 3 in this example) nearest neighbour to construct the graph from the observed pixels, represented by gray circles.

we adaptively aggregate neighboured information for each node i in G as:

$$\begin{aligned} \alpha^{i,j} &= \frac{exp(\mathbf{W}^{i,j})}{\sum_{k\in\mathcal{N}_i} exp(\mathbf{W}^{i,k})}, \\ \mathbf{F}''^{i}_{Io} &= \sum_{j\in\mathcal{N}_i} \alpha^{i,j}\mathbf{F}'^{i}_{Io}, \end{aligned} \tag{3.3}$$

where $\alpha^{i,j}$ is the computed attentional weight, $\mathcal{N}_i$ represents the nearest neighbours of the node i, and $\mathbf{W}^{i,j}$ is the adaptive weight between neighboured nodes i and j. Here, inspired by the works (Wu *et al.*, 2019; Wang *et al.*, 2019b; Li *et al.*, 2018c) on point cloud, we exploit the self-attention mechanism (Vaswani *et al.*, 2017) to learn $\mathbf{W}^{i,j}$ adaptively by modeling the relationship between the connected nodes. Mathematically, the mapping function f_w between $\mathbf{F}'_{Io}$ and $\mathbf{W}^{i,j}$ can be expressed as:

$$\mathbf{W}^{i,j} = f_w([\Delta p^{i,j}||\Delta\mathbf{F}'^{i,j}_{Io}]), j \in \mathcal{N}_i, \tag{3.4}$$

where $[\cdot||\cdot]$ represents the concatenation operation, $\Delta p^{i,j} = p^j - p^i$ and $\Delta\mathbf{F}'^{i,j}_{Io} = \mathbf{F}'^{j}_{Io} - \mathbf{F}'^{i}_{Io}$ denote the spatial and feature distances between node i and j, respectively. The f_w is implemented by a two-layer MLP, the first one followed by one LeakyReLU activation function (Maas *et al.*, 2013). Note that, permutation

variant operations like convolution are not allowed here due to the unordered input. After obtaining $\mathbf{F}''_{Io}$, the features of unobserved pixels are enhanced by a standard convolutional operation. In addition, a residual connection (He *et al.*, 2016) is also utilized to preserve early information. We can use the same algorithm to conduct propagation in the depth stream.

As shown in Figure 3.3, in the CGPM in our encoder, we learn the adaptive weights $\mathbf{W}_S$ and $\mathbf{W}_I$ by considering both information from the image stream and the sparse depth stream, inspired by the co-attention mechanism (Lu *et al.*, 2016). Therefore, in each CGPM, Eq. 3.4 can be re-written as:

$$\begin{aligned} \mathbf{W}_S^{i,j} &= f_{Sw}([\Delta p^{i,j}||[\Delta \mathbf{F}_{So}^{'i,j}||\Delta \mathbf{F}_{Io}^{'i,j}]]), j \in \mathcal{N}_i, \\ \mathbf{W}_I^{i,j} &= f_{Iw}([\Delta p^{i,j}||[\Delta \mathbf{F}_{Io}^{'i,j}||\Delta \mathbf{F}_{So}^{'i,j}]]), j \in \mathcal{N}_i. \end{aligned} \tag{3.5}$$

3.3.4 Symmetric Gated Fusion (SGFM)

For obtained features $\mathbf{F}_S$ and $\mathbf{F}_I$, we develop an effective fusing strategy to adaptively absorb complementary information from the multi-modal contextual representations. For example, depth features encode the scene geometry structure, *e.g.,* the distance from the camera to partial spatial locations. It contributes to inferring the depth of unobserved locations directly. In addition, RGB features contain semantic information and provide prior appearance knowledge of unobserved pixels. Instead of concatenating or summing them together directly with or without attention mechanism, we exploit the proposed SGFM with a symmetric structure, as shown in Figure 3.3. More specifically, at the beginning of the decoder, we employ the convolutional operation followed by a Sigmoid function on $\mathbf{F}_S^L$ to generate the adaptive gating weight $\mathbf{G}_S^L$. By applying the adaptive attention mechanism, the network can absorb meaningful information from the RGB branch and filter out the unrelated. Then we we feed the initial fused feature $[\mathbf{F}_S^L||\mathbf{G}_S^L * \mathbf{F}_I^L]$ into the Residual Block (*abbr.* ResBlock) (He *et al.*, 2016) to obtain the final fused features $\mathbf{Q}_{SI}^L$, which is then fed into a

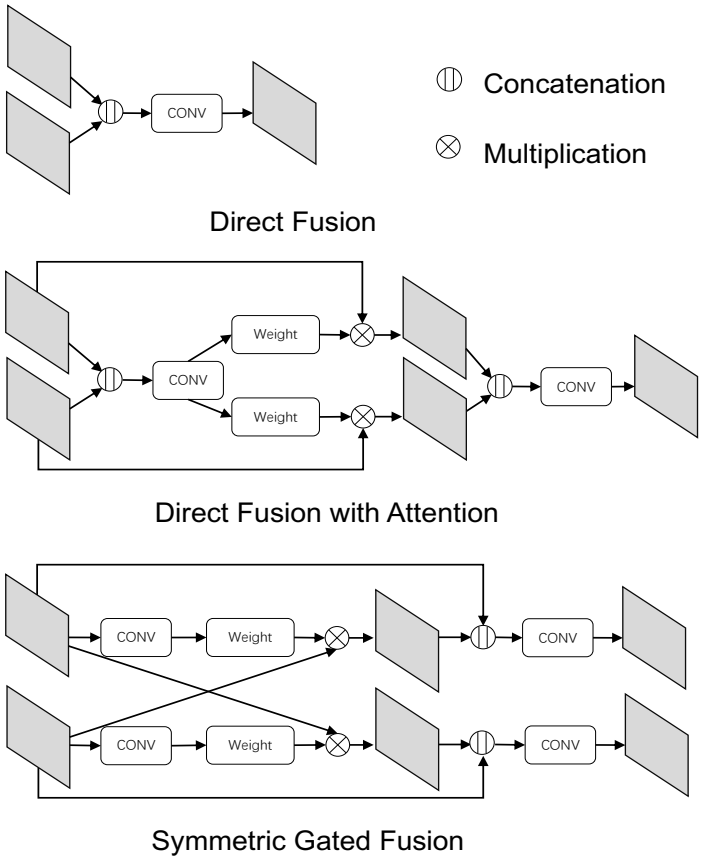


Figure 3.5. Different fusion strategies. Note that, in implementation, we consider the features in both encoder and decoder.

deconvolutional layer to generate $\mathbf{Q}_{SI}^{L,\uparrow}$. Therefore, the depth features can be improved by the complementary information automatically. At the other levels, there is a slight difference in learning the adaptive weights. In specific, at level $l \in \{L-1, L-2, ..., 0\}$, we learn the gating weights $\mathbf{G}_S^l$ using $\mathbf{Q}_{SI}^{l+1,\uparrow}$, rather than $\mathbf{F}_S^l$. Moreover, we feed the concatenated feature $[\mathbf{Q}_{SI}^{l+1,\uparrow}||[\mathbf{F}_S^l||\mathbf{G}_S^l * \mathbf{F}_I^l]]$ into the ResBlock at $l \in \{L-1, L-2, ..., 1\}$ or one convolutional layer at $l = 0$ to get the fused feature. Due to the symmetry of the structure, a similar procedure is employed in the image branch. To illustrate the difference between the proposed fusion strategy and the existing ones, *e.g.,* direct fusion and direct attention fusion, we provide the visual and quantitative comparisons among them in Figure 3.5 and the ablation study, respectively.

3.3.5 Branch Integration

By applying the proposed symmetric gated fusion modules, we obtain two sets of features, one from the image branch and the other from the depth branch. Here, we consider two methods, *i.e.,* end-integration and feature-integration, to integrate them together and then obtain the final prediction result.

3.3.5.1 End-integration

For each branch, we can predict a dense depth map, *i.e.,* $\hat{\mathbf{Y}}_S, \hat{\mathbf{Y}}_I \in \mathbb{R}^{H\times W}$. Since the two branches focus on different information, the reliability of the two predictions varies across the whole image plane. To integrate them adaptively, following (Qiu *et al.*, 2019; Van Gansbeke *et al.*, 2019), we further predict two confidence maps $\mathbf{C}_S, \mathbf{C}_I \in \mathbb{R}^{H\times W}$, which indicate the reliability of the predictions. Therefore, the final dense depth map can be obtained as follows:

$$\hat{\mathbf{Y}} = \frac{exp(\mathbf{C}_S) * \hat{\mathbf{Y}}_S + exp(\mathbf{C}_I) * \hat{\mathbf{Y}}_I}{exp(\mathbf{C}_S) + exp(\mathbf{C}_I)}, \tag{3.6}$$

where $*$ represents the element-wise multiplication.

3.3.5.2 Feature-integration

Apart from the integration in the end, we can also combine the features extracted by the two branches. In specific, as shown in Figure 3.6, we fuse the intermediate features $\mathbf{Q}_{SI}$ and $\mathbf{Q}_{IS}$ through several convolutional operations to obtain $\mathbf{Q}_F$ progressively, and lastly obtain the final prediction $\hat{\mathbf{Y}}$ by applying one convolutional operation on the final integrated features.

3.3.6 Loss Function

The network is mainly driven by a masked mean squared error (MSE) loss between the ground truth semi-dense depth map $\mathbf{Y}$ and the prediction $\hat{\mathbf{Y}}$, which is

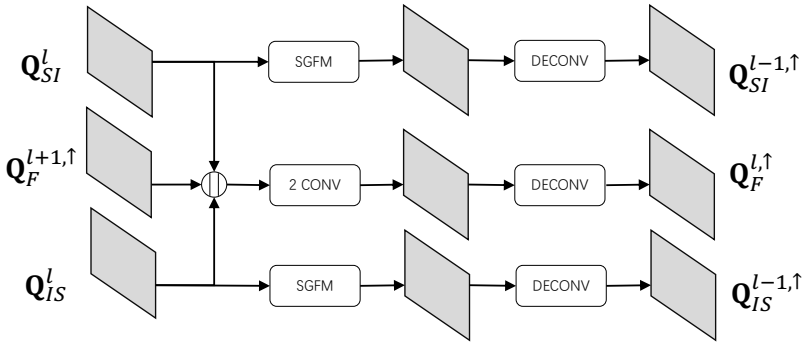


Figure 3.6. Feature-integration. Note that, we ignore some inputs of the SGFM for simplicity.

defined as:

$$\mathcal{L}_{mse}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N_p} \sum_{i,j} \mathbb{I}(\mathbf{Y}^{i,j} > 0)(\mathbf{Y}^{i,j} - \hat{\mathbf{Y}}^{i,j})^2, \tag{3.7}$$

where $\mathbb{I}(\cdot)$ denotes the indication function, and N_p represents the number of labeled pixels. In addition, similar to previous works (Godard *et al.*, 2017), we also apply an edge-aware smoothness loss to encourage depths to preserve spatial continuity:

$$\mathcal{L}_{sm}(\hat{\mathbf{Y}}; \mathbf{X}_I) = \frac{1}{N_s} ||\nabla \hat{\mathbf{Y}}||_1 e^{-||\nabla \mathbf{X}_I||_1}, \tag{3.8}$$

where N_s denotes the number of pixels in the whole image space, and ∇ represents first derivative along spatial directions. Finally, the full objective is:

$$\begin{aligned} \mathcal{L}(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}_S, \hat{\mathbf{Y}}_I, \mathbf{Y}; \mathbf{X}_I) = & \mathcal{L}_{mse}(\hat{\mathbf{Y}}, \mathbf{Y}) + \\ & \gamma_1 \mathcal{L}_{mse}(\hat{\mathbf{Y}}_S, \mathbf{Y}) + \\ & \gamma_1 \mathcal{L}_{mse}(\hat{\mathbf{Y}}_I, \mathbf{Y}) + \\ & \gamma_2 \mathcal{L}_{sm}(\hat{\mathbf{Y}}; \mathbf{X}_I), \end{aligned} \tag{3.9}$$

where γ_1 and γ_2 are the trade-off factors, and are set to 0.5 and 0.01 in our experiments, respectively.

3.4 Experiments

In this section, we first introduce the datasets (Geiger *et al.*, 2012a; Silberman *et al.*, 2012) used in our experiments, and the implementation details. Then we evaluate our method by making comparisons against state-of-the-art methods. Finally, we conduct several ablations to analyze our framework.

3.4.1 Benchmark Datasets

KITTI Depth Completion Benchmark (Geiger *et al.*, 2012a). It is currently the main benchmark for depth completion. The dataset consists of over $90,000$ frames with the ground truth semi-dense depth map for training and validation, and $1,000$ frames without the ground-truth for test. We train depth completion models on the training set, and then evaluate the performance on the official selected validation and test sets. During training, we crop all training data (images and depth maps, 375×1242) to the size of validation and test data, *i.e.,* 352×1216. For evaluation, we adopt the official error metrics: root mean squared error (RMSE in mm, main metric for ranking), mean absolute error (MAE in mm), root mean squared error of the inverse depth (iRMSE in $1/km$), and mean absolute error of the inverse depth (iMAE in $1/km$).

NYU-v2 (Silberman *et al.*, 2012). This dataset consists of RGB and depth images collected from 464 different indoor scenes. According to the official data split strategy, 249 scenes are used for training, and 654 labeled images are selected for evaluating the final performance (Eigen *et al.*, 2014; Laina *et al.*, 2016). In our experiments, we sample around $48k$ images with annotations from the training set for training. Adopting similar experimental setting as (Ma and Karaman, 2018; Cheng *et al.*, 2018), we firstly down-sample all images to half and center-crop them to 304×228, and then sample 500 sparse LiDAR points from the provided dense depth map randomly as the sparse depth data. We exploit root mean square error (RMSE in $meter$), mean absolute relative error

Method	PAR.	FLOPs	Time (s)	Mem.	**RMSE**	MAE	iRMSE	iMAE
SparseConv	-	-	-	-	1601.33	481.27	4.94	1.78
MorphNet	-	-	-	-	1045.45	310.49	3.84	1.57
CSPN	17.41	-	-	-	1019.64	279.46	2.93	1.15
Spade-RGBsD	∼5.3	-	-	-	917.64	234.81	2.17	0.95
HMSNet	-	-	-	-	841.78	253.47	2.73	1.13
DDP	18.8	-	-	-	832.94	203.96	2.10	0.85
NConv-CNN-L2	0.36	305	0.05	5.2	829.98	233.26	2.60	1.03
Sparse2Dense	26.10	1247	0.07	3.7	814.73	249.95	2.80	1.21
PwP	28.99	-	-	-	777.05	235.17	2.42	1.13
Certainty	2.55	111	0.02	5.4	772.87	215.02	2.19	0.93
DeepLiDAR	53.44	3070	0.04	4.0	758.38	226.05	2.56	1.15
UberATG-FuseNet	1.89	-	-	-	752.88	221.19	2.34	1.14
CSPN++	∼26	-	-	-	743.69	209.28	2.07	0.90
NLSPN	25.84	1353	0.14	3.3	741.68	199.59	1.99	0.84
ACMNet	4.9	544	0.08	2.9	744.91	206.09	2.08	0.90

Table 3.1. Quantitative results on the test set of KITTI depth completion benchmark, ranked by **RMSE**. Our method performs better than most of previous methods, and yields close performance to CSPN++ and NLSPN with a much smaller model size (PAR./M). Our model also runs faster than NLSPN, and has lower FLOPs (G) and consumes less GPU memory (Mem./G) than most of approaches during inference. For fair comparison, we run the methods with released code and pretrained models on one Tesla V100 GPU.

(REL in $meter$), and the percentage of relative errors inside a certain threshold (δ_t, $t \in \{1.25, 1.25^2, 1.25^3\}$) as evaluation metrics.

3.4.2 Implementation Details

Graph Construction. For KITTI dataset, we build the graphs at three scales with 10000, 5000, and 2500 observed pixels randomly sampled from the down-sampled sparse depth maps, respectively, and we calculate 6 nearest neighbours for each node. For NYU-v2, we randomly sample 250, 125, and 60 points, respectively. Note that, we can create the graphs using either the 3D coordinates (*e.g.,* camera coordinates) or the 2D coordinates (*e.g.,* pixel coordinates). Here, we use the 3D coordinates, and we will study the differences in ablation studies.

Architecture Details. At each level of the encoder, we employ two CGPMs, and in the decoder, two ResBlocks are utilized in the symmetric gated fusion

module at each scale. The feature channels in the modules are set to 64. Our final results are obtained using the feature-integration, and in this case, we use two convolutional layers, each with 64 output channels at each scale.

Training Details. We implement our depth completion framework in *PyTorch*. In specific, we optimize our network with the momentum of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the initial learning rate of $\alpha = 0.0005$ using the ADAM solver (Kingma and Ba, 2014b). The model is trained for around 40 epochs with a batch size of 8, and the learning rate is delayed by 0.5 every 10 epochs during training.

3.4.3 Comparison against the State-of-the-art

KITTI Dataset. In Table 3.1, we report the number of parameters as well as the performance of our approach and previous peer-reviewed works on KITTI depth completion benchmark. The comparison methods include SparseConv (Uhrig *et al.*, 2017), MorphNet (Dimitrievski *et al.*, 2018), CSPN (Cheng *et al.*, 2018), Spade-RGBsD (Jaritz *et al.*, 2018), HMSNet (Huang *et al.*, 2020), DDP (Yang *et al.*, 2019b), NConv-CNN-L2 (Eldesokey *et al.*, 2019), Sparse2Dense (Ma *et al.*, 2019), PwP (Xu *et al.*, 2019), Certainty (Van Gansbeke *et al.*, 2019), DeepLiDAR (Qiu *et al.*, 2019), UberATG-FuseNet (Chen *et al.*, 2019), CSPN++ (Cheng *et al.*, 2020a), and NLSPN (Park *et al.*, 2020). Note that, some of the existing approaches employ additional data during training. For example, DeepLiDAR renders $50K$ training samples using an open urban driving simulator to train the surface normal prediction network, and Certainty utilizes a pre-trained sematic segmentation model on Cityscapes (Cordts *et al.*, 2016) as network initialization, which can provide high-level semantic information for depth completion. In contrast to these approaches, we train our network from scratch without any additional data. Nevertheless, our approach obtains a convincible improvement over most of the previous methods. In comparison to the latest works, *i.e.,* CSPN++ and NLSPN, our model achieves very close performance, but our model has fewer parameters. Specifically, the RMSE errors of NLSPN

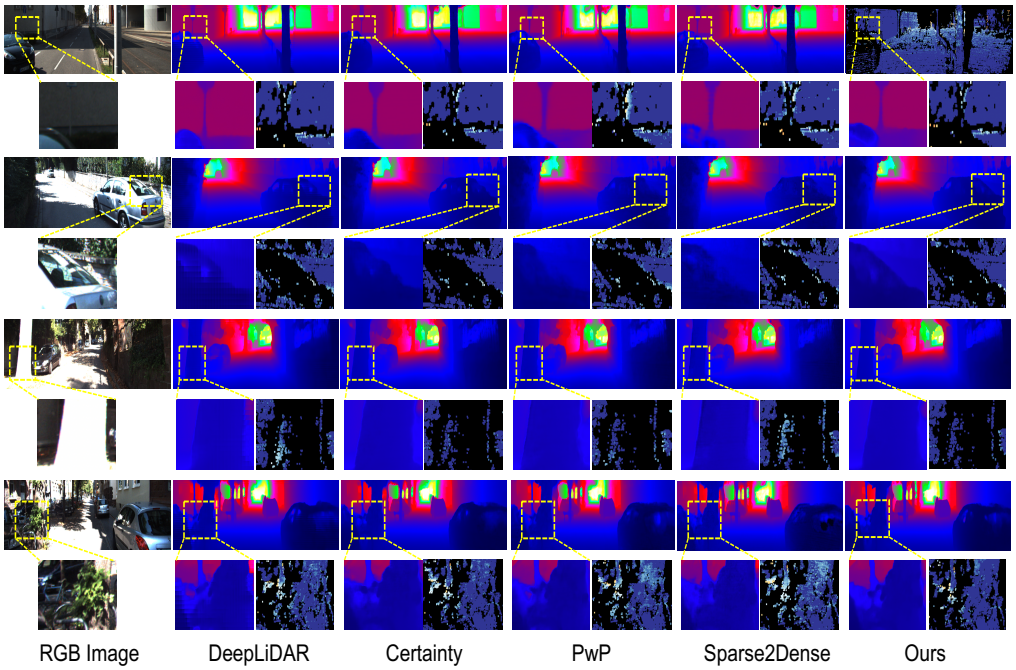


Figure 3.7. Qualitative comparison of our method against four state-of-the-art approaches on KITTI test set. Left to right: RGB image, results of DeepLiDAR, Certainty, PwP, Sparse2dense, and ACMNet, respectively. For better comparison, we show color images, dense predictions, and zoom-in views of details and error maps (darker, better). Best viewed in color.

and CSPN++ are $3mm$ and $1mm$ less than ours, respectively, but the number of their parameters is around four times larger than ours. Moreover, our method runs faster than NLSPN, and has lower FLOPs and GPU memory consumption than most of approaches during inference.

Figure 3.7 shows some qualitative results of ACMNet and other four state-of-the-art methods (Qiu *et al.*, 2019; Van Gansbeke *et al.*, 2019; Xu *et al.*, 2019; Ma *et al.*, 2019). Benefiting from our proposed co-attention guided graph propagation and symmetric gated fusion strategy, which exploit observed pixels' information and capture the heterogeneity of the two modalities efficiently, ACMNet is capable of yielding high-performing dense depth map, preserving more details over boundary regions (*e.g.,* the 2*nd* and 3*rd* examples), and performing better on the tiny/thin objects (the 1*st* example).

Method	RMSE	REL	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
TGV (Ferstl *et al.*, 2013)	0.635	0.123	81.9	93.0	96.8
Bilateral (Silberman *et al.*, 2012)	0.479	0.084	92.4	97.6	98.9
Zhang *et al.* (Zhang and Funkhouser, 2018)	0.228	0.042	97.1	99.3	99.7
Ma *et al.* (Ma and Karaman, 2018)	0.204	0.043	97.8	99.6	99.9
CSPN (Cheng *et al.*, 2018)	0.117	0.016	99.2	**99.9**	**100.0**
DeepLiDAR (Qiu *et al.*, 2019)	0.115	0.022	99.3	**99.9**	**100.0**
Xu *et al.* (Xu *et al.*, 2019)	0.112	0.018	99.5	**99.9**	**100.0**
NLSPN (Park *et al.*, 2020)	**0.092**	**0.012**	**99.6**	**99.9**	**100.0**
ACMNet	0.105	0.015	99.4	**99.9**	**100.0**

Table 3.2. Quantitative results on NYU-v2 with the setting of 500 sparse depth samples. RMSE, REL: lower better; δ_t: higher better.

Models	RMSE	MAE	PARAMs (M)	Time (s)	FLOPs (G)
- FI (D)	789.72	216.65	1.13	0.057	151
- FI (T)	785.97	213.24	1.17	0.068	153
- SG (D)	806.87	220.97	0.72	0.053	86
- SG (T)	801.76	219.18	0.76	0.063	88
- GP	794.13	218.56	1.32	0.018	189
Full (D)	786.89	216.24	1.35	0.061	191
Full (T)	781.66	212.61	1.39	0.072	193

Table 3.3. Investigation on the model with one module disabled. - FI: using end-integration instead of feature-integration, *i.e.,* feat-integration disabled; - SG: removing SGFM and using the direct fusion strategy instead; - GP: removing CGPM. D: default attention operator in CGPM; T: point transformer in CGPM.

NYU-v2 Dataset. As shown in Table 3.2, most of latest works have close performance on this dataset. Our method performs better than almost all of methods except NLSPN (Park *et al.*, 2020), but as stated above the number of our model's parameters is far less than it. In more specific, on NYU-v2, the PARAMs (M), FLOPs (G), Running Time (s), and Inference Memory (GB) of ours / NLSPN / CSPN is 4.9 / 25.8 / 21.8, 122 / 220 / 262, 0.02 / 0.03 / 0.05, and 1.3 / 1.7 / 2.5, respectively.

Method	RMSE	MAE	iRMSE	iMAE
Baseline	815.61	224.43	2.59	1.02
+GP	806.87	220.97	2.42	0.97
+GP/D	810.85	224.64	2.45	0.99
+GP/W	809.09	221.44	2.42	0.97
+SG	796.79	219.86	2.39	0.97
+GP+SG	789.72	216.65	2.32	0.96
+GP/D+SG	792.49	215.14	2.33	0.95
+GP/W+SG	790.75	217.34	2.39	0.97

Table 3.4. Quantitative results on KITTI validation set for ablation study on Graph Propagation. Noticeable improvements gained by +GP demonstrate the effectiveness of our proposed graph propagation module.

Method	RMSE	MAE	iRMSE	iMAE
DF	815.61	224.43	2.59	1.02
DAF	807.35	224.70	2.46	1.00
SG	796.79	219.86	2.39	0.97
GP+DF	807.49	218.74	2.39	0.96
GP+DAF	804.69	221.09	2.44	0.99
GP+SG	789.72	216.65	2.32	0.96

Table 3.5. Investigation for different fusion strategies. DF: direct fusion; DAF: direct fusion with attention mechanism; SG: our proposed adaptive symmetric gated fusion strategy.

3.4.4 Ablation Study

Here, we conduct comprehensive ablation studies on KITTI selected validation dataset to verify the effectiveness of our proposed components. In following experiments, we set the channels of intermediate layers in networks to 32 to speed up model training. Unless otherwise specified, we exploit the end-integration in most cases.

Investigation on the model with one module disabled. At first, to better understand the performance improvement brought by the proposed modules, we conduct a series of experiments by removing one component from the full model each time and observe how the performance changes. The results are shown in

Table 3.3, and we can observe the performance drops with any component disabled. In specific, SGFM can yield lower RMSE than CGPM and requires less time, but CGPM has fewer parameters and lower FLOPs. Therefore, the proposed two modules have both strengths and shortcomings.

In addition, currently there are various ways of modeling the attention guided propagation. As a result, we wonder whether there exist attention operators which can extract the observed contextual information better. To address this problem, we re-implement the attentional weights learning and neighboured information aggregation (Eq. 3.3, Eq. 3.4, and Eq. 3.5) using the Point Transformer operator, which is proposed by the latest work (Zhao *et al.*, 2020a). We refer the original implementation as Default (D), and the re-implementation as Transformer (T). As shown in Table 3.3, the performance can be improved further by the new attention operator with slight increase of FLOPs and PARAMs, which demonstrates the potential of CGPM for bringing improvement. In the following, we make detailed analysis for each module.

