SUB CRUCE LUMEN

THE UNIVERSITY
of ADELAIDE

Efficient Scene Parsing with Imagery and Point Cloud Data

Tong HE

A thesis submitted for the degree of
DOCTOR OF PHILOSOPHY
The University of Adelaide

November 21, 2020

Contents

List of Figures

List of Tables

University of Adelaide

Abstract

Efficient Scene Parsing with Imagery and Point Cloud Data

by Tong HE

Scene parsing, aiming to provide a comprehensive understanding of the scene, is a fundamental task in the field of computer vision and remains a challenging problem for the unconstrained environment and open scenes. The results of scene parsing can generate semantic labels, location distribution, as well as for instance shape information for each element, which has shown great potential in the applications like automatic driving, video surveillance, just to name a few. Also, the efficiency of the methods determines whether it can be used on a large scale. With the easy availability of various sensors, more and more solutions resort to different data modalities according to the requirements of the applications. Imagery and point cloud are two representative data sources. How to design efficient frameworks in separate domains remains an open problem and more importantly, lays a solid foundation for multimodal fusion. In this thesis, we study the task of scene parsing under different data modalities, i.e., imagery and point cloud data, by deep neural networks.

The first part of this thesis addresses the task of efficient semantic segmentation in 2D image data. The aim is to improve the accuracy of small models while maintaining their fast inference speed without introducing extra computation overhead. To achieve this, we propose a knowledge-distillation-based method tailored for semantic segmentation to improve the performance of the small Fully Convolution Network (FCN) model by injecting compact feature representation and long-tail dependencies from the large complex FCN model (incorporated in Chapter 3).

The second part of this thesis addresses the task of semantic and instance segmentation on point cloud data. Compared to rasterized image data, point cloud data often suffer from two problems: (1) how to efficiently extract and aggregate context information. (2) how to solve the forgetting issue Lin et al., 2017c caused by extreme data imbalance. For the first problem, we study the influence of instance-aware knowledge by proposing an Instance-Aware Module by learning discriminative instance embedding features via metric learning (incorporated in Chapter 4). We also address the second problem by proposing a memory-augmented network to learn and memorize the representative prototypes that cover diverse samples universally (incorporated in Chapter 5).

Declaration of Authorship

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree. I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works. I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time. I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Tong He

Sep 2020

acknowledgements

My deepest gratitude goes first and foremost to my principle supervisor, Prof. Chunhua Shen, for his constant encouragement and instructive advice on my research. Prof. Shen acted as a role model to me not only academically but also in life. If not his dedication, motivation and generous support, this thesis and many others not contained in this thesis would undoubtedly never have achieved fruition.

I would like to thank other members in my advisory team, Dr. Changming Sun, Dr. Lingqiao Liu and Dr. Dong Gong, for their kindness help with the all kinds of difficulties in the last three years.

I would also like to thank my supportive friends and colleagues in the University of Adelaide. They are Zhi Tian, Hao Chen, Yuliang Liu, Bohan Zhuang, Xinyu Wang, Xinyu Zhang, Chee Kheng Ch'ng, Shinfang Ch'ng, Jiawang Bian, Wei Yin, Libo Sun, Hu Wang, Huangying Zhan, and Zhipeng Cai. Not only for supporting me in the hard times, but the happy hours we spend together.

Finally, I would like to take this opportunity to thanks my family, who created a conformable environment that I could focus on my research. It would have been impossible for me to go through the last three years without their support and love.

publications

This thesis contains the following work that has been published or prepared for publication:

- *Knowledge Adaptation for Efficient Semantic Segmentation.*
  Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun and Youliang Yan. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

- *Instance-Aware Embedding for Point Cloud Instance Segmentation.*
  Tong He, Yifan Liu, Xinlong Wang and Chunhua Shen. European Conference on Computer Vision (ECCV), 2020. (incorporated as Chapter

- *Learning and Memorizing Representative Prototypes for 3D Point Cloud Semantic and Instance Segmentation.*
  Tong He, Dong Gong, Zhi Tian and Chunhua Shen. European Conference on Computer Vision (ECCV), 2020. (incorporated as Chapter)

- *An End-to-End TextSpotter with Explicit Alignment and Attention.*
  Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, Changming Sun. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. (Not incorporated in the thesis)

In addition, I have co-authored the below papers:

- *FCOS: Fully Convolutional One-Stage Object Detection.*
  Zhi Tian, Chunhua Shen, Hao Chen and **Tong He**. IEEE International Conference on Computer Vision (ICCV), 2019.

- *Decoders Matter for Semantic Segmentation: Data-Dependent Decoding Enables Flexible Feature Aggregation.*
  Zhi Tian, **Tong He**, Chunhua Shen and Youliang Yan. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

- *ABCNet: Real-time Scene Text Spotting with Adaptive Bezier-Curve Network.*
  Yuliang Liu, Hao Chen, Chunhua Shen, **Tong He**, Lianwen Jin and Liangwei Wang. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

Statement of Authorship

Title of Paper	Learning and Memorizing Representative Prototypes for 3D Point Cloud Semantic and Instance Segmentation
Publication Status	□ Published ☑ Accepted for Publication □ Submitted for Publication □ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Accepted by 16TH EUROPEAN CONFERENCE ON COMPUTER VISION (ECCV2020)

Principal Author

Name of Principal Author (Candidate)	Tong He		
Contribution to the Paper	Design new methods and conduct the experiments.		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	13/07/2020

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Dong Gong		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	13/07/2020

Name of Co-Author	Zhi Tian		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	13/07/2020

Name of Co-Author	Chunhua Shen		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	13/7/2020

Title of Paper	Knowledge Adaptation for Efficient Semantic Segmentation
Publication Status	☑ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style

Publication Details	booktitle={2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)},
	title={Knowledge Adaptation for Efficient Semantic Segmentation},
	year={2019},
	pages={578-587},}

Principal Author

Name of Principal Author (Candidate)	Tong He		
Contribution to the Paper	Design new methods and conduct the experiments.		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	13/July/2020

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

vii. the candidate's stated contribution to the publication is accurate (as detailed above);

viii. permission is granted for the candidate in include the publication in the thesis; and

ix. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Chunhua Shen		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	13/07/2020

Name of Co-Author	Zhi Tian		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	13/07/2020

Name of Co-Author	Dong Gong		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	13/07/2020

Name of Co-Author	Changming Sun		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	13/07/2020

Name of Co-Author	Youliang Yan		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	14/07/2020

Title of Paper	Instance-Aware Embedding for Point Cloud Instance Segmentation
Publication Status	□ Published ☑ Accepted for Publication □ Submitted for Publication □ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Accepted by 16TH EUROPEAN CONFERENCE ON COMPUTER VISION (ECCV2020)

Principal Author

Name of Principal Author (Candidate)	Tong He		
Contribution to the Paper	Design new methods and conduct the experiments.		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	13/July/2020

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

iv. the candidate's stated contribution to the publication is accurate (as detailed above);

v. permission is granted for the candidate in include the publication in the thesis; and

vi. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Yifan Liu		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	13/July/2020

Name of Co-Author	Chunhua Shen		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	13/7/2020

Name of Co-Author	Xinlong Wang		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	2020/7/13

Name of Co-Author	Changming Sun		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	13/07/2020

Chapter 1

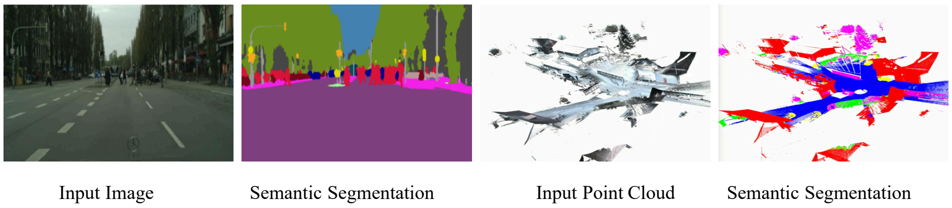
Introduction

1.1 Overview

Scene parsing is one of the most significant tasks in the field of computer vision, which aims to identify the object locations and provide pixel-level category distributions of objects, as shown in Figure 1.1. It remains challenging due to the complexity of the unconstrained environments. Besides, more and more applications have high demands on efficiency, such as automatic driving and video surveillance. With the development of hardware facilities, different sources of data have been collected and applied based on the demands of the applications. 2D image and 3D point cloud are two representative data sources. For example, the assisted driving system usually provides early warning information based on 2D images. It will detect lane lines, human, and cars from cameras with different locations and give precaution suggestions. On the other hand, if a higher degree of automation is required, the method needs to provide accurate distance measurement based on Light Detection and Ranging (LIDAR) data. In order to enhance the reliability of the whole auto-driving system, it is required to deal with both 2D imagery and 3D point cloud with high efficiency. Besides, deep learning technologies have demonstrated success in feature representation Chen et al., 2017b; Chen et al., 2017c and show superiority in both effectiveness and efficiency over traditional hand-crafted designation Lafferty, McCallum, and Pereira, 2001; Krahenbuhl and Koltun, 2011. In this thesis, we focus on the problem of efficient scene parsing using Deep Neural Networks (DNN) under different data modalities: 2D imagery and 3D point cloud.

1.2 Problem formulations

Both the 2D image and 3D point cloud data are common data formats. In this thesis, we focus on solving the tasks of (1) efficient semantic segmentation in the 2D image domain which needs to output dense semantic labels for each pixel with limited computation complexities. (2) efficient semantic and instance segmentation in the 3D



Input Image Semantic Segmentation Input Point Cloud Semantic Segmentation

FIGURE 1.1. Examples of scene parsing on imagery and point cloud data.

point cloud domain which requires predicting both per-point semantic label and the grouping information among the points.

1.2.1 Efficient Semantic Segmentation on Imagery data

Both accuracy and efficiency are of significant importance to the task of image semantic segmentation. Our task is to classify the regions into pre-defined categories. Before the emergence of the fully convolutional neural network (FCN Long, Shelhamer, and Darrell, 2015), deep-learning-based methods for semantic segmentation are often bottom-up approaches, which first segment object regions and apply convolutional neural networks for path-level classification. Compared to the above two-stage methods, FCN Long, Shelhamer, and Darrell, 2015 is more efficient and effective. It modifies the final fully connected layer as a 1×1 convolutional layer, allowing the network to have arbitrary input size and generating dense predictions in an end-to-end manner. Despite of its simplicity, many FCN-based approaches Chen et al., 2017b; Chen et al., 2017c; Lin et al., 2017a; Tian et al., 2019 have dominated the area of semantic segmentation. For example, the mean intersection-over-union (mIOU) has been boosted from 56.0 Long, Shelhamer, and Darrell, 2015 to 88.3 Chen et al., 2018 on one of the most popular datasets for semantic segmentation: Pascal VOC Everingham et al., 2014. However, the heavy computation overheads sacrifice the inference speed, which is critical for applications like automatic driving and video surveillance. In this thesis, we aim to find a better trade-off between effectiveness and efficiency: improving accuracy while maintaining the speed unchanged.

There are many different ways for efficient image segmentation, which can be roughly categorized into four classes: (1) Using powerful and efficient backbones. For example, MobileNet Howard et al., 2017 and MobileNetV2 Sandler et al., 2018 are proposed by adopting efficient convolution operations to reduce redundant computation, aiming to be deployed in mobile devices. (2) Designing new network architectures Paszke et al., 2016; Zhao et al., 2018 that can be more effectively addressing the problem of semantic segmentation. (3) Quantizing network Nagel et al., 2019; Xu et al., 2018 to use low-bit operations to deploy algorithms to different hardware platforms. (4) Applying knowledge distillation methods Hinton, Vinyals, and Dean, 2015; Liu et al., 2019a, which mainly contain two modules: a teacher module that has higher accuracy but low inference speed and a student module that is fast but with lower accuracy. The method is forcing the student model to mimic the performance of the teacher model while maintaining the inference speed. In this thesis, we proposed a novel knowledge-distillation-based method tailored for semantic segmentation that can largely reduce the gap between the teacher and the student models by addressing two issues: model discrepancies and long-range dependencies, which will be introduced in the following two subsections.

1.2.1.1 Model discrepancies

Teacher models are often selected from the ones with high-accuracy, which can be ResNet-50 He et al., 2016b, Xception-65 Chollet, 2016, or HRNet Sun et al., 2019. For a specific model, the performance can be further boosted by reducing output stride, which is controlled by an atrous step size Chollet, 2017. The student models, on the other hand, are selected by their efficiency and can be Resnet-18 He et al., 2016a, MobileNet Howard et al., 2017, or MobileNetV2 Sandler et al., 2018. The huge differences in network architectures make it hard to regularize intermediate layers of the student models. In addition, the discrepancies of the feature sizes fail to build dense correspondences between the two models. For the first problem, Hinton, Vinyals, and Dean, 2015 proposed to add a supervision signal from the logits layer, which is the last layer of the network. Compared to the one-hot representation, the soft probability distributions of the dense classifications from the teacher can introduce more knowledge to regularize student models. Huang and Wang, 2017 proposed to match the distributions of neuron selectivity patterns between the two models. However, it can be only utilized in the image recognition task and is hard to be transferred to dense segmentation. For the second problem, we propose to train an auto-encoder to compress the dense high-resolution information from the teacher network and distill the compact knowledge to the student network with a low-resolution output.

1.2.1.2 Long range dependencies

Capturing long-range dependencies is of significance in the task of semantic segmentation, as the label distributions disobey the assumptions of independent and identically distributed (i.i.d). For example, if a pixel in an image is labelled as ‘dog’, the neighbouring pixels probably share the same predictions. Many methods use graph models such as CRF Krahenbuhl and Koltun, 2011 as a post-processing step by taking pairwise connections into consideration to generate smooth predictions. To address the above issue we propose an affinity distillation module to regulate relationships among widely separated spatial regions between teacher and student. Compared to large models, small models with fewer parameters are hard to capture long-term dependencies and can be statistically brittle, due to the limited receptive field. The proposed affinity module alleviates the situation by explicitly computing pair-wise non-local interactions across the whole image.

1.2.2 Efficient Instance and Semantic Segmentation for Point Cloud

Different from rasterized images, point cloud is a set of spatial coordinates that are collected in an orderless manner. Compared to the well-studied techniques in images, deep-learning-based methods for point cloud have just started due to its unique properties Qi et al., 2017b:

- Unordered. Different from 2D pixel or 3D voxel, point cloud is a set of points without a specific order, which requires the method to be invariant to the permutations of the input points.
- Unstructured. A single point is meaningless. It can reflect useful context knowledge only when the points are combined with their neighbours. As a result, the method needs to capture local structures from the near points.
- Variant. Like CNN, which is translational invariant due to the shareable convolutional weights, the operations for point cloud need to be invariant to certain transformations. For example, the segmentation results should not be altered when a chair is placed in different locations of a room.

To solve the above problems, PointNet Qi et al., 2017b and PointNet++ Qi et al., 2017a are two pioneering works designed for directly processing orderless points. Based on this, many approaches Wang et al., 2018a; Wang et al., 2019b; Yi et al., 2018 have been proposed to solve the task of semantic and instance segmentation, which aims to predict both per-point semantic label and inter-point grouping information. However, previous methods often suffer from two problems:

- Limited contextual knowledge. PointNet++ can only extract local structure information, which ignores the instance-level representation and geometric knowledge.
- Easy to be dominated by some categories due to the data imbalance. Point cloud data are often distributed off-balance and previous methods often utilize focal loss Lin et al., 2017c, which can only bring limited improvements.

In this thesis, we build our methods on the above pipeline and address the task from two aspects: embed instance-aware geometric knowledge and learn representative prototypes for rare instances and categories.

1.2.2.1 Geometric Embedding

Instance geometric information is important for separating adjacent objects. Take a conference room, for example, most chairs have similar point distributions and textures. It is the location information that can significantly distinct and separate different chairs. In this thesis, geometric knowledge is explicitly encoded in the embedding space, which is an informative indicator to identify the points belonging to the same instance. Moreover, previous methods Wang et al., 2018a; Wang et al., 2019b contain operations that require heavy computation resources. SGPN Wang et al., 2018a predicts a large $N \times N$ similarity matrix to find points within one instance, where N denotes the number of input points. Due to the large consumption of memory and complex post-processing, it is hard to be transferred to applications that contain a large number of points. ASIS Wang et al., 2019b removes the similarity matrix and intertwines both semantic and instance segmentation tasks for mutual assistance. However, it involves large computations on neighbouring points (KNN) search. In our method, we constrain the operation within a small amount number of points, instead of searching neighbours of every input point exhaustively. Besides, we propose to learn instance-level context and realize the boundary and geometric information of each instance by introducing an Instance-Aware Module (IAM).

1.2.2.2 Prototypes Learning

Point cloud data often suffer from severe imbalance in both category-level and instance-level. For example, in an indoor scene, the proportions of the points belonging to the background (e.g., , wall) are much higher than the objects (e.g., , chairs). In S3DIS Armeni et al., 2016, the total amount of ceiling points is 50 times larger than the chair. To address this issue, previous methods use focal loss Lin et al., 2017c by setting lower loss weights to the well-classified classes. However, it only addresses the category-level imbalance and doesn't suitable to instance segmentation. In this thesis, we propose to learn and memorize the discriminative and representative prototypes covering all the samples, which is implemented as a memory-augmented network. It includes two branches for predicting point-level semantic labels and obtaining per-point embedding for instance grouping, respectively.

1.3 Main Contributions

The main contributions of this thesis include a set of algorithms for efficient scene parsing in both 2D RGB domain and 3D point cloud domain. To summarize the main contributions are listed below:

- For the semantic segmentation in 2D image, we propose a new knowledge distillation method tailored for semantic segmentation that reinterprets the output from the teacher network to a re-represented latent domain, which is easier to be learned by the compact student model. We also design an affinity distillation module to help the student network capture long-range dependencies from the teacher network. We validate the effectiveness of methods under various settings and show that our method can improve the performance of the student model by a large margin (%2) without introducing extra parameters or computations.

- For the semantic and instance segmentation in point cloud, we propose a novel Instance-Aware Module, which successfully encodes instance-dependent context information for point cloud instance segmentation. Our method explicitly encodes instance-related geometric information, which is informative and helpful to produce discriminative embedding features. The proposed framework can be trained in an end-to-end manner and shows superiority over previous methods on both efficiency and effectiveness.
- We also propose a memory-augmented network for point cloud instance segmentation (i.e., MPNet), which is trained to explicitly record the prototypes of the per-point features in a compact memory. The proposed MPNet is more effective and efficient than previous methods. The learned prototypes can consistently represent interpretable and meaningful concepts of various instances, alleviating the forgetting issue, especially for rare cases.

1.4 Thesis outline

The structure of this thesis is organized as follows.

In Chapter 2, we first review previous state-of-the-art methods for semantic segmentation in the RGB domain. Related knowledge distillation approaches are followed. Also, we provide a detailed literature review on point cloud semantic and instance segmentation.

In Chapter 3, an approach based on knowledge distillation is proposed for efficient semantic segmentation in the 2D RGB image. The proposed method reinterprets the output from the teacher network to a re-represented domain and can capture long-term dependencies from the teacher network via an affinity distillation module.

In Chapter 4, we propose a novel Instance-Aware Module for point cloud instance segmentation, which successfully encode the instance-level context and explicit geometric information.

In Chapter 5, we explore the influence of data imbalance in the task of semantic and instance segmentation on the point cloud. Moreover, we propose MPNet by utilizing a novel memory module, which is trained to explicitly record the prototypes of the per-point features in a compact memory.

In Chapter 6, the conclusion and the potential research directions are discussed.

Chapter 2

Literature Review

In this chapter, I will go through the related works in the literature. As the topics of the thesis are efficient semantic and instance segmentation in (1) 2D RGB, (2) 3D point cloud, I will first review traditional methods and deep-learning-based methods such as fully-convolutional-network for image semantic segmentation. Then, we introduce the most common techniques for fast and efficient semantic segmentation in the 2D image domain. Next, we introduce effective deep-learning-based methods on point cloud feature extraction. Finally, existing methods for semantic and instance segmentation on point cloud will be discussed.

2.1 Traditional Methods for Semantic Segmentation on 2D image

Semantic segmentation on the natural image is one of the fundamental tasks in the field of computer vision and has been studied for many years, which aims to classify every pixel of the image. It has brought many benefits to areas like medical image processing Ronneberger, Fischer, and Brox, 2015; Xu et al., 2018 and autopilot Zhao et al., 2018; Paszke et al., 2016; Treml et al., 2016.

Before the emergence of deep learning techniques, traditional approaches can be roughly categorised into two groups: (1) design representative features (2) design good classifiers.

2.1.1 Hand-crafted features

The choices of features are critical and significant for the final performance. In this part, we mainly introduce several methods for features extraction.

- Histogram of Oriented Gradients (HOG) Dalal and Triggs, 2005. HOG feature is proposed for capturing structure and shape knowledge of the input image. Different from the edge extraction, which only contains information of the edge area, HOG describes the feature with orientation and the gradient of a local region. Moreover, HOG separates the image into individual blocks and extracts features within each block, generating local and structured description. At last, a histogram will be generated for every separated block.

- Scale-invariant feature transform (SIFT) Lowe, 2004. SIFT is one of the most important hand-crafted features, which aims to extract scale and rotation invariant representations. It mainly contains four parts: (1) Find and search all potential rotation and scale-invariant points by using DOG(Difference of Gaussian kernel function) in different octaves of the image in the Gaussian Pyramid. (2) Key point localization is achieved by comparing local regions and find local extrema. (3) Based on the local gradient value and orientation, an orientation histogram covering 360 degrees is created and the peak with the highest value is chosen to be the principal direction. (4) Computing locale gradients as the keypoint descriptor.

- Local Binary Pattern (LBP) Ojala, Pietikainen, and Harwood, 1996. LBP is one of the most important texture descriptors. For each central point, neighbouring points are compared to generate 0-1 values. If the value of the neighbour is greater than the central value, the descriptor will output 0, otherwise 1. By concatenating all these 0-1 values in a clockwise manner, we can get a local descriptor. A histogram is calculated based on the statistics across the whole image.

2.1.2 Classifiers

Given the features of each point, the classifiers are to predict the categories of these points.

- Random Forest Breiman, 2001. The random forest classifier contains plenty of decision trees. Each decision tree only considers a random subset of the features and it only has access to a random subset of the training samples. The meanings of ‘Random’ have two aspects: feature-level and data-level. These randomnesses can significantly boost the robustness and increase diversity during the training process. Given a well-trained model, random forest averages all predictions from individual decision trees. Kang and Nguyen, 2019 proposed a random forest framework that successfully encodes shapes and sparsities knowledge. Shotton, Johnson, and Cipolla, 2008 investigated not only local features but also context-rich features in the random forest and found that multiple features enrich the capability of the classifier and achieved higher accuracy.

- Conditional Random Field Lafferty, McCallum, and Pereira, 2001. CRF is a discriminative approach and is widely applied especially when the inputs are dependent. Image semantic segmentation is one of the cases: neighbouring pixels are more likely to have identical labels. In CRF, there usually contains two parts: unary term and pairwise term (some methods contain high-order term). The unary term represents the potential between the output label and the input feature. The pairwise term denotes the relationship between neighbouring points, which is critical to smoothen the output predictions. Krahenbuhl and Koltun, 2011 came up with a highly efficient approximate inference method for the fully connected CRF and prove the dense connection can largely enhance the accuracy of semantic segmentation. Farabet et al., 2013 first segments the image into superpixels and a CRF is defined over a set of superpixels.

2.2 Convolutional Neural Networks

blueCompared to traditional methods that utilize hand-crafted features like SIFT Lowe, 2004 and HOG Dalal and Triggs, 2005 to capture low-level features such as gradients and texture information, deep neural networks are more desirable scene parsing for its capability of grasping high-level semantic context. In 2006, Geoffrey Hinton used greedy learning, combined with other learning methods, to learn discriminative network (Deep Belief Network Hinton, 2010), which is regarded as the beginning era of the deep learning. By 2011, with the rapid development of the GPU, it is possible to train the deep network in an end-to-end manner, instead of layer-by-layer. AlexNet Krizhevsky, Sutskever, and Hinton, 2012 attracted a lot of attention from both industry and academia, due to its outstanding performance on the ImageNet Deng et al., 2009. It surpassed the traditional hand-crafted based methods by a large margin. Nowadays, deep neural networks have become one of the most effective methods for feature learning. Convolutional Neural Networks, also known as CNN, is a category of deep neural networks and especially useful for image processing. Due to its weights sharing mechanism, CNN maintains several important properties, e.g., translation invariant. Inspired by the response of a neuron in the visual cortex to a specific stimulus, CNN is stacked with multiple layers. In the beginning, shallow layers can only process low-level information and have a small receptive field. Each neuron can extract one specific knowledge, for example, color and boundary. As the depth increases, the receptive field becomes large and the network is able to extract highly abstracted information according to the specific task and loss function. In this part, we list several types of building blocks used in the deep CNN.