The effectiveness of the graph propagation. We demonstrate the effectiveness of the proposed co-attention guided graph propagation by comparing the performance in four cases, *i.e.,* (1) Baseline: no propagation used in the encoder and direct fusion in the decoder; (2) +GP: graph propagation in the encoder and direct fusion in the decoder; (3) +SG: no propagation in the encoder and symmetric gated fusion in the decoder; (4) +GP+SG: our whole model with the end-integration. As shown in Table 3.4, +GP and +GP+SG outperform Baseline and +SG, respectively, which demonstrates that the proposed graph propagation module better captures the spatial contextual information from sparse LiDAR data.

Furthermore, we carry out four additional experiments to analyze in which stage, such as the encoder (*i.e.,* +GP and +GP+SG), decoder (referred as +GP/D and +GP/D+SG), or whole network (referred as +GP/W and +GP/W+SG), the

Graph	RMSE	MAE	iRMSE	iMAE
10K_2D_6NN	792.56	216.31	2.34	0.95
10K_3D_6NN	789.72	216.65	2.32	0.96
10K_3D_3NN	792.13	216.64	2.35	0.96
10K_3D_9NN	795.09	216.57	2.37	0.96
08K_3D_6NN	794.59	216.64	2.36	0.95
12K_3D_6NN	793.61	215.81	2.34	0.95

Table 3.6. Ablation study on the coordinate system and the number of nearest neighbours and sampled points.

Method	RMSE	MAE	iRMSE	iMAE
End-Integration	789.72	216.65	2.32	0.96
Feature-Integration	786.89	216.24	2.28	0.96
EI/Depth	802.66	219.88	2.40	0.97
EI/Image	807.34	223.26	2.47	1.00

Table 3.7. Investigation for the two proposed integration methods.

graph propagation module performs better. As shown in Table 3.4, the comparisons (+GP *v.s.* +GP/D, and +GP+SG *v.s.* +GP/D+SG) indicate that applying the propagation module in the feature extraction stage is more effective in modeling the contextual information. Additionally, we can also observe that compared to +GP (+GP+SG), +GP/W (+GP/W+SG) causes some performance drop. This might be because in the decoder the structure of the observed pixels is not well-preserved after several operations in the encoder.

The effectiveness of the symmetric gated fusion. To verify that the proposed symmetric gated fusion strategy performs better than direct fusion, *e.g.,* concatenation with or without attention (referred as DAF and DF, respectively), we compare six models, *i.e.,* DF (namely Baseline), DAF, SG, GP+DF (namely Baseline+GP in Table 3.4), GP+DAF, and GP+SG. As shown in Table 3.5, SG outperforms both DAF and DF, demonstrating that the proposed symmetric gated fusion strategy is capable of combining the multi-modal information more effectively. Moreover, the comparisons between GP+SG, GP+DAF, and GP+DF can further support this conclusion.

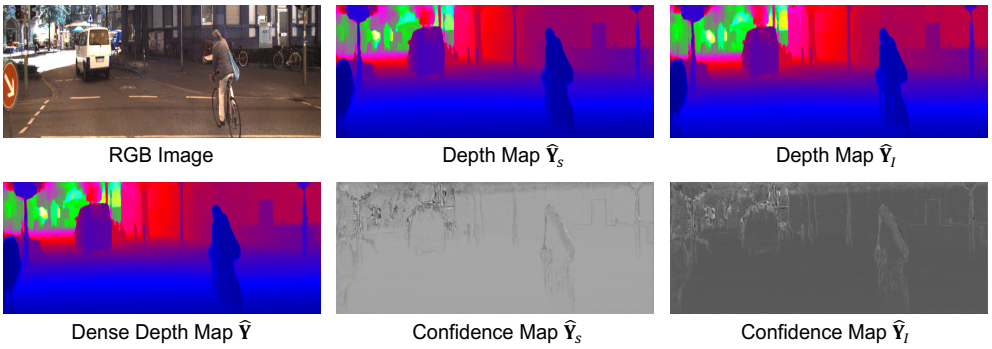
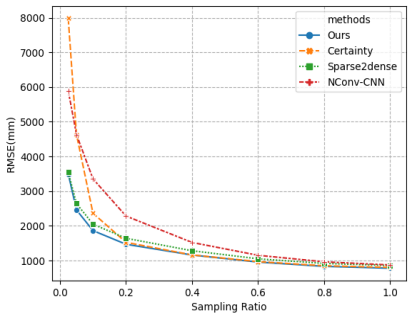


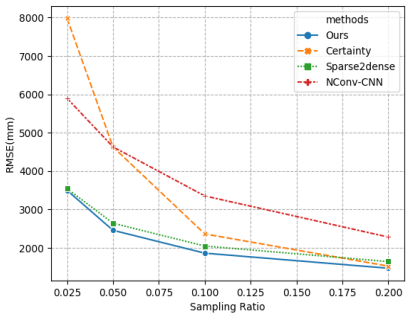
Figure 3.8. Qualitative example of the end-integration. First row: input image, prediction of EI/Depth and EI/Image, respectively; Second row: final prediction, and confidence maps corresponding to the predictions in the first row. We can find that each branch can capture different information.

Analysis of graph construction. Here, we investigate the impacts of three factors involved in constructing graphs. Note that, we conduct the following experiments using our final model with the end-integration. We report the results in Table 3.6.

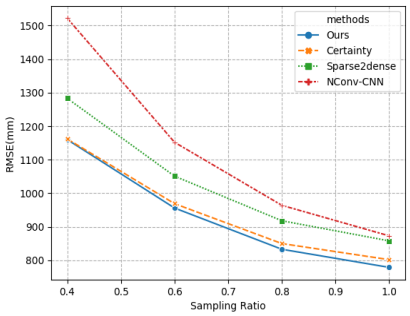
Firstly, since we aim at capturing more observed multi-modal information to enhance the features of unobserved pixels by finding their spatial neighbours, it is interesting to explore the selection of the coordinate system, *i.e.,* pixel coordinate system or camera coordinate system. In specific, for a set of observed pixels, we can construct a graph according to their 2D coordinates $\{(u_i, v_i)\}_{i=0}^{n-1}$ directly or 3D coordinates $\{(x_i, y_i, z_i)\}_{i=0}^{n-1}$, which are obtained according to Eq. 3.10, where f_x, f_y, c_x, c_y denote the camera parameters, and d_i represents the depth value. In Table 3.6, we compare two models (10K_2D_6NN *v.s.* 10K_3D_6NN), where 6-nearest neighbours algorithm is utilized to construct graphs and $10,000$ points are sampled at the first scale. We can find 10K_3D_6NN slightly outperforms 10K_2D_6NN on the RMSE metric. It is mainly because



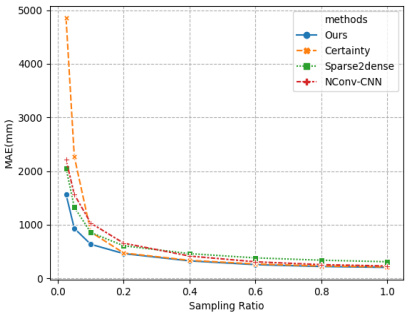
RMSE on All Sampling Ratios



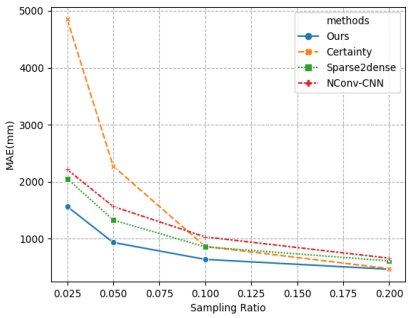
RMSE on Sampling Ratios $0.025, 0.05, 0.1, 0.2$



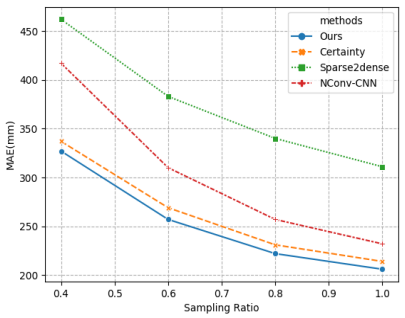
RMSE on Sampling Ratios $0.4, 0.6, 0.8, 1.0$



MAE on All Sampling Ratios



MAE on Sampling Ratios $0.025, 0.05, 0.1, 0.2$



MAE on Sampling Ratios $0.4, 0.6, 0.8, 1.0$

Figure 3.9. Performances under different levels of sparsity. For better comparison, we also show the performances on lower (the second and fifth) and larger (the third and sixth) densities separately. In comparison to Certainty, Sparse2dense, and NConv-CNN, ACMNet performs better under all input densities.

propagation in the camera (3D) coordinate system can learn the scene's geometric structure.

$$\begin{aligned} z_i &= d_i \\ x_i &= \frac{z_i(u_i - c_x)}{f_x} \\ y_i &= \frac{z_i(v_i - c_y)}{f_y} \end{aligned} \tag{3.10}$$

Secondly, we discuss the performance of the model under different numbers of nearest neighbours. By setting k (k nearest neighbours) to different values, *i.e.,* $3, 6, 9$, we train three models, *i.e.,* 10K_3D_3NN ($k = 3$), 10K_3D_6NN ($k = 6$), and 10K_3D_9NN ($k = 9$), all of which propagate features in the camera coordinate system. As shown in Table 3.6, in comparison to 10K_3D_3NN and 10K_3D_6NN, 10K_3D_9NN causes a slight decrease in the performance, it might be because increasing the number of nearest neighbours encourages the model to see unrelated contexts.

Lastly, we study the number of sampled points. In specific, we sample $10,000$, $8,000$, and $12,000$ points at the first scale, respectively, and at the following scales, half of points are sampled from the last scale. From Table 3.6, we can observe that more or fewer points might degrade the performance on the RMSE metric.

In a nutshell, the selection of coordinate system, the number of nearest neighbours and sampled points might affect the performance, but in most settings, the model performs well.

Analysis of branch integration. In Section 3.3.5, we introduce two methods for the integration of the two branches. Here, we analyze their performances. As shown in Table 3.7, the comparison (RMSE: 786 *v.s.* 789) between Feature-Integration (*abbr.* FI) and End-Integration (*abbr.* EI) shows that integration at the feature level is more powerful than the end in learning the reliability of the two branches.

In addition, we also evaluate the performance of the two branches. Taking the end-integration as an example, we report the performance of EI/Depth fusing the RGB information into the depth, and EI/Image doing the opposite. Although the two branches yield close scores on all metrics, by learning confidence maps to fuse them together, a significant improvement on all metrics is obtained. To

understand the two branches deeply, we provide a qualitative example in Figure 3.8. It can be seen that the depth branch is able to generate dense depth map with higher confidence in most locations, while the image branch performs better in capturing the boundary information. This result also further supports that the two modalities are complementary to each other.

3.4.5 Generalization Capabilities

Lastly, we evaluate the generalization capabilities of our method on the sparsity, including the number of observed points and the sparse data pattern.

Number of known points. To show the generalization capabilities of ACMNet on different levels of sparsity, we evaluate our approach and other three state-of-the-art methods with publicly available code, *i.e.*, Certainty (Van Gansbeke *et al.*, 2019), Sparse2dense (Ma *et al.*, 2019), and NConv-CNN (Eldesokey *et al.*, 2019), on KITTI selected validation set under different input densities. In specific, we first uniformly sub-sample the raw LiDAR depth by ratios of $0.8, 0.6, 0.4, 0.2, 0.1, 0.05$, and 0.025 to generate sparse depth maps with different densities, and then test pretrained models on the generated sparse depth maps. Note that, all the models are trained on KITTI training set under the original sparsity (sampling ratio of 1.0) but not fine-tuned on the new sparse depth maps. Figure 3.9 shows that our approach performs better under all input densities in terms of both RMSE and MAE metrics, which demonstrates the impressive generalization capabilities of our approach under different levels of sparsity.

Sparsity pattern. The NYU-v2 dataset provides dense depth maps, so we can evaluate the model on different sparsity patterns through applying different sampling method to generate the sparse depth map. Here, we compare our method with NLSPN (Park *et al.*, 2020) and CSPN (Cheng *et al.*, 2018), which released the code and pretrained models. In specific, we firstly define three patterns, *i.e.,*

Method	RMSE	REL	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
Uniform					
CSPN (Cheng *et al.*, 2018)	0.117	0.016	99.2	99.9	100.0
NLSPN (Park *et al.*, 2020)	0.092	0.012	99.6	99.9	100.0
ACMNet	0.105	0.015	99.4	99.9	100.0
Gaussian					
CSPN (Cheng *et al.*, 2018)	0.121	0.017	99.1	99.8	100.0
NLSPN (Park *et al.*, 2020)	0.093	0.013	99.5	99.9	100.0
ACMNet	0.110	0.017	99.3	99.9	100.0
Grid					
CSPN (Cheng *et al.*, 2018)	0.123	0.017	99.2	99.8	100.0
NLSPN (Park *et al.*, 2020)	0.095	0.013	99.5	99.9	100.0
ACMNet	0.090	0.012	99.6	99.9	100.0

Table 3.8. Quantitative results on NYU-v2 with different sparsity patterns. RMSE, REL: lower better; δ_t: higher better.

Uniform, Gaussian, and Grid. As shown in Figure 3.10, Uniform indicates that we randomly sample n ($= 500$) points from the dense depth map as the known points and each point could be selected with equal probability; Gaussian means that the closer the point is to the central location, the larger the probability of being selected is; Grid means we sample the points regularly. All models, including NLSPN, CSPN, and ACMNet are trained on the Uniform patterns, and then evaluated on all three patterns. As shown in Table 3.8, the scores of all models on RMSE and MAE drop slightly in the Gaussian pattern, since the number of points in two sides of the sparse depth map with Gaussian pattern is fewer than Uniform pattern. In comparison, on the Grid pattern, which can be considered as the easier version of the Uniform, our model generates higher scores. In contrast, other two methods still yield lower scores, which exploit the standard convolutional operation to extract the features. Therefore, our method generalizes well to different sparsity patterns.

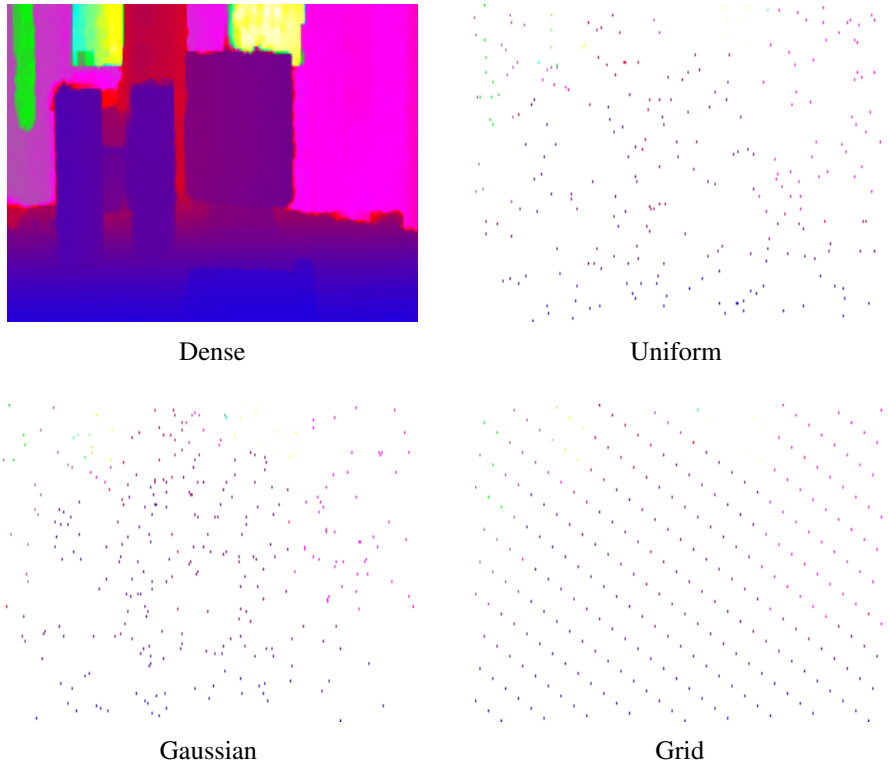


Figure 3.10. Different sparsity patterns. Zoom in for best view.

3.5 Conclusion

In this chapter, we have developed an Adaptive Context-Aware Multi-Modal Network (ACMNet) to recover a dense depth map from sparse LiDAR data and dense RGB data. The critical issue in depth completion is how to exploit the observed spatial contexts from multi-modal data efficiently. To this end, we apply the co-attention guided graph propagation within multiple graphs constructed from observed pixels, which adaptively extracts multi-scale and multi-modal features and contributes to the feature enhancement for unobserved pixels. Furthermore, to fuse the multi-modal features in an effective way, we propose the symmetric gated fusion strategy, which has the capability of learning the heterogeneity of the two modalities. Finally, we implement our ACMNet, where a stack of CGPMs are employed in the encoder and SGFMs are used in the decoder. Benefiting from the two new modules, ACMNet is capable of generating

high-quality dense depth maps. Our extensive experiments have demonstrated the effectiveness of the network as well as the network components.

CHAPTER 4

Adaptive Edge-to-Edge Interaction Learning for Point Cloud Analysis

---

Previous chapters study 3D structure prediction from single images and multi-modal data, *i.e.,* monocular depth estimation and depth completion, respectively. In this and next chapter, we are investigating 3D information understanding through exploring local shape representation and model generalization for 3D point cloud analysis, respectively. Recent years have witnessed the great success of deep learning on various point cloud analysis tasks, *e.g.,* classification and semantic segmentation. Since point cloud data is sparse and irregularly distributed, one key issue for point cloud data processing is extracting useful information from local regions. To achieve this, previous works mainly extract the points' features from the local region through learning the relation between each pair of adjacent points. However, these works ignore the relation between edges in the local region, which encodes the local shape information. Associating the neighbouring edges could potentially make the point-to-point relation more aware of the local structure and more robust. To explore the role of the relation between edges, this chapter proposes a novel **A**daptive **E**dge-to-**E**dge **I**nteraction **L**earning module (AE^{2}IL), which aims to enhance the point-to-point relation through modeling the edge-to-edge interaction in the local region adaptively. We further extend the AE^{2}IL module to a symmetric version, named SymAE^{2}IL, to capture the local structure more thoroughly. Taking advantage of the proposed modules, we develop two networks, AE^{2}INetCls and AE^{2}INetSeg,

for shape classification and segmentation tasks, respectively. Various experiments on several public point cloud datasets show that our models achieve state-of-the-art performance for point cloud analysis.

4.1 Introduction

In recent years, a lot of works have been made to exploit deep learning for 3D point cloud analysis, which is important for real-world applications, such as autonomous driving (Li *et al.*, 2019c) and robotics manipulation (Kim and Sukhatme, 2014). Since point cloud data does not have a regular structure like images, it cannot be processed straightforwardly in the deep convolutional neural networks (DCNNs). To address this issue, some works (Maturana and Scherer, 2015; Wu *et al.*, 2015; Wang *et al.*, 2017) propose to voxelize the points and obtain the volumetric representation, which can be fed into the conventional CNNs. However, these voxel-based approaches often suffer from quantization loss of the structure due to the low resolution caused by voxelization.

Another solution is designing a deep model that can learn representations from the raw point cloud directly. The pioneering work of this clue, *i.e.,* PointNet (Qi *et al.*, 2017a), extracts the features from each point directly using the multi-layer perceptron (MLP). Despite being efficient, however, it omits the local structure, which is significant for learning discriminative representations. PointNet++ (Qi *et al.*, 2017b), as an extension of PointNet, attempts to model the local shape by introducing a hierarchical encoder-decoder structure with point sampling and feature propagation operations.

Following these two works, a lot of variants (Wang *et al.*, 2019d; Liu *et al.*, 2019d; Fujiwara and Hashimoto, 2020; Li *et al.*, 2018e; Wang *et al.*, 2019b) have been proposed to extract discriminative features from the local regions. A typical clue they exploit is aggregating the information of a point into its neighbours according to their relationships, *i.e.,* the edge's features. For example,

DGCNN (Wang *et al.*, 2019d) proposes the EdgeConv operation that extracts the local features from the center point and the edges (*e.g.,* spatial relative position) between it and its neighbours. Another interesting work, RS-CNN (Liu *et al.*, 2019d) maps the predefined geometric priors between two adjacent points into a high-level relation expression, and then considers it as weights to aggregate the local contextual information. These methods improve the performance of several typical point cloud tasks, including shape classification, part segmentation, and semantic segmentation, remarkably.

However, as these methods solely model the point-to-point relation for each pair of adjacent points, the learned representation for the edge might lack the local structure information, making the relation not discriminative and not robust. For example, given two point pairs (p_1, p_2) and (p_3, p_4), if we calculate the relation for each pair respectively, then we might get close results, even though they locate in different objects. In comparison, if we exploit the local structure to enhance the point-to-point relation ($p1$ and p_2, p_3 and p_4) through considering other edges in the same local region, then the results could be more distinctive.

Based on the analysis above, we propose an Adaptive Edge-to-Edge Interaction Learning (AE^{2}IL) module. Specifically, for a point, we first find its K neighbours, and thus there are K edges emanating from it to its neighbours. Then, for each edge we consider other edges' information through modeling the edge-to-edge interaction in three steps: 1) find its nearest neighbours from the K edges; 2) learn the relation between it and its neighbours; 3) use these learned relations to update its information. Furthermore, to model the local structure thoroughly and explore the reverse edges, we extend the AE^{2}IL to a symmetric version, namely SymAE^{2}IL, the details of which can be found in the third section.

Taking advantage of the (symmetric) adaptive edge-to-edge interaction learning modules, we develop two networks, *i.e.,* AE^{2}INetCls and AE^{2}INetSeg, for shape classification and segmentation, respectively. The experimental results

show that the designed models outperform previous approaches on several public point cloud datasets, and achieve state-of-the-art performance.

4.2 Related Work

Currently, many efforts have been made to exploit deep learning on point cloud processing. In this section, we briefly review some of them, especially those point based methods. We refer to the survey (Guo *et al.*, 2020) for a thorough understanding.

Following PointNet (Qi *et al.*, 2017a), which is the first attempt to apply deep learning directly on the sparse and unstructured point sets, and its extension PointNet++ (Qi *et al.*, 2017b), a lot of efforts (KIM *et al.*, 2020; Landrieu and Simonovsky, 2018; Wu *et al.*, 2019; Lang *et al.*, 2020; Liu *et al.*, 2019a; Liu *et al.*, 2020; Wang *et al.*, 2018a; Wong and Vong, 2020; Huang *et al.*, 2018; Atzmon *et al.*, 2018; Le *et al.*, 2020; Nezhadarya *et al.*, 2020; Wang *et al.*, 2019c; Hua *et al.*, 2018; Xiang Zhang, 2021; Liu *et al.*, 2019b; Xu *et al.*, 2018c; Komarichev *et al.*, 2019; Chen *et al.*, 2021) have investigated the feature extraction of the local structure. Most of these point based works mainly focus on one or more of the following components: 1) points sampling, 2) relation learning, and 3) convolutional operation.

Points Sampling. To capture the contextual information in a hierarchical structure, PointNet++ (Qi *et al.*, 2017b) exploits the *farthest point sampling* (FPS) algorithm to sample a subset from the input points. Since the FPS algorithm is permutation-variant and samples points from low-dimension Euclidean space, PAT (Yang *et al.*, 2019a) proposes Gumbel Subset Sampling to select the subset, which is more robust to outliers. In comparison, to overcome the issue existing in FPS, PointASNL (Yan *et al.*, 2020) proposes an adaptive sampling

strategy to refine the initial sampled points, which considers both low- and high-dimension embedding space. Interestingly, a recent work, RandLA-Net (Hu *et al.*, 2020a) compares several point sampling approaches, and observes that the Random Sampling strategy is more suitable for large-scale point clouds.

Relation Learning. To represent the local structure, PointNet++ (Qi *et al.*, 2017b) extracts the features of each point in the local region using the MLP and then exploits the Max-Pooling operation to get the local region feature vector. However, it ignores the geometric relationships between points, which causes the limitation on the modeling of local structures. To improve this, DGCNN (Wang *et al.*, 2019d) exploits the proposed EdgeConv on the constructed local neighbourhood graph to model local geometric structures. EdgeConv aggregates the features of the edges emanating from the central point of the local region as its new representation. In comparison, some works aim to map the edge features into weights for feature association. For example, extending regular 2D CNN to irregular configuration, RS-CNN (Liu *et al.*, 2019d) encodes the predefined geometric priors, *e.g.,* the spatial distance, between two adjacent points as a high-level relation expression, *i.e.,* weight vector. Similarly, GAC (Wang *et al.*, 2019b) learns attentional weights from both spatial and feature distances. One point in common among these works is that in a local region, only the edges connecting the central point to the others are considered. In comparison, PointWeb (Zhao *et al.*, 2019a) constructs a densely-connected graph and aims to find the interaction between all adjacent points for better description of the local structure. This chapter attempts to learn the edge-to-edge interaction adaptively for the enhancement of the point-to-point relation.

Convolutional Operation. Motivated by the standard 2D convolution kernel, KPConv (Thomas *et al.*, 2019) designs a set of 3D kernel points. The kernel points are used to define the area where the kernel weights are applied to extract

the features. A recent work, PAConv (Xu *et al.*, 2021a), proposes to construct the convolutional kernels by dynamically assembling basic weight matrices in a pre-defined Weight Bank. In addition, some works attempt to apply projection in the feature space so that the projected features can be processed by the standard convolutional operation directly. Fox example, PointCNN (Li *et al.*, 2018e) transforms the points in a local region to the canonical order through learning a transformation matrix and then the traditional convolution can be applied. In comparison with PointCNN, FPConv (Lin *et al.*, 2020a) learns a weight map to softly project local points onto a 2D grid, which is further processed by the regular 2D convolutional operation.

4.3 Our Approach

In this section, we present the proposed novel adaptive edge-to-edge interaction learning modules, AE^{2}IL and its symmetric version SymAE^{2}IL in detail. In addition, we also provide an analysis for the differences between an existing work and our approach in the last.

4.3.1 AE^{2}IL Module

Let $\mathcal{P} = \{p_1, p_2, ..., p_N\} \subset \mathbb{R}^3$, where p_i represents the point's spatial position, denote the processed point cloud consisting of N points, $\mathcal{F} = \{f_1, f_2, ..., f_N\} \subset \mathbb{R}^{C_{in}}$ denote the corresponding feature set, where C_{in} is the number of channels. Our new AE^{2}IL module takes $\mathcal{P}$ and $\mathcal{F}$ as the input, and outputs the enhanced representation $f_i^o \in \mathbb{R}^{C_{out}}$ for each point p_i in $\mathcal{P}^s$. $\mathcal{P}^s$ is a subset sampled from $\mathcal{P}$ via the FPS technique (Qi *et al.*, 2017b), and C_{out} is the channel number.

Specifically, for a certain point p_i, we find its K neighbours $\mathcal{N}(p_i) \subseteq \mathcal{P}$ from $\mathcal{P}$. This is implemented by the K-nearest neighbour (K-NN) algorithm according to the spatial distance. To extract the local features, we follow (Liu *et al.*, 2019d;

Wang *et al.*, 2019c; Wang *et al.*, 2019b) to first encode the point-to-point relation (h_{ij}) based on the spatial relative position and difference between features as:

$$h_{ij} = \sigma([(p_j - p_i)||(f_j - f_i])),\ p_j \in \mathcal{N}(p_i), \tag{4.1}$$

where $[\cdot||\cdot]$ is the concatenation operation, and $\sigma(\cdot)$ denotes a mapping function. In this chapter, we use MLP as the function. Given the learned point-to-point relation h_{ij}, we can integrate the point feature f_i and its K neighbouring features f_j following (Liu *et al.*, 2019d; Wang *et al.*, 2019c; Wang *et al.*, 2019b). However, as discussed before, only studying the relations between two points solely may fail to model the local structure well. We thus seek a solution by adaptively investigating the interaction between the edges.

In detail, for the central point p_i we have K directed edges[1]. Among them, the edge e_{ij} emanates from point p_i to its neighbour $p_j \in \mathcal{N}(p_i)$. We then utilize K-NN algorithm to find the K_e nearest neighbours for each edge according to the distance between edges. Here, the distance between e_{ij} and e_{ik} is computed as:

$$Dist(e_{ij}, e_{ik}) = \sqrt{||p_j - p_k||_2}, \tag{4.2}$$

i.e., the spatial Euclidean distance between the terminal points of the edges. For a specific edge e_{ij}, we denote its neighbours as $e_{ik} \in \mathcal{N}(e_{ij})$, where e_{ik} represents the edge from p_i to its neighbour $p_k \in \mathcal{N}(p_i)$. We take $E_{i,jk}$ to represent the *edge* from e_{ij} to e_{ik}, and define $H_{i,jk}$ as their interaction. These symbolic marks are illustrated in Figure 4.1.

To obtain $H_{i,jk}$, we extract the information of *edge* $E_{i,jk}$. Here, we consider three kinds of information. The first two are the spatial relative position $D^s_{i,jk}$ and difference between features $D^f_{i,jk}$. Before computing $D^f_{i,jk}$, we first use

[1]Note that, in this chapter the edge e_{ij} between p_i and p_j is directed, where p_i is the staring / emanating point and p_j is the terminal point.

MLPs (ϕ and ψ) to encode the edges' features, and then calculate the two relations as:

$$\begin{aligned} D^s_{i,jk} &= p_k - p_j, \\ D^f_{i,jk} &= \phi(h_{ik}) - \psi(h_{ij}). \end{aligned} \tag{4.3}$$

Apart from $D^s_{i,jk}$ and $D^f_{i,jk}$, we further compute the surface normal $D^c_{i,jk}$ as follows:

$$D^c_{i,jk} = (p_k - p_i) \times (p_j - p_i), \tag{4.4}$$

where $\times$ denotes the cross product operation. By taking $D^s_{i,jk}$, $D^f_{i,jk}$, and $D^c_{i,jk}$ as the inputs, we first encode the spatial information into a new feature vector and then capture $H_{i,jk}$ via a summation operation:

$$H_{i,jk} = D^f_{i,jk} + \gamma(D^s_{i,jk}||D^c_{i,jk}), \tag{4.5}$$

where γ is an MLP. Therefore, the edge-to-edge interaction $H_{i,jk}$ not only encodes the relations between edges in both low- (Euclidean) and high- (feature) dimensional space, but also considers the plane information. We refer to the ablation studies for more analysis.

The next goal is to update the point-to-point relation h_{ij} using the learned edge-to-edge interactions between e_{ij} and its K_e neighbours $\mathcal{N}(e_{ij})$ via the attention technique (Hu *et al.*, 2020a; Vaswani *et al.*, 2017). In specific, we first calculate the attentional weights as follows:

$$\begin{aligned} w_{i,jk} &= \alpha(H_{i,jk}), \\ w^{'}_{i,jk} &= \frac{exp(w_{i,jk})}{\sum_{l \in \dot{\mathcal{N}}(e_{ij})} exp(w_{i,jl})}, \end{aligned} \tag{4.6}$$

where α is an MLP, and $\dot{\mathcal{N}}(e_{ij})$ denotes an index set containing the index of the terminal point of the edges in $\mathcal{N}(e_{ij})$. We then aggregate $\{h_{ik}\}_{k \in \dot{\mathcal{N}}(e_{ij})}$ and the spatial relation $\gamma(D^s_{i,jk}||D^c_{i,jk})$ using the learned attentional weights as follows:

$$h^{'}_{ij} = \sum_{k \in \dot{\mathcal{N}}(e_{ij})} w^{'}_{i,jk} \cdot (\beta(h_{ik}) + \gamma(D^s_{i,jk}||D^c_{i,jk})), \tag{4.7}$$

where $\cdot$ denotes the element-wise multiplication operation, and β is an MLP.

Now, we achieve the interaction between the edges emanating from p_i, and get the enhanced representation $h_{ij}^{'}$ for each directed edge e_{ij}. Lastly, we employ three consecutive operations, *i.e.,* a shared MLP μ and a max-pooling operation, and a residual connection (Hu *et al.*, 2020a; He *et al.*, 2016), to update the features of p_i so that it contains the extracted local structure information, *i.e.,*

$$\begin{aligned} f_{ij} &= \mu([f_i||h_{ij}^{'}]), \\ \langle f_i^o \rangle_c &= \max_{j \in [1,2,...,K]} \langle f_{ij} \rangle_c + \langle \rho(f_i) \rangle_c, \end{aligned} \tag{4.8}$$

where $c \in [1, 2, ..., C_{out}]$, ρ is an MLP, and $\langle f \rangle_i$ is the i^{th} element of the feature vector f.

4.3.2 SymAE^{2}IL Module

For a local region centering on point p_i, AE^{2}IL exploits the local structure to enhance the point-to-point relation h_{ij} through learning the edge-to-edge interactions between e_{ij} and its neighbours $\mathcal{N}(e_{ij})$. Taking an example of one of its neighbours e_{ik} ($j \neq k$) starting from p_i to p_k. AE^{2}IL models the interaction between e_{ij} and e_{ik}, but overlooks the reverse edge e_{ji}. We find that further studying the interaction between e_{ji} and e_{jk} could exploit the local structure better. From another perspective, through modeling the two interactions, we can also extract the structure information contained in the triangle constructed by p_i, p_j, and p_k. We call this new module as SymAE^{2}IL, *i.e.,* Symmetric Adaptive Edge-to-Edge Interaction Learning, which simultaneously learns the representations of both e_{ij} and e_{ji}.

To achieve this, we first define a new edge set $\hat{\mathcal{N}}(e_{ji})$, which contains all edges emanating from p_j to the terminal point of the edges in $\mathcal{N}(e_{ij})$, as the neighbours of the edge e_{ji}. Then, we compute the feature of each edge in $\hat{\mathcal{N}}(e_{ji})$, and update the feature of e_{ji} as we do to update the feature of e_{ij}, *i.e.,* from Eq. 4.3

to Eq. 4.7. This process is illustrated by the boxes in the bottom of Figure 4.1. Denoting the output feature of e_{ji} by $\hat{h}'_{ji}$, then we reformulate Eq. 4.8 as:

$$\begin{aligned} f_{ij} &= \mu([f_i||(h'_{ij} + \hat{h}'_{ji})]), \\ \langle f_i^o \rangle_c &= \max_{j \in [1,2,...,K]} \langle f_{ij} \rangle_c + \langle \rho(f_i) \rangle_c. \end{aligned} \tag{4.9}$$

We can use AE^{2}IL or SymAE^{2}IL as a basic operator to construct deep networks for point cloud analysis, including shape classification and segmentation. In our experiments, we use SymAE^{2}IL as the core operation, and we will study the two modules in the ablation. The networks for segmentation and classification are named as AE^{2}INetSeg and AE^{2}INetCls, respectively. The structures of the networks and detailed architectures of AE^{2}IL, SymAE^{2}IL, AE^{2}INetCls, and AE^{2}INetSeg are provided in Sec. 4.3.4.

4.3.3 Relation to PointWeb

In this chapter, we consider the edge-to-edge interaction, which involves the point-to-point relation between the neighbours of the central point as well as the central point and its neighbours. Technically, the feature of an edge are updated with its neighbouring edges, while not in most of existing works. However, an interesting work, PointWeb (Zhao *et al.*, 2019a), develops the Adaptive Feature Adjustment (AFA) module to update all points in the local region before the feature aggregation. As a result, the point-to-point relation in PointWeb can be viewed as being updated. Here, we analyze the differences between the AFA strategy in PointWeb and ours.

Firstly, we consider a simplified implementation of our AE^{2}IL, defined as:

$$h'_{ij} = \sum_{k \in \dot{\mathcal{N}}(e_{ij})} \omega(h_{ik} - h_{ij}) \cdot h_{ik}, \tag{4.10}$$

where $h_{ij} = \sigma(f_j - f_i)$, and ω is an MLP followed by a *SoftMax* function. We can re-write the equation as:

$$\begin{aligned} h_{ij}^{'} &= \sum_{k \in \dot{\mathcal{N}}(e_{ij})} \omega(h_{ik} - h_{ij}) \cdot h_{ik} \\ &= \sum_{k \in \dot{\mathcal{N}}(e_{ij})} \omega(\sigma(f_k - f_i) - \sigma(f_j - f_i)) \cdot \sigma(f_k - f_i) \\ &= \sum_{k \in \dot{\mathcal{N}}(e_{ij})} \omega(\sigma(f_k - f_j)) \cdot \sigma(f_k - f_i) \\ &= \sum_{k \in \dot{\mathcal{N}}(e_{ij})} \delta(\tau(f_k - f_j)) \cdot \sigma(f_k - f_i), \end{aligned} \tag{4.11}$$

where δ denotes the *SoftMax* function and τ is an MLP. Similarly, the symmetric version of the simplified AE^{2}IL can be written as:

$$\begin{aligned} h_{ij}^{'} = &\sum_{k \in \dot{\mathcal{N}}(e_{ij})} \delta(\tau(f_k - f_j)) \cdot \sigma(f_k - f_i) + \\ &\sum_{k \in \dot{\mathcal{N}}(e_{ij})} \delta(\tau'(f_k - f_i)) \cdot \sigma'(f_k - f_j), \end{aligned} \tag{4.12}$$

where τ' and σ' are both MLPs. Using our notations, we can represent the $h_{ij}^{'}$ in AFA module as follows:

$$\begin{aligned} h_{ij}^{'} =& f_j^{'} - f_i^{'} \\ =& f_j + \sum_{k \in \dot{\mathcal{N}}(e_{ij})} \xi(f_j - f_k) \cdot (f_j - f_k) - [f_i + \sum_{k \in \dot{\mathcal{N}}(e_{ij})} \xi(f_i - f_k) \cdot (f_i - f_k)] \\ =& \sum_{k \in \dot{\mathcal{N}}(e_{ij})} [\xi(f_j - f_k) \cdot (f_j - f_k) - \xi(f_i - f_k) \cdot (f_i - f_k)] + h_{ij} \\ =& \sum_{k \in \dot{\mathcal{N}}(e_{ij})} [\xi(f_j)(f_j - f_k) + \xi(f_i)(f_k - f_i) + \xi(f_k)(f_i - f_j)] + h_{ij}, \end{aligned} \tag{4.13}$$

where ξ is an MLP. Note that, PointWeb considers $i = k$ and $j = k$ as a special case, while we ignore that for simplicity, which does not affect the conclusion.

Taking an example of three points $\{p_i, p_j, p_k\}$, where $p_j, p_k \in \mathcal{N}(p_i)$ and $k \in \dot{\mathcal{N}}(e_{ij})$, we analyze the differences between PointWeb and ours. Firstly, comparing Eq. 4.11, Eq. 4.12, and Eq. 4.13, we can find that when using the edge e_{ik} to update e_{ij}, our method exploits the relation between the two edges, which is the difference between the terminal points' features in the simplified version. Although PointWeb also exploits the neighbouring edges, it does not involve the interaction between the edges. Instead, it sums the edges' features directly taking the points' features as the weights, or Eq. 4.13 can be explained as that it sums the new points' features taking the difference between points' features as the weights. In comparison, we update the edge's feature according to the relation between edges adaptively. In addition, through modeling the edge-to-edge interaction directly, we can exploit complicated functions (Eq. 4.3-Eq. 4.7 *v.s.* Eq. 4.10), such as the learning of both low- and high-level relations between edges, easily. As a result, in comparison to PointWeb, our method is able to learn the edge-to-edge interaction more effectively and thus model the local structure better.

4.3.4 Network Details

In the following, we introduce the architecture details of the proposed two networks, *i.e.,* AE^{2}INetCls and AE^{2}INetSeg, respectively. Before that, We first introduce the architectures of AE^{2}IL and SymAE^{2}IL, which are used as the basic operator to construct deep networks for point cloud analysis. In our experiments, we use SymAE^{2}IL as the core operation.

Architectures of AE^{2}IL and SymAE^{2}IL. The main operations (functions) in AE^{2}IL and SymAE^{2}IL include MLP, Leaky-ReLU, and batch normalization (BN). In specific,

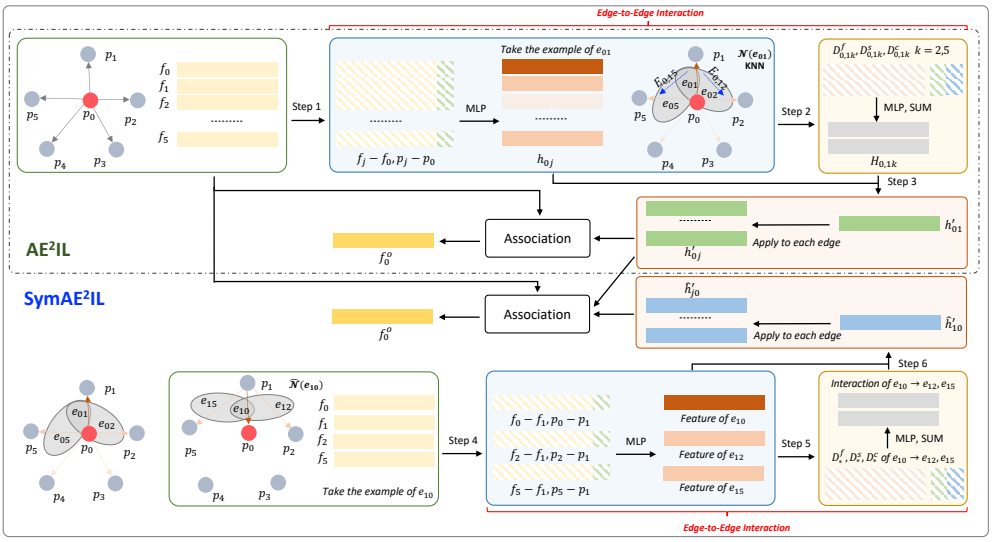


Figure 4.1. Illustration of AE^{2}IL and SymAE^{2}IL. AE^{2}IL marked by the gray dotted line consists of Step 1, Step 2, and Step 3, while SymAE^{2}IL marked by the gray solid line contains all steps. The edges located in one gray ellipse are neighbours. In this example, p_0 is the central point, and p_1-p_5 are its neighbours. Note that, from Step 2 to Step 6, we take the example of the edges e_{01} and e_{10}. The blue dotted arrow denotes the $edge$ between edges. The two association operations are formulated as Eq. 4.8 (for the top one) and Eq. 4.9 (for the below one), respectively. Notations are identical to the text. Best viewed in color (zoom in for details).

- σ in Eq. 4.1 contains a shared single-layer MLP followed by one BN layer and one Leaky-ReLU layer;
- ρ and μ in Eq. 4.8 and Eq. 4.9 contain a shared single-layer MLP followed by one BN layer, and there is a Leaky-ReLU layer after the summation operation in Eq. 4.8 and Eq. 4.9;
- ϕ and ψ in Eq. 4.3, and β in Eq. 4.7 contain a shared single-layer MLP;
- γ in Eq. 4.5 and α Eq. 4.6 in contain a shared two-layer MLP, where the first one is followed by one BN layer and one Leaky-ReLU.

AE^{2}INetCls for Classification. AE^{2}INetCls contains three consecutive SymAE^{2}ILs. In the l^{th} layer, given N_{l-1} points as the input, we sample N_l points ($N_{l-1} > N_l$) using the FPS algorithm (Qi *et al.*, 2017b). Taking the samples as the central point, the SymAE^{2}IL module is able to extract the structure information of the corresponding local regions. For the ModelNet40 dataset (Wu *et al.*, 2015),

we feed $1,024$ points ($N_0 = 1,024$) into the network. In all layers, we consider 32 ($K = 32$) nearest neighbours for each sampled central point. The number of central points sampled in each layer is $512 \rightarrow 128 \rightarrow 32$, respectively. The output feature dimension (C_{out}) is 128, 256, and 512, respectively. In addition, in each module, for the features obtained from the functions $\{\sigma\}$ and$\{\phi, \psi, \alpha, \beta, \gamma\}$, we set the number of channels as $\frac{C_{out}}{2}$ and $\frac{C_{out}}{4}$, respectively.

AE^{2}INetSeg for Segmentation. For the segmentation tasks, including part segmentation and semantic segmentation, we build the networks on the encoder-decoder architecture with skip connections, which is exploited widely by most previous works (Qi *et al.*, 2017b; Zhao *et al.*, 2019a; Zhang *et al.*, 2019b). The encoder part consists of a stack of SymAE^{2}ILs, which is identical to AE^{2}INetCls. After the encoder, the original point is sub-sampled, so we need to upsample the encoded features into the original resolution progressively. In specific, for the upsampling layer corresponding to the l^{th} layer in the encoder, we aim to propagate point features from N_l points to N_{l-1} points, which can be achieved by the Feature Propagation Operation in PointNet++ (Qi *et al.*, 2017b). The output feature dimension (C_{out}) of each module in the encoder is 64, 128, 256, 512, and 512, respectively. As in AE^{2}INetCls, in each module, for the features obtained from the functions $\{\sigma\}$ and$\{\phi, \psi, \alpha, \beta, \gamma\}$, we set the number of channels as $\frac{C_{out}}{2}$ and $\frac{C_{out}}{4}$, respectively. For ShapeNetPart (Yi *et al.*, 2016) ($N_0 = 2,048$), S3DIS (Armeni *et al.*, 2016) ($N_0 = 14,000$), and ScanNet v2 (Dai *et al.*, 2017a) ($N_0 = 14,000$), in each SymAE^{2}IL, we sample $1,024/4,096/4,096 \rightarrow 512/1,024/1,024 \rightarrow 256/256/256 \rightarrow 128/128/128 \rightarrow 64/64/64$ points, respectively. In each layer of the encoder, we consider 16 ($K = 16$) nearest neighbours for each sampled central point. In each layer of the decoder, we exploit the FPS to upsample the point features, which is followed by a shared two-layer MLP to update the points' features. The output feature dimensions in the decoder are symmetrical to the input feature dimensions in the encoder.

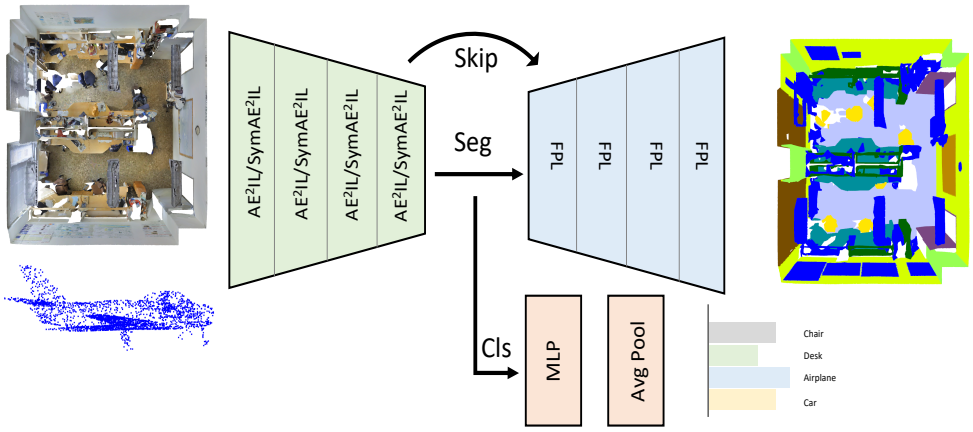


Figure 4.2. AE^{2}INetCls and AE^{2}INetSeg. FPL, Seg, Cls, and Skip denote feature propagation layer (Qi *et al.*, 2017b), segmentation, classification, and skip connections, respectively.

The structures of AE^{2}INetCls and AE^{2}INetSeg are shown in Figure 4.2.

4.4 Experiments

To examine the effectiveness of our point cloud analysis approach, we conduct experiments on several tasks, including semantic segmentation, part segmentation, and classification, on widely studied benchmarks, such as S3DIS (Armeni *et al.*, 2016), ScanNet v2 (Dai *et al.*, 2017a), ShapeNetPart (Yi *et al.*, 2016), and ModelNet40 (Wu *et al.*, 2015). The results and ablations for our modules are reported in the following, while some detailed results and visualization examples are provided in Sec. 4.5.

4.4.1 Implementation Details

We train all networks with an initial learning rate of 0.1 using the SGD optimization algorithm. For the semantic segmentation task on S3DIS (ScanNet v2), we train for 100 (1000) epochs with a batch size of 8 (10) and decay the learning

Method	6-fold			Area 5		
	oA	mAcc	mIoU	oA	mAcc	mIoU
PointNet	78.6	66.2	47.6	-	49.0	41.1
PointNet++	81.0	67.1	54.5	-	-	-
PointCNN	88.1	75.6	65.4	85.9	63.9	57.3
DGCNN	84.1	-	56.1	-	-	-
PointWeb	87.3	76.2	66.7	87.0	66.6	60.3
HPEIN	88.2	76.3	67.8	87.2	68.3	61.9
KPConv*	-	79.1	70.6	-	72.8	67.1
FPConv	-	-	68.7	88.3	68.9	62.8
SegGCN	-	-	-	88.2	70.4	63.6
RandLA-Net	88.0	82.0	70.0	-	-	-
Point2Node	89.0	79.1	70.0	88.8	70.0	63.0
SCF-Net*	88.4	82.7	71.6	-	-	-
PAConv*	-	78.7	69.3	-	73.0	66.6
AE^{2}INetSeg	89.6	81.7	73.0	89.7	73.5	67.3
AE^{2}INetSeg*	89.9	82.6	73.7	89.9	74.3	68.0

Table 4.1. The mIoU (%), mAcc (%) and oA (%) on S3DIS dataset. The mark '*' denotes that the voting scheme (Thomas *et al.*, 2019) is adopted at testing.

rate by 0.1 after 60 (600) epochs and 80 (800) epochs, respectively. For classification (ModelNet40) and part segmentation (ShapeNetPart) tasks, we reduce the learning rate until $1e-3$ using the cosine annealing (Loshchilov and Hutter, 2016) policy. We train the networks for 250 / 200 epochs with a batch size of 32 / 16 on ModelNet40 / ShapeNetPart. Following previous works (Xu *et al.*, 2021a; Liu *et al.*, 2019d; Qi *et al.*, 2017b; Thomas *et al.*, 2019), we exploit data augmentations. In specific, for classification and part segmentation, we augment the point cloud with 1) random anisotropic scaling in a range from -0.66 to 1.5 and 2) random translation in a range from -0.2 to 0.2, while for semantic segmentation, we exploit random rotation along the vertical axis, scaling in a range from 0.8 to 1.1, and gaussian jittering.

4.4.2 3D Scene Semantic Segmentation

Here, we study 3D scene segmentation on S3DIS (Armeni *et al.*, 2016) and ScanNet v2 (Dai *et al.*, 2017a) to evaluate the capacity of AE^{2}INetSeg.

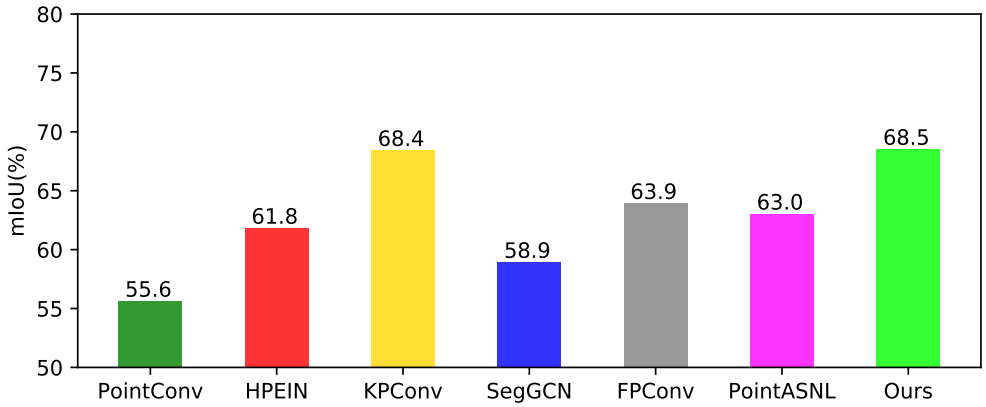


Figure 4.3. The mIoU on ScanNet v2 (Dai *et al.*, 2017a). We make comparisons against PointConv (Wu *et al.*, 2019), HPEIN (Jiang *et al.*, 2019), KPConv (Thomas *et al.*, 2019), SegGCN (Lei *et al.*, 2020), FPConv (Lin *et al.*, 2020a), and PointASNL (Yan *et al.*, 2020).