- Convolutional layer. The convolutional layer is an essential part of feature extraction for CNN. It defines a local region (receptive field) and is fully connected to learnable and shareable weights (also denoted as the kernel) in a sliding window manner. To generate representative features with various information, different kernels are learned to convolve the input signal. The results are passed to the next layer. Due to its shareable property, the convolutional layer can output translation-invariant features.

- Pooling layer. It is designed to reduce the dimensions of the data and enlarges the receptive field by grouping local features. According to different operations, the pooling layer can be max-pooling or average-pooling.
- Fully connected layer. It connects each input neuron to all the neurons of the output layer, which is utilized to capture global information.

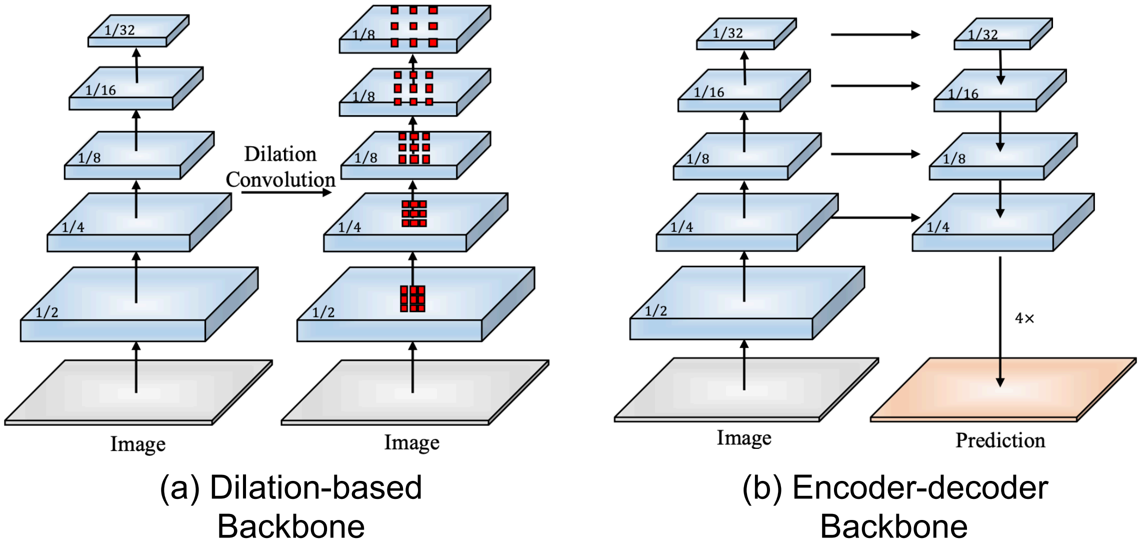
2.3 Deep-Learning-Based Semantic Segmentation on 2D Image

Semantic segmentation is a fundamental and challenging task in computer vision, aiming to identify pixel-wise classification. Due to its numerous applications, such as autonomous driving and medical image diagnostics, more and more methods Tian et al., 2019; Lin et al., 2018; Zhang et al., 2018b; Chen et al., 2018; Sandler et al., 2018; Chen et al., 2017b; Lin et al., 2017a; Zhao et al., 2017 have been proposed. One of the most important deep-learning-based work is Fully Convolutional Network (FCN) Long, Shelhamer, and Darrell, 2015.

2.3.1 Fully Convolutional Network

Different from image recognition, which needs to know what is in the input image, semantic segmentation, on the other hand, requires to identify not only what is in the image, but also where. Previous methods that use CNN for semantic segmentation often find enclosing region or object, e.g., superpixel Li and Chen, 2015; Veksler, Boykov, and Mehrani, 2010; Bergh et al., 2012, and classify the local region with one unique label Moore et al., 2009; Vu and Manjunath, 2008; Chen et al., 2013; Hariharan et al., 2011. The drawbacks are obvious: (1) patchwise training Lin et al., 2017a: the models need to have identical input sizes, (2) complex postprocessing Chen et al., 2017b: segmented regions or objects need to be interlaced for densely structured output. (3) ensembles Zhang et al., 2018b: a large number of heuristic hyper-parameters and clues need to be fused. Fully convolutional network (FCN) Long, Shelhamer, and Darrell, 2015 shows that it can be trained end-to-end with arbitrary input size for pixel-wise dense prediction on the task of semantic segmentation. Not only it stresses the above issues it also exceeds the state-of-the-art by a large margin with a much higher computation efficiency. FCN utilizes locally connected layers such as convolution, pooling, and activation layers. Meanwhile, it transfers the knowledge extracted from image-level classification to dense pixel-level prediction via replacing fully connected layers with 1×1 convolution layers. Building upon FCN, many network architectures have been proposed, which are summarized below:

- Dilation-based Chen et al., 2017c; Chen et al., 2017b; Zhao et al., 2017: As can be seen in Figure 2.1(a), the network replaces down-sampling operations with



(a) Dilation-based Backbone

(b) Encoder-decoder Backbone

FIGURE 2.1. Illustration of different backbone architectures, which can be roughly categorized into two groups: dilation and encode-decoder (Figure is originated from Yu et al., 2020).

stride-convolutions. In order to keep large a receptive field, dilated convolutions are applied. The advantage of this kind of architecture is to maintain the fine-grained details. The drawback is heavy computation complexity and memory footprint.

- Encoder-decoder network Chen et al., 2018; Tian et al., 2019; Lin et al., 2017a; Lin et al., 2018: The network architecture is shown in Figure 2.1(b). It mainly contains three parts: (1) down-sampling encoder to extract high-level representation, (2) top-down decoder to get high-resolution outputs, (3) lateral connections to incorporate low-level knowledge in the decoding process.

2.3.2 Accuracy Oriented Methods

The goal of this line is to significantly boost the accuracy of both pixel-level and category-level classification. Chen *et al.*proposed DeepLab-CRF Chen et al., 2017b, which applies a dense CRF as a post-processing step to refine the segmentation results and capture better boundaries on the top of CNN. This method is extended by CRF-RNN Zheng et al., 2015, in which CRF is implemented as an inner layer embedded in a network for end-to-end learning. Lin et al., 2017a proposed a multi-path RefineNet to output high-resolution results, by exploiting long-range residual modules to capture all information when down sample operations are performed. Recently, Chen *et al.*proposed DeepLabV3 Chen et al., 2017c and DeepLabV3+ Chen et al., 2018 that applied atrous convolution operation to effectively enlarge the reception field and capture rich semantic information. These methods improve the performance by outputting high-resolution feature maps to alleviate the loss of details and boundaries. However, considering the limit of GPU resources and computational efficiency,

$\frac{1}{8}$ or even denser $\frac{1}{4}$ size of inputs resolution are not realistic in the current model design. For example, when ResNet-101 He et al., 2016a uses the atrous convolution to output 16 times smaller feature maps, much more computation and storage will be used in the last 9 convolution layers. Even worse, 26 residual blocks (78 layers!) will be affected if the output features that are 8 times smaller than the input are desired.

2.3.3 Efficiency Oriented Methods

Recently, more and more researches Liu et al., 2019a; Zhao et al., 2018; Paszke et al., 2016; Treml et al., 2016; Sandler et al., 2018; Yu et al., 2018 focus on the efficiency of the model, for the increasing demand for real-time applications. The common ways of enhancing efficiency are summarized as below:

- **Depthwise Separable Convolution.** This technique has been widely utilized in light-weight models, such as X-ception Chollet, 2016 and MobileNet Howard et al., 2017; Sandler et al., 2018. Different from normal convolution, the depthwise separable convolution factorizes the operation into two separate convolutions: a depth-wise convolution and a 1×1 point-wise convolution.

- **New Architecture Design.** These methods often come up with new architectures and find a better trade-off between accuracy and efficiency. ICNet Zhao et al., 2018 cascades image as inputs to extract both abstracted and appearance features. BiSegNet Yu et al., 2018 encodes both low-level spatial details and high-level semantic knowledge in two separate branches.

- **Network Quantization.** Quantizing the network Xu et al., 2018; Nagel et al., 2019; Tang et al., 2018 can reduce the consumption of the computation resources and can be deployed in mobile devices with fast inference time. With the technique of quantization, a high precision number, e.g., float64 or float32, can be represented with low-bit, e.g., int8, int2, while maintaining high accuracy.

- **Knowledge Distillation.** This kind of method often involves two networks: one is the teacher network and the other is student network. The teacher network can be a large and complex model, which usually has more parameters and higher accuracy. Knowledge distillation is forcing a small student to mimic the teacher model while maintaining the speed unchanged.

2.4 Semantic and Instance Segmentation on 3D Point Cloud

3D point cloud processing technologies are evolving rapidly, as more and more scanning hardware is becoming readily accessible. For example, these techniques can be seen in more and more scenes such as medical imaging, 3D reconstruction, and robotics. In addition, 3D point cloud contains more precise spatial information, which is critical to the other domain. With the increasing accessibility of 3D point cloud data, efficient and effective processing methods are required.

2.4.1 Traditional methods for Segmentation in 3D Point Cloud

Segmentation on point cloud is the process of grouping and classifying points into multiple homogeneous regions, which can be either semantic or instance regions. Before the era of deep learning, most methods are built upon statistics and can be roughly categorized into five groups: edge-based, region-based, attribute-based, model-based, and graph-based.

- Edge-based. Edge often contains significant clues for segmenting objects, which inspires many methods to find boundaries between the semantic regions. Bhanu et al., 1986 detects lines and boundaries in 3D space by calculating the gradients and fitting straight lines and curves. Sappa and Devy, 2001 extracts contours for fast region segmentation. Although efficient, edge-based methods often suffer from the uneven density problem, resulting in low accuracy.

- Region-based. Region-based methods often search neighbouring points and group regions that share similar properties and separate regions that have dissimilarities. Besl and Jain, 1988 proposed a seed-based method, which first finds out seed points according to the curvature and applies region growing method to merge neighbouring points based on the feature similarity. Tovari and Pfeifer, 2005 came up with a method to process airborne laser scanning points by expanding the region based on the surface normal vector.

- Attributes-based. This kind of method often contains two steps: the first step is to compute attributes of each point. Then clustering methods are applied by taking both different attributes and spatial information into consideration. Vosselman and Dijkman, 2001 applied 3D Hough transform to extract planar surfaces. The author claimed that each point is treated as a plane in the 3D attribute space.

- Model-based. Model-based methods rely on pre-defined primitive shapes. Schnabel, Wahl, and Klein, 2007 presented a method to automatically detect shapes in unorganized point clouds. Each detected shape serves as a proxy for a set of corresponding points. Although it is robust to outliers to some extend, the method is not generalized well to the unseen cases.

- Graph-based. Graph-based methods are growing more and more popular for their efficiency and effectiveness, which treat each point as vertex and the connection with its neighbours as edges. Golovinskiy and Funkhouser, 2009 proposed to build a graph by using k-nearest neighbouring (KNN). The method also brought a function to encourage smooth segmentation by penalizing the regions that are weakly connected to the background.

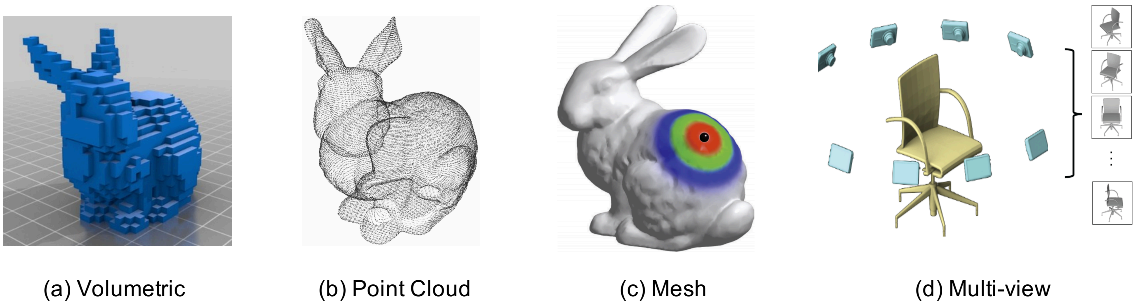


FIGURE 2.2. Illustration of different 3D data representations.

2.4.2 Deep Learning for 3D feature extraction

Although deep learning has pushed forward the progress in 2D image understanding, the counterpart of 3D scene analysis lags far behind and just started to be discussed in the last few years for its increasing potential applications. In this section, we briefly review some existing approaches that are related to this field.

As shown in Figure 2.2, deep-learning-based methods for 3D feature extraction can be roughly categorized into four classes: voxel-based, multi-view-based, mesh, and point-based. Voxel-based methods Maturana and Scherer, 2015; Wu et al., 2015; Riegler, Ulusoy, and Geiger, 2016; Graham, Engelcke, and Maaten, 2018 utilized 3D convolution neural networks for feature extraction on voxelized spatial grids. However, these methods are significantly influenced by the density of the points. Meanwhile, it is highly constrained by the huge memory occupation and lower running speed because a large proportion of computation is wasted on vacant voxels. Many approaches have been designed to address the problem Riegler, Ulusoy, and Geiger, 2016; Graham, Engelcke, and Maaten, 2018. Octree Riegler, Ulusoy, and Geiger, 2016 tries to modify the convolution operation by generating average hidden states in empty spaces. SparseConv Graham, Engelcke, and Maaten, 2018 is proposed to process spatially sparse data more efficiently by encoding with a Hash Table to avoid unnecessary memory usage in vacant space.

The second category is multi-view-based methods Hou, Dai, and Nießner, 2019; Qi et al., 2016; Su et al., 2015, which first project 3D shapes or point clouds into 2D images and utilize conventional 2D CNN for feature extraction. Hou *et al.*proposed 3D-SIS Hou, Dai, and Nießner, 2019 by leveraging both RGB 2D input and 3D geometrical information. 2D features are then back-projected into 3D grids.

The 3D mesh is a collection of vertices, edges, and faces that defines the surface of 3D shapes. The great success of graph representation via deep learning makes mesh popular to be used for encoding 3D structures. Ranjan et al., 2018 introduces a versatile model that learns a non-linear representation of a face using spectral convolutions on a mesh surface. Defferrard, Bresson, and Vandergheynst, 2016 presents a formulation of CNNs in the context of spectral graph theory.

Unlike the above methods, directly extracting features on point clouds is more efficient and straightforward. PointNet Qi et al., 2017b is the pioneering work that directly learns a spatial encoding of each point. A symmetrical function is used to process disordered point sets. However, PointNet failed to capture local structure knowledge which has been proven to be of great significance to obtain representative features. Many approaches Qi et al., 2017a; Thomas et al., 2019; Li et al., 2018; Wang et al., 2019a; Li et al., 2019 have been proposed to address the problem. Qi *et al.*proposed PointNet++ Qi et al., 2017a, which applied PointNet recursively on a nested partitioning of the input point clouds. Thomas *et al.*came up with KPConv Thomas et al., 2019 by designing a continuous weight space through interpolating with several kernel points.

2.4.3 Instance Segmentation on Point Cloud

Although the task of instance segmentation on 2D images has made huge progress since Mask-RCNN He et al., 2017a was proposed, its 3D point cloud counterpart lags far behind. SGPN Wang et al., 2018a is the first deep-learning-based method developed in this field. It tried to generate point cloud groups by predicting three objectives: the similarity matrix, the confidence map, and the semantic prediction map. However, due to the pair-wise term, the method occupies a large amount of GPU memories and suffers from slow running speed and small batch size for training. On the other hand, generating instance groups from three matrices requires many hyper-parameters, making it less stable for different scenarios.

Wang *et al.*proposed ASIS Wang et al., 2019b to address the problem by removing the pairwise prediction and introducing a discriminative loss for instance embedding. The loss pulls the embeddings of the same instance towards the cluster center and pushes the cluster centers away from each other. However, the method fails to utilize the geometrical information and is unaware of the spatial distribution of the instances. In addition, extensive use of neighbour searching makes it time-consuming. GSPN Yi et al., 2018, generates shape proposals using a generative model for instance segmentation. Due to its emphasis on geometric understanding for object proposal, it achieved satisfying performance on both indoor dataset and part instances dataset. However, the method requires a large amount of memory and needs a two-step training procedure, making it ineffective with limited resources.

Chapter 3

Knowledge Adaptation for Efficient Semantic Segmentation

3.1 Introduction

Semantic segmentation is a crucial and challenging task for image understanding Chen et al., 2017b; Chen et al., 2018; Chen et al., 2017c; Zhao et al., 2017; Lin et al., 2017a; Zhao et al., 2017; Liu et al., 2015; Long, Shelhamer, and Darrell, 2015; Tian et al., 2019. It aims to predict a dense labeling map for the input image, which assigns each pixel a unique category label. Apparently, efficient semantic segmentation is more attractive and has drawn a lot of attention for its potential applications that require fast inference speed, for example, autonomous driving and video surveillance. Although deep fully convolution network (FCN) based methods Chen et al., 2017b; Chen et al., 2018 have achieved remarkable results in semantic segmentation, extensive methods have been investigated to improve the performance by introducing sophisticated models with a large number of parameters. To preserve the detailed semantic structures in the dense estimation, many state-of-the-art FCN based methods Zhang et al., 2018b; Chen et al., 2017b; Chen et al., 2017c; Chen et al., 2018 maintain a series of high-resolution feature maps by applying a small overall stride Chen et al., 2018, which causes heavy computations and limits the practicability of semantic segmentation. For example, Chen proposed DeepLabV3+ Chen et al., 2018 and achieved state-of-the-art performance on many open benchmarks such as Pascal VOC Everingham et al., 2014 and Cityscapes Cordts et al., 2016. However, this is obtained back-boned on a large model: Xception-65 Chollet, 2016, which contains more than 41.0M parameters and 1857G FLOPS and runs at 1.3 FPS on a single 1080Ti GPU card if the output stride is set to 16. Even worse, 6110G FLOPS will be needed and running at 0.4 FPS with an output stride of 8. Similar situations can be found in lightweight models (see Figure 3.1).

One instant way to address this limitation is to reduce the resolution of a series of feature maps via sub-sampling operations like pooling and convolution striding.

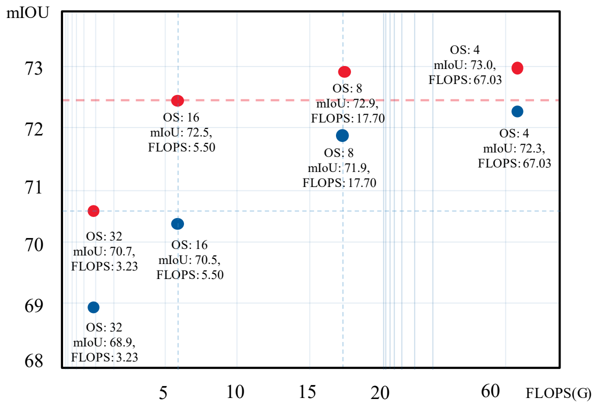


Figure 3.1. The relation between FLOPS and performance. Blue dots are the performance of the student model (MobilNetV2 Sandler et al., 2018), while red dots are the performance of the student model with our proposed knowledge distillation method. The performance is trained on the PASCAL VOC Everingham et al., 2014 *trainaug* set and tested on the *val* set. **OS** means output stride. With the help of our proposed method, the student model with low resolution (16s) of the feature maps outperforms the model with large feature maps (4s) by using only 8% FLOPS.

However, unsatisfactory estimation accuracy will be incurred for the huge loss of detailed information.

How to solve the dilemma and find a better trade-off between the accuracy and efficiency have been discussed for a long time. Knowledge distillation (KD), introduced by Hinton Hinton, Vinyals, and Dean, 2015 to the field of deep learning, has attracted much attention for its simplicity and efficiency. The knowledge in Hinton, Vinyals, and Dean, 2015 is defined as soft label output from a large teacher network, which contains more useful information, such as intra-class similarity, than one-hot encoding. The student network is supervised by both soft labels and hard one-hot labels simultaneously, reconciled by a hyper-parameter to adjust the loss weight. Following KD Hinton, Vinyals, and Dean, 2015, many methods Romero et al., 2015; Huang and Wang, 2017; Zagoruyko and Komodakis, 2017; Yim et al., 2017; Kim, Park, and Kwak, 2018 are proposed to regulate the intermediate features. However, these methods are mainly designed for the image-level classification task without considering the spatial context structures. Moreover, in the semantic segmentation task, the feature maps from the teacher and student usually have inconsistent context and mismatched features. Thus these methods are improper to be used for semantic segmentation directly.

In this chapter, we propose a new knowledge distillation method tailored for semantic segmentation. We aim to learn efficient compact FCNs (i.e., student) by distilling the rich and powerful knowledge from the accurate but heavy teachers with larger overall stride. Firstly, unlike other methods that force the student to mimic the output values from the teacher network directly, we rephrase the rich semantic knowledge from the teacher into a compact representation. The student is trained to match this implicit information. The knowledge translating is achieved relying on an auto-encoder pre-trained on the teacher features in an unsupervised manner, which reformulates the knowledge from the teacher to a compact format that is easier to be comprehended by the student network.

The behind intuitions are quite straightforward: Directly transferring the outputs from the teacher overlooks the inherent differences of network architecture between two models. Compact representation, on the other hand, can help the student focus on the most critical part by removing redundancy knowledge and noisy information. Furthermore, we also propose an affinity distillation module to regulate relationships among widely separated spatial regions between teacher and student. Compared to large models, small models with fewer parameters are hard to capture long-term dependencies and can be statistically brittle, due to the limited receptive field. The proposed affinity module alleviates the situation by explicitly computing pair-wise non-local interactions across the whole image.

We validate the effectiveness of methods under various settings. (1) Our method improves the performance of the student model by a large margin (%2) without introducing extra parameters or computations. (2) Our model achieves at least comparable or even better results with only 8% FLOPS compared to the model with large resolution outputs.

3.2 Background

3.2.1 MobileNet V1

MobileNet Howard et al., 2017 is built for deploying deep models on mobile devices. The efficiency comes from the architecture blocks that utilize depth-wise separable convolution. Take a feature map with M channels as input, and the output is a feature map with N channels. Traditional convolution operation will have flops calculated by:

$$\begin{aligned} Param_{conv} &= D_k \times D_k \times M \times N \\ flops_{conv} &= D_k \times D_k \times M \times N \times D_w \times D_h \end{aligned} \tag{3.1}$$

where D_k denotes the kernel size, $D_w and D_h$ are the sizes of input feature map. The convolution is fully connected within each sliding window.

Depthwise separable convolution, on the other hand, contains two parts: depthwise convolution and pointwise convolution. Depthwise convolution operates convolution in each channel, meaning that each channel has an independent kernel and the output channel is the same as the input. The flops and the number of parameters of depthwise convolution are calculated by:

$$
\begin{aligned}
Param_{dw} &= D_k \times D_k \times M \\
flops_{dw} &= D_k \times D_k \times M \times D_w \times D_h
\end{aligned}
\tag{3.2}
$$

which are $\frac{1}{N}$ of Equation. 3.1. The pointwise convolution then applies 1 × 1 convolution to create a linear combination of the features from the output of depthwise convolution:

$$
\begin{aligned}
Param_{pw} &= M \times N \\
flops_{pw} &= D_w \times D_h \times M \times N
\end{aligned}
\tag{3.3}
$$

As a result, the reduction in computation cost is:

$$
\begin{aligned}
&\frac{D_k \times D_k \times M \times D_w \times D_h + D_w \times D_h \times M \times N}{D_k \times D_k \times M \times N \times D_w \times D_h} \\
=&\frac{1}{N} + \frac{1}{D_k^2}
\end{aligned}
\tag{3.4}
$$

Take 3 × 3 convolution kernel, for example, the computation cost will have between 8 to 9 times less computation compared with standard convolution operation, which reduces the cost by a large margin while maintaining comparable accuracy.

3.2.2 MobileNetV2

Built upon MobileNetV1, MobileNetV2 Sandler et al., 2018 introduced residual structure, where bottleneck layers are connected. When using non-linear function ReLU, loss of information is inevitable. In order to control the situation, The intermediate expansion layer is proposed to serve as a source of non-linearity. Moreover, the author experimentally found that it is critical to remove non-linearities in the narrow layers to keep powerful representation.

3.2.3 Self-attention Mechanism

The attention mechanism Bahdanau, Cho, and Bengio, 2014 has been widely applied in natural language processing Vaswani et al., 2017; Devlin et al., 2018; Yang et al., 2019c and image understanding Liu et al., 2019a; Fu et al., 2019; Zhang et al., 2018a. The intuition behind it is quite simple: when humans observe an image, it is not likely to browse the whole image end-to-end, instead, it will focus on some specific objects according to the needs. Attention will be selectively paid to that particular part of the scene according to the content of the output. For example, if you are asked to count the number of people in a picture, you will pay attention to the areas with people and ignore the background or other irrelevant things. The attention mechanism in deep learning is considered to mimic the function of the brain in a simplified manner. Dual attention Fu et al., 2019 is proposed to utilize two types of attention modules: position attention and channel attention.Hu et al., 2019 aggregates multi-scale highlighted context information to provide better guidance.

3.2.4 Dilated Convolution

Dilated convolution Yu and Koltun, 2016 is specifically designed for dense prediction, such as semantic segmentation and depth prediction. It aggregates multi-scale contextual information by introducing a new parameter: dilated rate. It also supports exponential expansion of the receptive field without loss of resolution.

3.2.5 Knowledge Distillation

The research of Hinton, Vinyals, and Dean, 2015 is the pioneering work that exploits knowledge distillation for the image classification task. The knowledge is defined as the soft output from the teacher network which provides much more useful information, such as intra-class similarity and inner-class diversity, than one-hot encoding. The soften degree is controlled by a hyper-parameter temperature, T. The student network is supervised by two losses reconciled by a loss weight. Despite its effectiveness on image classification, there are some limitations for its application in the semantic segmentation task: (1) Authors in Romero et al., 2015 tried to force the student to mimic the output distribution of a teacher network in the decision space, where useful context information is cascaded. (2) The knowledge required for image-level classification is similar between the two models because both models capture global information. But the decision space may different for semantic segmentation, as two models have different abilities to capture context and long-range dependencies, depending on the network architecture. (3) The hyper-parameter temperature is sensitive to tasks and is hard to tune, especially on large benchmarks.

Following Hinton, Vinyals, and Dean, 2015, many other methods are proposed for knowledge distillation. Romero *et al.*proposed FitNet Romero et al., 2015, for the purpose of learning intermediate representation by directly aligning feature maps, which may not be a good choice for overlooking the inherent differences between two models, such as spatial resolution, channel numbers, and network architecture. Meanwhile, significantly different abstracting capability between the two models may make this situation severe. Attention transfer Zagoruyko and Komodakis, 2017 (AT) aims to mimic the attention map between student and teacher models. It is based on the assumption that the summation of feature maps across channel dimensions can represent attention distribution in the image classification task. However, this assumption may not suit pixel-wise segmentation task, because different channels are representing activations of different classes and simply summing up across channels will end up with mixed attention maps. Chen et al., 2017a proposed a novel detection system by utilizing knowledge distillation. A bounded loss is designed for the teacher

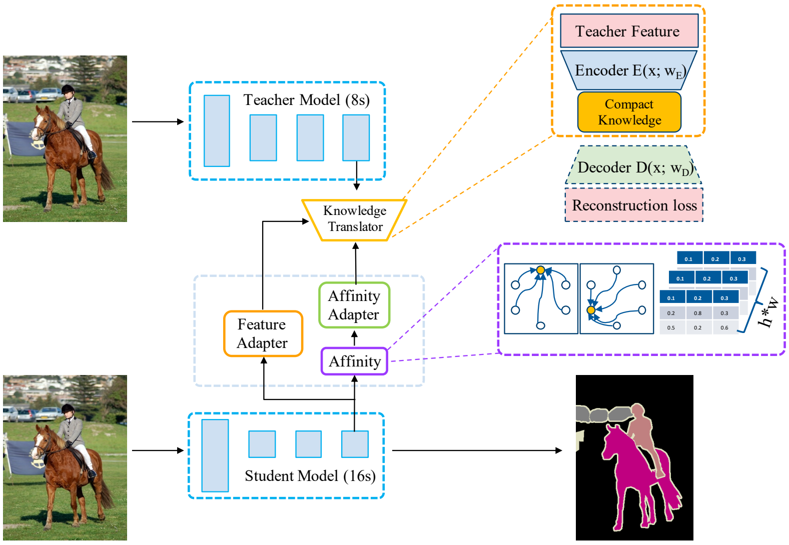


FIGURE 3.2. The detailed framework of our knowledge adaptation method tailored for semantic segmentation. The teacher network is frozen and outputs high-resolution feature maps. The student network outputs the small size of feature maps and is updated by both ground truth labels and the knowledge defined in a compressed space and affinity information.

model to encode better generalization ability and intermediate representation for the regression task.

3.3 Proposed Approach

With the help of the atrous convolution operation, a network with a small overall output stride often outperforms the one with a large overall output stride for capturing detailed information, as shown in Figure 3.1. Inspired by this, we propose a novel knowledge distillation method tailored for semantic segmentation. As shown in Figure 3.2, the whole framework involves two separate networks: one is the teacher network, which outputs features with larger resolution (e.g., 8s overall stride), the other is the student network, which has smaller outputs (e.g., 16s overall stride) for fast inference. The knowledge is defined as two parts: (1) The first part is designed for translating the knowledge from the teacher network to a compressed space that is more informative. The translator is achieved by training an auto-encoder to compress the knowledge to a compact format that is easier to be learned by the student network, otherwise much harder due to the inherent structural differences. (2) The second part is designed to capture long-range dependencies from the teacher network, which is difficult to be learned for small models due to the limited receptive field and abstracting capability. More details are provided in the following sections.

3.3.1 Knowledge Translation and Adaptation

Benefiting from the atrous convolution operation, FCNs can maintain detailed information while capturing a large receptive field. Although the performance is improved, large computation overheads are introduced and will grow exponentially as the output stride becomes smaller, as shown in Figure 3.1. In this section, we propose to utilize a large teacher model with high feature resolution to teach a lightweight student network with low feature resolution.

An auto-encoder, which tries to reconstruct the input, is capable of capturing useful and important information. We train an auto-encoder for mining the implicit structure information and translating the knowledge to a format that is easier to be comprehended and replicated by the student network. Compared with low-level and middle-level features, which are either general across different models or challenging to be transferred due to the inherent network differences, high-level features are more suitable for our situation. In our method, the auto-encoder takes the last convolution features from the teacher model as input and is composed of three strided convolution layers and symmetrical deconvolution layers. Suppose that we have two networks, namely, the student network S and the teacher network T and the last feature maps of the two models are Φ_s and Φ_t, respectively. The training process is completed by using a reconstruction loss in Eq. (3.5),

$$L_{\text{ae}} = \|\Phi_t - D(E(\Phi_t))\|^2 + \alpha\|E(\Phi_t)\|_1 \tag{3.5}$$

where $E(\cdot)$ and $D(\cdot)$ represent encoder and decoder, respectively. One common issue in training the auto-encoder is that the model may learn little more than an identity function, implying the extracted structure knowledge is more likely to share the same pattern with the input features. As the $l1$ norm is known to produce sparse representations, a similar strategy Ainde and Zurada, 2018 is utilized by regularizing both weights and the re-represented space. The weight for regularization loss α is set to 10^{-7} for all experiments. In order to solve the problem of feature mismatching and decrease the effect of the inherent network difference of two models, the feature adapter is utilized by adding a convolution layer.

Relying on the pre-trained auto-encoder, the transferring process is formalized in Eq. (3.6),

$$L_{\text{adapt}} = \frac{1}{|I|}\sum_{j\in I}\left\| \frac{C_f(\Phi_s^j)}{\|C_f(\Phi_s^j)\|_q} - \frac{E(\Phi_t^j)}{\|E(\Phi_t^j)\|_q} \right\|_p \tag{3.6}$$

where E represents the pre-trained auto-encoder. I denotes the indices of all student-teacher pairs in all positions. C_f is the adapter for the student features, which uses a 3 × 3 kernel with stride of 1, padding of 1, BN layer and ReLU activation function. The

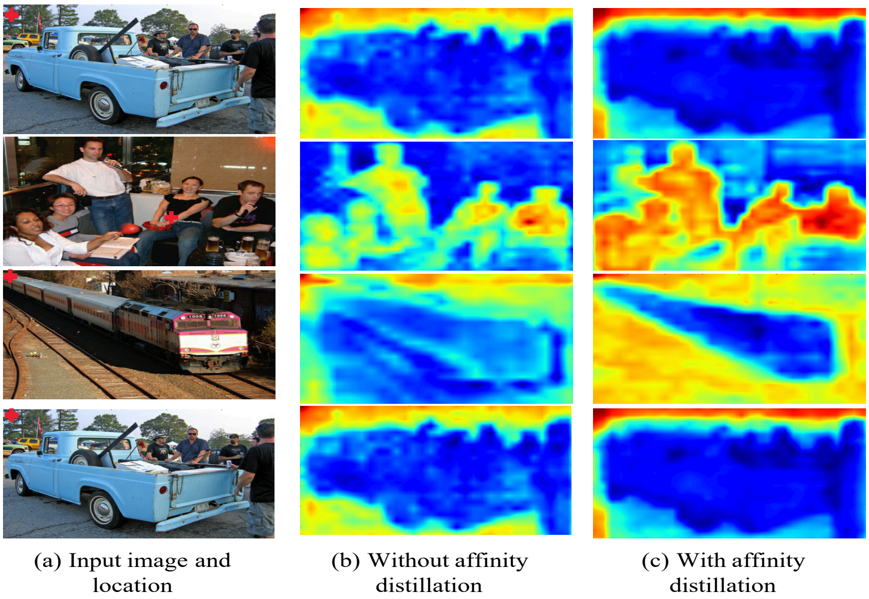

(a) Input image and location
(b) Without affinity distillation
(c) With affinity distillation

FIGURE 3.3. The effect of the affinity distillation module (better visualized in color). (a) input image and randomly selected points with red '+'. (b) affinity map of the given point of student model without affinity distillation module. (c) affinity map enhanced by our affinity distillation module.

features are normalized before matching. p and q are different normalization types to normalize the knowledge for stability.

3.3.2 Affinity Distillation Module

Capturing long-range dependency is important and can benefit the task of semantic segmentation. As described in Wang et al., 2018b, it is easier to be captured by deep-stacked convolution layers with a large receptive field. Small networks, on the other hand, have limited ability to learn this knowledge due to the deficient abstracting capability. We propose a new affinity distillation module by explicitly extracting long-range, non-local dependencies from the big teacher model. Details are described below.

In the case of studying, sometimes it would be more efficient to learn new knowledge by providing extra difference or affinity information. Inspired by this, we define the affinity in the network by directly computing interactions between any two positions, regardless of their spatial distances. As a result, the pixels with different labels will generate a low response and a high response for the pixels with the same labels. Let feature maps of the last layer to be Φ with size of $h \times w \times c$, where h, w and c represent the number of height, width, and channels, respectively. The affinity matrix $A \in \mathbb{R}^{m*m}$ can be calculated by Eq. (3.7), where m equals to $h{\times}w$, i and j are the indexes for vectorized Φ:

$$A(\Phi)_{i,j} = \frac{1}{h \times w} \cdot \frac{\Phi_i}{\|\Phi_i\|_2} \cdot \frac{\Phi_j}{\|\Phi_j\|_2}, \tag{3.7}$$

where $A(\Phi)$ denotes the affinity matrix corresponding to the feature map Φ with spectral normalization.

We use ℓ_2 loss to match affinity matrix between teacher and student models, which is defined as Eq. (3.8)

$$L_{\text{aff}} = \sum_i \|(A_s(C_a(\Phi_s))) - A_t(E(\Phi_t))\|_2 \tag{3.8}$$

where $E(\Phi_t)$ is the translated knowledge from teacher, C_a is the adapter for student affinity and i is the location index of the feature map.

To visualize the effect of the affinity distillation module, some examples are presented in Figure 3.3. Given one random selected point, the response between this point and all other separated spatial regions are shown in (b) and (c). As can be seen, the student network fails to capture this long-range dependency and only local similar patterns are highlighted. With the help of our method, long-range or even global information are captured and can be used to make a more robust decision.

3.3.3 Training Process

Our proposed method involves a teacher net and a student net. As presented in Algorithm 1, the teacher net is pre-trained and the parameters are kept frozen during the training the transferring process. The student net is supervised by three losses: cross-entropy loss L_{ce} with ground truth label, adaptation loss L_{adapt} in Eq. (3.6), and affinity transferring loss L_{aff} in Eq. (3.8). Three losses are reconciled by the loss weights of β and γ, which are set to 50 and 1 respectively in all our experiments. W_E, W_D and W_S denote the parameters for the encoder, decoder and student model, respectively.

Algorithm 1: Training Process of Our Method

```
Require:    Already trained teacher network $T$.
  STAGE 1: Training auto-encoder for teacher network.
  INPUTS: Knowledge from teacher network $\Phi_{t;W_t}$
          $W_E = \arg\min_{W_E, W_D} L_{\text{ae}}\left(\Phi_{t;W_t}\right)$
  STAGE 2: Training student network.
  INPUTS: Encoder Parameters $W_E$
          $W_S = \arg\min_{W_S} L_{\text{ce}} + \beta L_{\text{adapt}} + \gamma L_{\text{aff}}$
```

3.4 Experiments

In this section, we first introduce the datasets and the implementation details of our experiments. Extensive ablation studies are followed to investigate the effectiveness of our proposed methods. Finally, we report our results and make a comparison with other lightweight models on three popular benchmarks: Pascal VOC Everingham et al., 2014, Cityscapes Cordts et al., 2016 and Pascal Context Mottaghi et al., 2014.

3.4.1 Datasets

Pascal VOC. This dataset contains 1,464 images for training, 1,449 for validation, and 1,456 for testing. It contains 20 foreground objects classes and an extra background class. Besides, the dataset is augmented by extra coarse labeling provided by Hariharan et al., 2011. The final performance is measured in terms of pixel intersection-over-union (mIOU) averaged across the 21 classes.

Cityscapes. This dataset focuses on semantic understanding of urban street scenes, which contains high-resolution images with 1024×2048 pixels and sense pixel-wise annotations. The dataset includes 5,000 finely annotated images collected from 50 cities and is split with 2,975 for training, 500 for validation, and 1,525 for testing. Following the evaluation protocol, 19 output of 30 semantic labels are used for evaluation.

Pascal Context. The dataset contains 10,103 images in total, out of which 4,998 are used for training and 5,105 are used for validation. Following Mottaghi et al., 2014, methods are evaluated on the most frequent 59 classes with one background class.

3.4.2 Implementation Details

The MobileNetV2, recently proposed by Sandler *et al.*Sandler et al., 2018, has attracted much attention for its computation efficiency and optimal trade-offs between accuracy and the number of operations measured by FLOPS, actual latency, and the number of parameters. There are also MobileNetV2-1.3 and MobileNetV2-1.4, which are model variants with a width multiplier of 1.3 and 1.4, respectively. The mobile segmentation models in Sandler et al., 2018 use a reduced form of DeepLabV3 Chen et al., 2017c. Built on this strong baseline, our method significantly boosts the performance without introducing extra parameters and computation overheads.

Training teacher network. To demonstrate the effectiveness of our method, we select two different teacher models, ResNet-50 He et al., 2016a and Xception-41 Chollet, 2016. Both atrous convolution and atrous spatial pyramid pooling (ASPP) are utilized to obtain a series of feature maps with large size.

Knowledge Translator: In all our experiments, the auto-encoder is composed of three 2-D convolution layers, and three 2-D transposed convolution layers. The convolution strides of the first convolution layer in the encoder and the last convolution layer in the decoder are set to 2. All six layers use a 3×3 kernel with padding of 1, BN layer, and ReLU activation function. The channels of the six convolution layers are all equal to the channels of the last feature maps in the teacher model.

Knowledge Adapter: The adapter is utilized on the top of the last convolution features of the student model. It contains three convolution layers using a 3×3 kernel with a stride of 1, padding of 1, BN layer, and ReLU activation function. The spatial resolution of the feature maps of the three convolution layers remains unchanged and the channels are adjusted to the number of the features in the last layer of the teacher model.

Training Process: We follow the training strategy of DeepLabV3+ Chen et al., 2018 and MobileNetV2 Sandler et al., 2018 to train the teacher and the student models, respectively. For the Pascal VOC dataset, we first train the teacher and the student networks for 500,000 iterations on the COCO dataset. 30,000 iterations are followed by training on the *trainaug* dataset. The training process can be split into two steps. First, we train 300K iterations on the COCO dataset, then 30K iterations on the *trainaug* dataset Hariharan et al., 2011. We validate the performance on the *val* set. For the Pascal Context dataset, we train our teacher and student model on the *train* set for 30,000 iterations, and the performance is tested on the *val* set. The COCO dataset is not used for pre-training. For the Cityscapes dataset, we train our model for 90,000 iterations on the *train-fine* dataset, which is fine-tuned on the *trainval* and the *train-coarse* sets for another 90,000 iterations and the performance is evaluated on the *test* set. Our reported model is not pre-trained on the COCO dataset. We set the learning rate to 0.007 and the total batch size of 64 in all our experiments. We train our model by using 4 GPUs with a crop size of 513×513 on the Pascal VOC and Pascal Context and 769×769 on the Cityscapes. We use mini-batch stochastic gradient descent (SGD) with batch size 16 (at least 12), momentum 0.9, and weight decay 4×10^{-5} in training. Similar to Chen et al., 2018, we apply the poly learning rate strategy with power 0.9. General data augmentation methods are also used in network training, such as randomly flipping the images and randomly performing scale jitter.

Training auto-encoder. We finished the auto-encoder training within one epoch with a learning rate of 0.1. Large weight decay of 10^{-4} is used to attribute low energy to a smaller portion of the input points.

Training the whole system. Most of the training parameters are similar to the process of training the teacher network, except that our student network does not involve the ASPP and the decoder, which are the same as Sandler et al., 2018. With the help of atrous convolution, low-resolution feature maps are generated. During the training process, the parameters of the teacher net W_T and the parameters for auto-encoder W_E are fixed without updating.

3.4.3 Ablation for the knowledge adaption and the affinity distillation module.

In order to make use of rich spatial information, we propose to translate the knowledge from the teacher and force the student to mimic this compact format. The affinity distillation module is also proposed to make up the limited receptive field of the small student model. To show a better understanding, we visualize the effect of the affinity distillation module in Figure 3.3. It can be seen from Figure 3.3, that more context and long-range dependencies are captured with the help of our proposed method. We show the statistic results in Table 3.2, where performance is evaluated using mIOU. The

TABLE 3.1. Comparison with other lightweight models on the Pascal Context *val* set. "-" means not provided.

Method	FLOPS	Params	mIOU(%)
FCN-8s Long, Shelhamer, and Darrell, 2015	135.21G	1.48M	37.8
ParseNet Liu et al., 2015	162.82G	21.53M	40.4
Piecewise_CRF Lin et al., 2016	>100G	-	43.3
DAG_RNN Shuai et al., 2017	>100G	-	42.6
MobileNetV2 Sandler et al., 2018	5.52G	2.12M	39.9
Ours	5.52G	2.12M	41.2

TABLE 3.2. Ablations for the proposed method. **T**: The teacher model has a output stride of 8s. **S**: The student model (following the implementation of Sandler et al., 2018, without ASPP and decoder) has an output stride of 16s. **KA** represents knowledge adaption. The FLOPS is estimated with an input size of 513×513. For a fair comparison, all the models are trained on the Pascal VOC *trainaug* set Hariharan et al., 2011 tested on the *val* set without pre-training on the COCO dataset. As can be seen, our proposed method with small feature resolution outperforms the student model with large feature resolution by only 31% FLOPS.

Method	mIOU%)	FLOPS	Params
T: ResNet-50-8s He et al., 2016a	76.21	90.24B	26.82M
S1: MobileNetV2-16s Sandler et al., 2018	70.57	5.50B	2.11M
S2: MobileNetV2-8s Sandler et al., 2018	71.90	17.70B	2.11M
S1+affinit-16s	71.53	5.5B	2.11M
S1+KA+affinity-16s	72.50	5.5B	2.11M

Table 3.3. The performance on the Pascal VOC 2012 *val* data set with different student and teacher networks. MobilNetV2 is tailored with a width-multiplier. Performances are obtained by training on *trainaug* set.

Method	mIOU(%)	FLOPS	Params
T1: ResNet-50 He et al., 2016a	76.21	90.24B	26.82M
T2: Xception-41 Chollet, 2016	77.2	74.69B	27.95
S1: MobileNetV2-1.0 Sandler et al., 2018	70.57	5.50B	2.11M
S2: MobileNetV2-1.3 Sandler et al., 2018	72.60	9.02B	3.38M
S3: MobileNetV2-1.4 Sandler et al., 2018	73.36	10.29B	3.88M
T1+S1+our method	72.50	5.5B	2.11M
T2+S1+our method	72.40	5.5B	2.11M
T1+S2+our method	74.26	9.02B	3.38M
T1+S3+our method	74.07	10.29B	3.88M

model is tested in one single scale on the Pascal VOC *val* set without pretraining on the COCO dataset. As can be seen, the affinity distillation module boosts the performance from 70.57 to 71.53, and another 0.97% with the help of knowledge adaption. Because the affinity matrix mismatches if two models have different output features, in order to show the effect of a single affinity module, we resize the feature maps to the same dimension. Our MobileNetV2 with output stride of 16 even outperforms MobileNetV2 with an output stride of 8, using only 31% FLOPS. More comparisons with different output stride settings can be found in Figure 3.2, where our 16s model performs even better than the baseline model with 4s output by using only 8% FLOPS without introducing extra parameters.

3.4.4 Ablation for different networks.

From Sandler et al., 2018, MobileNetV2 tailors the framework to achieve different accuracies, by using width-multiplier as a tunable hyper-parameter, which is used to adjust the trade-off between accuracy and efficiency. In our experiments, we choose width-multipliers of 1.3 and 1.4, which are implemented with official pre-trained models on ImageNet. In order to validate the effectiveness of our proposed method, we choose two different network architectures, ResNet-50 He et al., 2016a and, Xception-41 Chollet, 2016. The results are shown in Table 3.3. The performance of MobileNetV2-1.0 gains 1.93 and 1.83 improvements under the guidance of ResNet-50 and Xception-41, respectively. Improvements of 1.66 and 0.71 are also observed with different student networks: MobileNetV2-1.3 and MobileNetV2-1.4.

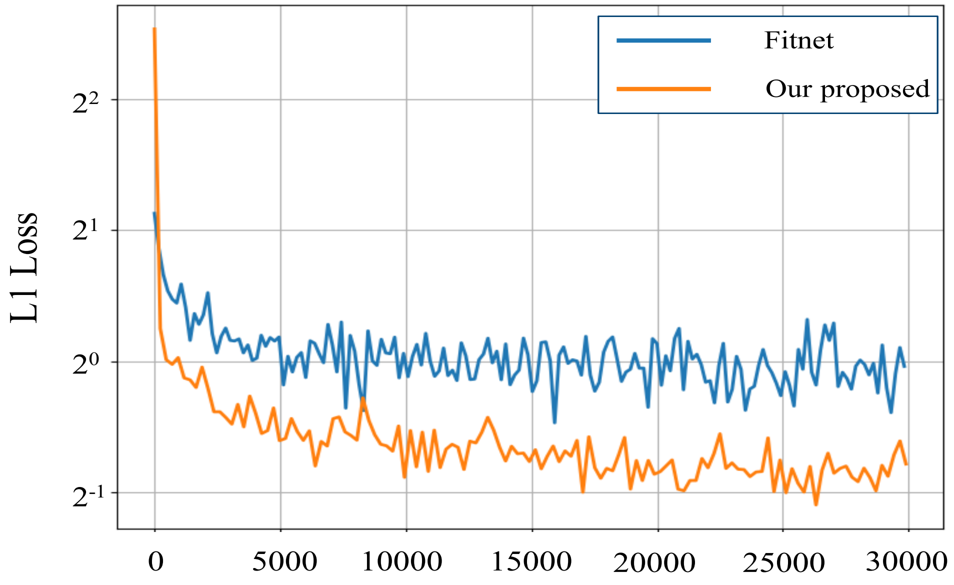


FIGURE 3.4. The L1 loss curve for the knowledge transferring process. Our method using translator and adapter makes it easier for student network to learn and replicate the knowledge.

3.4.5 Ablation for other method for knowledge distillation.

In this experiment, we make comparisons with other knowledge distillation methods: KD Hinton, Vinyals, and Dean, 2015 and FitNet Romero et al., 2015, which are designed for image-level classification. The knowledge defined in Hinton, Vinyals, and Dean, 2015 is the soft label output by a teacher network. The soften degree is controlled by a hyper-parameter temperature of t, which has a significant influence on the distillation and learning processes. We set t to 2, 4, 6. To make fair comparisons, we bilinearly upsample the logits map to the size of the teacher network. The results

TABLE 3.4. The performance on the Pascal VOC 2012 *val* set in comparison with KD Hinton, Vinyals, and Dean, 2015 and FitNet Romero et al., 2015. All the results are achieved by training only on the Pascal VOC *trainaug* set.