S3DIS contains 271 rooms captured from 6 areas. It provides 3D points and their corresponding RGB values. Each point is annotated with one of the semantic labels from 13 categories, such as table, wall, and sofa. In training time, we randomly select 14,000 points from a $2m \times 2m$ block on-the-fly. Each point is represented as a 9-dim vector with XYZ, RGB, and normalized position in the room. All points are evaluated at test time. We study two settings for the task, *i.e.,* 6-fold cross-validation and Area 5 validation. In Table 4.1, we make comparisons against the previous methods, including PointNet (Qi *et al.*, 2017a), PointNet++ (Qi *et al.*, 2017b), PointCNN (Li *et al.*, 2018e), DGCNN (Wang *et al.*, 2019d), PointWeb (Zhao *et al.*, 2019a), HPEIN (Jiang *et al.*, 2019), KPConv (Thomas *et al.*, 2019), FPConv (Lin *et al.*, 2020a), SegGCN (Lei *et al.*, 2020), RandLA-Net (Hu *et al.*, 2020a), Point2Node (Han *et al.*, 2020), SCF-Net (Fan *et al.*, 2021a), and PAConv (Xu *et al.*, 2021a). Since some works evaluates the model using the voting scheme at testing, which is helpful to improve the performance, we also report the results obtained with the voting scheme (marked by '*'). As shown in Table 4.1, we can find that both AE^{2}INetSeg and AE^{2}INetSeg* perform better than almost all of existing peer-reviewed works on all metrics, including overall Accuracy (oA), mean IoU

(mIoU), and mean class Accuracy (mAcc). It is worth noting that our method outperforms PointWeb (Zhao *et al.*, 2019a) by a large margin, which demonstrates the superiority of the proposed edge-to-edge interaction. The scores for each class are given in Sec. 4.5.

For ScanNet v2 (Dai *et al.*, 2017a), we train the model on the training set (1201 scans), and make evaluation on the test set (100 scans). There are 20 meaningful categories and one class for free space. During training, we randomly sample $14,000$ points from a $2m \times 2m$ block on-the-fly. Each point is represented as a 6-dim vector with XYZ and RGB. We mainly compare our model with previous point based methods. As shown in Figure 4.3, our model performs better than most of methods by a large margin. In the benchmark website, we can find some methods yield higher scores through exploring other clues, such as, rendering virtual views (Kundu *et al.*, 2020), training multiple tasks (Hu *et al.*, 2020b), and combining 2D and 3D domains (Hu *et al.*, 2021). See the detailed scores for each class in Sec. 4.5.

4.4.3 3D Shape Part Segmentation

The shape part segmentation task aims to predict part category label for each point in a 3D model. We evaluate the proposed AE^{2}INetSeg on the ShapeNet-Part dataset (Yi *et al.*, 2016). There are $16,881$ CAD models from 16 object categories, which are labeled with 50 parts in total. Following (Li *et al.*, 2018e; Zhang *et al.*, 2019b), we split the models into $14,006$ for training and 2,875 for testing. During training, we randomly sample $2,048$ points on mesh surfaces. In the inference stage, we sample $2,048$ points multiple times to make sure all the points have at least ten predictions. We present the instance average IoU (mIoU, %) and class average IoU (mcIoU, %) in Table 4.2. We observe that our AE^{2}INetSeg achieves competitive or even better scores compared with existing methods, including PointNet (Qi *et al.*, 2017a), PointNet++ (Qi *et al.*, 2017b), PCNN (Atzmon *et al.*, 2018), PointCNN (Li *et al.*, 2018e), DGCNN (Wang *et*

al., 2019d), RSCNN (Liu *et al.*, 2019d), DensePoint (Liu *et al.*, 2019c), KPConv (Thomas *et al.*, 2019), 3D-GCN (Lin *et al.*, 2020b), and PAConv (Xu *et al.*, 2021a). We can also find that the performance is improved slightly over the recent years. Specifically, while our model ranks third place w.r.t mcIoU, it yields higher mIoU (86.8%) than previous approaches.

Method	mcIoU	mIoU	air plane	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor bike	mug	pistol	rocket	skate board	table
PointNet	80.4	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++	81.9	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
PCNN	81.8	85.1	82.4	80.1	85.5	79.5	90.8	73.2	91.3	86.0	85.0	95.7	73.2	94.8	83.3	51.0	75.0	81.8
PointCNN	84.6	86.1	84.1	86.5	86.0	80.8	90.6	79.7	92.3	88.4	85.3	96.1	77.2	95.3	84.2	64.2	80.0	83.0
DGCNN	82.3	85.1	84.2	83.7	84.4	77.1	90.9	78.5	91.5	87.3	82.9	96.0	67.8	93.3	82.6	59.7	75.5	82.0
RSCNN	84.0	86.2	83.5	84.8	88.8	79.6	91.2	81.1	91.6	88.4	86.0	96.0	73.7	94.1	83.4	60.5	77.7	83.6
DensePoint	84.2	86.4	84.0	85.4	90.0	79.2	91.1	81.6	91.5	87.5	84.7	95.9	74.3	94.6	82.9	64.6	76.8	83.7
KPConv	85.1	86.4	84.6	86.3	87.2	81.1	91.1	77.8	92.6	88.4	82.7	96.2	78.1	95.8	85.4	69.0	82.0	83.6
3D-GCN	82.1	85.1	83.1	84.0	86.6	77.5	90.3	74.1	90.9	86.4	83.8	95.6	66.8	94.8	81.3	59.6	75.7	82.8
PAConv	84.6	86.1	84.3	85.0	90.4	79.7	90.6	80.8	92.0	88.7	82.2	95.9	73.9	94.7	84.7	65.9	81.4	84.0
AE^{2}INetSeg	84.4	86.8	84.9	84.9	88.4	81.8	91.9	76.5	92.0	88.8	86.4	96.1	75.5	95.6	84.1	63.9	76.3	83.5

Table 4.2. Quantitative results on ShapeNetPart dataset. Our method yields higher mIoU score than previous approaches, and competitive mcIoU score.

Method	Input	#Points	mA	oA	Aligned
PointNet++	PN	5k	-	91.9	No
PointNet	P	1k	86.2	89.2	No
PointNet++	P	1k	-	90.7	No
PointCNN	P	1k	88.1	92.2	No
AE^{2}INetCls	P	1k	89.9	92.4	No
PointASNL	PN	1k	-	93.2	Yes
DGCNN	P	2k	90.7	93.5	Yes
PointCNN	P	1k	88.8	92.5	Yes
DGCNN	P	1k	90.2	92.9	Yes
RSCNN	P	1k	-	93.6	Yes
ShellNet	P	1k	-	93.1	Yes
PointASNL	P	1k	-	92.9	Yes
GridGCN	P	1k	91.3	93.1	Yes
PAConv	P	1k	-	93.9	Yes
AE^{2}INetCls	P	1k	91.6	94.2	Yes

Table 4.3. The mA (%) and oA (%) on ModelNet40 dataset. P denotes Point, while PN denotes Point and Normal.

4.4.4 3D Shape Classification

We evaluate our model on the ModelNet40 (Wu *et al.*, 2015) shape classification benchmark. There are $9,843$ 3D models for training and $2,468$ for testing. During training, we uniformly sample $1,024$ points on the mesh surfaces. Noting that, a large percentage of 3D models in ModelNet40 have been pre-aligned to the common up direction and horizontal facing direction. As reported in PointCNN (Li *et al.*, 2018e), random horizontal rotation (*i.e.,* Not aligned) has a non-negligible impact on the performance. We thus consider both settings in our experiments, *i.e.,* Pre-aligned and Unaligned.

Here, we only make comparisons against several previous methods, including PointNet (Qi *et al.*, 2017a), PointNet++ (Qi *et al.*, 2017b), DGCNN (Wang *et al.*, 2019d), PointCNN (Li *et al.*, 2018e), PointASNL (Yan *et al.*, 2020), RSCNN (Liu *et al.*, 2019d), GridGCN (Xu *et al.*, 2020b), ShellNet (Zhang *et al.*, 2019b), and PAConv (Xu *et al.*, 2021a). More comparisons can be found in Sec. 4.5. As reported in Table 4.3, AE^{2}INetCls outperforms previous state-of-the-art approaches in terms of both oA (overall accuracy) and mA (mean per-class accuracy). It is worth noting that the improvement on the ModelNet40 classification benchmark is only around 1.1% w.r.t oA over the last two years. The SOTA methods are ShellNet in 2019 (93.1%), RSCNN in 2019 (93.6%), GridGCN in 2020 (93.1%), and PAConv in 2021 (93.9%), respectively. This observation may further show the significance of our approach.

4.4.5 Ablation Study

The comparisons against the state-of-the-art methods demonstrate the effectiveness of our model in exploiting the local structures. Here, we conduct extensive ablations to inspect the proposed modules and analyze their involved components on Area 5 of S3DIS (Armeni *et al.*, 2016).

Analysis	Config.	oA	mAcc	mIoU
#Near. Neb.	$K_e = 2$	89.3	73.0	66.2
	$K_e = 3$	89.8	73.1	66.8
	$K_e = 4$	89.7	73.5	67.3
	$K_e = 5$	89.4	73.1	66.8
Eff. of modules	Baseline	88.3	69.7	63.9
	+AE^{2}IL	88.7 (0.4 ↑)	72.1 (2.4 ↑)	65.3 (1.4 ↑)
	+SymAE^{2}IL	89.7 (1.4 ↑)	73.5 (3.8 ↑)	67.3 (3.4 ↑)
	AFA*	87.9	69.8	63.9
	AE^{2}IL*	88.1	70.0	64.1
	SymAE^{2}IL*	88.1	70.6	64.2
Relation learning	D^f	87.9	70.7	64.2
	$D^f + D^s$	89.1	72.8	65.9
	$D^f + D^c$	88.6	71.6	65.5
	$D^f + D^c + D^s$	89.7	73.5	67.3
Comb. w/others	RSConv	86.9	66.6	60.3
	+SymAE^{2}IL	88.4 (1.5 ↑)	69.2 (2.6 ↑)	63.5 (3.2 ↑)
	GAConv	86.0	67.3	59.8
	+SymAE^{2}IL	87.7 (1.7 ↑)	71.9 (4.6 ↑)	64.6 (4.8 ↑)

Table 4.4. Ablation study on the proposed modules.

Method	mIoU (A5 / 6F)	#Par./M	FLOPs/G	Mem./G	T./s
PointWeb	60.3 / 66.7	1.0	142	8.4	0.18
FPConv	62.8 / 68.7	17.4	1032	9.8	0.69
PAConv	66.6 / 69.3	0.6	52	13.2	0.28
Ours	68.0 / 73.7	3.6	312	9.4	0.37

Table 4.5. The mIoUs (Area 5 and 6-fold), the parameters (#Par.), FLOPs, Inference memory (Mem.), and Inference time (T.) of the segmentation models on S3DIS dataset. We calculate the FLOPs, Mem., and T. by processing 12 samples, each one containing $14,000$ points, on one Tesla v100 GPU.

Method	None	$90°$	$180°$	$270°$	$\times 0.8$	$\times 1.2$	0.5%	1%
PointWeb*	54.7	52.5	54.2	51.5	54.0	51.7	54.5	54.4
FPConv*	56.0	54.0	54.3	52.4	54.3	52.5	55.6	55.1
Ours*	58.1	56.5	58.0	55.6	58.0	56.3	57.9	57.7
PAConv	59.5	55.4	58.1	53.9	58.7	59.1	58.8	58.3
Ours	62.3	62.1	59.9	61.1	62.3	59.4	62.0	61.2

Table 4.6. Robustness analysis. We evaluate the robustness through performing rotation ($90°$,$180°$,$270°$), scaling ($\times 0.8$,$\times 1.2$), and adding noises ($0.5\%, 1\%$) on 20 rooms of S3DIS dataset. Since FPConv and PointWeb are trained without data augmentation (DA), for fair comparisons, we also re-train our model without DA (Ours*).

Number of nearest neighbours. We first study the impacts of the number of nearest neighbors for an edge, *i.e.,* K_e. As reported in Table 4.4, the model performs better when we set K_e to 4, which is used in all experiments. It is also worth noting that when we set K_e to 2 or 3, the model still outperforms most of existing works list in Table 4.1. When setting K_e to 5, we find that the performance drops, due to the aggregation of some unrelated information.

Effectiveness of the proposed modules. To study the effectiveness of the proposed AE^{2}IL and SymAE^{2}IL, we compare three model, *i.e,* Baseline (no edge-to-edge interaction), AE^{2}INetSeg with AE^{2}IL, and AE^{2}INetSeg with SymAE^{2}IL, in Table 4.4. The comparisons show that the basic edge-to-edge interaction can improve the performance and the symmetric information can bring a further improvement. In addition, to further provide supports for the analysis on the differences between ours and PointWeb experimentally, we re-implement the AFA module proposed by PointWeb in our framework (AFA*). The comparisons between AFA* and the simplified version of our modules (AE^{2}IL* and SymAE^{2}IL*) in Table 4.4 also show the superiority of our methods over AFA strategy in PointWeb.

Relations learning between edges. In Eq. 4.5, we introduce three kinds of information, *i.e.,* relative position D^s, difference between features D^f, and normal vector D^c, to learn the low and high-level relations between the edges. To evaluate their impacts, we revise SymAE^{2}IL by considering four kinds of feature combinations, including D^f, $D^f + D^s$, $D^f + D^c$, and $D^f + D^s + D^c$. The results in Table 4.4 show that the geometric relations (D^s and D^c) can bring improvements, and when all relations are exploited, the performance can be improved greatly.

Combination with other point cloud operations. To further show the effectiveness of our method, we compare four models by integrating our module (SymAE^{2}IL) with two point cloud operations, *i.e.,* RSConv (Liu *et al.*, 2019d) and GAConv (Wang *et al.*, 2019b). In specific, since they aim to learn (attentional) weights from the edge information, here we insert the edge-to-edge interaction into the two operations to enhance the edge representations. As show in Table 4.4, we can observe that our module can bring remarkable performance improvements for both two operations.

Model complexity. In Table 4.5, we report the mIoUs, parameters, FLOPs, inference memory, and inference time of our segmentation model and several state-of-the-art works, including PointWeb (Zhao *et al.*, 2019a), FPConv (Lin *et al.*, 2020a), and PAConv (Xu *et al.*, 2021a). We can observe that our model complexities and running time are competitive to recent approaches, while our model outperforms previous methods for various point cloud analysis tasks, as shown in Table 4.1, Table 4.2, Table 4.3, and Figure 4.3.

Robustness analysis. Our method aims to enhance the point-to-point relation through modeling the edge-to-edge interaction adaptively, which could make the relation aware of the local structure and thus more robust to the geometry transformation. We make robustness evaluation through performing rotation, scaling, and adding noises on the input data during inference. As shown in Table 4.6, our method performs better than previous methods under all settings.

4.5 Supplementary Experiments

4.5.1 More Quantitative Results

In Sec. 4.4.2, we presented the oA, mIoU, and mAcc obtained under the 6-fold cross-validation evaluation setting and Area-5 setting on S3DIS (Armeni *et al.*, 2016), and the mIoU on ScanNet v2 (Dai *et al.*, 2017a). Here, we provide the category-level scores in Table 4.7 (6-fold S3DIS), Table 4.8 (Area-5 S3DIS), and Table 4.9 (ScanNet v2). The compared methods include PointNet (Qi *et al.*, 2017a), PointCNN (Li *et al.*, 2018e), DeepGCN (Li *et al.*, 2019b), PointWeb (Zhao *et al.*, 2019a), PAT (Yang *et al.*, 2019a), KPConv (Thomas *et al.*, 2019), FPConv (Lin *et al.*, 2020a), PAConv (Xu *et al.*, 2021a), HPEIN (Jiang *et al.*, 2019), SAGC (Wong and Vong, 2020), and SegGCN (Lei *et al.*, 2020).

As shown in Table 4.7, Table 4.8, and Table 4.9, our model achieves competitive or state-of-the-art scores (ranking first or second) in comparison against previous methods for most of classes on both two datasets.

In addition, we provide more comparisons for the classification task on ModelNet40 dataset (Wu *et al.*, 2015) in Table 4.10.

Method	oA	mIoU	ceil.	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clutter
PointNet	78.6	47.6	88.0	88.7	69.3	42.4	23.1	47.5	51.6	54.1	42.0	9.6	38.2	29.4	35.2
PointCNN	88.1	65.4	94.8	97.3	75.8	63.3	51.7	58.4	57.2	69.1	71.6	61.2	39.1	52.2	58.6
DeepGCN	85.9	60.0	93.1	95.3	78.2	33.9	37.4	56.1	68.2	64.9	61.0	34.6	51.5	51.1	54.4
PointWeb	87.3	66.7	93.5	94.2	80.9	52.4	41.3	64.9	68.1	71.4	67.1	50.3	62.7	62.2	58.5
PAT	-	64.3	93.0	98.4	73.5	58.5	38.9	77.4	67.7	62.7	67.3	30.6	59.6	66.6	41.4
KPConv	-	70.6	93.6	92.4	83.1	63.9	54.3	66.1	76.6	64.0	57.8	74.9	69.3	61.3	60.3
FPConv	-	68.7	94.8	97.5	82.6	42.8	41.8	58.6	73.4	71.0	81.0	59.8	61.9	64.2	64.2
PAConv	-	69.3	94.3	93.5	82.8	56.9	45.7	65.2	74.9	74.6	59.7	61.8	67.4	65.8	58.4
AE^{2}INetSeg	89.9	73.7	95.0	97.5	82.9	60.9	46.8	68.6	75.4	76.0	77.7	72.2	70.5	70.9	64.5

Table 4.7. Semantic segmentation scores on S3DIS 6-fold cross-validation.

4.5.2 Visualization Examples

We provide several visualization examples on ShapeNetPart (Yi *et al.*, 2016) (Figure 4.4) and S3DIS (Armeni *et al.*, 2016) (Figure 4.5). For S3DIS, we

Method	oA	mIoU	ceil.	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clutter
PointNet	-	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	58.9	52.6	5.9	40.3	26.4	33.2
PointWeb	87.0	60.3	92.0	98.5	79.4	0.0	21.1	59.7	34.8	76.3	88.3	46.9	69.3	64.9	52.5
PAT	-	60.1	93.0	98.5	72.3	1.0	41.5	85.1	38.2	57.7	83.6	48.1	67.0	61.3	33.6
HPEIN	87.2	61.9	91.5	98.2	81.4	0.0	23.3	65.3	40.0	75.5	87.7	58.5	67.8	65.6	49.4
KPConv	-	67.1	92.8	97.3	82.4	0.0	23.9	58.0	69.0	81.5	91.0	75.4	75.3	66.7	58.9
FPConv	87.5	62.8	94.6	98.5	80.9	0.0	19.1	60.1	48.9	80.6	88.0	53.2	68.4	68.2	54.9
SAGC	87.5	60.1	93.3	95.4	78.3	43.7	27.6	50.3	68.1	69.2	71.2	30.6	57.6	41.0	54.6
SegGCN	88.2	63.6	93.7	98.6	80.6	0.0	28.5	42.6	74.5	80.9	88.7	69.0	71.3	44.4	54.3
PAConv	-	66.6	94.6	98.6	82.4	0.0	26.4	58.0	60.0	80.4	89.7	69.8	74.3	73.5	57.7
AE^{2}INetSeg	89.9	68.0	95.3	98.5	83.2	0.0	21.8	59.4	63.4	81.7	91.4	77.5	75.8	76.6	58.7

Table 4.8. Semantic segmentation scores on S3DIS Area 5.

Method	mIoU	bath.	bed	book.	cab.	cha.	cou.	cur.	des.	door	floor	oth.	pic.	refr.	show.	sink	sofa	tab.	toi.	wall	wind.
HPEIN	61.8	72.9	66.8	64.7	59.7	76.6	41.4	68.0	52.0	52.5	94.6	43.2	21.5	49.3	59.9	63.8	61.7	57.0	89.7	80.6	60.5
KPConv	68.4	84.7	75.8	78.4	64.7	81.4	47.3	77.2	60.5	59.4	93.5	45.0	18.1	58.7	80.5	69.0	78.5	61.4	88.2	81.9	63.2
FPConv	63.9	78.5	76.0	71.3	60.3	79.8	39.2	53.4	60.3	52.4	94.8	45.7	25.0	53.8	72.3	59.8	69.6	61.4	87.2	79.9	56.7
SegGCN	58.9	83.3	73.1	53.9	51.4	78.9	44.8	46.7	57.3	48.4	93.6	39.6	6.1	50.1	50.7	59.4	70.0	56.3	87.4	77.1	49.3
AE^{2}INetSeg	68.5	81.9	77.2	71.6	65.6	82.9	49.0	81.3	62.4	60.5	95.1	48.4	26.7	56.6	59.8	70.2	75.0	60.7	92.4	82.9	69.5

Table 4.9. Semantic segmentation scores on ScanNet v2 test set. Our model yields higher mIoU score than previous works.

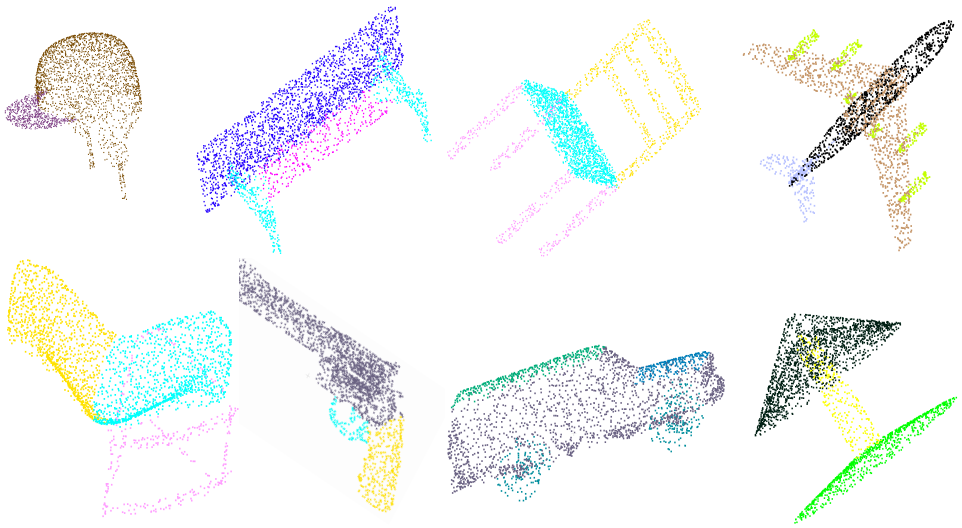


Figure 4.4. Visualization examples on ShapeNetPart dataset. Best viewed in color.

compare our method and the baseline model (*i.e.,* no edge-to-edge interaction used). We can observe that our model generates more accurate prediction results for some objects (marked with red dotted bounding box), like door (the 4^{th} and 7^{th} examples), board (5^{th}), bookcase (1^{st}), wall (2^{nd}, 6^{th} and 8^{th}), and column (2^{nd}, 3^{rd}, and 5^{th}).

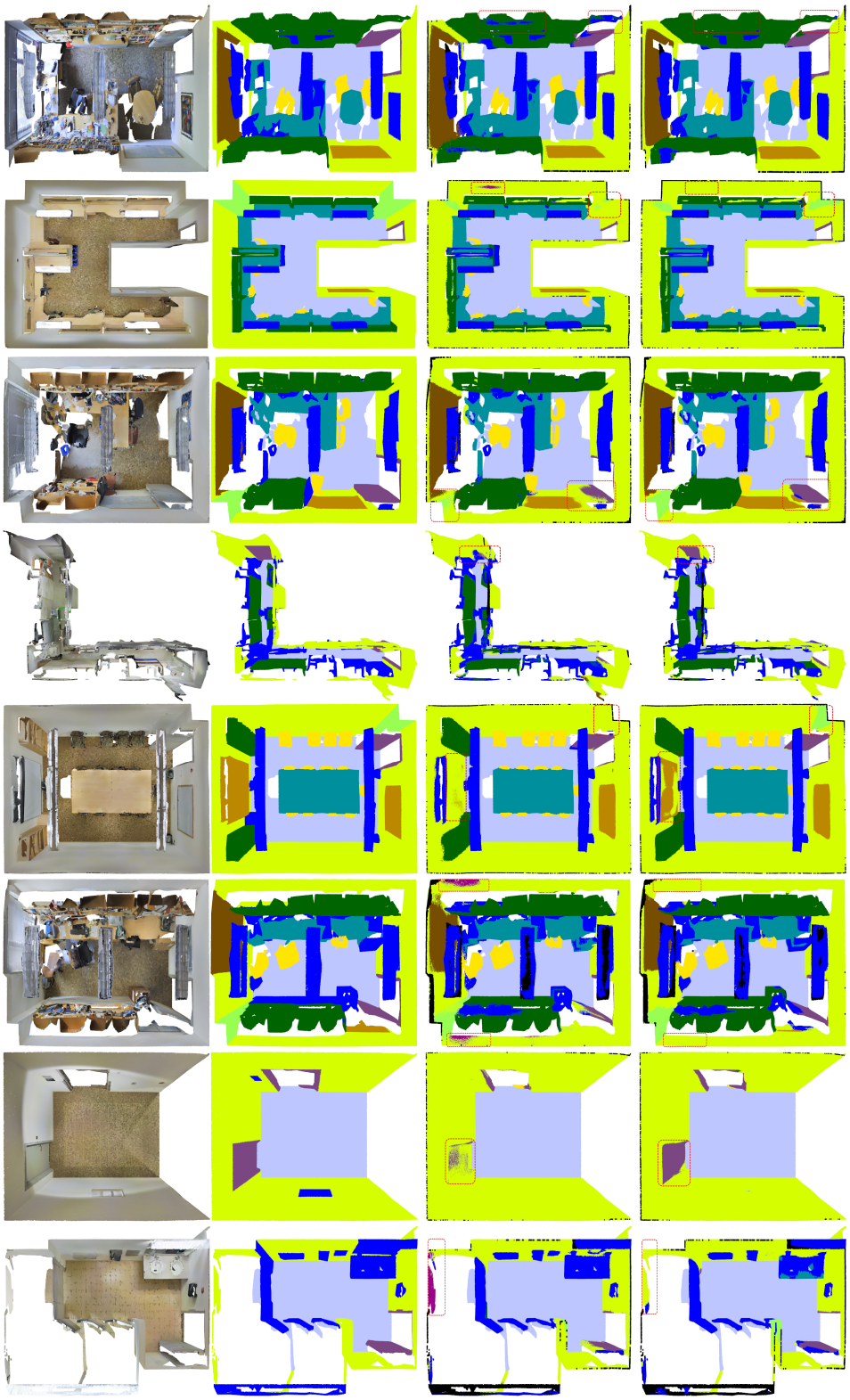


Figure 4.5. Segmentation examples on S3DIS dataset. Best viewed in color.