Method	mIOU(%)
T: ResNet-50 He et al., 2016a	76.21
S: MobileNetV2 Sandler et al., 2018	70.57
S+KD Hinton, Vinyals, and Dean, 2015 (t=2)	71.32
S+KD Hinton, Vinyals, and Dean, 2015 (t=4)	71.21
S+KD Hinton, Vinyals, and Dean, 2015 (t=8)	70.74
S+FitNet Sandler et al., 2018	71.30
S+Ours	72.50

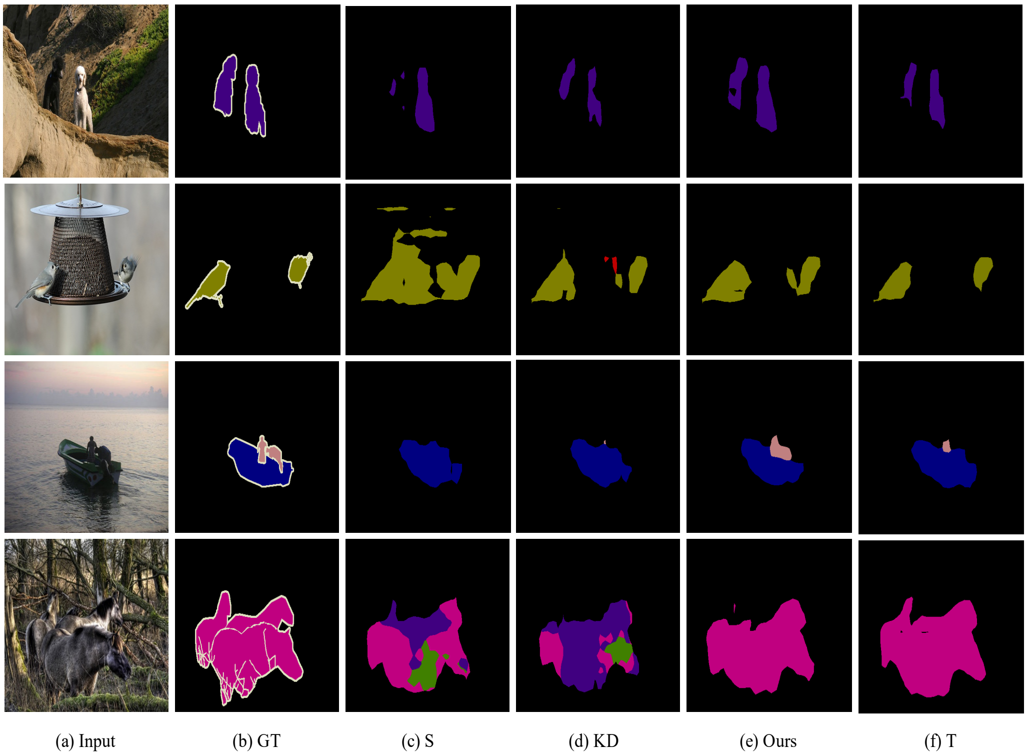


FIGURE 3.5. Comparison of segmentation results. (a) Input image. (b) Ground truth. (c) The results of the student network, MobileNetV2 Sandler et al., 2018. (d) Results of the knowledge distillation Hinton, Vinyals, and Dean, 2015 with MobileNetV2 Sandler et al., 2018. (e) Results of our proposed method with MobileNetV2 Sandler et al., 2018. (f) Results of the teacher network, which is ResNet50 He et al., 2016a.

are evaluated on the Pascal VOC *val* dataset. All results are achieved without pretraining on the COCO dataset.

FitNet Romero et al., 2015, different from KD, tries to match the intermediate representation between two models. But this requires a similar network design. In our experiments, we directly upsample the feature map of the last layer and add a ℓ_2 loss. The loss curve is shown in Figure 3.4. Our proposed method successfully translates the knowledge from the teacher to a format that is easier to be learned.

As shown in Table 3.4, the fluctuation of mIOU is observed with different settings of T. Our method achieves better performances than KD, with all the hyper-parameters fixed across all experiments and datasets. Our method also outperforms FitNet by 1.2 points, indicating that the knowledge defined by our method alleviates the inherent difference of two networks. Compared with the traditional methods, the qualitative segmentation results in Figure 3.5 visually demonstrate the effectiveness of our distillation method for objects that require more context information, which is captured by our proposed affinity transfer module. On the other hand, the knowledge translator and adapter reduce the loss of the detailed information and produce more consistent and detail-preserving predictions, as shown in Figure 3.6.

3.4.6 Comparing with other lightweight models.

We first test our method on the Pascal Context dataset. The results are shown in Table 3.1. Our proposed method boosts the baseline by 1.3 points.

Then we compare our proposed method with other state-of-the-art lightweight models on the Pascal VOC *val* dataset. The results are shown in Table 3.5. Our model yields mIOU 75.8, which is quantitatively better than several methods that do not care about speed. It also improves the baseline of MobileNetV2 by about 1 point. Finally, we testify the effectiveness of our method on the Cityscapes dataset. It achieves 70.3 and 72.7 mIOU on the *val* and *test* data sets, respectively. Even built on a highly competitive baseline, our method boosts the performance by 2.1 and 2.5 points, without introducing extra parameters and computations overheads, as shown in Table 3.6.

3.5 Conclusion

In this chapter, we present a novel knowledge distill framework tailored for semantic segmentation. We improve the performance of the student model by translating the high-level feature to a compact format that is easier to be learned. Extensive experiments have been done to testify the effectiveness of our proposed method. Even built upon a highly competitive baseline, our method (1) improves the performance of the student model by a large margin without introducing extra parameters or computations (2) achieves better results with much less computation overhead.

TABLE 3.5. Comparison with other lightweight models on the Pascal VOC 2012 *val* data set. Speed is tested on single 1080Ti GPU with input size of 513 × 513. The baseline is our implementation of MobileNetV2.

Method	basemodel	FPS	mIOU(%)
CRF-RNN Zheng et al., 2015	VGG-16	7.6	72.9
MultiScale Yu and Koltun, 2014	VGG-16	16.7	73.9
DeeplabV2 Chen et al., 2017b	VGG-16	16.7	75.2
MobileNetV2 Sandler et al., 2018	MobileNet	120.7	75.3
Baseline	MobileNet	120.7	74.8
Ours	MobileNet	120.7	75.8

TABLE 3.6. Performance and computation comparisons of our proposed method against other light-weight models on the Cityscapes *val* and *test* data sets. The running times are all computed with input size of 1025 × 2049. "-" means not provided.

Method	Year	Time	mIOU(%)	
			val	*test*
DeepLabV2 Chen et al., 2017b	2016	652.9ms	-	71.4
Dilation-10 Yu and Koltun, 2014	2017	3549.5ms	-	67.1
PSPNet Zhao et al., 2017	2017	2647.4ms	-	80.2
ResNet38 Wu, Shen, and Hengel, 2016	2017	3089.9ms	77.86	78.4
SegNet Badrinarayanan, Kendall, and Cipolla, 2017	2015	89.2ms	-	57.0
ENet Paszke et al., 2016	2016	19.3ms	-	58.3
SQ Treml et al., 2016	2016	-	-	59.8
ICNet Zhao et al., 2018	2018	33.0ms	67.7	70.6
MobilenetV2 Sandler et al., 2018	2018	38.0ms	68.9	70.2
Ours	-	38.0ms	71.0	72.7

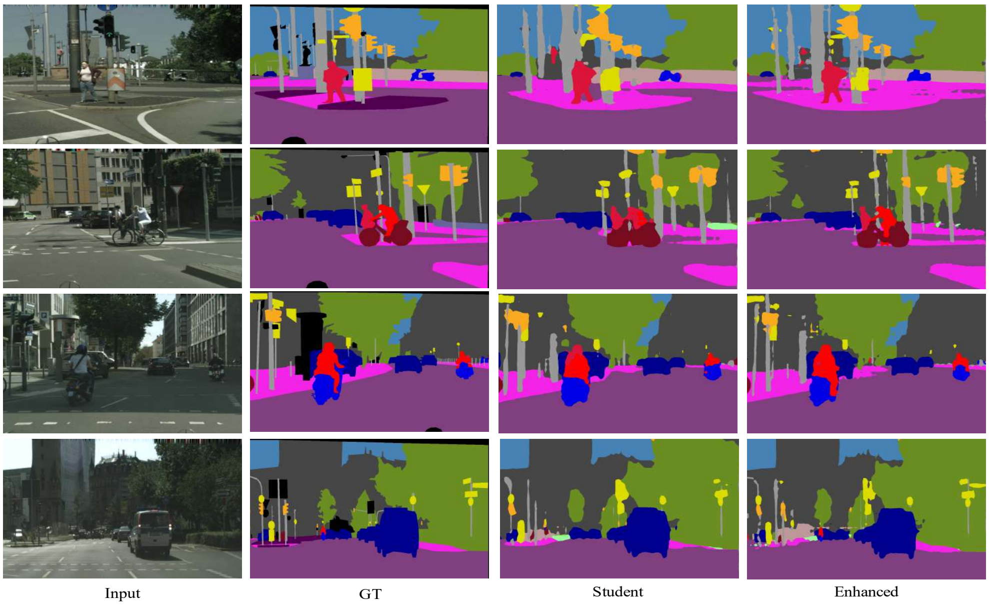


FIGURE 3.6. Example results on the Cityscapes dataset. From left to right are: (1) Input images, (2) Ground truth, (3) The results of the student net (4) The results of our proposed method.

Chapter 4

Instance-Aware Embedding for Point Cloud Instance Segmentation

4.1 Introduction

The task of instance segmentation has recently gained popularity. As an extension to semantic segmentation, this task needs to separate pixels/points that have identical categories into individual groups. In the 2D image domain, many approaches He et al., 2017a; Dai, He, and Sun, 2016; Dai et al., 2017b have been proposed and achieve promising results. With the growth of the availability of 3D sensors, more and more researches have focused on 3D scene understanding, which is a fundamental necessity for robotic vision, autonomous driving, and virtual reality. Although instance segmentation in the 3D domain has started to draw attention and has been discussed in Wang et al., 2018a; Wang et al., 2019b; Yi et al., 2018; Pham et al., 2019; Yang et al., 2019a, it still lags behind its 2D image counterpart and far from being solved.

Similar to the tasks of dense prediction in 2D images Chen et al., 2018; Lin et al., 2017b, context is also important in 3D the domain. For 3D point clouds, PointNet++ Qi et al., 2017a is the first work that captures local structure information and has been successfully utilized in the task of semantic segmentation. It maintains an encoder-decoder architecture, which includes several set-abstraction layers and feature-propagation layers for down-sampling and up-sampling, respectively. Algorithms such as radius search and k nearest neighbours (K-NN) search are utilized for aggregating local context knowledge. Built upon this powerful network, many methods Wang et al., 2018a; Wang et al., 2019b; Pham et al., 2019 have been proposed to tackle the task of instance segmentation on point clouds. To encode meaningful context information, ASIS Wang et al., 2019b is proposed to associate two tasks together so they can cooperate with each other. JSIS3D Pham et al., 2019 applied multi-value Conditional Random Field (CRF) that formulates a joint optimization for semantic segmentation and instance segmentation in a unified framework. However, these methods fail to explicitly encode the *instance contextual knowledge* and *geometric information*, which are extremely critical for separating adjacent instances and handling complex situations. For example, two neighbouring chairs can be easily confused and grouped as one united instance if boundaries and geometric information are not encoded in the embedding space (e.g., the second row in Figure 4.1). In this chapter, we address the problem by proposing an Instance-Aware Module (IAM) to learn the instance level context by locating representative regions for each input point. Moreover, geometric knowledge is explicitly encoded in the embedding space, which is an informative indicator to identify the points belonging to the same instance. The whole framework can be trained in an end-to-end manner to tackle instance segmentation and semantic segmentation simultaneously with little computation resource overhead.

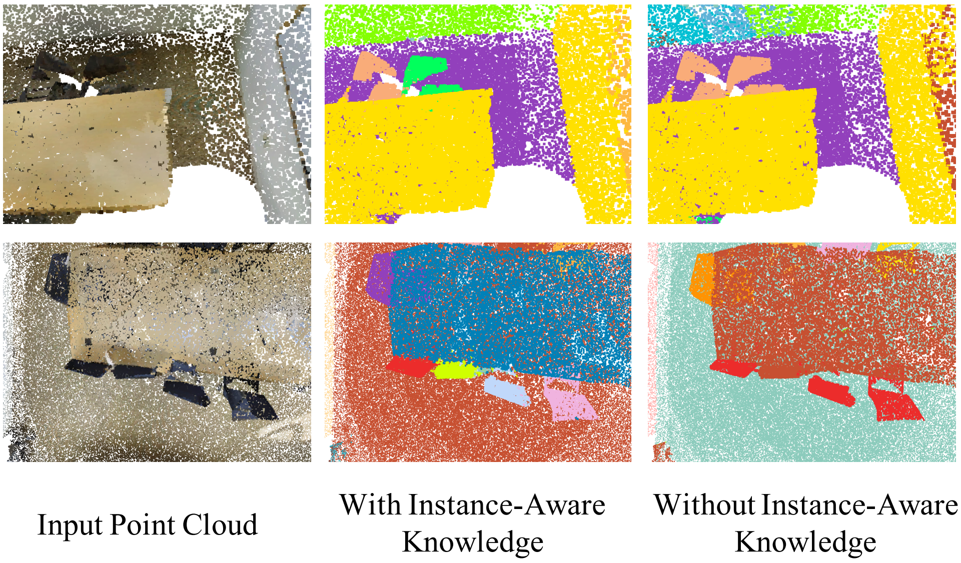

Input Point Cloud
With Instance-Aware Knowledge
Without Instance-Aware Knowledge

FIGURE 4.1. Comparison of the instance segmentation results with and without the proposed Instance-Aware Module (IAM). The proposed IAM successfully encodes instance-aware information and geometric knowledge, which are critical for separating adjacent instances. Note that different instances can be presented in different colours.

Specifically, our method maintains an encoder-decoder architecture. Different from previous methods that only maintain an instance grouping branch and a semantic segmentation branch, we come up with a novel light-weight instance-aware module, which localizes representative points within the same instance for each input point.

The information from these representative points is then aggregated into the decoding process of the instance branch, generating instance-aware contexts for learning discriminative point-level embeddings. Moreover, the normalized geometric centroids of these representative points (predicted by every input point feature), are directly added to the embedding space, which provides critical geometric knowledge for identifying and reducing the ambiguity of adjacent instances.

The training of the instance-aware module is regularized jointly by the bounding box and instance segmentation supervision, such that the meaningful semantic regions can be tightly bounded by the spatial extension of the instance and guided towards representative regions of the instance.

Compared with the conventional representation of an instance by using vertexes to represent a bounding box, learning semantically meaningful regions helps to remove unrelated background and noise information. As it is applied in the bottleneck layer, very few additional computations are introduced. Compared with ASIS Wang et al., 2019b, which needs to search neighbours of every input point exhaustively, our approach shows superiority in both efficiency and effectiveness.

To validate the effectiveness of our proposed method, extensive experiments have been conducted on three popular benchmarks. The flexibility of our method allows it to be applied in not only indoor scenes but objects with fine-grained part labels. State-of-the-art performances are achieved on these datasets.

4.2 Background

Point clouds are a set of spatial points that can be collected by a laser scanner and are represented in an identical coordinate system. Other than spatial locations, point clouds can also contain rich information such as color, density, and normal vector. Previous methods for extracting features of the point cloud are handcrafted. One of the most significant properties of the point cloud is non-grid, making the extracted feature to maintain invariance under different transformations. Many methods have been proposed which can be roughly categorized as local features and global features. In many situations, finding a better combination of different features are necessary and non-trivial. Recently, deep-learning-based approaches have achieved huge success in 2D images. However, it is non-trivial to extend these methods to the 3D domain for the following reasons: (1) the network needs to be invariant to the inputs as they are unordered sets of vectors. (2) the method should have the ability to extract both local and global representations, which is similar to CNN to build hierarchical features. In this section, we mainly introduce two pioneering works: PointNet Qi et al., 2017b and PointNet++ Qi et al., 2017a

4.2.1 PointNet

PointNet is the first deep-learning-based method for feature extraction on the point cloud, which shows superiority in the tasks of classification and semantic segmentation. The whole architecture has three core parts: (1) alignment sub-network, aiming to remove certain transformations and force the learned features to be invariant to different transformations. (2) local and global feature combination structure, which concatenates two features for the label predictions. (3) Symmetry function for unordered point sets, which extracts point-wise representation via a multi-layer perceptron (MLP) and generates features that are invariant to input permutation by using a max-pooling layer. PointNet, while simple and effective, achieves state-of-the-art performance on various tasks.

4.2.2 PointNet++

The basic idea of PointNet is to learn point-wise features and achieves invariance property by applying a max-pooling layer. However, local structures are ignored, limiting its ability to capture local patterns and shape information. To address this problem, PointNet++ Qi et al., 2017a is proposed by applying PointNet recursively on nested local regions, which is critical to get hierarchical representations.

Every abstract level contains the following layers:

- **Sampling layer:** One scan of laser data contains numerous points. Extracting local feature for every single point requires large computation cost. The author proposed to downsample a set of points by utilizing the farthest point sampling (FPS) strategy. Compared to the random sampling, FPS is more robust to cover the entire point set.
- **Grouping layer:** Given the sampling points, the grouping layer is defined to find the local context by a hyper-parameter denoting the euclidean distance. In convolutional networks, the local context is aggregated within a certain kernel size of the pixel. The grouping layer uses ball-search or k nearest neighbour search for gathering local information.
- **PointNet Layer:** In this layer, each local region in the output is abstracted by its centroid and local feature that encodes the centroid's neighbourhood.

4.2.3 Single-stage instance segmentation

Instance segmentation is a challenging task that not only needs to classify the category information of every pixel (or point), it also requires to distinguish separated instances. Compared to the two-stage methods that follow the pipeline of MaskRCNN He et al., 2017a, single-stage frameworks have several advantages: (1) It is faster and stable than two-stage frameworks as the running time does not heavily rely on the detection results. (2) The mask branch is not cascaded on the proposal. Brabandere, Neven, and Gool, 2017 is one of the typical one-stage methods for instance segmentation. It proposes to cluster instances based on the per-pixel embedding representation, which pulls the pixels from the same instance towards the cluster center and pushes cluster centers away from each other. A similar idea is adopted in this chapter.

4.3 Proposed Method

In this section, we describe our proposed Instance-Aware Module (IAM), which can encode both instance-aware context and instance-related geometric information. Details of the approach are presented below.

4.3.1 Network Framework

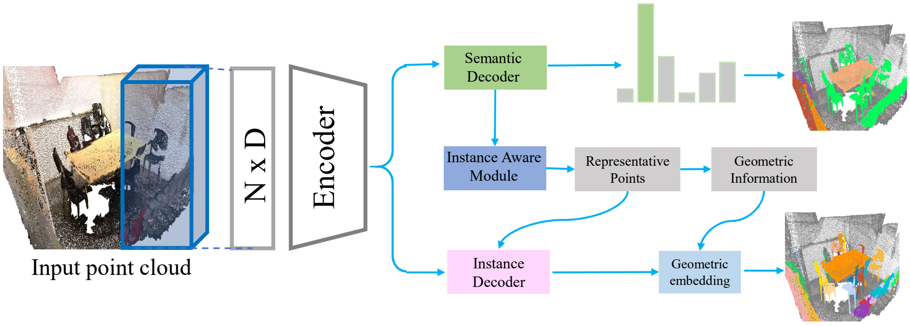


FIGURE 4.2. The whole framework of our proposed one-stage method, which is a simple and clear encoder-decoder architecture. The input point clouds first go through a shared encoder network, and two parallel decoders are followed: one for semantic segmentation, one for instance grouping. A novel instance aware module (IAM) is proposed to generate representative points for instance segmentation. We use the coordinates of representative points to select argument features for instance segmentation module and the geometric information of the coordinates to extend the instance embedding. The whole framework is end-to-end trainable.

As shown in Figure 4.2, we apply an encoder-decoder architecture. The encoder is shared by two tasks and takes point sets $P \in \mathbb{R}^{N \times D}$ as input, where N denotes the total number of the points and D refers to the input feature dimension. The input features can consist of colour and position information, e.g., X, Y, Z, R, G, and B. The decoder contains two parallel branches: one for semantic segmentation, one for instance embedding. The semantic segmentation branch generates per-point classification results $S \in \mathbb{R}^{N \times D_c}$, where D_c is the category number. Focal loss Lin et al., 2017c L_{fl} is applied to address the category imbalance during the training process.

Besides, the instance branch outputs per-point embedding features $E \in \mathbb{R}^{N \times D_e}$ for learning a distance metric, where D_e is the embedding dimension. The embeddings belonging to the same instance should end up close together, and the embeddings belonging to the different instances should end up far apart. During the inference, a clustering algorithm is applied to obtain the final grouping results. A novel Instance-Aware-Module (IAM) for producing instance aware knowledge is achieved by detecting the spatial extension of an instance. Through IAM, representative points locating on the corresponding instance provide instance-aware knowledge, which contains two parts: (1) instance-related contextual information via detection a set of regions that are tightly covering the spatial extension of an instance. (2) instance geometric knowledge that is critical for separating adjacent objects.

4.3.2 Instance-Aware Module

Inspired by Yang et al., 2019b, which provides a finer representation of objects as a set of sampling points, we tailor it for 3D point clouds and propose to regularize the features from the same instance to have identical geometric centroids with minor variation. We propose an instance-aware module (IAM) mainly for selecting representative points that capture spatial instance context. For point p_i with position x_i, y_i and z_i, point-level offsets are predicted by the contextual detection branch to represent the spatial extension of the instance, denoted as $\{\Delta x_i^k, \Delta y_i^k,\ \Delta z_i^k\}_{k=1}^K$. Representative regions of the instance predicted by p_i is $\mathcal{R}_i$, which can be simply represented as:

$$\mathcal{R}_i = \{(x_i + \Delta x_i^k, y_i + \Delta y_i^k, z_i + \Delta z_i^k)\}_{k=1}^K, \tag{4.1}$$

where K is the number of representative points and i represent the i-th point. The axis-aligned bounding box predicted by every point can be formulated as $\mathcal{B}_i$ through a min-max function F: $\mathcal{B}_i = F(\mathcal{R}_i)$

Learning these representative regions is jointly driven by both the spatial bounding boxes and the instance grouping labels, such that $\mathcal{R}_i$ can tightly compass the instance. To achieve this, three losses are provided: L_{bnd}, L_{cen} and L_{ins} (the last two will be discussed in the next section).

L_{bnd} is to maximize the overlaps of the bounding boxes between the prediction and the ground truth. 3D IoU loss is utilized in our method:

$$L_{bnd} = \frac{1}{N}\sum_{i=1}^{N} 1 - IoU(GT_i, \mathcal{B}_i), \tag{4.2}$$

where N is the total number of points, $\mathcal{B}_i$ is the predicted bounding box of the i-th point and GT_i is the 3D axis-aligned bounding box ground truth of the i-th point. To have a better understanding of the detection branch, we visualize $\mathcal{R}_i$ in Figure 4.3. Green points are selected p_i, and red points are the predicted $\mathcal{R}_i$. We choose the number of representative points as 12, which empirically works well in our experiments. Employing more points will have limited improvements. Therefore, in terms of efficiency, we choose $K = 12$. Instance related regions are located and successfully cover the spatial extension. In the next section, we provide details of how to incorporate these instance contextual information.

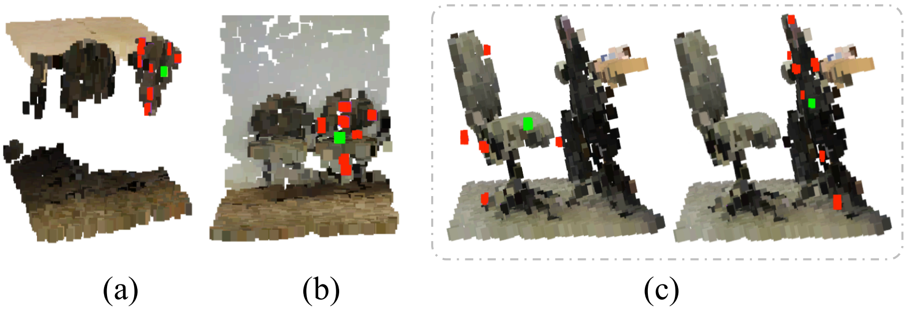


FIGURE 4.3. Visualization of detected representative points. The green point is randomly selected, and the red points are the corresponding meaningful regions output by the IAM. Due to the encoded instance context information, our method can separate adjacent objects. (Figure best viewed in color)

4.3.3 Instance Decoder

Conventionally, the inputs of the instance decoder are down-sampled bottleneck points $P_b \subseteq P$, and the corresponding features are denoted as F_b. These features are gradually propagated to the full set of points through several up-sampling layers. To encode the instance context during the propagation process, we utilize the meaningful semantic regions of $\mathcal{R}_b$ for the bottleneck points.

Representations of F_b are augmented by aggregating information from $\mathcal{R}_b$ that covers the instance spatial extent. As these detected points are not necessarily located on the input points, the features of $\mathcal{R}_b$ are interpolated by using K-NN. The interpolated features are then added to the original F_b, generating features containing both local representation and instance context. Compared with ASIS Wang et al., 2019b, which has to search neighbours for every input point, our method, on the other hand, is more efficient. As K-NN is applied in the bottleneck layer, the searching space in P_b is much smaller than that in P, introducing very limited computation overhead. The combined features are gradually upsampled during the decoding process, propagating the instance-aware context through all points.

Geometric information is critical for identifying two close objects. To learn a discriminative embedding feature, we directly concatenate the normalized centroids of coordinates to the embedding space. Considering the centroid $C(\mathcal{B}_i)$ predicted by point p_i, where $C(\cdot)$ is the function for computing geometric centroids of a given bounding box, the final per-point embedding feature can be represented as $\hat{E}_i = \text{Concat}(E_i, C(\mathcal{B}_i))$, where E_i is the embedding feature produced from the instance branch. Besides, to force the geometric information to be consistent for the points that have identical instance label, we pull the predicted geometric centroids from the same instance towards the cluster center by:

$$L_{cen} = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N_m} \sum_{i=1}^{N_m} [\|C(\mathcal{B}_i) - \mu_m\| - \sigma_v]_+^2, \tag{4.3}$$

where M is the total number of instances, and N_m is the point number for m-th instance. μ_m refers to the average predicted geometric centroids of m-th instance. $[x]_+$ is defined as$[x]_+ = \max(0, x)$ and σ_v is the loose margin. The L_{cen} is designed for forcing the additional geometric information to have less variation and to be informative for separating adjacent objects.

The informative per-point embedding $\{\hat{E}\}_{n=1}^{N}$ is applied for learning a distance metric that could pull intra-instance embedding toward the cluster center and push instances centers away from each other. The loss function is formulated as:

$$\begin{aligned} L_{ins} = &\underbrace{\frac{1}{M(M-1)}\sum_{a=1}^{M}\sum_{\substack{b=1\\b\neq a}}^{M}[2\sigma_d - \|\mu_a - \mu_b\|]_+^2}_{inter-instance} \\ &+ \underbrace{\frac{1}{M}\sum_{i=1}^{M}\frac{1}{N_m}\sum_{m=1}^{N_m}[\|\mu_m - \hat{E}_m\| - \sigma_v]_+^2}_{intra-instance}, \end{aligned} \tag{4.4}$$

where M is the total instance number, N_m is the point number of the m-th instance. σ_d and σ_v are relaxation margins. During the training process, the first term pushes instance clusters away from each other and the second term pulls the embedding towards the cluster center.

During the inference process, a fast mean-shift algorithm is applied for clustering different instances in the embedding spaces.

To summarize, our method is end-to-end trainable and supervised by four losses. The loss weights for the four losses are all set to 1 in all our experiments.

$$L = L_{fl} + L_{bnd} + L_{cen} + L_{ins}, \tag{4.5}$$

4.4 Experiments

In this section, we evaluate the effectiveness of our proposed method. Both qualitative and quantitative experiments are conducted and reported.

4.4.1 Datasets

We introduce three popular datasets that have instance annotations: Stanford 3D Indoor Semantic Dataset (S3DIS) Armeni et al., 2016, ScanNetV2 Dai et al., 2017a, and PartNet Mo et al., 2019. S3DIS is collected in 6 large-scale indoor areas, covering 272 rooms. The whole dataset contains more than 215 million points and is consisted of 13 common semantic categories. ScanNetV2 Dai et al., 2017a is an RGB-D video dataset. It contains more than 1500 scans, which is split into 1201, 300, and 100 scans for training, validation, and testing, respectively. The dataset contains 40 classes in total, and 13 common categories are evaluated. Different from the above two datasets, PartNet Mo et al., 2019 is a consistent large-scale dataset with fine-grained object annotations. It consists of more than 570k part instances covering 24 object categories. Each object contains 10000 points. Similar to GSPN Yi et al., 2018, we select five categories that have the largest number of training examples.

4.4.2 Evaluation Metrics

On S3DIS dataset, we conduct 6-fold cross-validation. Similar to SGPN Wang et al., 2018a and ASIS Wang et al., 2019b, the performance on Area-5 is also reported. On ScanNetV2 Dai et al., 2017a, we report our results on the validation set, which contains more instances and has more stable results. On PartNet Mo et al., 2019 dataset, five selected categories are Chair, Storage, Table, Lamp, and Vase. Both coarse and fine-grained results are included. Different levels of different categories are trained separately and independently. The evaluation metrics for semantic segmentation are the overall pixel-wise accuracy ($mAcc$), category-wise mean accuracy ($oAcc$) and average intersection-over-union ($mIoU$). The instance segmentation is evaluated by the average instance-wise coverage ($mCov$), mean weighted instance-wise coverage ($mWCov$), mean instance precision ($mPrec$) and recall ($mRec$) with IoU threshold of 0.5. The weights for $mWCov$ is calculated by $w_i = \frac{|N_i|}{\sum_k |N_k|}$, where i is the i-th instance and N_k is the point number of k-th ground truth instance.

4.4.3 Implementation Details

For S3DIS Armeni et al., 2016 and ScanNetV2 Dai et al., 2017a, each scan contains millions of points, making it hard to process all data at one time. In our experiments, we split each scene into $1m \times 1m$ overlapped blocks with $0.5m$ stride. Then, 4,096 points are randomly sampled across each block. Similar to SGPN Wang et al., 2018a, every point is represented by a 9-D feature (X, Y, Z, R, G, B, and normalized positions in blocks N_X, N_Y, N_Z). PartNet Mo et al., 2019, on the other hand, is proposed for shape analysis which contains 10,000 points for each instance. We randomly select 8,000 for training and 10,000 for testing.

Although our method is not restricted to any specific network, all experiments are conducted with vanilla PointNet++ Qi et al., 2017a as the backbone (without multi-scale grouping) and leave the other choices for future study. One single GTX1080Ti GPU card is used for training with the batch size set to 16. The initial learning rate is set to 0.001 and divided by 2 in every 300k iterations. We use Adam optimizer with momentum set to 0.9, and the whole network is trained for 100 epochs. The hyper-parameters for discriminative loss are identical with original setting in Wang et al., 2019b: $\sigma_v = 0.5$, $\sigma_d = 1.5$. Besides, for testing the whole scene on S3DIS and

ScanNetV2, a method named BlockMerging is used for grouping blocks according to the segmentation information of the overlapped areas. The whole algorithm is shown in Algorithm 2. Each training scene is divided into $1m \times 1m$ blocks. We slide the window with 0.5m overlap in a snake pattern, as shown in Figure 4.4. The entire scene is also divided into a $401 \times 401 \times 401$ voxel V and V_k is used to represent the instance label of the k_{th} voxel, where $0 \leq k < 401 \times 401 \times 401$. For each voxel, we use $PL_{i,j}$ to indicate the predicted instance label for the j_{th}point in the i_{th} block. The details of the block merging algorithm is shown in Algorithm 2

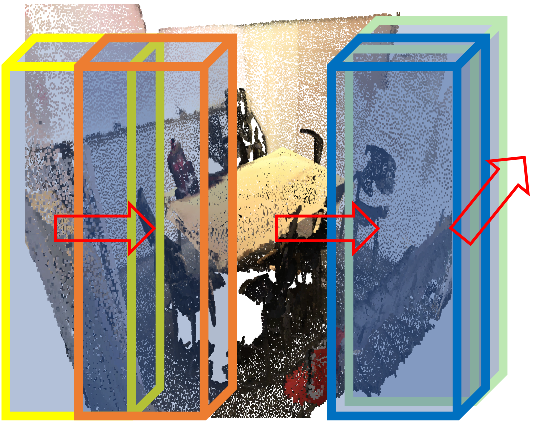


FIGURE 4.4. Illustration of block merging algorithm.

We first build a strong baseline that contains two decoder branches: one is the semantic segmentation, and the other is the instance embedding branch. Two losses are used for supervising the two branches: the cross-entropy loss for the segmentation task and the discriminative loss for instance grouping. The discriminative loss forces points belonging to the same instance to lie close together in the embedding space and keep a large margin for points belonging to different instances. The loss weights are set to 1.0. We conduct our experiments on the ScanNetV2 validation set.

4.4.4 Ablations on Focal Loss

Focal loss Lin et al., 2017c is first proposed in object detection task to address the problem of data imbalance between positive and negative samples. Due to the imbalance of categories introduced in the point cloud, we apply focal loss in the segmentation branch with default parameters identical to Lin et al., 2017c. The results are shown in Table 4.1, and the focal loss can improve the results by 2.0 for AP_{50}, from 22.0 to 24.0.

4.4.5 Ablations on Instance Aware Module

We study the influence of the proposed instance-aware module, which first finds out representative points of the instance, and then features from these sampled points are

Algorithm 2: BlockMeriging Wang et al., 2018a

```
Input   : $V$, $PL$
Output: Point instance labels for the whole scene $L_{ins}$
Initialize $V_k = -1, k \in [0, 401 \times 401 \times 401)$ ;
$GroupCount \leftarrow 0$;
for every block $i$ do
    if $i$ is the 1st block then
        for every point $P_j$ in block $i$ do
            Define $k$ where $P_j$ is located in the $k$th cell of $V$;
            $V_k \leftarrow PL_{1j}$;
        end
    else
        for every instance $I_j$ in block $i$ do
            Define $V_{I_j}$ points in $I_j$ are located in cells $V_{I_j}$;
            $V_t \leftarrow$ the cells in $V_{I_j}$ that do not have value $-1$;
            if the frequency of the mode in $V_t < 30$ then
                $V_{I_j} \leftarrow GroupCount$;
                $GroupCount \leftarrow GroupCount + 1$;
            else
                $V_{I_j} \leftarrow$ the mode of $V_t$;
            end
        end
    end
end
for every point $P_j$ in the whole scene do
    Define $k$ where $P_j$ is located in the $k$th cell of $V$;
    $L_j \leftarrow V_k$;
end
```

TABLE 4.1. Ablation study on ScanNetV2 dataset. Both AP_{50} and AP_{25} are reported on the validation set. **FL** refers to focal loss. **InsContext** refers to instance-aware context. $\mathbf{L_{cen}}$ refers to centroid constrain loss in Eq. 4.3. **GE** refers to geometric embedding.

Method	FL	InsContext	L_{cen}	GE	AP_{50}	AP_{25}
Baseline					22.0	45.2
	✓				24.0	45.5
	✓	✓			27.6	48.2
	✓	✓	✓		28.9	48.9
Ours	✓	✓	✓	✓	**31.5**	**50.4**

aggregated. Encoding the spatial extension knowledge helps to separate and distinguish close instances. As shown in Table 4.1, the instance aware decoder boosts the performance by a large margin, improving AP_{50} from 24.0 to 27.6 and AP_{25} from 45.5 to 48.2. Besides, simply enlarging the dimension of the embedding space can not bring further improvement in performance (presented in ASIS Wang et al., 2019b). The proposed geometric embedding provides informative knowledge, which brings about 2.6% improvements in AP_{50}, demonstrating the effectiveness of our proposed method. The qualitative result is shown in Figure 4.5. Our method shows robustness to the intensive scenes, which require more discriminative features to separate different instances.

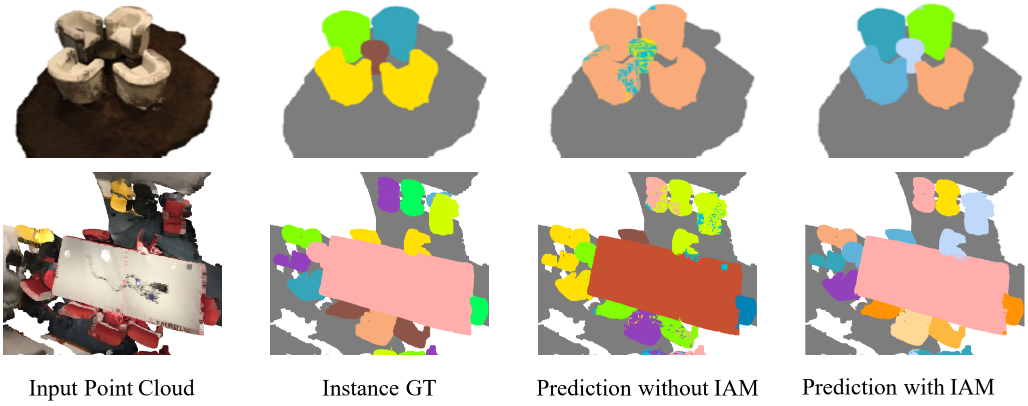


FIGURE 4.5. Comparison of the results with and without the Instance-Aware Module. Due to the successfully encoded instance context and geometric information, our method generates discriminative results, especially for the nearby objects.

Furthermore, we visualize the distribution of the predicted positions. As shown in Figure 4.3, it can learn a geometric representation of the objects and tend to localize at representative positions of the objects, covering instance-level context information. Although there is no explicit supervision, our method can still learn useful knowledge which is important for separating closing instances.

4.4.6 Ablations on Centroid Constrain Loss

The centroid constraint loss L_{cen} is designed for maintaining consistency for points belonging to the same instance. The loss function serves as a regularizer to constrain the embedding features from the same instance to have a small variance. Moreover, it also helps stabilize the centroids when concatenated to the embedding space. As can be inferred from Table 4.1, the utilization of L_{cen} improves the AP_{50} from 27.6 to 28.9. By further combing the geometric embeddings with the per-point features, we achieve an improvement on the AP_{50} from 28.9 to 31.5.

4.4.7 Ablations on Training and Testing Efficiency

As the first method to solve instance segmentation on the point cloud, SGPN Wang et al., 2018a needs to predict a pair-wise similarity matrix, which requires a lot of memory. Each sample requires about 2.7G for training. GSPN Yi et al., 2018 needs two training stages, and each sample has to take about 6G memory for training due to the generative network. ASIS Wang et al., 2019b addresses the problem by removing the memory consuming parts and learning a discriminative embedding. However, due to the massive usage of K-NN for every point, training ASIS requires a memory of more than 700M for every sample and the inference time for the network requires 60ms for each block. As we only utilize K-NN in the bottleneck layer, training IAM needs only about 400M for each sample and reduces the running time to 42ms for each block, showing the superiority in both the effectiveness and efficiency of our method.

4.4.8 Comparison with State-of-the-art Methods

In this section, we make a comprehensive comparison with other state-of-the-art methods on three popular benchmarks. Our method can not only be applied to indoor scenes but also achieved promising results on the hierarchical 3D part dataset. The results on S3DIS Armeni et al., 2016, ScanNetV2Dai et al., 2017a, and PartNet Mo et al., 2019 show the superiority of our method on both efficiency and effectiveness.

4.4.9 Ablations on Quantitative Results on S3DIS

Instance segmentation performance on Area-5 and k-fold cross validation results are reported in Table 4.2.

We compare our method with other start-of-the-art results. Equipped with instance-aware knowledge, 2.4% and 7.7% improvement are achieved with metric $mPrec$ and $mRec$ for instance segmentation. Although employing a simple backbone, our method surpasses previous methods, which need more complex operations and more memories for training. Moreover, we also report the performance on the semantic segmentation

TABLE 4.2. Instance segmentation results on S3DIS dataset. Both Area-5 and 6-fold performance are reported. **mCov**: mean instance-wise IoU coverage. **mWCov**: mean size-weighted IoU coverage. **mPrec**: mean precision with IoU threshold 0.5. **mRec**: mean recall with IoU threshold of 0.5. All our results are achieved on a vanilla PointNet++ Qi et al., 2017a backbone without multi-scale grouping for fair comparison.

Method	Year	mCov	mWCov	mPrec	mRec
Test on Area 5					
SGPN Wang et al., 2018a	2018	32.7	35.5	36.0	28.7
ASIS Wang et al., 2019b	2019	44.6	47.8	55.3	42.4
3D-BoNet Yang et al., 2019a	2019	-	-	57.5	40.2
Baseline	-	46.7	49.9	53.8	43.9
Ours	-	**49.9**	**53.2**	**61.3**	**48.5**
Test on 6-fold					
SGPN Wang et al., 2018a	2018	37.9	40.8	31.2	38.2
MT-PNet Pham et al., 2019	2019	-	-	24.9	-
MV-CRF Pham et al., 2019	2019	-	-	36.3	-
ASIS Wang et al., 2019b	2019	51.2	55.1	63.6	47.5
3D-BoNet Yang et al., 2019a	2019	-	-	65.6	47.6
PartNet Mo et al., 2019	2019	-	-	56.4	43.4
Ours	-	**54.5**	**58.0**	**67.2**	**51.8**

task in Table 4.3. The results are evaluated with 6-fold cross-validation. Our method is built upon vanilla PointNet++ Qi et al., 2017a and achieves better results compared with methods that applied multi-view Engelmann et al., 2017 or even graph CNN Wang et al., 2019c; Li et al., 2019. The qualitative instance grouping results are shown in Figure 3.5. We compare the performance of our method with ASIS Wang et al., 2019b, showing the effectiveness of the encoded instance-aware knowledge.

4.4.10 Quantitative Results on ScanNetV2

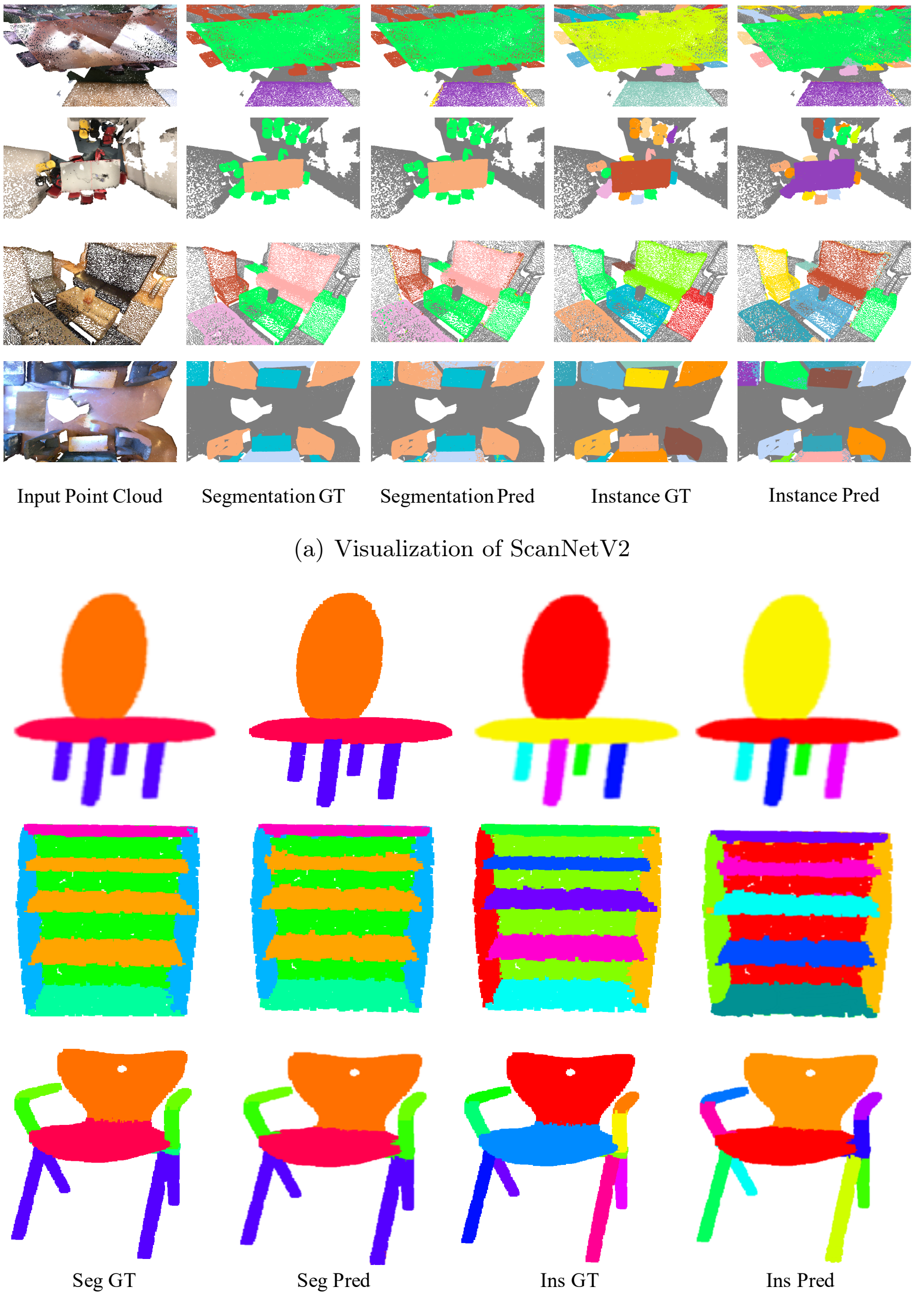
The quantitative performance of ScanNetV2 is presented in Table 4.4. It is evaluated on the validation set. Both mAP@0.25 and mAP@0.5 are reported. The results of ASIS Wang et al., 2019b and R-PointNet Yi et al., 2018 are reproduced via the open source code. For fair comparison, methods based on PointNet Qi et al., 2017b or PointNet++ Qi et al., 2017a are reported. Compared with state-of-the-art ASIS Wang et al., 2019b, our method achieves promising results and boosts mAP@0.25 and mAP@0.5 with a significant improvement, by 8.4% and 6.5%, respectively. Figure 4.6(a) shows qualitative results of instance segmentation on ScanNetV2.

TABLE 4.3. Comparison per-class performance of our proposed method with the state-of-the-art on S3DIS semantic segmentation task, tested on all areas (6-fold). Our result utilizes the vanilla PointNet++ Qi et al., 2017a without a multi-scale grouping strategy. Even with a simple baseline, the proposed method surpassed the complex graph-based methods. **mAcc:** mean pixel-wise accuracy. **mIoU:** mean category-wise IoU.

Method	mAcc	mIOU	cei.	flo.	wall	beam	col.	win.	door	tab.	cha.	sofa	boo.	boa.	clu.
PNet Qi et al., 2017b	78.5	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
MS+CU Engelmann et al., 2017	79.2	47.8	88.6	**95.8**	67.3	36.9	24.9	48.6	52.3	51.9	45.1	10.6	36.8	24.7	37.5
G+RCU Engelmann et al., 2017	81.1	49.7	90.3	92.1	67.9	44.7	24.2	52.3	51.2	58.1	47.4	6.9	39.0	30.0	41.9
PNet++ Qi et al., 2017a	-	53.2	90.2	91.7	73.1	42.7	21.2	49.7	42.3	62.7	59.0	19.6	45.8	48.2	45.6
PNei Engelmann et al., 2018	-	58.3	92.1	90.4	**78.5**	37.8	35.7	51.2	65.4	64.0	**61.6**	25.6	51.6	49.9	53.7
DGCNN Wang et al., 2019c	84.1	56.1	-	-	-	-	-	-	-	-	-	-	-	-	-
ResGCN Li et al., 2019	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**
Ours	**86.5**	**60.2**	**94.0**	94.1	76.6	**53.4**	33.6	54.2	62.7	**70.2**	60.2	**36.6**	**53.4**	**54.3**	53.5

TABLE 4.4. Instance segmentation results on ScanNetV2 benchmark (validation set). Both mAP@0.25 and mAP@0.5 are reported. All methods except Engelmann et al., 2018 are based on PointNet or PointNet++.