Method	Input	#Points	mA	oA	Aligned
PointNet++ (Qi *et al.*, 2017b)	Point+Normal	5k	-	91.9	No
SpiderCNN (Xu *et al.*, 2018c)	Point+Normal	1k	-	92.4	No
KPConv (Thomas *et al.*, 2019)	Point	7k	-	92.9	No
PointNet (Qi *et al.*, 2017a)	Point	1k	86.2	89.2	No
PointNet++ (Qi *et al.*, 2017b)	Point	1k	-	90.7	No
3D-GCN (Lin *et al.*, 2020b)	Point	1k	-	92.1	No
PointCNN (Li *et al.*, 2018e)	Point	1k	88.1	92.2	No
AE^{2}INetCls	Point	1k	89.9	92.4	No
SO-Net (Li *et al.*, 2018c)	Point+Normal	5k	90.8	93.4	Unknown
PAT (Yang *et al.*, 2019a)	Point+Normal	1k	-	91.7	Unknown
PConv (Wu *et al.*, 2019)	Point+Normal	1k	-	92.5	Unknown
FPConv (Lin *et al.*, 2020a)	Point	1k	-	92.5	Unknown
CN (Yang *et al.*, 2020b)	Point	1k	-	93.3	Unknown
PointASNL (Yan *et al.*, 2020)	Point+Normal	1k	-	93.2	Yes
DGCNN (Wang *et al.*, 2019d)	Point	2k	90.7	93.5	Yes
PointCNN (Li *et al.*, 2018e)	Point	1k	88.8	92.5	Yes
PCNN (Atzmon *et al.*, 2018)	Point	1k	-	92.3	Yes
DGCNN (Wang *et al.*, 2019d)	Point	1k	90.2	92.9	Yes
Point2Seq (Liu *et al.*, 2019b)	Point	1k	90.4	92.6	Yes
RSCNN (Liu *et al.*, 2019d)	Point	1k	-	93.6	Yes
PointWeb (Zhao *et al.*, 2019a)	Point	1k	89.4	92.3	Yes
DensePnt (Liu *et al.*, 2019c)	Point	1k	-	93.2	Yes
ShellNet (Zhang *et al.*, 2019b)	Point	1k	-	93.1	Yes
PointASNL (Yan *et al.*, 2020)	Point	1k	-	92.9	Yes
GridGCN (Xu *et al.*, 2020b)	Point	1k	91.3	93.1	Yes
WCPNet (Nezhadarya *et al.*, 2020)	Point	1k	90.5	92.4	Yes
PosPool (Liu *et al.*, 2020)	Point	1k	-	93.2	Yes
GDANet (Xu *et al.*, 2020a)	Point	1k	-	93.8	Yes
PAConv (Xu *et al.*, 2021a)	Point	1k	-	93.9	Yes
AE^{2}INetCls	Point	1k	91.6	94.2	Yes

Table 4.10. Mean per-class accuracy (mA) and overall accuracy (oA) on ModelNet40 dataset.

4.6 Conclusion

In this chapter, we propose a novel adaptive edge-to-edge interaction learning module, *i.e.,* AE^{2}IL, for point cloud analysis. Through learning the interaction between edges, the module makes the point-to-point relation aware of the local shape, which is beneficial to capture the discriminative local structure information. Moreover, to model the local structure more thoroughly, we further extend the AE^{2}IL to a symmetric version, *i.e.,* SymAE^{2}IL. To examine the effectiveness of the proposed module, we design two networks, *i.e.,* AE^{2}INetCls and

AE^{2}INetSeg, for point cloud classification and segmentation, respectively. The experimental results on several typical point cloud tasks and ablations show the models' capability of representing the local structure.

CHAPTER 5

Domain Generalization via Entropy Regularization

Domain generalization aims to learn from multiple source domains a predictive model that can generalize to unseen target domains. One essential problem in domain generalization is to learn discriminative domain-invariant features. To arrive at this, some methods introduce a domain discriminator through adversarial learning to match the feature distributions in multiple source domains. However, adversarial training can only guarantee that the learned features have invariant marginal distributions, while the invariance of conditional distributions is more important for prediction in new domains. To ensure the conditional invariance of learned features, we propose an *entropy regularization* term that measures the dependency between the learned features and the class labels. Combined with the typical task-related loss, *e.g.,* cross-entropy loss for classification, and adversarial loss for domain discrimination, our overall objective is guaranteed to learn conditional-invariant features across all source domains and thus can learn classifiers with better generalization capabilities. We demonstrate the effectiveness of our method through comparison with state-of-the-art methods on simulated 3D and 2D object classification datasets and real-world 2D object recognition datasets.

5.1 Introduction

Recent years have witnessed the remarkable success of modern machine learning techniques in various applications. However, a fundamental problem machine learning suffers from is that the model learned from training data often does not generalize well on data sampled from a different distribution, due to the existence of data bias (Torralba and Efros, 2011; Fang *et al.*, 2020) between the training and test data. To tackle this issue, a significant effort has been made in domain adaptation, which reduces the discrepancy between source and target domains (Zhang *et al.*, 2013; Tzeng *et al.*, 2014; Ganin and Lempitsky, 2015; Sun and Saenko, 2016; Bousmalis *et al.*, 2016; Zhao *et al.*, 2018). The main drawback of this approach is that one has to repeat training for each new dataset, which can be time-consuming. Therefore, *domain generalization* (Blanchard *et al.*, 2011) is proposed to learn generalizable models by leveraging information from multiple source domains (Muandet *et al.*, 2013; Ghifary *et al.*, 2015; Li *et al.*, 2018b; Arjovsky *et al.*, 2019).

Since there is no prior information about the distribution of the target domain during training, it is difficult to match the distributions between source and target domains, which makes domain generalization more challenging. To improve the generalization capabilities of learned models, various solutions have been developed from different perspectives. A classic but effective solution to domain generalization is learning a domain-invariant feature representation (Ghifary *et al.*, 2015; Li *et al.*, 2018b; Li *et al.*, 2018d; Muandet *et al.*, 2013; Matsuura and Harada, 2020a; Li *et al.*, 2018d) across source domains. Muandet *et al.* (Muandet *et al.*, 2013) presented a kernel-based optimization algorithm, called Domain-Invariant Component Analysis, to learn an invariant transformation by minimizing the dissimilarity across domains. Ghifary *et al.* (Ghifary *et al.*, 2015) proposed to learn features robust to variations across domains by introducing multi-task auto-encoders. Another line of research explores various

data augmentation strategies (Shankar *et al.*, 2018; Volpi *et al.*, 2018; Carlucci *et al.*, 2019). For example, Shankar *et al.* (Shankar *et al.*, 2018) presented a gradient-based domain perturbation strategy to perturb the input data. By augmenting the original feature space, Blanchard *et al.* (Blanchard *et al.*, 2017) viewed the problem of domain generalization as a kind of supervised learning problem. Then, they developed a kernel-based method that predicts classifiers from the augmented feature space. To make theoretical complementary to these empirically supported approaches, Deshmukh *et al.* (Deshmukh *et al.*, 2019) proved the first known generalization error bound for multi-class domain generalization through studying a kernel-based learning algorithm. Apart from the clues aforementioned, some recent works (Dou *et al.*, 2019; Li *et al.*, 2019a; Balaji *et al.*, 2018; Li *et al.*, 2018a) attempted to exploit meta-learning for domain generalization. A latest work, MASF (Dou *et al.*, 2019), proposed a model-agnostic episodic learning procedure to regularize the semantic structure of the feature space.

In this chapter, we revisit the domain-invariant feature representation learning methods. Most of existing methods assume that the marginal distribution $P(X)$ changes while the conditional distribution $P(Y|X)$ stays stable across domains [1]. Therefore, significant effort has been made in learning a feature representation $F(X)$ that has invariant $P(F(X))$, either by traditional moment matching (Peng *et al.*, 2019) or modern adversarial training (Matsuura and Harada, 2020a; Li *et al.*, 2018d). To ensure the universality of $F(X)$ and also make it discriminative, a joint classification model is trained on all the source domains and can be used for prediction in new datasets. However, the stability of $P(Y|X)$ is often violated in real applications, leading to sub-optimal solutions. Li *et al.* (Li *et al.*, 2018d) proposed to learn invariant class-conditional distribution $(P(F(X)|Y))$ by doing adversarial training for each class. However, the method becomes less effective as the number of classes increases.

[1] Here, X and Y represent the sample and corresponding label, respectively.

To tackle the aforementioned issues, we propose an entropy-regularization approach which directly learns features that have invariant $P(Y|F(X))$ across domains. In specific, the conditional entropy term $H(Y|F(X))$ measures the dependency between $F(X)$ and class label Y, and we aim to minimize the dependency by maximizing the conditional entropy. We show theoretically that our entropy-regularization together with the cross-entropy classification loss effectively minimize the divergence between $P(Y|F(X))$ in all source domains. In addition, we show that $H(Y|F(X))$ can be effectively estimated by assuming a multinomial distribution for $P(Y|F(X))$, which is a weak assumption for discrete class labels. Together with the adversarial training on $P(F(X))$, our approach can guarantee the invariance of the joint distribution $P(F(X), Y)$ and thus has a better generalization capability. We demonstrate the effectiveness of our approach through conducting comprehensive experiments on several 2D object recognition datasets, including simulated and real-world scenes. Moreover, since currently there is no work studying domain generalization in 3D shape analysis, we also test our model on a simulated 3D object classification dataset as a tentative exploration to this problem.

5.2 Related Work

5.2.1 Domain Generalization

According to the number of source domains, we can divide the domain generalization problem into two sub-problems, *i.e.,* multi-soruce domain generalization and single domain generalization. Multi-source domain generalization (Ghifary *et al.*, 2015; Li *et al.*, 2018b; Li *et al.*, 2018d; Li *et al.*, 2019a; Balaji *et al.*, 2018), which this chapter focuses on, refers to domain generalization using multiple source domains. Here, we briefly categorize previous efforts into three groups, *i.e.,* domain-invariant features learning, data augmentation, and meta-learning. We refer to a recent survey about domain generalization (Zhou *et*

al., 2021) for a thorough understanding of more detailed categorization. Early methods mainly follow the first clue through aligning the distributions across source domains. For example, Li *et al.* (Li *et al.*, 2018b) minimize MMD distance (Gretton *et al.*, 2012) to align the distributions across source domains, and force the aligned distribution to be similar to a pre-defined prior distribution via adversarial learning. Matsuura *et al.* (Matsuura and Harada, 2020b) use a multi-class domain discriminator to learn domain-invariant features. To learn domain-invariant class-conditional distribution, Li *et al.* (Li *et al.*, 2018d) exploit adversarial training for each category separately as well as for global datasets. Recently, data augmentation technique for domain generalization has been studied extensively. For instance, Zhou *et al.* (Zhou *et al.*, 2020b) propose to increase the diversity of the training data through training a synthetic data generator. To generate new data, the generator is required to be distant from the known source domains and to preserve the semantic content. Taking advantage of the property of the Fourier transformation that the phase component preserves high-level semantic content while the amplitude component contains low-level statistics, Xu *et al.* (Xu *et al.*, 2021b) first get the Fourier transformation of the images, then interpolate between the amplitude spectrums of two images using the MixUp strategy (Zhang *et al.*, 2018a), and lastly use the new amplitude spectrum to generate new data. Another solution to domain generalization (Dou *et al.*, 2019; Li *et al.*, 2019a; Balaji *et al.*, 2018; Li *et al.*, 2018a) is using meta-learning (Hospedales *et al.*, 2021), where the training domains are split into meta-train and meta-test at each iteration, to simulate domain shift. To learn semantically consistent features across source domains, Dou *et al.* (Dou *et al.*, 2019) propose to regularize the semantic structure of the feature space in a model-agnostic episodic learning framework. In this chapter, we follow the first clue, *i.e.,* learning domain-invariant features, and exploit the proposed entropy regularization to learn a classifier and a feature extractor with better generalization capabilities.

In comparison with multiple domain generalization, single domain generalization (Volpi *et al.*, 2018) is more challenging. To achieve domain generalization from a single source domain, existing works often exploit the data augmentation technique (Volpi *et al.*, 2018; Qiao *et al.*, 2020; Wang *et al.*, 2021) to create fictitious domains using adversarial training. For instance, Qiao *et al.* (Qiao *et al.*, 2020) exploit adversarial training to augment the samples and train the model on original data and augmented data using meta-learning. To increase the diversity between the original data and the augmented data, Wang *et al.* (Wang *et al.*, 2021) synthesize samples with unseen styles out of original distributions and enlarge the domain shifts gradually via a proposed style-complement module. Batch normalization (Ioffe and Szegedy, 2015) has widely been used in most of modern neural networks, which can reduce the internal covariate shift by normalizing each layer's input using the statistics. However, in single domain generalization, due to the domain shift from source domain to target domain, the source domain statistics and target domain statistics are usually different, which might cause significant performance drops. To address this issue, Fan *et al.* (Fan *et al.*, 2021b) study the statistics of normalization layers and propose an adaptive normalization approach, where the standardization and rescaling statistics are enabled to be adaptive to each individual input sample.

5.2.2 Domain Adaptation in Point Cloud Analysis

Currently, domain generalization in 2D image analysis, like object classification and semantic segmentation, has been studied extensively (Dou *et al.*, 2019; Fan *et al.*, 2021b; Qiao *et al.*, 2020), while domain generalization in 3D point cloud analysis is still largely under-explored. In comparison, some efforts to address domain adaptation in several 3D point cloud tasks, like object detection (Saltori *et al.*, 2020), classification (Achituve *et al.*, 2021), and semantic segmentation (Langer *et al.*, 2020; Jaritz *et al.*, 2020; Yi *et al.*, 2021), have been made recently. The domain shifts in 2D image data are mainly from the changes

in style, like texture and color, while in 3D point cloud, the domain shifts largely result from, such as, the density, object size, and sensor location. For domain adaptation in 3D objection detection from point cloud, Yang *et al.* (Yang *et al.*, 2021) exploit a random object scaling strategy to mitigate the negative effects resulting from the object size bias and use self-training to improve the detector on target domain. For domain adaptation in 3D semantic segmentation, Zhao *et al.* (Zhao *et al.*, 2020b) learn the simulation-to-real domain adaption, *i.e.,* from synthetic data to real-world data, at two levels, *i.e.,* pixel level through self-supervised dropout noise rendering and feature level using feature alignment between the simulation and real domains. To adapt a 3D shape classifier from source domain to target domain, Achituve *et al.* (Achituve *et al.*, 2021) design a self-supervised task, *i.e.,* deformation reconstruction, to capture the structure of the target data, and propose a Point Cloud MixUp strategy to learn robust feature representations. Although there is no work to cope with domain generalization in 3D point cloud analysis tasks, we believe previous attempts to domain adaptation in related tasks can motivate the researchers to study the more challenging problem. In this chapter, we study the general domain generalization problem, and evaluate our method's generalization capabilities on both 3D and 2D object classification datasets.

5.3 Method

5.3.1 Problem Definition

Let $\mathcal{X}$ and $\mathcal{Y}$ be the feature and label spaces, respectively. In the domain generalization subject, there are K source domains $\{\mathcal{D}_i\}_{i=1}^{K}$ and L target domains[2] $\{\mathcal{D}_i\}_{i=K+1}^{L+K}$. The goal is to generalize the model learned using data samples of source domains to unseen target domains. In the following, we denote the joint distribution of domain i by $P_i(X, Y)$ (defined on $\mathcal{X} \times \mathcal{Y}$). During the training

[2]Source/Target: seen/unseen during training.

process, there are K datasets $\{S_i\}_{i=1}^{K}$ available, where $S_i = \{(\mathbf{x}_j^{(i)}, y_j^{(i)})\}_{j=1}^{N_i}$. Here, N_i is the number of samples of S_i, which are sampled from the i^{th} domain. In the test stage, we evaluate the generalization capabilities of the learned model on L datasets sampled from the L target domains, respectively. This chapter mainly studies domain generalization for image classification, where the label space $\mathcal{Y}$ contains C discrete labels $\{1, 2, \cdots, C\}$.

5.3.2 Domain Generalization Through Adversarial Learning

We first present how domain generalization can be learned in an adversarial learning framework.

For the classification subject, the model consists of one feature extractor F parameterized by θ and one classifier T parameterized by ϕ. We can optimize θ and ϕ on the K source datasets by minimizing a cross-entropy loss:

$$\begin{aligned}\min_{F,T} \mathcal{L}_{cls}(\theta, \phi) &= -\sum_{i=1}^{K} \mathop{\mathbb{E}}_{(X,Y)\sim P_i(X,Y)} [\log(Q^T(Y|F(X)))] \\ &= -\sum_{i=1}^{K}\sum_{j=1}^{N_i} \mathbf{y}_j^{(i)} \cdot \log(T(F(\mathbf{x}_j^{(i)})),\end{aligned} \tag{5.1}$$

where $\mathbf{y}_j^{(i)}$ is the one-hot vector of the class label $y_j^{(i)}$, "$\cdot$" represents the dot product operation, and $Q^T(Y|F(X))$ denotes the predicted label distribution (conditioned on $F(X)$) corresponding to domain i.

However, optimized by the classification loss solely, the model cannot learn domain-invariant features, and thus shows limitations in generalizing to the unseen domains. By exploiting the adversarial learning (Goodfellow *et al.*, 2014),

we can alleviate the issue. Specifically, we further introduce a domain discriminator D parameterized by ψ, and train D and F in a minimax game as follows:

$$\begin{aligned}\min_{F}\max_{D}\mathcal{L}_{adv}(\theta,\psi) &= \sum_{i=1}^{K}\mathop{\mathbb{E}}_{X\sim P_i(X)}[\log D(F(X))] \\ &= \sum_{i=1}^{K}\sum_{j=1}^{N_i}\mathbf{d}_j^{(i)}\cdot\log(D(F(\mathbf{x}_j^{(i)}))),\end{aligned} \tag{5.2}$$

where $\mathbf{d}_j^{(i)}$ is the one-hot representation of the domain label i.

Although optimizing Eq. 5.2 can lead to invariant marginal distributions *i.e.,* $P_1(F(X)) = P_2(F(X)) = \cdots = P_K(F(X))$, it cannot guarantee the conditional distribution $P(Y|F(X))$ is invariant across domains. This would degrade the generalization capabilities of the model. Even though the classifier attempts to cluster the samples from the same category together in the feature space, which benefits to the learning of the invariant conditional distribution, there still exists an issue. We take the simulated data for example. Firstly, we sample data from two 2D-distributions (shown in Figure 5.1) as the Domain_0 and Domain_1, respectively. The marginal distributions of the first dimension (x_0) in the two domain are the same, while the second (x_1) comes from different marginal distributions. Each domain consists of three components. We take each dimension as the input to train a classifier using Eq. 5.1 and Eq. 5.2, and we find that the classifier distinguishes the second dimension better than the first (loss: -0.34 *v.s.* -0.16). This indicates that the classifier might not select the domain-invariant feature, but select the features easier to discriminate. Therefore, it is challenging for the typical classification loss to achieve a balance between learning domain-invariant features and discriminative features.

5.3.3 Entropy Regularization

Description. To address the issues aforementioned, we propose to regularize the distributions of the features by minimizing the KL divergence between

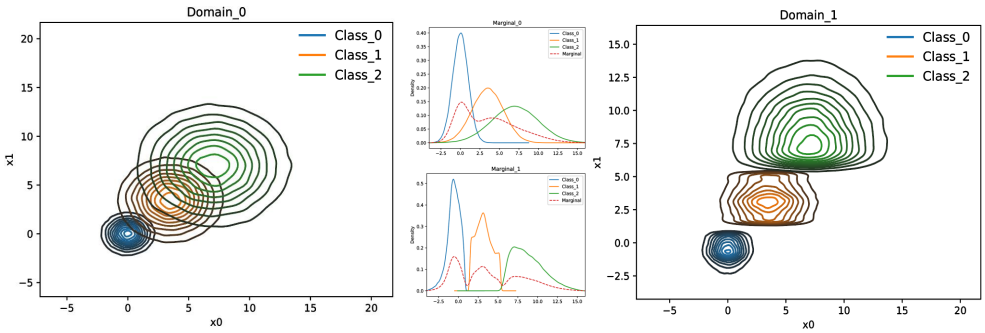


Figure 5.1. Simulated data. We create two domains from the two 2D-distributions (left and right), respectively. The data in Domain_0 and Domain_1 is two-dimensional. In specific, the first dimensions in two domains are both sampled from Marginal_0 (top-middle), while the second dimension in Domain_0 and Domain_1 is sampled from Marginal_0 and Marginal_1 (bottom-middle), respectively.

the conditional distribution $P_i(Y|F(X))$ in the i^{th} domain and the conditional distribution $Q^T(Y|X)$. $P_i(Y|F(X))$ denotes the predicted label distribution conditioned on the learned features. By matching any conditional distribution $P_i(Y|F(X))$ to a common distribution $Q^T(Y|F(X))$, we can obtain the domain-invariant conditional distribution $P(Y|F(X))$. For the purpose, we define an optimization problem as follows:

$$\min_{F,T} \sum_{i=1}^{K} KL(P_i(Y|F(X))||Q^T(Y|F(X))). \tag{5.3}$$

By using the definition of the KL divergence, we have:

$$\begin{aligned}
&\min_{F,T} \sum_{i=1}^{K} KL(P_i(Y|F(X))||Q^T(Y|F(X))) \\
&= \sum_{i=1}^{K} \mathop{\mathbb{E}}_{(X,Y)\sim P_i(X,Y)} [\log \frac{P_i(Y|F(X))}{Q^T(Y|F(X))}] \\
&= \sum_{i=1}^{K} \mathop{\mathbb{E}}_{(X,Y)\sim P_i(X,Y)} [\log P_i(Y|F(X))] - \sum_{i=1}^{K} \mathop{\mathbb{E}}_{(X,Y)\sim P_i(X,Y)} [\log Q^T(Y|F(X))].
\end{aligned} \tag{5.4}$$

The second term is actually the cross-entropy classification loss (Eq. 5.1), while the first one is the sum of K negative conditional entropy terms $\sum_{i=1}^{K} -H_{P_i}(Y|F(X))$. However, it is difficult to optimize $-H_{P_i}(Y|F(X))$ directly, since we do not

know the conditional distribution $P_i(Y|F(X))$. To overcome this issue, we first provide the following theorem to exploit the relationship between the negative conditional entropy term and the Jensen-Shannon divergence (JSD) between the conditional distributions $\{P_i(F(X)|Y=c)\}_{c=1}^{C}$.

THEOREM 1. *Assuming that all classes are equally likely, minimizing* $-H_{P_i}(Y|F(X))$ *is equivalent to minimizing the JSD between the conditional distributions* $\{P_i(F(X)|Y=c)\}_{c=1}^{C}$. *The global minimum is achieved if and only if* $P_i(F(X)|Y=1) = P_i(F(X)|Y=2) = \cdots = P_i(F(X)|Y=C)$. *Note that, if the dataset is balanced, it is easy to make the assumption satisfied. Otherwise, we can enforce it through biased batch sampling.*

The proof is given in Sec. 5.5. Inspired by Theorem 1 and the minimax game proposed in GAN (Goodfellow *et al.*, 2014) and conditional GAN (Gong *et al.*, 2019), we introduce K additional classifiers $\{T_i'\}_{i=1}^{K}$, and then present the following minimax game:

$$\min_{F} \max_{\{T_i'\}_{i=1}^{K}} V(F, T_1', T_2', \cdots, T_K') = \sum_{i=1}^{K} \mathop{\mathbb{E}}_{(X,Y)\sim P_i(X,Y)} [\log Q_i^{T_i'}(Y|F(X))], \tag{5.5}$$

where T_i' parameterized by ϕ_i' represents a classifier trained on data sampled from domain $\mathcal{D}_i$, and $Q_i^{T_i'}(Y|F(X))$ denotes the conditional distribution induced by T_i'. The following theorem (the proof can be found in Sec. 5.5) shows that the minimax game is equal to minimizing the JSD between the conditional distributions $\{P_i(F(X)|Y=c)\}_{c=1}^{C}$. According to Theorem 1, we can thus achieve the optimization of $\sum_{i=1}^{K} -H_{P_i}(Y|F(X))$.

THEOREM 2. *If* $U(F)$ *is the maximum value of* $V(F, T_1', T_2', \cdots, T_K')$, *i.e.,*

$$U(F) = \max_{\{T_i'\}_{i=1}^{K}} V(F, T_1', T_2', \cdots, T_K'), \tag{5.6}$$

the global minimum of the minimax game is attained if and only if $P_i(F(X)|Y = 1) = P_i(F(X)|Y = 2) = \cdots = P_i(F(X)|Y = C)$. *At this point,* $U(F)$ *attains the value* $-KC \log C$.

Therefore, our proposed entropy regularization loss can be defined as:

$$\min_{F} \max_{\{T_i'\}_{i=1}^K} \mathcal{L}_{er}(\theta, \{\phi_i'\}_{i=1}^K) = \sum_{i=1}^{K} \mathop{\mathbb{E}}_{(X,Y)\sim P_i(X,Y)} [\log Q_i^{T_i'}(Y|F(X))]. \tag{5.7}$$

Combining Eq. 5.7 with the classification loss (Eq. 5.1) and the domain discrimination loss (Eq. 5.2), we obtain the training objective:

$$\begin{aligned} &\min_{F,T} \max_{D,\{T_i'\}_{i=1}^K} \mathcal{L}(\theta, \phi, \psi, \{\phi_i'\}_{i=1}^K) \\ =&\mathcal{L}_{cls}(\theta, \phi) + \alpha_1 \mathcal{L}_{adv}(\theta, \psi) + \alpha_2 \mathcal{L}_{er}(\theta, \{\phi_i'\}_{i=1}^K), \end{aligned} \tag{5.8}$$

where α_1 and α_2 are trade-off parameters.

Algorithm. In our experiments, we observed that directly optimizing the loss Eq. 5.8 may show instability, since the minimax game in Eq. 5.7 encourages the learned features not to be distinguished by the classifiers. That may impede the optimization of the classification loss. To alleviate this issue, we introduce additional classifiers $\{T_i\}_{i=1}^K$ and add a new cross-entropy loss $\mathcal{L}_{cel}$:

$$\begin{aligned} \min_{F,\{T_i\}_{i=1}^K} \mathcal{L}_{cel}(\theta, \{\phi_i\}_{i=1}^K) = &- \sum_{i=1}^{K} \mathop{\mathbb{E}}_{(X,Y)\sim P_i(X,Y)} [\log Q_i^{T_i}(Y|\bar{F}(X))] \\ &- \sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} \mathop{\mathbb{E}}_{(X,Y)\sim P_j(X,Y)} [\log Q_i^{\bar{T}_i}(Y|F(X))], \end{aligned} \tag{5.9}$$

where $Q_i^{T_i}(Y|F(X))$ denotes the conditional distribution induced by T_i. Here, $\bar{F}$ and $\bar{T}_i$ mean that we fix the parameters of F and T during the training procedure, respectively. Specifically, we feed the learned features in the i^{th} domain into T_i to optimize its parameters ϕ_i. Additionally, we expect the feature extractor can map the data in domains $\{\mathcal{D}_j\}_{j=1,j\neq i}^K$ to a representation, which can

be distinguished by T_i accurately. This strategy, on the one hand, can impose regularization on the feature distribution of domains $\{\mathcal{D}_j\}_{j=1,j\neq i}^{K}$. On the other hand, the new loss can be considered as a complementary of $\mathcal{L}_{cls}$.

Thus, our final objective is formulated as:

$$\begin{aligned}&\min_{F,T,\{T_i\}_{i=1}^{K}}\max_{D,\{T_i'\}_{i=1}^{K}}\mathcal{L}(\theta,\phi,\psi,\{\phi_i\}_{i=1}^{K},\{\phi_i'\}_{i=1}^{K})\\ =&\mathcal{L}_{cls}+\alpha_1\mathcal{L}_{adv}+\alpha_2\mathcal{L}_{er}+\alpha_3\mathcal{L}_{cel},\end{aligned}\tag{5.10}$$

where α_3 is a weighting factor. To illustrate the training process clearly, we provide the pseudo-code of our algorithm in Alg. 1.

Algorithm 1: Training algorithm for domain generalization via entropy regularization.

Input: $\{S_i\}_{i=1}^{K}$: K source training datasets

Input: α_1, α_2, α_3: weighting factors

Output: F: feature extractor; T, $\{T_i\}_{i=1}^{K}$, $\{T_i'\}_{i=1}^{K}$: classifier; D: discriminator

while *training is not end* **do**

  Sample data from each training dataset respectively

  Update θ, ϕ, and ψ by optimizing the first and second terms of Eq. 5.10

  for i *in* $1:K$ **do**

    Sample data from the i^{th} dataset S_i

    Update $\{\phi_i\}_{i=1}^{K}$ by optimizing the forth term of Eq. 5.10

    Update θ, and $\{\phi_i'\}_{i=1}^{K}$ by optimizing the third term of Eq. 5.10

    Sample data from datasets $\{S_j\}_{j=1,j\neq i}^{K}$

    Update θ by optimizing the forth term of Eq. 5.10.

  end

end

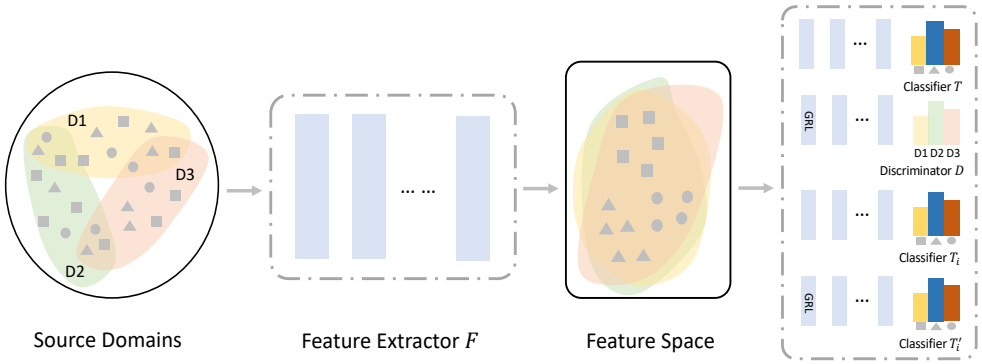


Figure 5.2. Illustration of our framework. GRL represents the gradient reversal layer. All components are trained, but only F and T are preserved for test.

Framework. Here, we provide an illustration of our framework in Figure 5.2 for better understanding of the proposed components. The main module consists of a feature extractor F and a classifier T. In addition, we exploit a domain discriminator D to discriminate domains, and $2K$ classifiers ($\{T_i\}_{i=1}^K$ and $\{T_i'\}_{i=1}^K$) to regularize the generated features. We insert a gradient reversal layer (GRL) (Ganin and Lempitsky, 2015) between F and D, and F and T_i', respectively. In the inference stage, only the main module (F and T) is required.

Discussion. In comparison with the typical classification loss, our entropy regularization loss can push the network to learn domain-invariant features. For instance, in the example of simulated data in Figure 5.1, the summation of the classification loss, the regularization loss and the domain adversarial loss is -0.16 in classifying the first dimension, and is -0.02 in classifying the second dimension. Therefore, our training objective can enforce the learned features to be domain-invariant.

5.4 Experiments

In this section, we study domain generalization on four datasets, including three simulated datasets, *i.e.,* Rotated MNIST (Ghifary *et al.*, 2015), Rotated CIFAR-10, and Rotated ModelNet40 (Wu *et al.*, 2015), and two real-world datasets, *i.e.,*

VLCS (Ghifary *et al.*, 2015) and PACS (Li *et al.*, 2017). We make comparisons against state-of-the-art methods to demonstrate the effectiveness of the proposed algorithm. We conduct extensive ablations to discuss our method comprehensively.

5.4.1 Simulated 2D Datasets

Rotated MNIST. Following the setting in (Ghifary *et al.*, 2015), we first randomly choose 100 samples for each category ($1,000$ in total) from the original dataset (LeCun *et al.*, 1998) to form the domain M_0. Then, we create 5 rotating domains $\{M_{15}, M_{30}, M_{45}, M_{60}, M_{75}\}$ by rotating each image in M_0 five times with 15 degrees intervals in clock-wise direction. As done by previous works (Li *et al.*, 2019d; Shankar *et al.*, 2018), we conduct leave-one-domain-out experiments by selecting one domain to hold out as the target. For fair comparisons, we exploit the standard MNIST CNN, where the feature network consists of two convolutional layers and one fully-connected (FC) layer, and the classifier has one FC layer. We train our model with the learning rate of $1e-4$ (F, T, and D), and $1e-5$ ($\{T_i, T_i'\}_{i=1}^5$) for $3,000$ iterations. We set the weighting factors to 0.5 (α_1), 0.005 (α_2), and 0.01 (α_3), respectively. We repeat all of the experiments 10 times, and report the average mean and standard deviation of recognition accuracy in Table 5.1.

Rotated CIFAR-10. We randomly choose 500 samples per category ($5,000$ in total) from the original CIFAR-10 dataset (Krizhevsky *et al.*, 2009), and then create additional 5 domains using the same strategy as stated in Rotated MNIST. We use AlexNet (Krizhevsky *et al.*, 2012) as our backbone network. In specific, the feature extractor F consists of the top layers of AlexNet model till the POOL5 layer, while T contains FC6, FC7, and an additional FC layer. For $\{T_i, T_i'\}_{i=1}^5$ and D, we use a similar architecture to T. We train the whole network *from scratch* with the learning rate of $1e-3$ (F, T, and D) and $1e-4$

Target	CrossGrad	MetaReg	Reptile	Feature-Critic	DeepAll	Basic-Adv	Ours
M_0	86.03	85.70	87.78	87.04	88.37 ± 1.19	88.88 ± 1.08	$\mathbf{90.09 \pm 1.25}$
M_{15}	98.92	98.87	99.44	**99.53**	99.13 ± 0.41	99.10 ± 0.19	99.24 ± 0.37
M_{30}	98.60	98.32	98.42	**99.41**	99.28 ± 0.27	99.25 ± 0.14	99.27 ± 0.16
M_{45}	98.39	98.58	98.80	**99.52**	99.09 ± 0.29	99.25 ± 0.17	99.31 ± 0.21
M_{60}	98.68	98.93	99.03	99.23	99.14 ± 0.28	99.16 ± 0.32	$\mathbf{99.45 \pm 0.19}$
M_{75}	88.94	89.44	87.42	**91.52**	87.48 ± 1.01	89.06 ± 1.54	90.81 ± 1.35
Avg.	94.93	94.97	95.15	96.04	95.42	95.78	**96.36**

Table 5.1. Results on MNIST dataset with object recognition accuracy (%) averaged over 10 runs.

Method	$M0$	$M15$	$M30$	$M45$	$M60$	$M75$	*Avg.*
DeepAll	71.28 ± 1.59	97.94 ± 0.32	99.14 ± 0.04	99.06 ± 0.19	99.07 ± 0.40	76.59 ± 0.89	90.51
Basic-Adv	75.85 ± 1.45	99.03 ± 0.18	99.16 ± 0.06	99.14 ± 0.11	99.29 ± 0.13	$\mathbf{81.14 \pm 1.34}$	92.27
Ours	$\mathbf{77.91 \pm 0.83}$	$\mathbf{99.05 \pm 0.22}$	$\mathbf{99.33 \pm 0.09}$	$\mathbf{99.39 \pm 0.14}$	$\mathbf{99.40 \pm 0.29}$	80.12 ± 0.60	**92.53**

Table 5.2. Results on CIFAR-10 dataset with object recognition accuracy (%) averaged over 5 runs.

($\{T_i, T_i'\}_{i=1}^5$) using the Adam optimizer (Kingma and Ba, 2014a) for $2,000$ iterations. The weighting factors (α_1, α_2, α_3) are set to 0.5, 0.001, and 0.1, respectively. We repeat all experiments 5 times, and provide the results in Table 5.2.

Results. We make comparisons against several recent works, including CrossGrad (Shankar *et al.*, 2018), MetaReg (Balaji *et al.*, 2018), Reptile (Nichol *et al.*, 2018), and Feature-Critic (Li *et al.*, 2019d), on Rotated MNIST. To better illustrate the generalization capabilities of our model, we also evaluate the performance of two additional models, *i.e.,* DeepAll and Basic-Adv, on both Rotated MNIST and Rotated CIFAR-10. DeepAll trains F and T on all of the source domains without performing any domain generalization (Eq. 5.1), while Basic-Adv is the basic solution through adversarial learning (Eq. 5.1 and Eq. 5.2). We can find all of the algorithms perform well on Rotated MNIST from Table 5.1, which means the generated domains have similar distributions. Nevertheless, our approach still performs better than existing approaches. Furthermore, the higher accuracy compared with DeepAll and Basic-Adv on both Rotated MNIST and Rotated CIFAR-10 shows the better generalization capabilities of the proposed algorithm.

Method	M_0 $([0°, 0°, 0°])$	M_1 $([30°, 15°, 45°])$	M_2 $([80°, 60°, 75°])$	M_3 $([120°, 90°, 60°])$	M_4 $([270°, 210°, 180°])$	*Avg.*
DGCNN						
DeepAll	34.81 ± 0.96	34.83 ± 1.82	54.04 ± 2.13	26.87 ± 1.21	29.30 ± 1.03	35.91
Basic-Adv	36.17 ± 1.48	$\mathbf{45.06 \pm 2.27}$	56.24 ± 1.09	29.28 ± 1.70	$\mathbf{31.00 \pm 2.69}$	39.55
Ours	$\mathbf{38.00 \pm 1.67}$	44.13 ± 1.26	$\mathbf{59.17 \pm 1.49}$	$\mathbf{30.28 \pm 1.45}$	30.85 ± 2.01	**40.49**
PointNet						
DeepAll	16.47 ± 0.44	20.37 ± 0.57	33.83 ± 0.32	12.80 ± 0.75	16.72 ± 0.11	20.04
Basic-Adv	$\mathbf{19.68 \pm 0.69}$	22.91 ± 0.93	34.86 ± 0.71	$\mathbf{14.44 \pm 0.67}$	18.24 ± 0.52	22.02
Ours	16.56 ± 0.59	$\mathbf{25.31 \pm 0.23}$	$\mathbf{35.66 \pm 0.32}$	13.96 ± 0.56	$\mathbf{19.90 \pm 0.21}$	**22.28**

Table 5.3. Results on ModelNet40 dataset with 3D shape recognition accuracy (%) averaged over 5 runs.

Method	Pascal VOC2007	LabelMe	Caltech	SUN09	Average
MLP					
D-MATE (Ghifary *et al.*, 2015)	63.90	60.13	89.05	61.33	68.60
DBADG (Li *et al.*, 2017)	65.58	58.74	92.43	61.85	69.65
CCSA (Motiian *et al.*, 2017)	67.10	62.10	92.30	59.10	70.15
MetaReg (Balaji *et al.*, 2018)	65.00	60.20	92.30	64.20	70.43
CrossGrad (Shankar *et al.*, 2018)	65.50	60.00	92.00	64.70	70.55
DANN (Ganin *et al.*, 2016)	66.40	64.00	92.60	63.60	71.65
MMD-AAE (Li *et al.*, 2018b)	67.70	62.60	94.40	64.40	72.28
MLDG (Li *et al.*, 2018a)	67.70	61.30	94.40	65.90	72.33
Epi-FCR (Li *et al.*, 2019a)	67.10	**64.30**	94.10	65.90	72.85
DeepAll	70.07 ± 0.79	60.54 ± 1.02	93.83 ± 1.08	65.95 ± 1.13	72.60
Basic-Adv	70.47 ± 0.59	60.94 ± 0.94	93.84 ± 1.00	66.05 ± 0.91	72.82
Ours	$\mathbf{70.54 \pm 0.55}$	60.81 ± 1.38	$\mathbf{94.44 \pm 0.98}$	$\mathbf{66.11 \pm 0.75}$	**72.97**
E2E					
DBADG (Li *et al.*, 2017)	69.99	63.49	93.64	61.32	72.11
JiGen (Carlucci *et al.*, 2019)	70.62	60.90	96.93	64.30	73.19
MMLD (Matsuura and Harada, 2020a)	71.96	58.77	96.66	68.13	73.88
CIDDG (Li *et al.*, 2018d)	73.00	58.30	97.02	68.89	74.30
DeepAll	73.11 ± 0.67	58.07 ± 0.52	$\mathbf{97.15 \pm 0.40}$	68.79 ± 0.44	74.28
Basic-Adv	72.79 ± 0.67	$\mathbf{58.53 \pm 0.69}$	97.00 ± 0.50	68.70 ± 0.69	74.26
Ours	$\mathbf{73.24 \pm 0.49}$	58.26 ± 0.82	96.92 ± 0.40	$\mathbf{69.10 \pm 0.46}$	**74.38**

Table 5.4. Results on VLCS dataset with object recognition accuracy (%) averaged over 20 runs.

5.4.2 Simulated 3D Dataset

Rotated ModelNet40. Here, we evaluate our method on a simulated 3D shape classification dataset, *i.e.,* ModelNet40 (Wu *et al.*, 2015), which consists of $9,843$ 3D models for training and $2,468$ for testing. We uniformly sample $1,024$ points as input for each 3D model on the mesh surfaces. In addition, the 3D models in ModelNet40 have been pre-aligned to the common up direction and horizontal facing direction. To evaluate the generalization capability and speed up the evaluation, we sample 20 categories ($5,112$ samples for training,

Method	Art Painting	Cartoon	Photo	Sketch	Average
D-MATE (Ghifary *et al.*, 2015)	60.27	58.65	91.12	47.68	64.48
CrossGrad (Shankar *et al.*, 2018)	61.00	67.20	87.60	55.90	67.93
DBADG (Li *et al.*, 2017)	62.86	66.97	89.50	57.51	69.21
MLDG (Li *et al.*, 2018a)	66.23	66.88	88.00	58.96	70.01
Epi-FCR (Li *et al.*, 2019a)	64.70	72.30	86.10	65.00	72.03
Feature-Critic (Li *et al.*, 2019d)	64.89	71.72	89.94	61.85	71.20
CIDDG (Li *et al.*, 2018d)	66.99	68.62	90.19	62.88	72.20
MetaReg (Balaji *et al.*, 2018)	69.82	70.35	91.07	59.26	72.62
JiGen (Carlucci *et al.*, 2019)	67.63	71.71	89.00	65.18	73.38
MMLD (Matsuura and Harada, 2020a)	69.27	**72.83**	88.98	66.44	74.38
MASF (Dou *et al.*, 2019)	70.35	72.46	**90.68**	67.33	75.21
DeepAll	68.35 ± 0.80	70.14 ± 0.87	90.83 ± 0.32	64.98 ± 1.92	73.57
Basic-Adv	71.34 ± 0.81	70.11 ± 1.18	88.86 ± 0.50	70.91 ± 0.94	75.31
Ours	$\mathbf{71.34 \pm 0.87}$	70.29 ± 0.77	89.92 ± 0.42	$\mathbf{71.15 \pm 1.01}$	**75.67**

Table 5.5. Results on PACS dataset with object recognition accuracy (%) averaged over 5 runs.

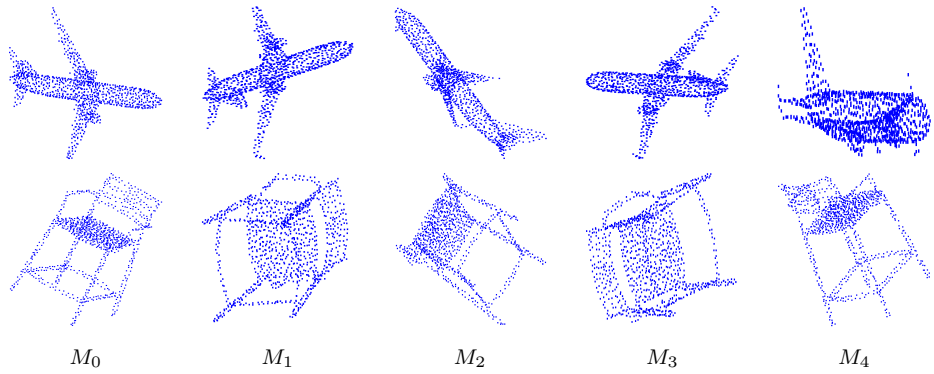


Figure 5.3. Data Visualization on Rotated ModelNet40 dataset. For better observation, we select different viewpoints for the two objects.

$1,202$ samples for testing) from the original ModelNet40 dataset (M_0), and then create additional 4 domains by rotating each point cloud in M_0 four times: 1) M_1 ($[30°, 15°, 45°]$), 30 degrees in x-axis, 15 degrees in y-axis, and 45 degrees in z-axis; 2) M_2 ($[80°, 60°, 75°]$), 80 degrees in x-axis, 60 degrees in y-axis, and 75 degrees in z-axis; 3) M_3 ($[120°, 90°, 60°]$), 120 degrees in x-axis, 90 degrees in y-axis, and 60 degrees in z-axis; 4) M_4 ($[270°, 210°, 180°]$), 270 degrees in x-axis, 210 degrees in y-axis, and 180 degrees in z-axis. Visualization for data in the simulated dataset is shown in Figure 5.3. We use DGCNN (Wang *et al.*, 2018b) and PointNet (Qi *et al.*, 2017a) as the backbone network, respectively. Take an example of DGCNN. In detail, the feature extractor F consists of shape

representation layers of DGCNN model, while T contains the remaining linear layers. For $\{T_i, T_i'\}_{i=1}^4$ and D, we use a similar architecture to that in Rotated CIFAR-10. We train the whole network *from scratch* with the learning rate of $1e-3$ (F, T, and D) and $1e-4$ ($\{T_i, T_i'\}_{i=1}^4$) using the SGD optimizer for 80 epochs. The weighting factors (α_1, α_2, α_3) are set to 0.5, 0.01, and 0.1, respectively. For PointNet, we exploit similar configurations. We repeat all experiments 5 times, and provide the results in Table 5.3.

Results. To illustrate the generalization capabilities of our model, we also evaluate the performance of two additional models, *i.e.,* DeepAll and Basic-Adv as we do on Rotated CIFAR-10 and Rotated MNIST. As shown in Table 5.3, we can observe that all three methods, especially when PointNet is used as the backbone, yield low scores. Nevertheless, our approach still outperforms other two methods. In addition, the accuracy of the model DGCNN (PointNet) trained on $\{M_1, M_2, M_3, M_4\}$ is 96.01% (72.58%) on the validation dataset, while 38.00% (16.56%) on the test set, *i.e.,* M_0. The great performance drops mean that the domain shift, like geometric changes, in 3D shape classification is a very serious problem for current deep networks, which is worth further investigating. In addition, the comparisons between DGCNN and PointNet show that the local shape representation operations also have significant impacts on the generalization capability, which would motivate us to develop more effective operations.

5.4.3 Real-World Datasets

VLCS. VLCS (Ghifary *et al.*, 2015) contains images from four well-known datasets, *i.e.,* Pascal VOC2007 (V) (Everingham *et al.*, 2010), LabelMe (L) (Russell *et al.*, 2008), Caltech (C) (Fei-Fei *et al.*, 2004), and SUN09 (S) (Choi *et al.*, 2010). There are five categories, including bird, car, chair, dog, and person. Following previous works (Ghifary *et al.*, 2015; Li *et al.*, 2019a; Dou *et al.*, 2019), we randomly split each domain data into training (70%) and test (30%) sets, and

do the leave-one-out evaluation. For the configuration of the network, we consider two cases, *i.e.,* MLP and E2E. In specific, in MLP, we use the pre-extracted DeCAF6 features (4096-dimensional vector) as the input, and F consists of two FC layers with latent dimensions of 1024 and 128. For the classifiers T and $\{T_i, T_i'\}_{i=1}^3$, we use one FC layer, respectively. For the discriminator D, we utilize three FC layers with the output dimensions of 128, 64, and 3 (the number of source domains). In this case, we train our model with the learning rate of $1e-3$ for 30 epochs using the SGD optimizer. We set all trade-off parameters to 0.1. In another setting (E2E), we employ the same network configuration as used on Rotated CIFAR-10, but use the model pre-trained on ImageNet (Krizhevsky *et al.*, 2012). We set the learning rate to $1e-4$, and the weighting factors α_1, α_2, and α_3 to 0.1, 0.001, and 0.05, respectively. We train the model with the batch size of 64 for each source domain for 60 epochs and repeat all of the experiments 20 times.

PACS. PACS (Li *et al.*, 2017) is proposed specially for domain generalization. It contains four domains, *i.e.,* Photo (P), Art Painting (A), Cartoon (C), and Sketch (S), and seven categories: dog, elephant, giraffe, guitar, house, horse, and person. For a fair comparison, we use the same training and validation split as presented in (Li *et al.*, 2017). Our network configuration is the same as that used for VLCS (E2E), and we set the weighting factors to 0.5 (α_1), 0.01 (α_2), and 0.05 (α_3), respectively. Then we train the model with the learning rate of $1e-3$ (F, T, D) and $1e-4$ ($\{T_i, T_i'\}_{i=1}^3$) for 60 epochs. We repeat all experiments 5 times, and report the results in Tabel 5.5.

Results. As shown in Table 5.4, although the baselines (DeepAll and Basic-Adv) are competitive with previous methods in both cases (MLP and E2E), our proposed entropy regularization still improves the performance further on VLCS. Furthermore, the highest average score and the highest score on several domains of PACS can also demonstrate the effectiveness of our approach. For example, Table 5.5 shows that our method improves the average accuracy by

2.1% on PACS over DeepAll, and improves 6.17% and 2.99% on Sketch and Art Painting, respectively. In addition, from the results in Table 5.4 and Table 5.5, we can observe that the performance (Ours *v.s.* DeepAll and Basic-Adv *v.s.* DeepAll) gains obtained by our regularization policy on PACS are more notable than those on VLCS. A possible reason we guess is that only one domain (C) in VLCS is object-centric, while others are all scene-centric. This makes the generalization of the model difficult, although the domain shifts in VLCS are small (Li *et al.*, 2017). In contrast, the images in all domains of PACS are mostly object-centric, and objects in different domains mainly have different styles and shapes. This can better evaluate the generalization capabilities of the model.

5.4.4 Ablation Studies

The experimental results above have demonstrated the effectiveness of our proposed algorithm for domain generalization. Here, we provide the ablation studies on the designed loss and network backbone to analyze the contributions of the proposed entropy regularization further.

Different Weighting Factors. We conduct various experiments with different weighting factors on PACS to examine their impacts. We report the average accuracy of 5 trials in Table 5.7. The results marked by the "gray" color correspond to the results reported in Table 5.5. "-" means the corresponding loss term is ignored. As shown in Table 5.7, in most cases, our proposed conditional entropy regularization ($\alpha_2 \neq 0$) can yield some improvements. Besides, by optimizing the full objective, our approach can further improve the generalization capabilities of the model.

Deeper Networks. We further study the generalization capabilities of our model by taking deeper networks, *e.g.,* ResNet-18 and RestNet-50 (He *et al.*, 2016), as the backbone network. The models are pre-trained on ImageNet, and fine-tuned

on PACS using the proposed loss. In specific, we take the last FC layer as our task network T, and other layers as the feature extractor F. We use three FC layers with output dimensions of 1024, 256, and the number of source domains / categories to construct the discriminator D and classifiers $\{T_i, T_i'\}_{i=1}^3$, respectively. For both ResNet-18 and ResNet-50, we use the same hyper-parameters, *i.e.*, $\alpha_1 = 0.1$, $\alpha_2 = 0.001$, $\alpha_3 = 0.05$, and the learning rate of $1e-3$ (F, T, D) and $1e-4$ ($\{T_i, T_i'\}_{i=1}^3$). We learn models for 100 epochs, and report the average scores of 5 trials. As shown in Table 5.6, even though we take deeper networks as our backbones, our approach still yield higher scores than the two baselines.