Method	mAP @0.25	mAP @0.5	bat	bed	shelf	cab	cha	cou	curt	desk	door	other	pict	refrig	scurt	sink	sofa	tab	toil	win
MaskRCNN He et al., 2017a	26.1	5.8	33.3	0.2	0.0	5.3	0.2	0.2	2.1	0.0	4.5	2.4	**23.8**	6.5	0.0	1.4	10.7	2.0	11.0	0.6
3D-BEVIS Elich et al., 2019	-	24.8	**66.7**	56.6	7.6	3.5	39.4	2.7	3.5	9.8	9.9	3.0	2.5	9.8	37.5	12.6	**60.4**	18.1	85.4	17.1
SGPN Wang et al., 2018a	35.1	14.3	20.8	39.0	**16.9**	6.5	27.5	2.9	6.9	0.0	8.7	4.3	1.4	2.7	0.0	11.2	35.1	16.8	43.8	13.8
R-Point Yi et al., 2018	40.0	23.5	51.3	52.3	12.5	15.2	61.8	0.0	1.5	7.6	**29.0**	11.7	14.7	**25.0**	3.7	14.0	34.5	18.1	53.0	16.1
ASIS Wang et al., 2019b	47.4	26.8	57.3	52.1	1.4	18.5	46.1	**19.2**	**20.3**	**13.3**	13.8	18.8	6.6	17.6	33.1	8.8	32.1	39.0	67.2	16.9
Ours	**50.4**	**31.5**	63.0	**60.9**	0.2	**22.9**	**67.2**	10.2	18.6	10.5	15.5	**22.7**	9.5	16.5	**55.2**	**13.6**	34.3	**41.8**	**87.3**	**17.7**



(a) Visualization of ScanNetV2

(b) Visualization of Partnet

Figure 4.6. Visualization of the instance segmentation results on (a) ScanNetV2 and (b) Partnet. Our method successfully discriminates adjacent objects that are difficult to separate. Noting: different instances are presented with different colors, and the same instance in different methods are not necessarily sharing the same color.

4.4.11 Quantitative Results on PartNet

The performance on PartNet Mo et al., 2019 is shown in Table 4.5. Different from indoor scenes, PartNet provides fine-grained and hierarchical object parts annotations. Level-1 contains the coarsest annotations and level-3 contains the finest annotations. Similar to GSPN Yi et al., 2018, we report the performance of the five categories that have the largest number of training samples: Chair, Storage, Table, Lamp, and Vase. mAP@0.5 is reported. Each category on different levels is trained separately. Our method achieved state-of-the-art results on most categories and levels, substantially improving the performance. Figure 4.6(b) shows qualitative results of instance segmentation on Partnet. Different categories and fine-grained levels are provided.

4.5 Conclusion

In this chapter, we present a novel method for solving point cloud instance segmentation and semantic segmentation simultaneously. An instance-aware module (IAM) is proposed to encode both instance-aware context and geometric information. Extensive experimental results show that our method has achieved state-of-the-art performance on several benchmarks and shown superiority in both effectiveness and efficiency.

TABLE 4.5. Instance segmentation results on PartNet. We report part-category mAP (%) under IoU threshold 0.5. There are three different levels for evaluation: coarse-grained level, middle-grained level, and fine-grained level. We select five categories with the most data amount for training and evaluation. We put short lines for the levels that are not defined.

Method	Year	Level1					Level 2					Level 3				
		Chair	StorageTable		Lamp	Vase	Chair	StorageTable		Lamp	Vase	Chair	StorageTable		Lamp	Vase
SGPN Wang et al., 2018a	2018	72.4	32.9	49.2	32.7	46.6	25.4	30.5	18.9	21.7	-	19.4	21.5	14.6	14.4	36.5
PartNet Mo et al., 2019	2019	74.4	**45.2**	54.2	**37.2**	49.8	35.5	35.0	31.0	26.9	-	29.0	27.5	23.9	18.7	52.0
GSPN Yi et al., 2018	2019	-	-	-	-	-	-	-	-	-	-	26.8	26.7	21.9	18.3	-
ASIS Yi et al., 2018	2019	77.1	43.2	55.0	34.1	48.5	36.0	35.5	31.3	24.8	-	26.8	26.7	21.9	18.3	51.9
Ours	-	**79.5**	44.2	**56.1**	36.1	**49.9**	**38.6**	**37.1**	**33.0**	**26.9**	-	**31.2**	**28.9**	**25.5**	**19.4**	**53.1**

Chapter 5

Memorizing Representative Prototypes for 3D Point Cloud Instance Segmentation

5.1 Introduction

Based on the pioneering works of PointNet Qi et al., 2017b and PointNet++ Qi et al., 2017a, directly processing point sets becomes simpler, more memory-efficient, and flexible than handling the volumetric grids with 3D convolution Hou, Dai, and Nießner, 2019; Wu et al., 2015; Maturana and Scherer, 2015. Some following approaches Wang et al., 2018a; Wang et al., 2019b; Yang et al., 2019a; Yi et al., 2018 propose to handle semantic and instance segmentation in an end-to-end network jointly for the fine-grained description of the observation. Specifically, discriminative instance embeddings are learned to measure the instance-level clustering patterns of the points Wang et al., 2019b; Pham et al., 2019.

Although existing methods have achieved some impressive results, we still can observe performance bottlenecks on different datasets Armeni et al., 2016; Dai et al., 2017a, especially on the non-dominant classes with fewer samples (see Figure 5.5). It has been proved that deep networks tend to *forget* the non-dominant rare cases easily while learning on a dataset distributed off balance and diversely Toneva et al., 2018. On point cloud data, imbalance issue usually appears as the *category imbal-*

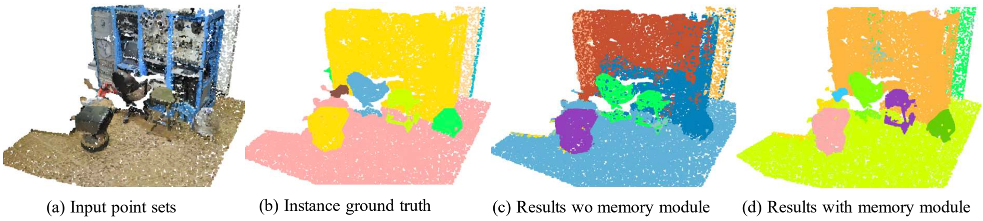

(a) Input point sets (b) Instance ground truth (c) Results wo memory module (d) Results with memory module

FIGURE 5.1. Comparison of instance segmentation results. The performance of our proposed method shows strong robustness against non-dominant cases.

ance and *pattern imbalance*, which is severer than that on 2D images Yang et al., 2019a. Defining and measuring the category imbalance is easier, which appears as a significant discrepancy among the proportions of different categories. For example, in an indoor scene (as shown in Figure 5.1), the proportions of the points belonging to the background (e.g., , wall) are much higher than the objects (e.g., , chairs). In S3DIS Armeni et al., 2016, the total amount of ceiling points is 50 times larger than the chair. The pattern imbalance can be observed on the (non-dominant) rare cases may appearing both dominant and non-dominant categories, which are usually in the minority of the datasets. It is often caused by complex geometric factors, such as positions, shapes, and relative relationships. For example, chairs are usually placed near a desk, while may occasionally appear with arbitrary positions (e.g., stacking and back-to-back near the cabinet) in an office, as shown in Figure 5.1. Conventional methods Yang et al., 2019a ignore this issue or simply resort to the focal loss Lin et al., 2017c, by down weighing the well-learned samples during training. However, it is hard to find a balance between the dominant and non-dominant samples in the dynamic training process.

To address the above issues, we propose to learn and **m**emorize the discriminative and representative **p**rototypes covering all the samples, which is implemented as a memory-augmented network, referred to as **MP**Net. The proposed MPNet includes two branches for predicting point-level semantic labels and obtaining per-point embedding for instance grouping, respectively. As shown in Figure 5.2, the two branches access the shared memory via two separate memory readers, which associates the two tasks via the shared memory.

Given an input, MPNet retrieves the most relevant items in the memory for the extracted per-point features and feeds only retrieved embedding to the following segmentation tasks. In MPNet, the memory is maintained as a compact dictionary shared by diverse points. Driven by the task-specific training objectives and the proposed geometry-aware regularization, the compact memory is pushed to record the prototypes encoding the geometric and semantic information that is the most representative for all samples. During training, the memory slots are dynamically associated with both the dominant (common) and non-dominant (rare) categories (and cases) seen in mini-batches, alleviating the example forgetting issue Toneva et al., 2018. In testing, the distorted observations and rare cases can thus be augmented by retrieving the stored prototypes, leading to better robustness and generalization.

Additionally, different from previous methods relying on either pairwise relations computation Wang et al., 2018a or KNN based feature aggregation Wang et al., 2019b, the proposed MPNet is free from complex and time-consuming operations, which is more efficient.

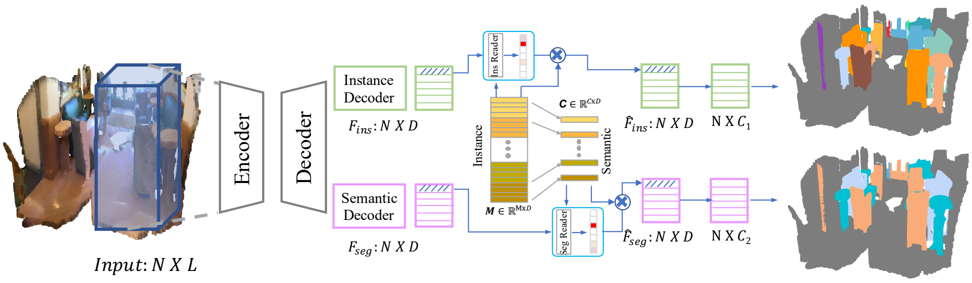


Figure 5.2. The framework of our proposed MPNet, which contains two parallel branches with a shared encoder. A memory module is proposed to memorize representative prototypes that are shared by all samples. The maintained memory module is shared with all instances across different categories. Both distorted and rare cases can be augmented by retrieving the stored prototypes.

5.2 Background

Memory Networks Memory-based approaches have been discussed for solving various problems. NTM Graves, Wayne, and Danihelk, 2014 is proposed to improve the generalization ability of the network by introducing an attention-based memory module. Gong *et al.*Gong et al., 2019 proposed a memory augmented auto-encoder for anomaly detection, which is detected by represented the input with prototypical elements of the normal data maintained in a memory module. However, the memory model in Gong et al., 2019 only includes a single memory pool in autoencoder for unsupervised representation, which may not work for the other tasks. Prototypical Network He et al., 2017b maintains category-wise templates for the problem of few-shot classification. Liu Liu et al., 2019b proposed an OLTR algorithm to solve the open-ended and long-tail problem by associating a memory feature that can be transferred to both head and tail classes adaptively. These two methods are designed for the task of classification and are not suitable for the task of instance segmentation.

5.3 Proposed Method

5.3.1 Overview of the Proposed MPNet

We propose a memory-augmented network for joint point cloud semantic and instance segmentation, which learns and memorizes the prototypes of the point sets to alleviate the influence of the imbalanced distribution of the data. As shown in Figure 5.2, the proposed memory-augmented network (i.e., MPNet) adopts an encoder-decoder architecture, which is free from the specific design of the encoder and decoder. In the proposed MPNet, we use PointNet++ Qi et al., 2017a to implement the encoder for per-point feature extraction. Two parallel decoders for instance segmentation and semantic segmentation are built upon the shared encoder. The memory is implemented as a dictionary to record the discriminative and representative prototypes as bases, which are optimized driven by the task-specific objective and the proposed instance-aware geometric regularization.

Given a set of input points $\{\mathbf{p}_i\}_{i=1}^{N}$ with $\mathbf{p}_i \in \mathbb{R}^L$, we can formulate the input of the network as a matrix $\mathbf{P} \in \mathbb{R}^{N\times L}$, where L denotes the input feature dimension and N denotes the total number of input points. The input features of each point may consist of both geometry and appearance information, i.e., 3D coordinate (x, y, z) and RGB values. The two decoder branches produce features $\mathbf{F}_{\text{seg}} \in \mathbb{R}^{N\times D}$ and $\mathbf{F}_{\text{ins}} \in \mathbb{R}^{N\times D}$, respectively, where D denotes the dimension of the features. Instead of directly using $\mathbf{F}_{\text{seg}}$ and $\mathbf{F}_{\text{ins}}$ to perform semantic and instance segmentation tasks, respectively, MPNet applies them as queries to retrieve the most relevant prototypes in the memory and then obtains features $\widehat{\mathbf{F}}_{\text{seg}}$ and $\widehat{\mathbf{F}}_{\text{ins}}$, which are delivered to the following semantic classifier and instance embedding module. The memory is randomly initialized and optimized during training. The two branches access the memory with specifically designed reading heads.

5.3.2 Memory Representation for Prototypes

The *prototype memory* is designed as a matrix $\mathbf{M} \in \mathbb{R}^{M\times D}$, where M is a hyper-parameter that defines the number of memory slots and D is the feature dimension that is identical with the outputs from the two branches. The M memory slots are trainable parameters and are used to restore the prototypes shared by all instances across all categories. To easily represent the semantic characteristics, we define a *semantic memory* $\mathbf{C} \in \mathbb{R}^{C\times D}$, where C denotes the number of categories in semantic segmentation task and each row of $\mathbf{C}$ represents the summary of a class. $\mathbf{C}$ can be seen as the semantic summary of $\mathbf{M}$ and are generated from $\mathbf{M}$. We equally associate the M memory slots in $\mathbf{M}$ with C categories and thus define $M = M_c \times C$, where M_c is the number of per-category prototypes. As shown in Figure 5.2, the i-th row in $\mathbf{C}$ is defined as the average of the i-th subsegment (i.e., , rows from $(i-1)\times M_c + 1$ to $i \times M_c$) in $\mathbf{M}$:

$$\mathbf{c}_i = \frac{1}{M_c}\sum\nolimits_{j=(i-1)\times M_c+1}^{i\times M_c} \mathbf{m}_j, \tag{5.1}$$

where $\mathbf{m}_j$ denotes the j-th row vector of $\mathbf{M}$, which is trainable parameters.

Given the query features $\mathbf{F}_{\text{ins}}$ and $\mathbf{F}_{\text{seg}}$, the instance grouping branch directly accesses the prototypes memory $\mathbf{M}$ and the semantic labeling branch accesses the semantic summary $\mathbf{C}$, with two specifically designed readers. $\mathbf{M}$ can be seen as a dictionary to restore the representative bases shared by all instances, as the instances cross different categories can share some common basic components and characteristics. Because the semantic memory $\mathbf{C}$ is a re-parameterization of $\mathbf{M}$, the two tasks are naturally associated together, without computation-consuming operations as Wang et al., 2019b. The memorized prototypes are discriminative for both tasks.

5.3.3 Memory-augmented Instance Embedding

5.3.3.1 Reading memory for instance embedding.

Given $\mathbf{F}_{\text{ins}}$, the proposed MPNet reads the most relevant items from $\mathbf{M}$ to obtain instance embedding for instance grouping. For each per-point feature $\mathbf{f}_{\text{ins},i}$ (i.e., , the i-th row of $\mathbf{F}_{\text{ins}}$), we calculate the memory addressing weights $\mathbf{w}_i \in \mathbb{R}^M$ according to the similarity between $\mathbf{f}_{\text{ins},i}$ and the prototypes stored in memory $\mathbf{M}$:

$$w_{ij} = \frac{\exp(d(\mathbf{f}_{\text{ins},i}, \mathbf{m}_j))}{\sum_{j=1}^{M} \exp(d(\mathbf{f}_{\text{ins},i}, \mathbf{m}_j))}, \tag{5.2}$$

where w_{ij} denotes the j-th element of $\mathbf{w}_i$, $\mathbf{m}_j$ is the j-th row in $\mathbf{M}$, and $d(\cdot,\cdot)$ denotes the similarity measurement function. We use cosine similarity as $d(\cdot,\cdot)$ in this work. $\mathbf{w}_i$ can also been seen as a soft-attention weight vector indicating the relevance of each memory item to the query $\mathbf{f}_{\text{ins},i}$. With $\mathbf{w}_i$, we calibrate $\mathbf{f}_{\text{ins},i}$ with the memory $\mathbf{M}$ and obtain the augmented feature as $\widehat{\mathbf{f}}_{\text{ins},i} = \sum_{j=1}^{M} w_{ij}\mathbf{m}_j$.

5.3.3.2 Instance-aware geometric regularization

Different from previous memory-based representation methods, which are designed for either classification Liu et al., 2019b or unsupervised tasks Gong et al., 2019, we propose an instance-aware geometric regularization loss tailored for instance segmentation in the point cloud, in the hope that the prototypes in the memory module can encode informative geometric information. To achieve this, we force the memory-augmented features from the same instance to have identical geometric predictions.

We first introduce an instance centroids estimator $G(\cdot)$ that will be trained to predict the instance centroids based on the augmented features as $G(\widehat{\mathbf{f}}_{\text{ins},n})$ and try to enforce the predicted centroids to be grouped around the corresponding geometric centers of the instances. The instance-aware regularization loss R_{ins} is defined as:

$$R_{\text{ins}} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{n=1}^{N_k}\|G(\widehat{\mathbf{f}}_{\text{ins},n}) - GT_k\|^2, \tag{5.3}$$

where K is the instance number, N_k is the number of the points of k-th instance, and GT_k denotes the ground truth centroid of the k-th instance. $\widehat{\mathbf{f}}_{\text{ins},n}$ denotes the augmented feature of a point belonging to the k-th instance. $G(\cdot)$ is implemented as an MLP and can be trained in an end-to-end manner.

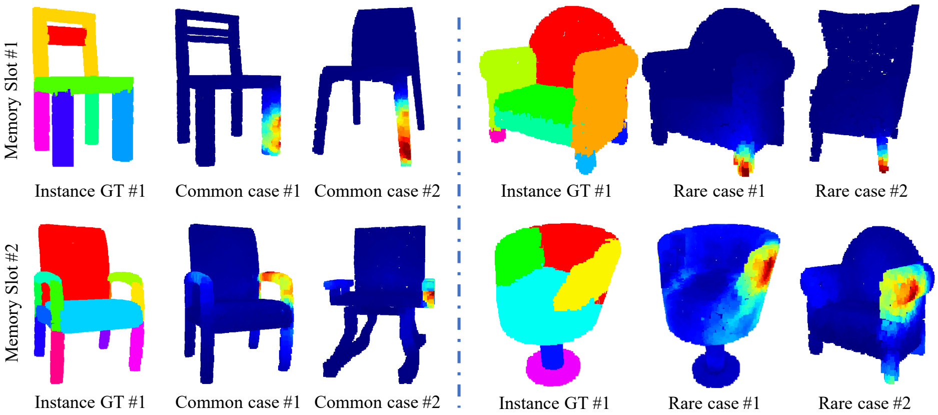


Figure 5.3. Visualization of the memory slots in $\mathbf{M}$. We visualize what the memory has learned with the instance segmentation on PartNet dataset Mo et al., 2019, in which the parts (e.g., , the chair legs) of the object are treated as instances. For a specific memory slot (i.e., , slot #1 and #2 in the figure), we visualize the addressing weights of the points from common and rare cases in pseudo color. The correlation between a specific memory slot and the "visual concepts" (e.g., , the components type and relative position) of the most related points are consistent across diverse examples, including common and rare cases, which implies the memory captures meaningful and interpretable semantic and geometric prototypes.

5.3.3.3 What are learnt and stored in memory

To have a clear understanding of the learned memory prototypes, we select the category of 'Chair' in PartNet Mo et al., 2019 for training and visualization due to its largest number of training samples, as shown in Figure 5.3. For each memory item, the points that are addressing it have consistent semantic meaning, implying the capability of the memory module to capture the discriminative and unified representation for both dominant and rare cases.

5.3.4 Memory-augmented Semantic Labeling

5.3.4.1 Reading memory for semantic segmentation

Similar to the instance grouping branch, the semantic branch read the embedding from semantic memory $\mathbf{C}$ for classification. For each $\mathbf{f}_{\text{seg},i}$ from $\mathbf{F}_{\text{seg}}$, we obtain the soft memory addressing weights $\boldsymbol{\gamma}_i \in \mathbb{R}^C$ by calculating the similarity between $\mathbf{f}_{\text{seg},i}$ and each $\mathbf{c}_j$ (i.e., , each row of $\mathbf{C}$), similar to Eq. (5.2). Then we can obtain $\widehat{\mathbf{F}}_{\text{seg}}$ through $\widehat{\mathbf{f}}_{\text{seg},i} = \boldsymbol{\gamma}_i^{\mathsf{T}}\mathbf{C} = \sum_{j=1}^{C} \gamma_{ij}\mathbf{c}_j$, where γ_{ij} denotes the j-th item in $\boldsymbol{\gamma}_i$.

5.3.4.2 Semantic memory regularization

We apply an additional regularization term on the semantic memory to enforce the centroids of different categories (i.e., , the semantic summarization $\mathbf{c}_i$'s) separately distributed. Specifically, a large margin loss Liu et al., 2019b is used to encourage the embedding close to its category centroid in memory and far away from others. Given the feature embedding read from memory $\widehat{\mathbf{f}}_{\text{seg},i}$ and its class label y_i, the regularization term R_{seg} is calculated as:

$$R_{\text{seg}} = \max(0, \sum_{j=y_i} \|\widehat{\mathbf{f}}_{\text{seg},i} - \mathbf{c}_j\| - \sum_{j\neq y_i} \|\widehat{\mathbf{f}}_{\text{seg},i} - \mathbf{c}_j\| + m), \tag{5.4}$$

where m is the margin, which is set as 5 in our implementation. Each $\mathbf{c}_j$ performs like an anchor point and pulls the features with identical semantic labels close to it and pushes the features with different semantic labels away from it.

5.3.5 Loss Functions

5.3.5.1 Classification loss

We use the traditional cross-entropy loss L_{cd} for the semantic segmentation task.

5.3.5.2 Instance discriminative loss

Given the per-point memory augmented features $\{\widehat{\mathbf{f}}_{\text{ins},i}\}_{i=1}^{N}$, point-level embeddings $\{\mathbf{g}_{\text{ins},i} \in \mathbb{R}^{c'}\}_{i=1}^{N}$ are generated by a simple MLP layer, where c' is the dimension of the embedding features. Similar to Brabandere, Neven, and Gool, 2017; Wang et al., 2019b, we set $c' = 5$ and use the instance discriminative loss for instance grouping embedding. For instance grouping, embeddings from the same instance shoule be forced to group together. A soft margin σ_v is introduced to allow these embeddings distributing on a local manifold rather than having to converge to a single point. Moreover, instance embedding centers are no longer repulsed if their distances are larger than $2\sigma_d$. The instance discriminative loss is formulated as:

$$L_{\text{dis}} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{n=1}^{N_k}[\|\mathbf{g}_{\text{ins},n} - \boldsymbol{\mu}_k\| - \sigma_v]_+^2 + \frac{1}{K(K-1)}\sum_{\substack{i=1\\ i\neq j}}^{K}\sum_{j=1}^{K}[2\sigma_d - \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|]_+^2\,, \tag{5.5}$$

where K is the total instance number, and N_K is the point number of the k-th instance. $\boldsymbol{\mu}_k$ is the average embedding of the k-th instance, which is calculated by $\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N_k}\mathbf{g}_{\text{ins},n}$. σ_v and σ_d in Eq. (5.5) are the margins. During testing, a simple mean shift clustering algorithm is adopted to group the points in the embedding space.

5.3.5.3 Training objective

As all operations are differentiable, memory can be updated through back-propagation in an end-to-end manner. By combining the four losses discussed above, the training objective is formulated as:

$$L = L_{\text{ce}} + L_{\text{dis}} + R_{\text{seg}} + \lambda R_{\text{ins}}, \tag{5.6}$$

where the loss weight λ is set to 0.1 to avoid large gradients. Moreover, as $\mathbf{C}$ is a re-parameterization of $\mathbf{M}$, the supervisions jointly update $\mathbf{M}$ and then influence the two tasks in turn. The two tasks are thus naturally associated together, free from the complex and time-consuming operation, as introduced in Wang et al., 2019b.

5.4 Experiments

To validate the effectiveness of our proposed method, both qualitative and quantitative experiments are conducted on two public datasets: Stanford 3D Indoor Semantic Dataset (S3DIS) Armeni et al., 2016 and ScanNetV2 Dai et al., 2017a.

5.4.1 Datasets

S3DIS dataset Armeni et al., 2016 is collected in 6 large-scale indoor areas, including 13 classes. ScanNetV2 is another large-scale dataset for point cloud instance segmentation, which consists of 1613 indoor scans from 40 categories. The dataset is split into 1201, 312, and 100 for training, validating, and testing, respectively.