Class Imbalance. We address the class imbalance issue by using the weighted cross-entropy loss according to the number of each class in each batch. If not using the weighted loss *i.e.,* setting the weight to 1 for each class, the model yields a lower average accuracy of 75.58% (weighted loss used: 75.67%) on PACS, but still has better generalization capabilities.

Feature Visualization. To better understand the distribution of the learned features, we exploit t-SNE (Maaten and Hinton, 2008) to analyze the feature space learned by DeepAll, Basic-Adv, and Ours, respectively. We conduct this study on PACS, and in specific, we take the Photo dataset as the target, and others as the source. As shown in Figure 5.4, both Ours and Basic-Adv are capable of minimizing the distance between the distributions of the domains. For example, in DeepAll (Domains), we can observe that the Sketch (Green) is far away from other domains, while in Ours (Domains) and Basic-Adv (Domain), domains are clustered better. Furthermore, the comparison between Ours (Classes, Domains) and Basic-Adv (Classes, Domains) can show that our approach also discriminates the data from different categories better than Basic-Adv.

Method	Art Painting	Cartoon	Photo	Sketch	Average
ResNet-18					
DeepAll	78.93 ± 0.46	75.02 ± 0.89	96.60 ± 0.16	70.48 ± 0.84	80.25
Basic-Adv	80.54 ± 1.71	75.21 ± 0.92	$\mathbf{96.67 \pm 0.21}$	70.65 ± 1.91	80.77
Ours	$\mathbf{80.70 \pm 0.71}$	$\mathbf{76.40 \pm 0.34}$	96.65 ± 0.21	$\mathbf{71.77 \pm 1.27}$	**81.38**
ResNet-50					
DeepAll	86.18 ± 0.34	76.79 ± 0.33	98.14 ± 0.15	74.66 ± 0.93	83.94
Basic-Adv	87.11 ± 1.08	78.65 ± 1.13	98.22 ± 0.17	$\mathbf{76.48} \pm 1.09$	85.11
Ours	$\mathbf{87.51 \pm 1.03}$	$\mathbf{79.31 \pm 1.40}$	$\mathbf{98.25 \pm 0.12}$	76.30 ± 0.65	**85.34**

Table 5.6. Results of deeper networks on PACS dataset with object recognition accuracy (%) averaged over 5 runs.

5.5 Proofs

In this section, we provide the proofs of Theorem 1 and Theorem 2.

5.5.1 Proof of Theorem 1

PROOF. According to the definition of mutual information and under the assumption that all classes are equally likely, we have:

$$
\begin{aligned}
&- H_{P_i}(Y|F(X)) \\
=&I_{P_i}(Y, F(X)) - H(Y) \\
=&H_{P_i}(F(X)) - H_{P_i}(F(X)|Y) - H(Y) \\
=&-\frac{1}{C}\sum_{c=1}^{C} \mathop{\mathbb{E}}_{X' \sim P_i^F(X|Y)} \log P_i(X') + \frac{1}{C}\sum_{c=1}^{C} \mathop{\mathbb{E}}_{X' \sim P_i^F(X|Y)} \log P_i(X'|Y=c) - H(Y) \\
=&\frac{1}{C}\sum_{c=1}^{C} \mathop{\mathbb{E}}_{X' \sim P_i^F(X|Y)} \log \frac{P_i(X'|Y=c)}{P_i(X')} - H(Y) \\
=&\frac{1}{C}\sum_{c=1}^{C} KL(P_i(X'|Y=c)||P_i(X')) - H(Y) \\
=&\frac{1}{C}\sum_{c=1}^{C} KL(P_i(F(X)|Y=c)||P_i(F(X))) - H(Y) \\
=&JSD(P_i(F(X)|Y=1), P_i(F(X)|Y=2), \cdots, P_i(F(X)|Y=C)) - H(Y).
\end{aligned}
\tag{5.11}
$$

Since $H(Y)$ is a constant, then minimizing $-H_{P_i}(Y|F(X))$ is equivalent to minimizing $JSD(P_i(F(X)|Y = 1), P_i(F(X)|Y = 2), \cdots, P_i(F(X)|Y = C))$, the global minimum of which is achieved at $P_i(F(X)|Y = 1) = P_i(F(X)|Y = 2) = \cdots = P_i(F(X)|Y = C)$. □

5.5.2 Proof of Theorem 2

PROPOSITION 1. *Let* $V(F, \{T_i'\}) = \sum_{i=1}^{K} \mathop{\mathbb{E}}_{(X,Y)\sim P_i(X,Y)} [\log Q_i^{T_i'}(Y|F(X))]$. *Then the optimal prediction probabilities of* T_i' *are*

$$\langle T_i'^*(\mathbf{x}_i')\rangle_c = Q_i^{T_i'^*}(Y = c|\mathbf{x}_i') = \frac{P_i(\mathbf{x}_i'|Y = c)}{\sum_{c=1}^{C} P_i(\mathbf{x}_i'|Y = c)}, \tag{5.12}$$

where $\langle \mathbf{z}\rangle_i$ *denotes the* i^{th} *element of* $\mathbf{z}$, *and* $\mathbf{x}_i' = F(\mathbf{x}_i)$.

PROOF. For a fixed F, $\min_F \max_{\{T_i'\}} V(F, \{T_i'\})$ reduces to maximizing $V(F, \{T_i'\}_{i=1}^{K})$ w.r.t. $\{T_1', T_2', \cdots, T_K'\}$[3]:

$$\begin{aligned} &\{\langle T_i'^*(\mathbf{x}')\rangle_1, \langle T_i'^*(\mathbf{x}')\rangle_2, \cdots, \langle T_i'^*(\mathbf{x}')\rangle_C\} \\ =&arg \max_{\{\langle T_i'(\mathbf{x})\rangle_c\}_{c=1}^{C}} \sum_{c=1}^{C} \int_{\mathbf{x}_i'} P_i(\mathbf{x}_i'|Y = c) \log(\langle T_i'(\mathbf{x}_i')\rangle_c) d\mathbf{x}_i', \\ &s.t. \sum_{c=1}^{C} \langle T_i'(\mathbf{x}_i')\rangle_c = 1. \end{aligned} \tag{5.13}$$

Maximizing the value function point-wisely and applying Lagrange multipliers, we obtain the following problem:

$$\begin{aligned} &\{\langle T_i'^*(\mathbf{x}')\rangle_1, \langle T_i'^*(\mathbf{x}')\rangle_2, \cdots, \langle T_i'^*(\mathbf{x}')\rangle_C\} \\ =&arg \max_{\{\langle T_i'(\mathbf{x}')\rangle_c\}_{c=1}^{C}} \sum_{c=1}^{C} P_i(\mathbf{x}_i'|Y = c) \log(\langle T_i'(\mathbf{x}_i')\rangle_c) + \lambda_i(\sum_{c=1}^{C} \langle T_i'(\mathbf{x}_i')\rangle_c - 1). \end{aligned} \tag{5.14}$$

Setting the derivative of Eq. 5.14 w.r.t. $\langle T_i'(\mathbf{x}_i')\rangle_c$ to zero, we obtain $\langle T_i'^*(\mathbf{x}_i)\rangle_c = -\frac{P_i(\mathbf{x}_i'|Y=c)}{\lambda_i}$. Through substituting the value of $\langle T_i'^*(\mathbf{x}_i)\rangle_c$ into the constraint

[3]Here, we only consider T_i' for simplicity.

$\sum_{c=1}^{C}\langle T_i'(\mathbf{x}_i')\rangle_c = 1$, we can obtain $\lambda_i = -\sum_{c=1}^{C} P_i(\mathbf{x}_i'|Y=c)$, and thus get the optimal solution $\langle T_i'^*(\mathbf{x}_i')\rangle_c = \frac{P_i(\mathbf{x}_i'|Y=c)}{\sum_{c=1}^{C} P_i(\mathbf{x}_i'|Y=c)}$. □

THEOREM 3. *If $U(F)$ is the maximum value of $V(F, \{T_i'\}_{i=1}^{K})$, i.e.,*

$$U(F) = \sum_{i=1}^{K}\sum_{c=1}^{C} \mathop{\mathbb{E}}_{X_i\sim P_i(X)}[\log \frac{P_i(X_i'|Y=c)}{\sum_{c=1}^{C} P_i(X_i'|Y=c)}], \tag{5.15}$$

the global minimum of the minimax game is attained if and only if $P_i(X_i'|Y=1) = P_i(X_i'|Y=2) = \cdots = P_i(X_i'|Y=C)$ for any $i \in \{1, 2, \cdots, K\}$, where $U(F)$ achieves the value $-KC\log C$.

PROOF. Adding $KC\log C$ to $U(F)$ can obtain:

$$\begin{aligned} U(F) + KC\log C &= \sum_{i=1}^{K}\sum_{c=1}^{C}\{\mathop{\mathbb{E}}_{X_i\sim P_i(X)}[\log \frac{P_i(X_i'|Y=c)}{\sum_{c=1}^{C} P_i(X_i'|Y=c)}] + \log C\} \\ &= \sum_{i=1}^{K}\sum_{c=1}^{C} \mathop{\mathbb{E}}_{X_i\sim P_i(X)}[\log \frac{P_i(X_i'|Y=c)}{\frac{1}{C}\sum_{c=1}^{C} P_i(X_i'|Y=c)}] \\ &= \sum_{i=1}^{K}\sum_{c=1}^{C} KL(P_i(X_i'|Y=c)||\frac{1}{C}\sum_{c=1}^{C} P_i(X_i'|Y=c)). \end{aligned} \tag{5.16}$$

According to the definition of the Jensen-Shannon divergence, we can obtain $U(F) = -KC\log C + \sum_{i=1}^{K} C\cdot JSD(P_i(X_i'|Y=1), P_i(X_i'|Y=2), \cdots, P_i(X_i'|Y=C))$. Since the JSD between multiple distributions is always non-negative, and zero iff they are equal, then we have

$$\begin{aligned} &P_1(X_1'|Y=1) = P_1(X_1'|Y=2) = \cdots = P_1(X_1'|Y=C), \\ &P_2(X_2'|Y=1) = P_2(X_2'|Y=2) = \cdots = P_2(X_2'|Y=C), \\ &\cdots \\ &P_K(X_K'|Y=1) = P_K(X_K'|Y=2) = \cdots = P_K(X_K'|Y=C), \end{aligned} \tag{5.17}$$

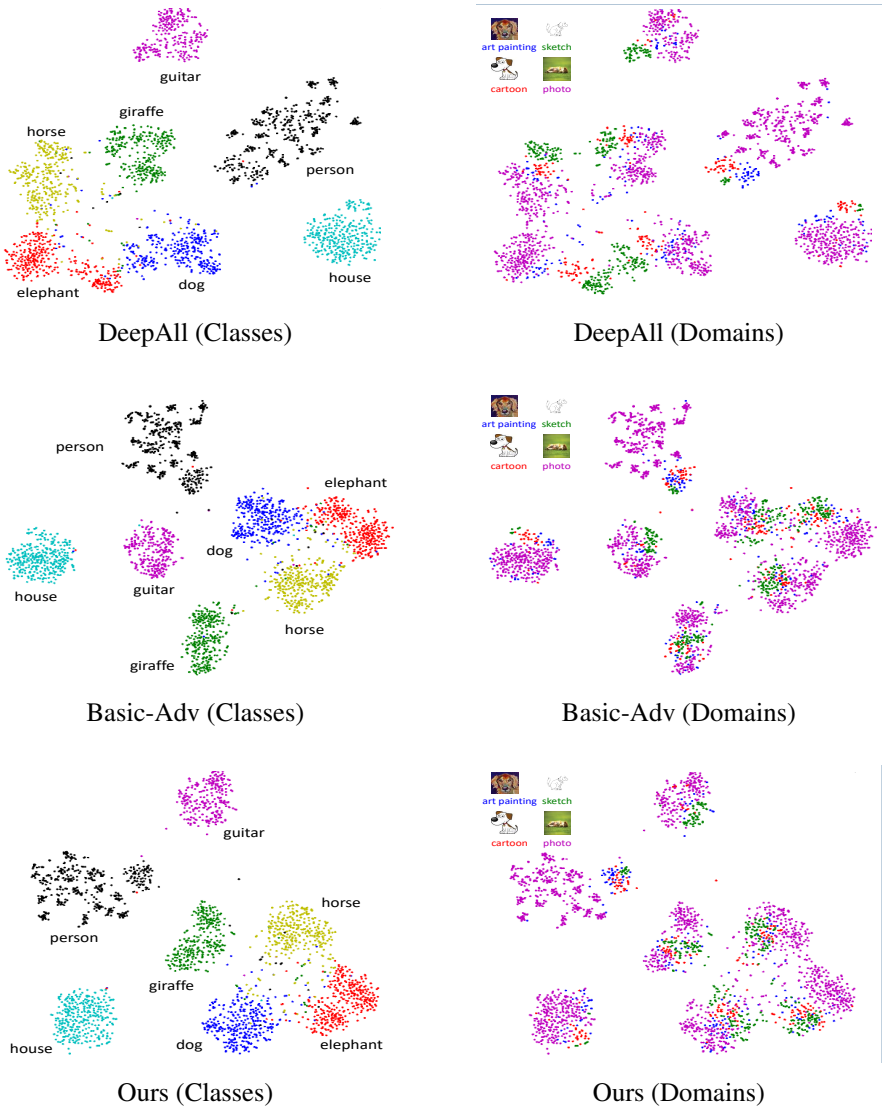
and the global minimum of $U(F)$ is $-KC\log C$. □

$\alpha_1, \alpha_2, \alpha_3$	Art Painting	Cartoon	Photo	Sketch	Average
- , - , -	68.35 ± 0.80	70.14 ± 0.87	90.83 ± 0.32	64.98 ± 1.92	**73.57**
1.0 , - , -	64.46 ± 3.80	64.07 ± 3.01	83.48 ± 1.39	66.70 ± 2.64	69.68
0.5 , - , -	71.35 ± 0.81	70.11 ± 1.18	88.86 ± 0.50	70.91 ± 0.94	**75.31**
0.1 , - , -	68.22 ± 0.89	70.13 ± 0.67	90.60 ± 0.37	64.61 ± 1.93	73.39
0.5 , 0.05 , -	70.83 ± 1.35	70.06 ± 0.98	89.25 ± 0.38	71.34 ± 0.82	**75.37**
0.5 , 0.01 , -	71.05 ± 1.62	70.29 ± 0.88	89.44 ± 0.36	70.06 ± 1.80	75.21
0.5 , 0.001 , -	71.72 ± 0.77	69.84 ± 1.65	88.88 ± 0.42	70.85 ± 0.83	75.32
0.5 , - , 0.5	68.92 ± 0.59	69.62 ± 0.51	89.99 ± 0.38	70.04 ± 0.63	74.74
0.5 , - , 0.1	71.04 ± 0.96	69.78 ± 0.98	89.68 ± 0.51	70.95 ± 0.81	**75.36**
0.5 , - , 0.05	71.59 ± 1.01	68.97 ± 1.42	89.57 ± 0.23	69.81 ± 3.45	74.99
0.5 , 0.05 , 0.1	71.09 ± 1.10	69.55 ± 0.54	89.56 ± 0.33	71.31 ± 0.90	75.37
0.5 , 0.01 , 0.1	70.91 ± 0.81	70.05 ± 1.33	89.80 ± 0.44	71.46 ± 0.46	75.56
0.5 , 0.005 , 0.1	70.95 ± 0.77	69.78 ± 0.91	89.56 ± 0.64	71.00 ± 1.12	75.32
0.5 , 0.05 , 0.05	70.55 ± 1.17	69.57 ± 1.14	89.33 ± 0.55	70.40 ± 2.88	74.96
0.5 , 0.01 , 0.05	71.34 ± 0.87	70.29 ± 0.77	89.92 ± 0.42	71.15 ± 1.02	**75.67**
0.5 , 0.005 , 0.05	70.51 ± 2.26	69.60 ± 0.58	89.69 ± 0.39	71.51 ± 0.84	75.33

Table 5.7. Results with different weighting factors on PACS.

5.6 Conclusion

In this chapter, we aim at learning the domain-invariant conditional distribution, which the basic adversarial learning based solutions cannot reach. We analyze the issues existed in related works, and propose an entropy regularization term, *i.e.,* the conditional entropy $H(Y|F(X))$, as the remedy. Our approach can produce domain-invariant features by optimizing the proposed regularization term coupled with the cross-entropy loss and the domain adversarial loss, and thus has a better generalization capability. The experimental results on both simulated and real-world datasets demonstrate the effectiveness of our proposed method. In the future, we can extend our approach to other challenging tasks, like semantic segmentation.



DeepAll (Classes) DeepAll (Domains)

Basic-Adv (Classes) Basic-Adv (Domains)

Ours (Classes) Ours (Domains)

Figure 5.4. Feature visualization. Left: different colors represent different classes; Right: different colors indicate different domains (Target: Photo). Best viewed in color (Zoom in for details).

CHAPTER 6

Conclusions

In this thesis, we studied the problem of 3D information prediction and understanding in the deep learning framework. In detail, our goal was to investigate several crucial issues in deep 3D information prediction and understanding, such as multi-modal fusion, sparse data processing, relation learning, model generalization, and unsupervised learning, through studying four specific tasks, including monocular depth estimation, depth completion, point cloud analysis, and domain generalization.

First, we studied the domain adaptation from synthetic data to real data for unsupervised monocular depth estimation. We found that previous works ignore the geometric information of the natural images and thus the generated images might suffer from distortions in the image-to-image translation process, which then causes the performance drop of depth estimation model. To alleviate this issue, we proposed to jointly explore the ground truth data in synthetic data and the epipolar geometry in real data. We demonstrated our model is able to generate high-quality image-to-image translation results and depth maps through a comparison with previous related works.

Second, we analyzed the shortcomings of standard convolution with fixed size kernel in modeling the contextual information of sparse data, and proposed to exploit the graph propagation to capture rich multi-modal contexts effectively through developing a co-attention guided graph propagation module. Then, for

multi-modal information fusion, we introduced a symmetric gated fusion module, which has two branches, one focusing on fusing the depth information as supplementary into the RGB information and the other one doing the opposite. Taking advantage of the proposed two modules, our designed model achieves the state-of-the-art performance on two depth completion datasets, and at the same time has fewer parameters and lower computational costs.

Third, we developed an adaptive edge-to-edge interaction learning module for local shape representation of 3D point cloud data. We hypothesized that associating the neighbouring edges could potentially make the point-to-point relation more aware of the local structure and more robust. We, thereby, introduced an edge-to-edge interaction learning strategy to enhance the representation of point-to-point relation. Then, we extended the basic interaction module into its symmetric version to modeling the local structure thoroughly. At last, we designed the models for classification and segmentation using the proposed modules. To show the effectiveness of our method in point cloud analysis, we evaluated the performances on several public point cloud datasets. Our models outperform previous related works on almost all of metrics.

Finally, we investigated the domain generalization from multiple source domains to unseen target domains for a basic task, *i.e.,* object classification. We first revisited previous works which aim to learn domain-invariant representations across source domains, and found that these methods cannot ensure the conditional invariance of the learned features. To address this issue, we proposed an entropy regularization term which measures the dependency between the learned features and the category. Together with the adversarial training on the marginal distribution of the learned features, our model can guarantee the invariance of the joint distribution of learned features and category. We evaluated the generalization capability on a 3D object classification dataset as an initial step to the study on domain generalization in point cloud analysis as well

the 2D object classification datasets. The experimental results can show the effectiveness of our method.

This thesis suggests two possible further studies on deep 3D information prediction and understanding. First, since we can obtain the 3D information, such as depth data, from stereo data, multi-view images, or video, and then further analyze the 3D information, like detecting objects and labelling each pixel, it is worthwhile to study how to complete these two tasks in a unified end-to-end deep learning framework. As a result, learning the two tasks jointly would potentially enhance each other. Second, the domain generalization in 2D object classification has been studied extensively, while for 3D information prediction and understanding, like depth estimation, 3D segmentation, and 3D reconstruction, there is hardly any work exploring the problem. In the future study, we can investigate the domain generalization in these more complex and challenging tasks through taking advantage of the domain knowledge.

References

[Achituve *et al*.2021] Idan Achituve, Haggai Maron, and Gal Chechik. 2021. Self-supervised learning for domain adaptation on point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 123–133.

[Ajakan *et al*.2014] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. 2014. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*.

[Arjovsky *et al*.2019] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

[Armeni *et al*.2016] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 2016. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543.

[Atapour-Abarghouei and Breckon2018] Amir Atapour-Abarghouei and Toby P Breckon. 2018. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2810.

[Atapour-Abarghouei and Breckon2019a] Amir Atapour-Abarghouei and Toby P Breckon. 2019a. To complete or to estimate, that is the question: A multi-task approach to depth completion and monocular depth estimation. In *2019 International Conference on 3D Vision (3DV)*, pages 183–193. IEEE.

[Atapour-Abarghouei and Breckon2019b] Amir Atapour-Abarghouei and Toby P. Breckon. 2019b. Veritatem dies aperit - temporally consistent depth prediction enabled by a multi-task geometric and semantic scene understanding approach. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Atzmon *et al*.2018] Matan Atzmon, Haggai Maron, and Yaron Lipman. 2018. Point convolutional neural networks by extension operators. *ACM Trans.*

Graph., 37(4).

[Bak *et al.*2018] Slawomir Bak, Peter Carr, and Jean-Francois Lalonde. 2018. Domain adaptation through synthesis for unsupervised person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 189–205.

[Balaji *et al.*2018] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. 2018. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pages 998–1008.

[Barron and Poole2016] Jonathan T Barron and Ben Poole. 2016. The fast bilateral solver. In *European Conference on Computer Vision*, pages 617–632. Springer.

[Bay *et al.*2008] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359.

[Ben-Younes *et al.*2017] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. 2017. Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2612–2620.

[Blanchard *et al.*2011] Gilles Blanchard, Gyemin Lee, and Clayton Scott. 2011. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in neural information processing systems*, pages 2178–2186.

[Blanchard *et al.*2017] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. 2017. Domain generalization by marginal transfer learning. *arXiv preprint arXiv:1711.07910*.

[Bousmalis *et al.*2016] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in neural information processing systems*, pages 343–351.

[Cao *et al.*2016] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. 2016. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *arXiv preprint arXiv:1605.02305*.

[Cao *et al.*2018] Y. Cao, T. Zhao, K. Xian, C. Shen, Z. Cao, and S. Xu. 2018. Monocular depth estimation with augmented ordinal depth relationships. *IEEE Transactions on Image Processing*, pages 1–1.

[Carlucci *et al*.2019] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. 2019. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238.

[Chang and Chen2018] Jia-Ren Chang and Yong-Sheng Chen. 2018. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418.

[Chen *et al*.2016] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. 2016. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738.

[Chen *et al*.2017a] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848.

[Chen *et al*.2017b] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. 2017b. Multi-view 3d object detection network for autonomous driving. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6526–6534.

[Chen *et al*.2018] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. 2018. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3339–3348.

[Chen *et al*.2019] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. 2019. Learning joint 2d-3d representations for depth completion. In *The IEEE International Conference on Computer Vision (ICCV)*, October.

[Chen *et al*.2021] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. 2021. Equivariant point network for 3d point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14514–14523.

[Cheng *et al*.2018] Xinjing Cheng, Peng Wang, and Ruigang Yang. 2018. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119.

[Cheng *et al*.2019] Xuelian Cheng, Yiran Zhong, Yuchao Dai, Pan Ji, and Hongdong Li. 2019. Noise-aware unsupervised deep lidar-stereo fusion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*,

June.

[Cheng *et al.*2020a] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. 2020a. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *AAAI*, pages 10615–10622.

[Cheng *et al.*2020b] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. 2020b. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems*, 33.

[Chodosh *et al.*2018] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. 2018. Deep convolutional compressed sensing for lidar depth completion. *arXiv preprint arXiv:1803.08949*.

[Choi *et al.*2010] Myung Jin Choi, Joseph J Lim, Antonio Torralba, and Alan S Willsky. 2010. Exploiting hierarchical context on a large database of object categories. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 129–136. IEEE.

[Choi *et al.*2021] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. 2021. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11580–11590.

[Choy *et al.*2019] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084.

[Chu *et al.*2018] Hang Chu, Wei-Chiu Ma, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. 2018. Surfconv: Bridging 3d and 2d convolution for rgbd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3002–3011.

[Cordts *et al.*2016] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223.

[Dai *et al.*2017a] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017a. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 5828–5839.

[Dai *et al*.2017b] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017b. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773.

[Dalal and Triggs2005] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee.

[Deng *et al*.2018] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. 2018. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 994–1003.

[Deshmukh *et al*.2019] Aniket Anand Deshmukh, Yunwen Lei, Srinagesh Sharma, Urun Dogan, James W Cutler, and Clayton Scott. 2019. A generalization error bound for multi-class domain generalization. *arXiv preprint arXiv:1905.10392*.

[Dimitrievski *et al*.2018] Martin Dimitrievski, Peter Veelaert, and Wilfried Philips. 2018. Learning morphological operators for depth completion. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 450–461. Springer.

[Dosovitskiy *et al*.2015a] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015a. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766.

[Dosovitskiy *et al*.2015b] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. 2015b. Learning to generate chairs with convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1538–1546.

[Dou *et al*.2019] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. 2019. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pages 6447–6458.

[Eigen and Fergus2015] David Eigen and Rob Fergus. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658.

[Eigen *et al*.2014] David Eigen, Christian Puhrsch, and Rob Fergus. 2014. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374.

[Eldesokey *et al*.2019] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. 2019. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*.

[Eldesokey *et al*.2020] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. 2020. Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Everingham *et al*.2010] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338.

[Fan *et al*.2021a] Siqi Fan, Qiulei Dong, Fenghua Zhu, Yisheng Lv, Peijun Ye, and Fei-Yue Wang. 2021a. Scf-net: Learning spatial contextual features for large-scale point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14504–14513, June.

[Fan *et al*.2021b] Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. 2021b. Adversarially adaptive normalization for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8208–8217.

[Fang *et al*.2020] Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. 2020. Rethinking importance weighting for deep learning under distribution shift. *arXiv preprint arXiv:2006.04662*.

[Fei-Fei *et al*.2004] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE.

[Ferstl *et al*.2013] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. 2013. Image guided depth upsampling using

anisotropic total generalized variation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 993–1000.

[Fu *et al*.2018] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. 2018. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011.