5.4.2 Evaluation

Following Wang et al., 2019b on the S3DIS dataset, the results on Area-5 and 6-fold cross-validation are reported in our experiments. For semantic segmentation, we present 1) the overall accuracy (oAcc), which measures the point-level accuracy, 2) the mean class accuracy (mAcc), which calculates the average category-level accuracy, and 3) the instance-wise mean intersection-over-union (mIoU). For instance segmentation, four evaluation metrics are calculated, namely, $mConv$, $mWConv$, $mPrec$, and $mRec$. $mConv$ is defined as the mean instance-wise matching IoU score between the ground truth and the prediction. Instead of treating every instance equally, $mWConv$ is calculated by weighting the size of each instance object. Moreover, traditional $mPrec$ and $mRec$ represent mean precision and recall with the IoU threshold of 0.5, respectively.

5.4.3 Implementation Details

For the datasets of S3DIS and ScanNetV2, each room is divided into $1m \times 1m$ blocks with a stride of $0.5m$. 4096 points are randomly sampled from each block during the training process. Without special notation, all experiments are conducted using vanilla PointNet++ Qi et al., 2017a as the backbone (without introducing any multi-scale grouping operation). We utilize the same setting as ASIS Wang et al., 2019b for training. The whole network is trained in an end-to-end manner for 100 epochs in total. During the inference time, blocks within each room are merged by utilizing the semantic and instance results of the overlapped region. Detailed settings of the algorithm are identical with Wang et al., 2018a.

5.4.4 Ablation Study

In this section, we study the influence of each integration of the aforementioned components. All the results are tested on S3DIS Area-5 for a fair comparison. We first build a strong baseline which is equivalent to the vanilla ASIS Wang et al., 2019b. Building upon the strong baseline, our MPNet surpasses it by a large margin via memorizing representative prototypes. In the following, we provide detailed analyses from different aspects.

5.4.4.1 Memory M and C

The representative and consistent prototypes are maintained in the prototypes memory $\mathbf{M}$, which is universal to represent the shared concepts of all instances. Besides, the semantic memory $\mathbf{C}$ is served as a semantic summary to represent the category characteristics efficiently. As shown in Table 5.1, using instance memory $\mathbf{M}$ alone can boost $mPre$ from 52.3% to 58.9% and $mRec$ from 41.4% to 47.0%. On the other hand, using the semantic memory $\mathbf{C}$ can bring another 1.3% and 0.4% improvements with the metrics of $mPrec$ and $oAcc$, respectively.

5.4.4.2 Visualizing the Effects of Memory on Instance Embedding

In Figure 5.4, we directly visualize the instance embedding $\mathbf{g}_{\text{ins},i}$ to show the positive effects of the memory, which cover both the common and rare scenes, i.e., office and lobby.

5.4.4.3 Memory Size

We study the influence of the memory size, i.e., , the hyper-parameter M or N_c equivalently, to the final performance. We evaluate three settings of N_c with 100, 150,

TABLE 5.1. Ablation study on S3DIS dataset with vanilla Pointnet++ as the backbone. **FL** refers to focal loss. **InsMem** indicates that the memory is updated by the instance information. **SegMem** means the memory is updated by the supervision from semantic segmentation. **Regul** refers to the regularizations used in learning the prototypes memory. Both instances and semantic segmentation results are provided.

Method	FL	InsMem	SegMem	Regul	mPre	mRec	oAcc
Baseline					52.3	41.4	86.2
	✓				55.2	43.0	86.9
		✓			58.9	47.0	87.7
		✓	✓		60.2	47.2	88.1
Ours		✓	✓	✓	62.5	49.0	88.2

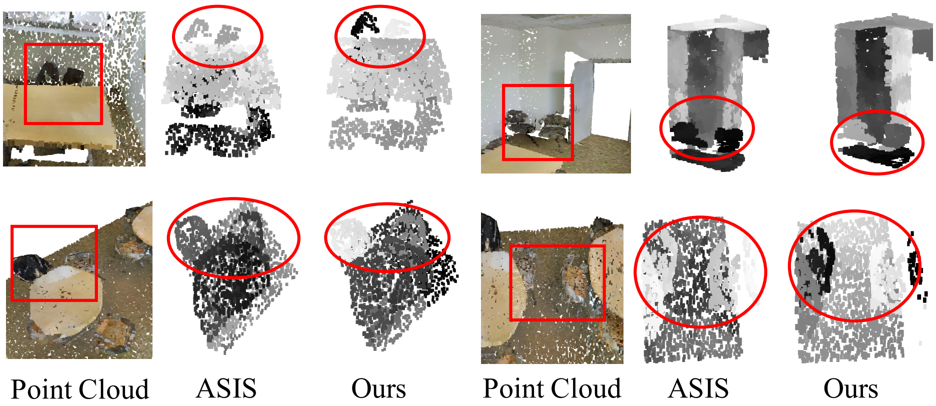


FIGURE 5.4. Barnes-Hut t-SNE Maaten, 2014 visualization of the instance embedding on S3DIS Area-5 set (Best viewed in color and zoomed in).

200 as the number of per-category prototypes. The *mPrec* on S3DIS Area-5 are 60.4, 62.7 and 62.5 respectively. The results show that the performance increases as the N_c grows, and becomes stable after when N_c is greater than 200. In all our experiments, N_c is set to 150.

5.4.4.4 Regularization Loss

To effectively learn representative and discriminative prototypes, regularization losses are proposed in Eq. (5.3) and Eq. (5.4), which directly work on the memory-augmented features for instance segmentation and semantic segmentation, respectively. The first one is designed for forcing the calibrated instance features to have identical geometric outputs. The second one is to keep a compact intra-class representation and a large margin between different categories. Both of them can be beneficial for both semantic and instance segmentation due to the mutual influence on the memory module. As shown in Table 5.1, the two regularization terms boost the $mPre$ and $mRec$ for about 2.3% and 1.8%, respectively.

5.4.4.5 Comparing with Focal Loss Lin et al., 2017c

The discrepancies among different categories are significant in the 3D point cloud. Focal loss Lin et al., 2017c has been widely used in different kinds of vision tasks due to the imbalanced distribution of the training data. It addresses the problem by down-weighting the well-classified samples. However, it only alleviates the category imbalance to some extent and fails to solve the diversely distributed patterns and cases. As shown in Table 5.1, the focal loss can only improve the $mPre$ and $mRec$ by 2.9% and 1.6%, respectively. Compared with the Focal Loss, our method surpasses the baseline model by a large margin, due to the memorized prototypical patterns and improves $mPre$ and $mRec$ by 10.2% and 7.6%, respectively.

5.4.5 Analysis on the Non-dominant Categories and Rare Cases

We study the instance segmentation performance gain brought by the proposed memory network specifically on the non-dominant categories and rare cases.

5.4.5.1 Analysis on Non-dominant (Rare) Categories

We compare the performance of our proposed MPNet with ASIS Wang et al., 2019b on non-dominant classes. We first sort the 13 categories on S3DIS according to their proportions in the training set and split the dataset into three levels: dominant classes (the first 4 classes), mid-dominant classes (the mid 5 classes), and non-dominant classes (the last 4 classes). The amount proportions of the three levels are 79.17%, 16.95%, and 3.88%, respectively. As shown in Figure 5.5, we report the improvements with two metrics $mPrec$ and $mRec$. Our method not only boosts the overall performances but also brings much more significant improvements to the non-dominant classes than the focal loss Lin et al., 2017c and ASIS Wang et al., 2019b.

In Figure 5.7, we plot the changes of the $mPrec$ scores of the model with or without the memory module during training. The results on both common category ("wall") and uncommon category ("sofa") from S3DIS are shown. With the proposed memory module, our method has the ability to alleviate the forgetting issue on the non-dominant samples.

5.4.5.2 Analysis on Rare Cases

Analyzing with the rare cases is not as easy as on the rare classes since it is not easy to define. We maintain a set of rare cases from "Area-5" in S3DIS Armeni et al.,

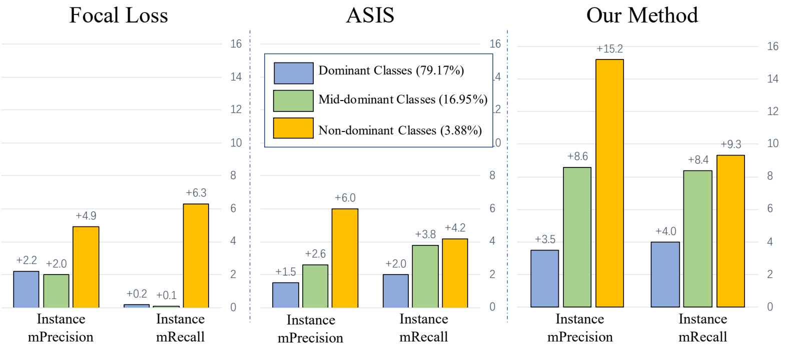


FIGURE 5.5. The comparison of the improvements on both common and uncommon categories. We compare the performance of *mPrec* and *mRec* with Focal Loss Lin et al., 2017c and ASIS Wang et al., 2019b.

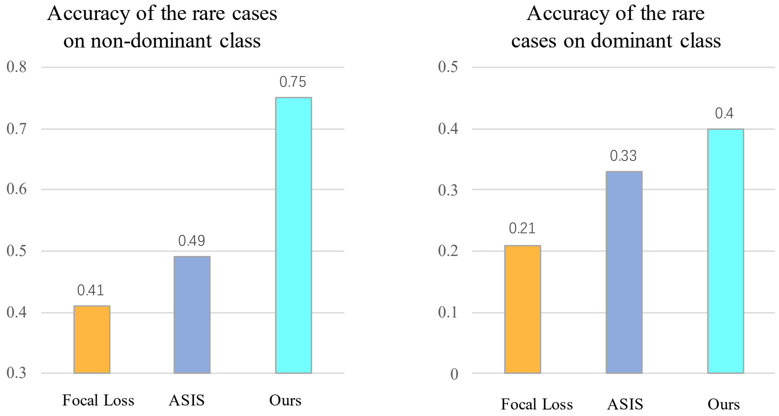


FIGURE 5.6. The instance precision of the rare cases. Both common and uncommon categories are presented. The rare instances are collected as the 20% hardest samples from the baseline model.

2016 by using the performance of the baseline model as the criterion. Specifically, we evaluate the instance-wise IoU score of vanilla ASIS Wang et al., 2019b and collect 20% of the instances with the lowest scores as the rare cases for further studies. in Figure 5.6, we show the performance of different methods on the rare cases from both a non-dominant class ("sofa") and a dominant class ("wall"). As shown in the figure, the proposed method is more effective to handle the rare cases. It brings much more improvements than other methods, especially on the rare cases from the non-dominant class, which has more diverse patterns.

5.4.6 Comparison with the State-of-the-art Methods

5.4.6.1 Performance on S3DIS

We first compare the instance segmentation performance on both Area-5 and 6-fold. The results are presented in Table 5.2. Our proposed MPNet achieves promising results and surpasses the previous state-of-the-art approaches substantially by a large

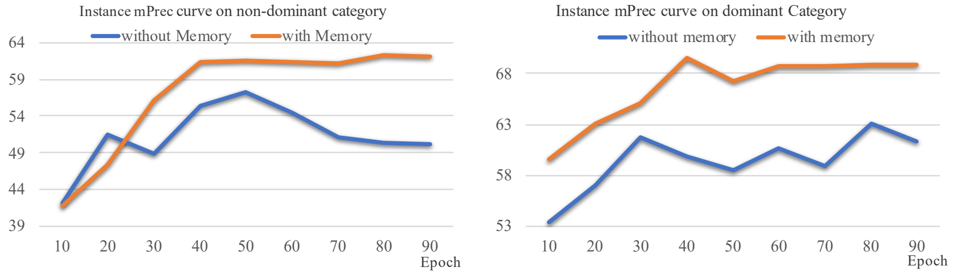


FIGURE 5.7. The training curve on both dominant ("wall") and non-dominant categories ("sofa"). The forgetting issue can be alleviated when associated with our proposed representative memory slots.

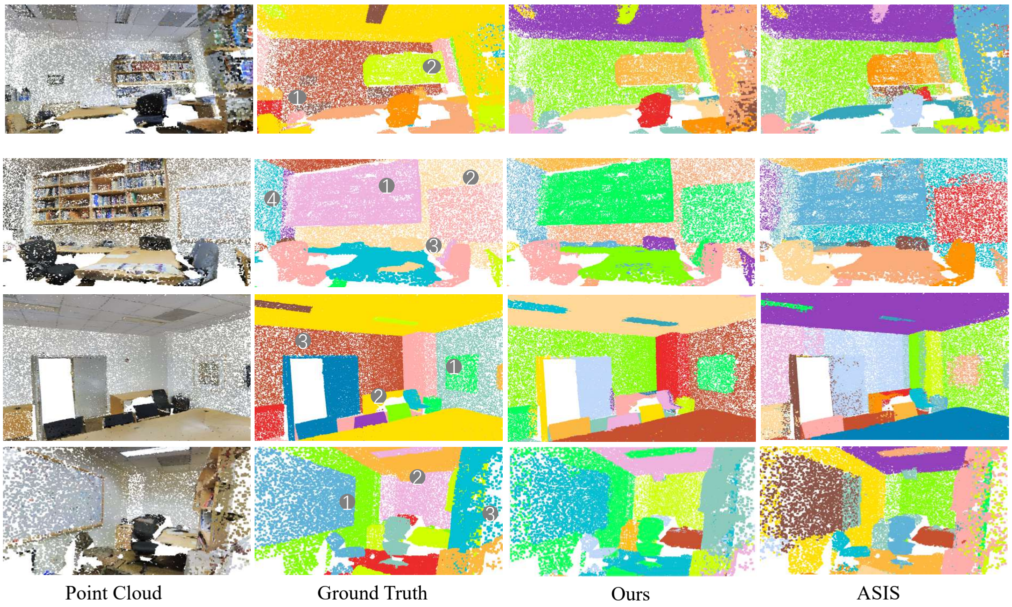


FIGURE 5.8. **Qualitative results of our method on S3DIS dataset.** From left to right are: input point cloud, instance segmentation ground truth, the results of our method and the results of Wang et al., 2019b.

margin. The large improvements are mainly beneficial from the strong ability of the proposed prototypes memory module. Qualitative results are shown in Figure 5.8.

Table 5.2. Instance Segmentation results on S3DIS dataset. Both Area-5 and 6-fold results are reported. All our results are achieved based on a vanilla PointNet++ backbone (without multi-scale grouping) for a fair comparison.

Method	Year	mCov	mWCov	mPrec	mRec
Test on Area 5					
SGPN Wang et al., 2018a	2018	32.7	35.5	36.0	28.7
ASIS Wang et al., 2019b	2019	44.6	47.8	55.3	42.4
3D-BoNet Yang et al., 2019a	2019	-	-	57.5	40.2
Ours	-	**50.1**	**53.2**	**62.5**	**49.0**
Test on 6-fold					
SGPN Wang et al., 2018a	2018	37.9	40.8	31.2	38.2
MT-PNet Pham et al., 2019	2019	-	-	24.9	-
MV-CRF Pham et al., 2019	2019	-	-	36.3	-
ASIS Wang et al., 2019b	2019	51.2	55.1	63.6	47.5
3D-BoNet Yang et al., 2019a	2019	-	-	65.6	47.6
PartNet Mo et al., 2019	2019	-	-	56.4	43.4
Ours	-	**55.8**	**59.7**	**68.4**	**53.7**

In addition to instance segmentation, we also report the results of semantic segmentation and compare them with other methods. The performance is tested on all areas (6-fold), as shown in Table 5.3. Although based on a simple PointNet++ backbone, we achieve better results than the other methods which are based on graph neural networks Li et al., 2019; Wang et al., 2019c.

5.4.6.2 Performance on ScanNetV2

In addition to S3DIS, we conduct experiments on ScanNetV2 Dai et al., 2017a. The instance segmentation results are reported in Table 5.4, which is tested on the validation set. To make a fair comparison, we select the methods that are based on PointNet or PointNet++. Our proposed MPNet outperforms previous methods and dominants in many categories.

5.4.6.3 Performance on PartNet

In Figure 5.3, to better understand the learned memory prototypes, we do visualization relying on the category of “Chair” in PartNet Mo et al., 2019. PartNet Mo et al., 2019 is a consistent dataset of 3D objects with fine-grained and hierarchical 3D part annotations. We report the quantitative results in Table 5.5. **Level-1** refers to the

Table 5.3. Comparison per-class performance of our proposed method with the state-of-the-art methods on the task of semantic segmentation on S3DIS. We use vanilla pointnet++ Qi et al., 2017a without multi-scale grouping. Even with a simple backbone, the proposed method surpasses the graph-based method by more than 1% mIOU (reported with 6-fold cross-validation).

Method	OA	miou	ceil	floor	wall	beam	colu	wind	door	table	chair	sofa	book	boar	clut
PNet Qi et al., 2017b	78.5	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
MS+CU Engelmann et al., 2017	79.2	47.8	88.6	**95.8**	67.3	36.9	24.9	48.6	52.3	51.9	45.1	10.6	36.8	24.7	37.5
G+RCU Engelmann et al., 2017	81.1	49.7	90.3	92.1	67.9	44.7	24.2	52.3	51.2	58.1	47.4	6.9	39.0	30.0	41.9
PNet++ Qi et al., 2017a	-	53.2	90.2	91.7	73.1	42.7	21.2	49.7	42.3	62.7	59.0	19.6	45.8	48.2	45.6
PNeigh Engelmann et al., 2018	-	58.3	92.1	90.4	**78.5**	37.8	35.7	51.2	65.4	64.0	61.6	25.6	51.6	49.9	53.7
DGCNN Wang et al., 2019c	84.1	56.1	-	-	-	-	-	-	-	-	-	-	-	-	-
ResGCN Li et al., 2019	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**
Ours PNet++	**86.8**	**61.3**	**94.0**	94.1	76.6	**53.4**	33.6	54.2	62.7	**70.2**	60.2	**36.6**	53.4	**54.3**	53.5

TABLE 5.4. Instance segmentation results on ScannetV2 benchmark (validation set). The results of mAP@0.25 and mAP@0.5 are reported. All methods except Engelmann et al., 2018 are based on PointNet or PointNet++ (3D-BEVIS Elich et al., 2019 is multi-view based method).

Method	mAP	mAP	bat.	bed	she.	cab.	cha.	cou.	cur.	des	doo	oth.	pic.	ref.	sho.	sin	sof	tab.	toi.	win.
	@0.25	@0.5																		
MRCNNHe et al., 2017a	26.1	5.8	33.3	0.2	0.0	5.3	0.2	0.2	2.1	0.0	4.5	2.4	**23.8**	6.5	0.0	1.4	10.7	2.0	11.0	0.6
SGPNWang et al., 2018a	35.1	14.3	20.8	39.0	**16.9**	6.5	27.5	2.9	6.9	0.0	8.7	4.3	1.4	2.7	0.0	11.2	35.1	16.8	43.8	13.8
3DBEVISElich et al., 2019	-	24.8	66.7	56.6	7.6	3.5	39.4	2.7	3.5	9.8	9.9	3.0	2.5	9.8	37.5	12.6	**60.4**	18.1	85.4	17.1
GSPNYi et al., 2018	40.0	23.5	51.3	52.3	12.5	15.2	61.8	0.0	1.5	7.6	**29.0**	11.7	14.7	**25.0**	3.7	**14.0**	34.5	18.1	53.0	16.1
ASISWang et al., 2019b	41.5	24.0	29.9	50.5	0.0	16.7	57.7	0.0	**18.4**	7.8	14.8	12.9	1.8	12.4	38.0	10.2	36.9	37.4	71.7	14.5
ours	**49.3**	**31.0**	**69.4**	**59.8**	2.7	**23.7**	**71.1**	**4.5**	8.4	**18.3**	11.6	**17.3**	4.8	21.8	**57.0**	13.4	27.7	**41.8**	**87.3**	**18.3**

coarsest annotation and **Level-3** refers to the most fine-grained annotation as defined in Mo et al., 2019.

Table 5.5. Comparison of the per-level performance of our method with the state-of-the-art methods on "Chair" category in PartNet Mo et al., 2019. The performance is evaluated using part-category mAP, with IoU threshold of 0.5. All the results are achieved with the same backbone: PointNet++ Qi et al., 2017a.

Method	Year	**Level-1**	**Level-2**	**Level-3**
SGPN Wang et al., 2018a	2019	72.4	25.4	19.4
PartNet Mo et al., 2019	2019	74.4	35.5	29.0
GSPN Yi et al., 2018	2019	-	-	26.8
Ours	-	**79.9**	**41.2**	**32.5**

For a fair comparison, all results are evaluated with the same backbone PointNet++ Qi et al., 2017a. Our method outperforms the previous methods by a large margin, showing the flexibility of our method to handle various types of input data. Moreover, visualization examples of the results are shown in Figure 5.10 and Figure 5.9, indicating that our method can handle both rare and common cases well.

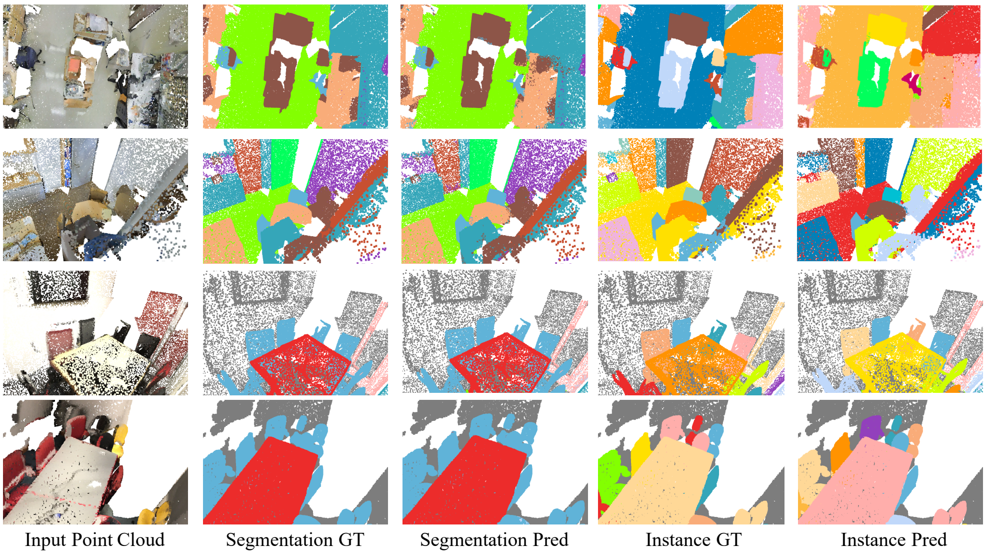


Figure 5.9. Visualization of the performance of on ScanNet.

5.4.6.4 Speed Analysis.

We compare the inference speed with the other two methods: SGPN Wang et al., 2018a and ASIS Wang et al., 2019b, as shown in Table 5.6. The whole evaluation process includes two parts: the network forward and instance grouping. The first part is to get per-point semantic labeling and instance embedding. The second part utilizes a grouping algorithm to find out instance groups. SGPN, which is based on PointNet, predicts a pair-wise affinity matrix to group points into instance clusters, requiring

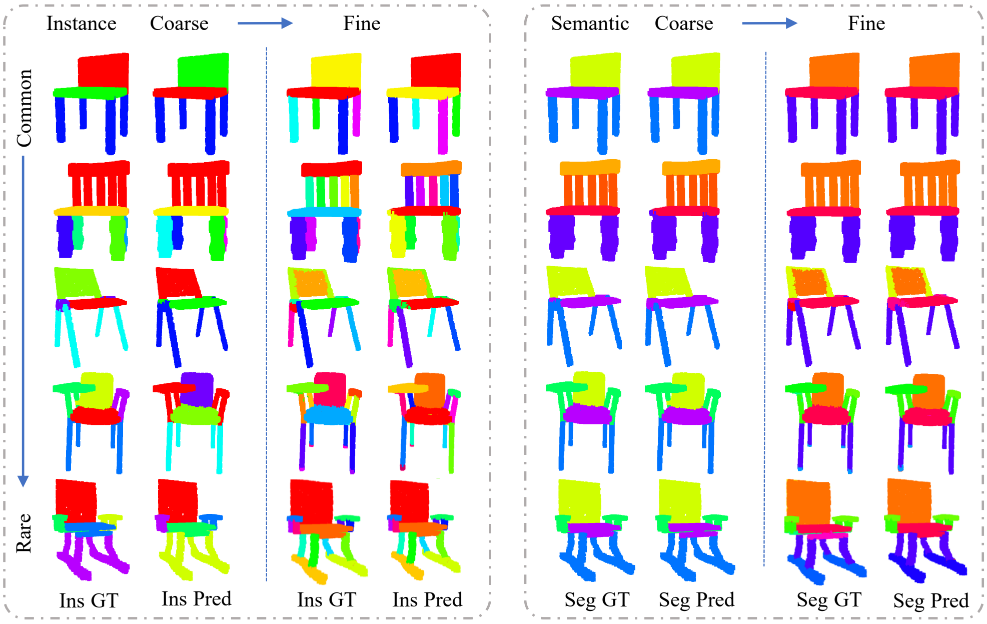


FIGURE 5.10. Visualization of the performance of on PartNet Mo et al., 2019. Both coarse and fine-grained results are provided. Note that different instance are shown with different colors, and the same instance are not necessarily have the same color in ground truth and prediction presentation.

a huge memory buffer. Different from SGPN, ASIS utilizes mean-shift for clustering embeddings to instance groups. Meanwhile, ASIS applies KNN for fusing semantic context from a fixed number of neighboring points, which is utilized on every input point. This operation is extremely time-consuming and fails to take full advantage of computational resources. Compared with the above two approaches, our proposed MPNet is free from complex and time-consuming operations, showing superiority in both effectiveness and efficiency.