[Fu *et al*.2020] Keren Fu, Deng-Ping Fan, Ge-Peng Ji, and Qijun Zhao. 2020. Jl-dcf: Joint learning and densely-cooperative fusion framework for rgb-d salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Fujiwara and Hashimoto2020] Kent Fujiwara and Taiichi Hashimoto. 2020. Neural implicit embedding for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11734–11743.

[Gaidon *et al*.2016] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. 2016. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349.

[Ganin and Lempitsky2015] Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*.

[Ganin *et al*.2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

[Garcia and Bruna2017] Victor Garcia and Joan Bruna. 2017. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*.

[Garg *et al*.2016] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. 2016. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer.

[Geiger *et al*.2012a] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012a. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Geiger *et al*.2012b] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012b. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE.

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

[Girshick *et al.*2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

[Godard *et al.*2017] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. 2017. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279.

[Gong *et al.*2012] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE.

[Gong *et al.*2016] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. 2016. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848.

[Gong *et al.*2018] Mingming Gong, Kun Zhang, Biwei Huang, Clark Glymour, Dacheng Tao, and Kayhan Batmanghelich. 2018. Causal generative domain adaptation networks. *arXiv preprint arXiv:1804.04333*.

[Gong *et al.*2019] Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. 2019. Twin auxilary classifiers gan. In *Advances in Neural Information Processing Systems*, pages 1328–1337.

[Gong *et al.*2021] Jingyu Gong, Jiachen Xu, Xin Tan, Jie Zhou, Yanyun Qu, Yuan Xie, and Lizhuang Ma. 2021. Boundary-aware geometric encoding for semantic segmentation of point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1424–1432.

[Goodfellow *et al.*2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[Graham *et al.*2018] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 2018. 3d semantic segmentation with submanifold sparse convolutional networks. *CVPR*.

[Gretton *et al*.2009] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. 2009. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5.

[Gretton *et al*.2012] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773.

[Guo *et al*.2020] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. 2020. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*.

[Han *et al*.2020] Wenkai Han, Chenglu Wen, Cheng Wang, Xin Li, and Qing Li. 2020. Point2node: Correlation learning of dynamic-node for point cloud feature modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10925–10932.

[Hanocka *et al*.2019] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. Meshcnn: A network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):90:1–90:12.

[He *et al*.2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[He *et al*.2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[He *et al*.2018a] L. He, G. Wang, and Z. Hu. 2018a. Learning depth from single images with deep neural network embedding focal length. *IEEE Transactions on Image Processing*, 27(9):4676–4689.

[He *et al*.2018b] Lei He, Guanghui Wang, and Zhanyi Hu. 2018b. Learning depth from single images with deep neural network embedding focal length. *IEEE Transactions on Image Processing*.

[Herrera *et al*.2013] Daniel Herrera, Juho Kannala, Janne Heikkilä, et al. 2013. Depth map inpainting under a second-order smoothness prior. In *Scandinavian Conference on Image Analysis*, pages 555–566. Springer.

[Hori *et al*.2018] Chiori Hori, Takaaki Hori, Gordon Wichern, Jue Wang, Tengyok Lee, Anoop Cherian, and Tim K Marks. 2018. Multimodal attention for fusion of audio and spatiotemporal features for video description. In *CVPR Workshops*, pages 2528–2531.

[Hospedales *et al*.2021] Timothy M Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J Storkey. 2021. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Hu *et al*.2020a] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. 2020a. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117.

[Hu *et al*.2020b] Zeyu Hu, Mingmin Zhen, Xuyang Bai, Hongbo Fu, and Chiew-lan Tai. 2020b. Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds. In *ECCV*, pages 222–239.

[Hu *et al*.2021] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. 2021. Bidirectional projection network for cross dimension scene understanding. In *CVPR*.

[Hua *et al*.2018] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. 2018. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993.

[Huang *et al*.2018] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. 2018. Recurrent slice networks for 3d segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2635.

[Huang *et al*.2020] Z. Huang, J. Fan, S. Cheng, S. Yi, X. Wang, and H. Li. 2020. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *IEEE Transactions on Image Processing*, 29:3429–3441.

[Imran *et al*.2019] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. 2019. Depth coefficients for depth completion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Ioffe and Szegedy2015] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

[Jaderberg *et al*.2015] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. 2015. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025.

[Jaritz *et al*.2018] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. 2018. Sparse and dense data with cnns:

Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE.

[Jaritz *et al*.2020] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. 2020. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12605–12614.

[Ji *et al*.2020] Wei Ji, Xi Li, Lina Wei, Fei Wu, and Yueting Zhuang. 2020. Context-aware graph label propagation network for saliency detection. *IEEE Transactions on Image Processing*, 29:8177–8186.

[Jiang *et al*.2019] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. 2019. Hierarchical point-edge interaction network for point cloud semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10433–10441.

[Jiang *et al*.2020a] Haiyong Jiang, Feilong Yan, Jianfei Cai, Jianmin Zheng, and Jun Xiao. 2020a. End-to-end 3d point cloud instance segmentation without detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12796–12805.

[Jiang *et al*.2020b] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. 2020b. Pointgroup: Dual-set point grouping for 3d instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Karsch *et al*.2014] Kevin Karsch, Ce Liu, and Sing Bing Kang. 2014. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158.

[Khodabandeh *et al*.2019] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G Macready. 2019. A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 480–490.

[Kim and Sukhatme2014] David Inkyu Kim and Gaurav S Sukhatme. 2014. Semantic labeling of 3d point clouds with object affordance for robot manipulation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5578–5584. IEEE.

[Kim *et al*.2018] Y. Kim, H. Jung, D. Min, and K. Sohn. 2018. Deep monocular depth estimation via integration of global and local predictions. *IEEE Transactions on Image Processing*, 27(8):4131–4144.

[Kim *et al*.2019] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. 2019. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20.

[KIM *et al*.2020] SEOHYUN KIM, JaeYoo Park, and Bohyung Han. 2020. Rotation-invariant local-to-global representation learning for 3d point cloud. *Advances in Neural Information Processing Systems*, 33.

[Kingma and Ba2014a] Diederik Kingma and Jimmy Ba. 2014a. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12.

[Kingma and Ba2014b] Diederik P Kingma and Jimmy Ba. 2014b. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Komarichev *et al*.2019] Artem Komarichev, Zichun Zhong, and Jing Hua. 2019. A-cnn: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7421–7430.

[Krizhevsky *et al*.2009] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

[Krizhevsky *et al*.2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Kundu *et al*.2018] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R Venkatesh Babu. 2018. Adadepth: Unsupervised content congruent adaptation for depth estimation. *arXiv preprint arXiv:1803.01599*.

[Kundu *et al*.2020] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. 2020. Virtual multi-view fusion for 3d semantic segmentation. In *European Conference on Computer Vision*, pages 518–535. Springer.

[Kuznietsov *et al*.2017] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. 2017. Semi-supervised deep learning for monocular depth map prediction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6647–6655.

[Ladicky *et al*.2014] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. 2014. Pulling things out of perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–96.

[Lai *et al*.2017] Wei-Sheng Lai, Jia-Bin Huang, and Ming-Hsuan Yang. 2017. Semi-supervised learning for optical flow with generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 354–364.

[Laina *et al*.2016] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. 2016. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248. IEEE.

[Landrieu and Simonovsky2018] Loic Landrieu and Martin Simonovsky. 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4558–4567.

[Lang *et al*.2020] Itai Lang, Asaf Manor, and Shai Avidan. 2020. Samplenet: Differentiable point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7578–7588.

[Langer *et al*.2020] Ferdinand Langer, Andres Milioto, Alexandre Haag, Jens Behley, and Cyrill Stachniss. 2020. Domain transfer for semantic segmentation of lidar data using deep neural networks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8263–8270. IEEE.

[Le *et al*.2020] Eric-Tuan Le, Iasonas Kokkinos, and Niloy J. Mitra. 2020. Going deeper with lean point networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[LeCun *et al*.1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lei *et al*.2020] Huan Lei, Naveed Akhtar, and Ajmal Mian. 2020. Seggcn: Efficient 3d point cloud segmentation with fuzzy spherical kernel. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11611–11620.

[Li *et al*.2015] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. 2015. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127.

[Li *et al*.2017] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. 2017. Deeper, broader and artier domain generalization. In *Proceedings of*

the IEEE international conference on computer vision, pages 5542–5550.

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

[Li *et al*.2018b] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. 2018b. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409.

[Li *et al*.2018c] Jiaxin Li, Ben M Chen, and Gim Hee Lee. 2018c. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406.

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

[Li *et al*.2018e] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. 2018e. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830.

[Li *et al*.2019a] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. 2019a. Episodic training for domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1446–1455.

[Li *et al*.2019b] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2019b. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9267–9276.

[Li *et al*.2019c] Qing Li, Shaoyang Chen, Cheng Wang, Xin Li, Chenglu Wen, Ming Cheng, and Jonathan Li. 2019c. Lo-net: Deep real-time lidar odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8473–8482.

[Li *et al*.2019d] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. 2019d. Feature-critic networks for heterogeneous domain generalisation. In *The Thirty-sixth International Conference on Machine Learning*.

[Li *et al*.2020a] Ang Li, Zejian Yuan, Yonggen Ling, Wanchao Chi, Chong Zhang, et al. 2020a. A multi-scale guided cascade hourglass network for depth completion. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 32–40.

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

[Liao *et al*.2017] Yiyi Liao, Lichao Huang, Yue Wang, Sarath Kodagoda, Yinan Yu, and Yong Liu. 2017. Parse geometry from a line: Monocular depth estimation with partial laser observation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5059–5066. IEEE.

[Lin *et al*.2020a] Yiqun Lin, Zizheng Yan, Haibin Huang, Dong Du, Ligang Liu, Shuguang Cui, and Xiaoguang Han. 2020a. Fpconv: Learning local flattening for point convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4293–4302.

[Lin *et al*.2020b] Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang. 2020b. Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Liu *et al*.2010] Beyang Liu, Stephen Gould, and Daphne Koller. 2010. Single image depth estimation from predicted semantic labels. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1253–1260. IEEE.

[Liu *et al*.2011] Ce Liu, Jenny Yuen, and Antonio Torralba. 2011. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994.

[Liu *et al*.2014] Miaomiao Liu, Mathieu Salzmann, and Xuming He. 2014. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723.

[Liu *et al*.2016a] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. 2016a. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039.

[Liu *et al*.2016b] Sifei Liu, Jinshan Pan, and Ming-Hsuan Yang. 2016b. Learning recursive filters for low-level vision via a hybrid neural network. In *European Conference on Computer Vision*, pages 560–576. Springer.

[Liu *et al*.2019a] Jinxian Liu, Bingbing Ni, Caiyuan Li, Jiancheng Yang, and Qi Tian. 2019a. Dynamic points agglomeration for hierarchical point sets learning. In *Proceedings of the IEEE International Conference on Computer*

Vision, pages 7546–7555.

[Liu *et al*.2019b] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 2019b. Point2sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8778–8785.

[Liu *et al*.2019c] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. 2019c. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5239–5248.

[Liu *et al*.2019d] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. 2019d. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904.

[Liu *et al*.2020] Ze Liu, Han Hu, Yue Cao, Zheng Zhang, and Xin Tong. 2020. A closer look at local aggregation operators in point cloud analysis. *ECCV*.

[Long *et al*.2013] Mingsheng Long, Guiguang Ding, Jianmin Wang, Jiaguang Sun, Yuchen Guo, and Philip S Yu. 2013. Transfer sparse coding for robust image representation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 407–414.

[Long *et al*.2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

[Loshchilov and Hutter2016] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

[Lowe1999] David G Lowe. 1999. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee.

[Lu *et al*.2016] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297.

[Lu *et al*.2020] Kaiyue Lu, Nick Barnes, Saeed Anwar, and Liang Zheng. 2020. From depth what can you see? depth completion via auxiliary image reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Ma and Karaman2018] Fangchang Ma and Sertac Karaman. 2018. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE.

[Ma *et al*.2019] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. 2019. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3288–3295. IEEE.

[Maas *et al*.2013] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.

[Maaten and Hinton2008] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

[Matsuura and Harada2020a] Toshihiko Matsuura and Tatsuya Harada. 2020a. Domain generalization using a mixture of multiple latent domains. In *AAAI*.

[Matsuura and Harada2020b] Toshihiko Matsuura and Tatsuya Harada. 2020b. Domain generalization using a mixture of multiple latent domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11749–11756.

[Maturana and Scherer2015] Daniel Maturana and Sebastian Scherer. 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE.

[Menze and Geiger2015] Moritz Menze and Andreas Geiger. 2015. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070.

[Moon *et al*.2018] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. 2018. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Motiian *et al*.2017] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. 2017. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5715–5725.

[Mousavian *et al*.2019] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 2019. 6-dof graspnet: Variational grasp generation for object manipulation.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2901–2910.

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

[Nath Kundu *et al*.2018] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R Venkatesh Babu. 2018. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2656–2665.

[Nezhadarya *et al*.2020] Ehsan Nezhadarya, Ehsan Taghavi, Ryan Razani, Bingbing Liu, and Jun Luo. 2020. Adaptive hierarchical down-sampling for point cloud classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Nichol *et al*.2018] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

[Pan and Yang2009] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

[Pan *et al*.2010] Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

[Park *et al*.2020] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. 2020. Non-local spatial propagation network for depth completion. *arXiv preprint arXiv:2007.10042*.

[Peng *et al*.2014] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. 2014. Rgbd salient object detection: a benchmark and algorithms. In *European conference on computer vision*, pages 92–109. Springer.

[Peng *et al*.2019] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415.

[PNVR *et al*.2020] Koutilya PNVR, Hao Zhou, and David Jacobs. 2020. Sharingan: Combining synthetic and real data for unsupervised geometry estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Pons-Moll *et al.*2017] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. 2017. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (TOG)*, 36(4):1–15.

[Qi *et al.*2017a] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660.

[Qi *et al.*2017b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108.

[Qi *et al.*2018] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. 2018. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291.

[Qiao *et al.*2020] Fengchun Qiao, Long Zhao, and Xi Peng. 2020. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565.

[Qiu *et al.*2019] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. 2019. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3313–3322.

[Ranjan *et al.*2019] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. 2019. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12240–12249.

[Repala and Dubey2018] Vamshi Krishna Repala and Shiv Ram Dubey. 2018. Dual cnn models for unsupervised monocular depth estimation. *arXiv preprint arXiv:1804.06324*.

[Roy and Todorovic2016] Anirban Roy and Sinisa Todorovic. 2016. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5506–5514.

[Russell *et al.*2008] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. 2008. Labelme: a database and web-based tool for

image annotation. *International journal of computer vision*, 77(1-3):157–173.

[Saenko *et al*.2010] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer.

[Sakuma and Konishi2021] Hiroki Sakuma and Yoshinori Konishi. 2021. Geometry-aware unsupervised domain adaptation for stereo matching. *arXiv preprint arXiv:2103.14333*.

[Saltori *et al*.2020] Cristiano Saltori, Stéphane Lathuilière, Nicu Sebe, Elisa Ricci, and Fabio Galasso. 2020. Sf-uda 3d: Source-free unsupervised domain adaptation for lidar-based 3d object detection. In *2020 International Conference on 3D Vision (3DV)*, pages 771–780. IEEE.

[Sauder and Sievers2019] Jonathan Sauder and Bjarne Sievers. 2019. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32:12962–12972.

[Saxena *et al*.2006] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. 2006. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168.

[Saxena *et al*.2009] Ashutosh Saxena, Min Sun, and Andrew Y Ng. 2009. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840.

[Schneider *et al*.2016] Nick Schneider, Lukas Schneider, Peter Pinggera, Uwe Franke, Marc Pollefeys, and Christoph Stiller. 2016. Semantically guided depth upsampling. In *German Conference on Pattern Recognition*, pages 37–48. Springer.

[Shankar *et al*.2018] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. 2018. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations (ICLR)*.

[Shi *et al*.2019] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. 2019. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7912–7921.

[Silberman *et al*.2012] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer.

[Simonyan and Zisserman2014] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576.

[Simonyan and Zisserman2015] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

[Sinha and Poggio1996] Pawan Sinha and Tomaso Poggio. 1996. Role of learning in three-dimensional form perception. *Nature*, 384(6608):460–463.

[Stekovic *et al.*2020] Sinisa Stekovic, Shreyas Hampali, Mahdi Rad, Sayan Deb Sarkar, Friedrich Fraundorfer, and Vincent Lepetit. 2020. General 3d room layout from a single view by render-and-compare. In *European Conference on Computer Vision*, pages 187–203. Springer.

[Sun and Saenko2016] Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer.

[Sun *et al.*2016] Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of frustratingly easy domain adaptation. In *AAAI*, volume 6, page 8.

[Szegedy *et al.*2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[Taigman *et al.*2016] Yaniv Taigman, Adam Polyak, and Lior Wolf. 2016. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*.

[Tang *et al.*2019] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. 2019. Learning guided convolutional network for depth completion. *arXiv preprint arXiv:1908.01238*.

[Tchapmi *et al.*2019] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. 2019. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Thomas *et al.*2019] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420.

[Torralba and Efros2011] Antonio Torralba and Alexei A Efros. 2011. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE.

[Tzeng *et al.*2014] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

[Uhrig *et al.*2017] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. 2017. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20. IEEE.

[Ulyanov *et al.*2017] Dmitry Ulyanov, Andrea Vedaldi, and Victor S Lempitsky. 2017. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*, volume 1, page 3.

[Van Gansbeke *et al.*2019] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. 2019. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6. IEEE.

[Vaswani *et al.*2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, page 6000–6010.

[Volpi *et al.*2018] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. 2018. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems*, pages 5334–5344.

[Wang and Neumann2018] Weiyue Wang and Ulrich Neumann. 2018. Depth-aware cnn for rgb-d segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150.

[Wang *et al.*2004] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

[Wang *et al.*2015] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. 2015. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2809.

[Wang *et al.*2017] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. 2017. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):1–11.

[Wang *et al.*2018a] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. 2018a. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2589–2597.

[Wang *et al.*2018b] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2018b. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*.

[Wang *et al.*2019a] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. 2019a. Densefusion: 6d object pose estimation by iterative dense fusion. *arXiv preprint arXiv:1901.04780*.

[Wang *et al.*2019b] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. 2019b. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10296–10305.

[Wang *et al.*2019c] Xu Wang, Jingming He, and Lin Ma. 2019c. Exploiting local and global structure for point cloud semantic segmentation with contextual point representations. In *Advances in Neural Information Processing Systems*, pages 4571–4581.

[Wang *et al.*2019d] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019d. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12.

[Wang *et al.*2020] A. Wang, Z. Fang, Y. Gao, S. Tan, S. Wang, S. Ma, and J. Hwang. 2020. Adversarial learning for joint optimization of depth and ego-motion. *IEEE Transactions on Image Processing*, 29:4130–4142.

[Wang *et al.*2021] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. 2021. Learning to diversify for single domain generalization. *arXiv preprint arXiv:2108.11726*.

[Wong and Vong2020] Chi-Chong Wong and Chi-Man Vong. 2020. Efficient outdoor 3d point cloud semantic segmentation for critical road objects and distributed contexts. In *European Conference on Computer Vision*, pages 499–514.

[Wu *et al.*2015] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920.

[Wu *et al.*2019] Wenxuan Wu, Zhongang Qi, and Li Fuxin. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 9621–9630.

[Wu *et al*.2020] Yiming Wu, Omar El Farouk Bourahla, Xi Li, Fei Wu, Qi Tian, and Xue Zhou. 2020. Adaptive graph representation learning for video person re-identification. *IEEE Transactions on Image Processing*.

[Xiang Zhang2021] Zhouhui Lian Xiang Zhang, Xiao Sun. 2021. Bow pooling: A plug-and-play unit for feature aggregation of point clouds. *AAAI 2021*.

[Xie *et al*.2016] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pages 842–857. Springer.

[Xingjian *et al*.2015] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810.

[Xiong *et al*.2020] Xin Xiong, Haipeng Xiong, Ke Xian, Chen Zhao, Zhiguo Cao, and Xin Li. 2020. Sparse-to-dense depth completion revisited: Sampling strategy and graph construction. In *European Conference on Computer Vision (ECCV)*.

[Xu *et al*.2017] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. 2017. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of CVPR*, volume 1.

[Xu *et al*.2018a] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. 2018a. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3917–3925.

[Xu *et al*.2018b] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. 2018b. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Xu *et al*.2018c] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. 2018c. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102.

[Xu *et al*.2019] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. 2019. Depth completion from sparse lidar data with depth-normal constraints. In *The IEEE International Conference on Computer Vision (ICCV)*, October.

[Xu *et al.*2020a] Mutian Xu, Junhao Zhang, Zhipeng Zhou, Mingye Xu, Xiaojuan Qi, and Yu Qiao. 2020a. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. *arXiv preprint arXiv:2012.10921*.

[Xu *et al.*2020b] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. 2020b. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5670.

[Xu *et al.*2021a] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. 2021a. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3173–3182, June.

[Xu *et al.*2021b] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. 2021b. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14383–14392.

[Yan *et al.*2018] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Yan *et al.*2020] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. 2020. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5598.

[Yang and Soatto2018] Yanchao Yang and Stefano Soatto. 2018. Conditional prior networks for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 271–287.

[Yang *et al.*2019a] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. 2019a. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3323–3332.

[Yang *et al.*2019b] Yanchao Yang, Alex Wong, and Stefano Soatto. 2019b. Dense depth posterior (ddp) from single image and sparse range. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Yang *et al.*2020a] H. Yang, P. Chen, K. Chen, C. Lee, and Y. Chen. 2020a. Fade: Feature aggregation for depth estimation with multi-view stereo. *IEEE Transactions on Image Processing*, 29:6590–6600.

[Yang *et al.*2020b] Zetong Yang, Yanan Sun, Shu Liu, Xiaojuan Qi, and Jiaya Jia. 2020b. Cn: Channel normalization for point cloud recognition. In *European Conference on Computer Vision*, pages 600–616. Springer.

[Yang *et al.*2021] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. 2021. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378.

[Yi *et al.*2016] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. 2016. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*.

[Yi *et al.*2021] Li Yi, Boqing Gong, and Thomas Funkhouser. 2021. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15363–15373.

[Yin and Shi2018] Zhichao Yin and Jianping Shi. 2018. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2.

[Yoo *et al.*2020] Jin Hyeok Yoo, Yeocheol Kim, Ji Song Kim, and Jun Won Choi. 2020. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. *arXiv preprint arXiv:2004.12636*.

[You *et al.*2019] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. 2019. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In *International Conference on Learning Representations*.

[Yu and Koltun2015] Fisher Yu and Vladlen Koltun. 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

[Yue *et al.*2019] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. 2019. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2100–2110.

[Zhan *et al.*2018] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. 2018. Unsupervised learning of

monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349.

[Zhang and Funkhouser2018] Yinda Zhang and Thomas Funkhouser. 2018. Deep depth completion of a single rgb-d image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Zhang and Rabbat2018] Yingxue Zhang and Michael Rabbat. 2018. A graph-cnn for 3d point cloud classification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6279–6283.

[Zhang *et al.*2013] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. 2013. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827.

[Zhang *et al.*2017] Yang Zhang, Philip David, and Boqing Gong. 2017. Curriculum domain adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 2020–2030.

[Zhang *et al.*2018a] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018a. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

[Zhang *et al.*2018b] Z. Zhang, C. Xu, J. Yang, J. Gao, and Z. Cui. 2018b. Progressive hard-mining network for monocular depth estimation. *IEEE Transactions on Image Processing*, 27(8):3691–3702.

[ZHANG *et al.*2019a] Qiming ZHANG, Jing Zhang, Wei Liu, and Dacheng Tao. 2019a. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *Advances in Neural Information Processing Systems*, 32:435–445.

[Zhang *et al.*2019b] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. 2019b. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1607–1616.

[Zhang *et al.*2020] Weidong Zhang, Wei Zhang, and Yinda Zhang. 2020. Geolayout: Geometry driven room layout estimation based on depth maps of planes. In *European Conference on Computer Vision*, pages 632–648. Springer.

[Zhang *et al.*2021a] Weichen Zhang, Wen Li, and Dong Xu. 2021a. Srdan: Scale-aware and range-aware domain adaptation network for cross-dataset 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer*

Vision and Pattern Recognition (CVPR), pages 6769–6779, June.

[Zhang *et al.*2021b] Yuan-fang Zhang, Jiangbin Zheng, Wenjing Jia, Wenfeng Huang, Long Li, Nian Liu, Fei Li, and Xiangjian He. 2021b. Deep rgb-d saliency detection without depth. *IEEE Transactions on Multimedia*, pages 1–1.

[Zhao *et al.*2018] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. 2018. Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, pages 8559–8570.

[Zhao *et al.*2019a] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. 2019a. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5565–5573.

[Zhao *et al.*2019b] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. 2019b. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9788–9798.

[Zhao *et al.*2020a] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. 2020a. Point transformer. *arXiv preprint arXiv:2012.09164*.

[Zhao *et al.*2020b] Sicheng Zhao, Yezhen Wang, Bo Li, Bichen Wu, Yang Gao, Pengfei Xu, Trevor Darrell, and Kurt Keutzer. 2020b. epointda: An end-to-end simulation-to-real domain adaptation framework for lidar point cloud segmentation. *arXiv preprint arXiv:2009.03456*, 2.

[Zhao *et al.*2021] Yuyang Zhao, Zhun Zhong, Fengxiang Yang, Zhiming Luo, Yaojin Lin, Shaozi Li, and Nicu Sebe. 2021. Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6277–6286.

[Zheng *et al.*2018] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. 2018. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783.

[Zhong *et al.*2019] Yiqi Zhong, Cho-Ying Wu, Suya You, and Ulrich Neumann. 2019. Deep rgb-d canonical correlation analysis for sparse depth completion. In *Advances in Neural Information Processing Systems*, pages 5332–5342.

[Zhou *et al.*2017] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. 2017. Unsupervised learning of depth and ego-motion from video. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1851–1858.

[Zhou *et al*.2018] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*.

[Zhou *et al*.2020a] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. 2020a. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13025–13032.

[Zhou *et al*.2020b] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. 2020b. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pages 561–578. Springer.

[Zhou *et al*.2021] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2021. Domain generalization: A survey. *arXiv preprint arXiv:2103.02503*.

[Zhu *et al*.2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.

[Zhu *et al*.2019] Fengda Zhu, Linchao Zhu, and Yi Yang. 2019. Sim-real joint reinforcement transfer for 3d indoor navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

[Zou *et al*.2018] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305.