TABLE 5.6. Inferencing time comparison on S3DIS Area-5 set. Forward time is the network running time on GPU, whereas postprocessing time is the BlockMerging algorithm introduced in Wang et al., 2018a. ASIS is 45% slower than our method in the forward process due to the usage of KNN, which is extremely time-consuming. The reported time is running on a single 1080ti GPU with 4096 input points.

Method	Backbone	Inference Time (ms)			mPre	mRec
		Overall	Forward	Post		
SGPNWang et al., 2018a	PointNet	730	**22**	708	36.0	28.7
ASISWang et al., 2019b	PointNet2	183	58	**125**	55.3	42.4
Ours	PointNet2	**165**	40	**125**	**62.5**	**49.0**

5.5 Conclusion

In this chapter, we propose a memory-augmented network to handle both category and pattern imbalance in the task of point cloud instance and semantic segmentation. A memory module is introduced to alleviate the forgetting issue during the training process. The performance on the benchmarks shows the superiority of our method in both effectiveness and efficiency.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

In this thesis, we have discussed efficient scene parsing solutions under different modalities.

In Chapter 3, we address the fast semantic segmentation task in the 2D image domain and propose an approach based on knowledge distillation to boost the accuracy of a model while maintaining the inference speed unchanged. Previous methods often need to design new architectures, which often suffer from a worse trade-off between the accuracy and efficiency compared to the well-designed light-weighted models. Besides, some approaches utilize knowledge distillation technology but fail to solve the short-range problem introduced by the student model. In this thesis, we propose a knowledge distillation method tailored for semantic segmentation to improve the performance of the compact FCNs with large overall stride. To handle the inconsistency between the features of the student and teacher network, we optimize the feature similarity in a transferred latent domain formulated by utilizing a pre-trained autoencoder. Moreover, an affinity distillation module is proposed to capture the long-range dependency by calculating the non-local interactions across the whole image. To validate the effectiveness of our proposed method, extensive experiments have been conducted on three popular benchmarks under various settings.

In Chapter 4, we propose a novel Instance-Aware Module for point cloud instance segmentation, which successfully encode the instance-level context and explicit geometric information. Instance segmentation in point cloud just begin to draw attention in the computer vision field. The proposed IAM learns discriminative instance embedding features in two-fold: (1) Instance contextual regions, covering the spatial extension of an instance, are implicitly learned and propagated in the decoding process. (2) Instance-dependent geometric knowledge is included in the embedding space, which is informative and critical to discriminate adjacent instances. The proposed framework can be trained in an end-to-end manner and shows superiority over previous methods on both efficiency and effectiveness. With the proposed method, state-of-the-art results are achieved on different tasks, surpassing previous approaches by a large margin.

In Chapter 5, we analysis the forgetting issue and explore the influence of data imbalance in the task of semantic and instance segmentation on point cloud, which appears as both category imbalance and pattern imbalance. Different from the previous methods that address the problem by re-weighting the category-wise weights, we propose a memory-augmented network to learn and memorize the representative prototypes that cover diverse samples universally. Specifically, a memory module is introduced to alleviate the forgetting issue by recording the patterns seen in mini-batch training. The learned memory items consistently reflect the interpretable and meaningful information for both dominant and non-dominant categories and cases. The distorted observations and rare cases can thus be augmented by retrieving the stored prototypes, leading to better performances and generalization. We validate the effectiveness of our proposed method on various datasets.

6.2 Future Directions

We have presented some solutions to the issues stated above, but the methods are not perfect and remains some issues unresolved.

6.2.1 2D image domain

The issue of efficient semantic segmentation has been addressed in Chapter 3. However, the knowledge distillation is only utilized in the last convolution layer, which could be extended to the architectures that have structured output, such as ResNet. More supervision signals from the teacher model can be applied to the intermediate layers. Besides, the experiments show that increasing the accuracy of the teacher model can't bring more improvements to the small student model. At last, we model the structure knowledge as pair-wise cosine distances. Some papers have addressed the issue by introducing a generative adversarial network Liu et al., 2019a to force the structure of the two models to be close. Other structural information should be studied. Also, the current methods are good at grasping high-level semantic knowledge while failing to provide fine results on the boundary. One possible solution may be to combine different modalities, for example, point cloud, as it excels at measuring distance and tell apart the edges.

6.2.2 3D domain

Due to a large number of points, many methods split the whole scene into overlapped blocks ($1m \times 1m$), which not only limits the context information but also requires complex post-processing (as introduced in the BlockMerging algorithm 4). With the emergence of sparse convolution Graham, Engelcke, and Maaten, 2018, directly processing large-scale point sets is possible and the method is highly related to CNN. However, different from the 2D domain, where techniques have been carefully studied and discussed, the counterpart on 3D point cloud, on the other hand, lags far behind. For example, FPN Lin et al., 2017b is known to be useful to capture multi-scale contextual information. But recent works fail to study the effects of the classical techniques that are widely used in the 2D domain. Moreover, accurate point sets are captured by the laser reflection, which has some effects: (1) close objects are covered by a large number of points while the distant objects contain much fewer points, making it hard for predicting.

Besides, a single point, like pixel, is meaningless, unless incorporating surrounding points to introduce structure knowledge. Both PointNet Qi et al., 2017b and PointNet++ Qi et al., 2017a directly operate on point sets. However, the per-point representation is captured by using MLP layers, which is equivalent to 1 × 1 convolution in the 2D image. KPConv Thomas et al., 2019 and sparse convolution methods Graham, Engelcke, and Maaten, 2018 alleviate the problem to some extent. Because of the variety of geometry and topology of real-world shapes, I think it is more suitable to treat point sets as sampling samples from a continuous surface, which can be represented as a parameterized function.

The last and most important thing might be the multimodality fusion of the two domains. Point cloud data are better at distance measuring and imagery data are more efficient for understanding high-level context, which makes it promising to combine different sources of data to achieve automation.

Bibliography

Ainde, B. O. and J. M. Zurada (2018). “Deep Learning of Constrained Autoencoders for Enhanced Understanding of Data”. In: *IEEE Trans. Neural Netw. & Learn. Syst.*

Armeni, Iro, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese (2016). “3D Semantic Parsing of Large-Scale Indoor Spaces”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Badrinarayanan, V., A. Kendall, and R. Cipolla (2017). “SegNet: A Deep Convolutional Encoder-Decoder Architecture For Image Segmentation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.*

Bahdanau, D., K. Cho, and Y. Bengio (2014). “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473.*

Bergh, Michael Van den, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool (2012). “SEEDS: Superpixels Extracted via Energy-Driven Sampling”. In: *Proc. Eur. Conf. Comp. Vis.*

Besl, Paul J. and Ramesh C. Jain (1988). “Segmentation Through Variable-Order Surface Fitting”. In: *IEEE Trans. Pattern Anal. Mach. Intell.*

Bhanu, Bir, Sung-Kee Lee, Chih Cheng Ho, and Thomas C. Henderson (1986). “Range Data Processing: Representation Of Surfaces By Edges.” In: *Proc. Int. Conf. Patt. Recogn.*

Brabandere, Bert De, Davy Neven, and Luc Van Gool (2017). “Semantic Instance Segmentation with a Discriminative Loss Function”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Breiman, Leo (2001). “Random Forests”. In: *Machine Learning.*

Chen, Fei, Huimin Yu, Roland Hu, and Xunxun Zeng (2013). “Deep Learning Shape Priors for Object Segmentation”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Chen, Guobin, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker (2017a). “Learning Efficient Object Detection Models with Knowledge Distillation”. In: *Proc. Advances in Neural Inf. Process. Syst.*

Chen, L-C., G. Papandreou, I. Kokkinos, K. Murphy, and A.L. Yuille (2017b). “DeepLab: Semantic Image Segmentation With Deep Convolutional Nets, Atrous Convolution, And Fully Connected Crfs”. In: *IEEE Trans. Pattern Anal. Mach. Intell.*

Chen, L-C., G. Papandreou, F. Schroff, and H. Adam (2017c). “Rethinking Atrous Convolution for Semantic Image Segmentation”. In: *arXiv preprint arXiv:1706.05587.*

Chen, L-C., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam (2018). “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *Proc. Eur. Conf. Comp. Vis.*

Chollet, F. (2017). “Deep Learning With Depthwise Separable Convolutions”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Chollet, François (2016). “Xception: Deep Learning with Depthwise Separable Convolutions”. In: *arXiv preprint arXiv:1610.02357*.

Cordts, M., M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele (2016). “The Cityscapes Dataset For Semantic Urban Scene Understanding”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Dai, Angela, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner (2017a). “ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Dai, Jifeng, Kaiming He, and Jian Sun (2016). “Instance-aware semantic Segmentation via Multi-task Network Cascades”. In: *Proc. Eur. Conf. Comp. Vis.*

Dai, Jifeng, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei (2017b). “Deformable Convolutional Networks”. In: *Proc. IEEE Int. Conf. Comp. Vis.*

Dalal, Navneet and Bill Triggs (2005). “Histograms of Oriented Gradients for Human Detection”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst (2016). “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering”. In: *Proc. Advances in Neural Inf. Process. Syst.*

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). “ImageNet: A Large-Scale Hierarchical Image Database”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv preprint arXiv:1810.04805*.

Elich, Cathrin, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe (2019). “3D-BEVIS: Bird’s-Eye-View Instance Segmentation”. In: *arXiv preprint arXiv:1904.02199*.

Engelmann, Francis, Theodora Kontogianni, Alexander Hermans, and B. Leibe (2017). “Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds”. In: *Proc. IEEE Int. Conf. Comp. Vis. Workshops.*

Engelmann, Francis, Theodora Kontogianni, Jonas Schult, and Bastian Leibe (2018). “Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds”. In: *arXiv:1810.01151.*

Everingham, M., S.M.A. Eslami, L.V. Gool, C.K.I. Williams, J. Winn, and A. Zisserman (2014). “The Pascal Visual Object Classes Challenge – A Retrospective”. In: *Int. J. Comput. Vision.*

Farabet, Clement, Camille Couprie, Laurent Najman, and Yann LeCun (2013). "Learning Hierarchical Features for Scene Labeling". In: *IEEE Trans. Pattern Anal. Mach. Intell.*

Fu, Jun, Jing Liu, Haijie Tian, and Yong Li (2019). "Dual Attention Network for Scene Segmentation". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Golovinskiy, Aleksey and Thomas Funkhouser (2009). "Min-Cut Based Segmentation of Point Clouds". In: *Proc. IEEE Int. Conf. Comp. Vis. Workshops.*

Gong, Dong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel (2019). "Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection". In: *Proc. IEEE Int. Conf. Comp. Vis.*

Graham, Benjamin, Martin Engelcke, and Laurens van der Maaten (2018). "3D Semantic Segmentation with Submanifold Sparse Convolutional Networks". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Graves, Alex, Greg Wayne, and Ivo Danihelk (2014). "Neural Turing Machines". In: *arXiv preprint arXiv:1410.5401.*

Hariharan, B., P. Arbelaez, L Bourdev, S. Maji, and J. Malik (2011). "Semantic Contours From Inverse Detectors". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

He, K., X. Zhang, S. Ren, and J. Sun (2016a). "Deep Residual Learning For Image Recognition". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016b). "Deep Residual Learning for Image Recognition". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017a). "Mask R-CNN". In: *Proc. IEEE Int. Conf. Comp. Vis.*

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2017b). "Prototypical Networks for Few-shot Learning". In: *Proc. Advances in Neural Inf. Process. Syst.*

Hinton, G., O. Vinyals, and J. Dean (2015). "Distilling the Knowledge in a Neural Network". In: *arXiv preprint arXiv:1503.02531.*

Hinton, Geoffrey (2010). "Deep Belief Nets". In: *Encyclopedia of Machine Learning.* Boston, MA: Springer US, pp. 267–269. ISBN: 978-0-387-30164-8.

Hou, Ji, Angela Dai, and Matthias Nießner (2019). "3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Howard, A.G., M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam (2017). "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: *arXiv preprint arXiv:1704.04861.*

Hu, Tao, Pengwan Yang, Chiliang Zhang, Gang Yu, Yadong Mu, and Cees G. M. Snoek (2019). "Attention-Based Multi-Context Guiding for Few-Shot Semantic Segmentation". In: *Proc. AAAI Conf. Artificial Intell.*

Huang, Zehao and Naiyan Wang (2017). "Like What You Like: Knowledge Distill via Neuron Selectivity Transfer". In: *arXiv preprint arXiv:1707.01219.*

Kang, Byeongkeun and Truong Q. Nguyen (2019). "Random Forest with Learned Representations for Semantic Segmentation". In: *arXiv preprint arXiv:1901.07828.*

Kim, J., S. Park, and N. Kwak (2018). “Paraphrasing Complex Network: Network Compression via Factor Transfer”. In: *Proc. Advances in Neural Inf. Process. Syst.*

Krahenbuhl, Philipp and Vladlen Koltun (2011). “Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials”. In: *Proc. Advances in Neural Inf. Process. Syst.*

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Proc. Advances in Neural Inf. Process. Syst.*

Lafferty, John, Andrew McCallum, and Fernando C.N. Pereira (2001). “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”. In: *Proc. Int. Conf. Mach. Learn.*

Li, Guohao, Matthias Müller, Ali Thabet, and Bernard Ghanem (2019). “DeepGCNs: Can GCNs Go as Deep as CNNs?” In: *Proc. IEEE Int. Conf. Comp. Vis.*

Li, Yangyan, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen (2018). “PointCNN: Convolution On X-Transformed Points”. In: *Proc. Advances in Neural Inf. Process. Syst.*

Li, Zhengqin and Jiansheng Chen (2015). “Superpixel Segmentation using Linear Spectral Clustering”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Lin, D., Y. Ji, D. Linschinski, D. Cohen-Or, and H. Huang (2018). “Multi-Scale Context Intertwining for Semantic Segmentation”. In: *Proc. Eur. Conf. Comp. Vis.*

Lin, G., C. Shen, A. van den Hengel, and I. Reid (2016). “Efficient piecewise training of deep structured models for semantic segmentation”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Lin, G., A. Milan, C. Shen, and I. Reid (2017a). “RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Lin, Tsung-Yi, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie (2017b). “Feature Pyramid Networks for Object Detection”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár (2017c). “Focal Loss for Dense Object Detection”. In: *Proc. IEEE Int. Conf. Comp. Vis.*

Liu, Y., K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang (2019a). “Structured Knowledge Distillation for Semantic Segmentation”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Liu, Z., X. Li, P. Luo, C.C. Loy, and X. Tang (2015). “Semantic Image Segmentation Via Deep Parsing Network”. In: *Proc. IEEE Int. Conf. Comp. Vis.*

Liu, Ziwei, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu (2019b). “Large-Scale Long-Tailed Recognition in an Open World”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Long, J., E. Shelhamer, and T. Darrell (2015). “Fully Convolutional Networks For Semantic Segmentation”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Lowe, David G. (2004). “Distinctive Image Features from Scale-Invariant Keypoints”. In: *Int. J. Comput. Vision.*

Maaten, Laurens van der (2014). “Accelerating t-SNE using Tree-Based Algorithms”. In: *J. Mach. Learn. Res.*

Maturana, Daniel and Sebastian Scherer (2015). “VoxNet: A 3D Convolutional Neural Network for real-time object recognition”. In: *Proc. IEEE Int. Conf. Intelligent Robots Syst.*

Mo, Kaichun, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su (2019). “PartNet: A Large-scale Benchmark for Fine-grained and Hierarchical Part-level 3D Object Understanding”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Moore, Alastair P., Simon J. D. Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones (2009). “Scene Shape Priors for Superpixel Segmentation”. In: *Proc. IEEE Int. Conf. Comp. Vis.*

Mottaghi, R., X. Chen, X. Liu, N. Cho, S. Lee, S. Fidler, R. Urtasun, and A. Yuille (2014). “The Role of Context for Object Detection and Semantic Segmentation in the Wild”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Nagel, Markus, Mart van Baalen, Tijmen Blankevoort, and Max Welling (2019). “Data-Free Quantization Through Weight Equalization and Bias Correction”. In: *Proc. IEEE Int. Conf. Comp. Vis.*

Ojala, Timo, Matti Pietikainen, and David Harwood (1996). “A comparative study of texture measures with classification based on featured distributions”. In: *Pattern Recogn.*

Paszke, A., A. Chaurasia, S. Kim, and E. Culurciello (2016). “ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation”. In: *arXiv preprint arXiv:1606.02147.*

Pham, Quang-Hieu, Duc Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung (2019). “JSIS3D: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Qi, Charles R., Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas (2016). “Volumetric and Multi-View CNNs for Object Classification on 3D Data”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Qi, Charles R, Li Yi, Hao Su, and Leonidas J. Guibas (2017a). “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In: *Proc. Advances in Neural Inf. Process. Syst.*

Qi, Charles R, Hao Su, Kaichun Mo, and Leonidas J. Guibas (2017b). “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Ranjan, Anurag, Timo Bolkart, Soubhik Sanyal, and Michael J. Black (2018). “Generating 3D faces using Convolutional Mesh Autoencoders”. In: *Proc. Eur. Conf. Comp. Vis.*

Riegler, Gernot, Ali Osman Ulusoy, and Andreas Geiger (2016). "OctNet: Learning Deep 3D Representations at High Resolutions". In: *arXiv preprint arXiv:1611.05009.*

Romero, A., N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, and Y. Bengio (2015). "FitNets: Hints for Thin Deep Nets". In: *Proc. Int. Conf. Learn. Representations.*

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention Society.*

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L-C. Chen (2018). "MobileNetV2: Inverted Residuals and Linear Bottlenecks". In: *CVPR.*

Sappa, A. D. and M. Devy (2001). "Fast range image segmentation by an edge detection strategy". In: *Proc. Int. Conf. 3D Imaging and Modeling.*

Schnabel, Ruwen, Roland Wahl, and Reinhard Klein (2007). "Efficient RANSAC for Point-Cloud Shape Detection". In: *Computer Graphics Forum.*

Shotton, Jamie, Matthew Johnson, and Roberto Cipolla (2008). "Semantic Texton Forests for Image Categorization and Segmentation". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Shuai, B., Z. Zuo, B. Wang, and G. Wang (2017). "Scene Segmentation With Dagrecurrent Neural Networks". In: *IEEE Trans. Pattern Anal. Mach. Intell.*

Su, Hang, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller (2015). "Multi-view Convolutional Neural Networks for 3D Shape Recognition". In: *Proc. IEEE Int. Conf. Comp. Vis.*

Sun, Ke, Bin Xiao, Dong Liu, and Jingdong Wang (2019). "Deep High-Resolution Representation Learning for Human Pose Estimation". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Tang, Zhiqiang, Xi Peng, Shijie Geng, Lingfei Wu, Shaoting Zhang, and Dimitris Metaxas (2018). "Quantized Densely Connected U-Nets for Efficient Landmark Localization". In: *Proc. Eur. Conf. Comp. Vis.*

Thomas, Hugues, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas (2019). "KPConv: Flexible and Deformable Convolution for Point Clouds". In: *Proc. IEEE Int. Conf. Comp. Vis.*

Tian, Z., T. He, C. Shen, and Y. Yan (2019). "Decoders Matter for Semantic Segmentation: Data-Dependent Decoding Enables Flexible Feature Aggregation". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Toneva, Mariya, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon (2018). "An empirical study of example forgetting during deep neural network learning". In: *arXiv preprint arXiv:1812.05159.*

Tovari, D. and N. Pfeifer (2005). "Segmentation based robust interpolation - A new approach to laser data filtering". In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences.*

Treml, M., J. Arjona-Medina, T. Unterthiner, R. Durgesh, F. Friedmann, P. Schuberth, A. Mayr, M. Heusel, M. Hofmarcher, and M Widrich et al (2016). "Speeding

Up Semantic Segmentation For Autonomous Driving". In: *Proc. Advances in Neural Inf. Process. Syst Workshops.*

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, and Lukasz Kaiser (2017). "Attention Is All You Need". In: *Proc. Advances in Neural Inf. Process. Syst.*

Veksler, Olga, Yuri Boykov, and Paria Mehrani (2010). "Superpixels and Supervoxels in an Energy Optimization Framework". In: *Proc. Eur. Conf. Comp. Vis.*

Vosselman, G. and S. Dijkman (2001). "3D building model reconstruction from point clouds and ground plans". In: *Proceedings of the ISPRS Workshop.*

Vu, Nhat and B.S. Manjunath (2008). "Shape Prior Segmentation of Multiple Objects with Graph Cuts". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Wang, Lei, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan (2019a). "Graph Attention Convolution for Point Cloud Semantic Segmentation". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Wang, Weiyue, Ronald Yu, Qiangui Huang, and Ulrich Neumann (2018a). "SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Wang, X., R. Girshick, A. Gupta, and K. He (2018b). "Non-local Neural Networks". In: *CVPR.*

Wang, Xinlong, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia (2019b). "Associatively Segmenting Instances and Semantics in Point Clouds". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Wang, Yue, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon (2019c). "Dynamic Graph CNN for Learning on Point Clouds". In: *ACM Trans. On. Graphic.*

Wu, Z., C. Shen, and A. van den Hengel (2016). "Wider or Deeper: Revisiting the ResNet Model for Visual Recognition". In: *Pattern Recogn.*

Wu, Zhirong, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao (2015). "3D ShapeNets: A Deep Representation for Volumetric Shapes ". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Xu, Xiaowei, Qing Lu, Yu Hu, Lin Yang, Sharon Hu, Danny Chen, and Yiyu Shi (2018). "Quantization of Fully Convolutional Networks for Accurate Biomedical Image Segmentation". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Yang, Bo, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni (2019a). "Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds". In: *Proc. Advances in Neural Inf. Process. Syst.*

Yang, Ze, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin (2019b). "RepPoints: Point Set Representation for Object Detection". In: *Proc. IEEE Int. Conf. Comp. Vis.*

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le (2019c). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Proc. Advances in Neural Inf. Process. Syst.*

Yi, Li, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J. Guibas (2018). “GSPN: Generative Shape Proposal Network for 3d Instance Segmentation in Point Cloud”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Yim, J., D. Joo, J. Bae, and J. Kim (2017). “A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Yu, Changqian, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang (2018). “BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation”. In: *Proc. Eur. Conf. Comp. Vis.*

Yu, Changqian, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang (2020). “BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation”. In: *arXiv preprint arXiv:2004.02147.*

Yu, F. and V. Koltun (2014). “Multi-Scale Context Aggregation by Dilated Convolutions”. In: *arXiv preprint arXiv:1511.07122.*

Yu, Fisher and Vladlen Koltun (2016). “Multi-Scale Context Aggregation by Dilated Convolutions”. In: *Proc. Int. Conf. Learn. Representations.*

Zagoruyko, Sergey and Nikos Komodakis (2017). “Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer”. In: *Proc. Int. Conf. Learn. Representations.*

Zhang, Han, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena (2018a). “Self-Attention Generative Adversarial Networks”. In: *arXiv preprint arXiv:1805.08318.*

Zhang, Z., X. Zhang, C. Peng, D. Cheng, and J. Sun (2018b). “ExFuse: Enhancing Feature Fusion for Semantic Segmentation”. In: *Proc. Eur. Conf. Comp. Vis.*

Zhao, H., J. Shi, X. Qi, X. Wang, and J. Jia (2017). “Pyramid Scene Parsing Network”. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Zhao, H., X. Qi, X. Shen, J. Shi, and J. Jia (2018). “ICNet for Real-Time Semantic Segmentation on High-Resolution Images”. In: *Proc. Eur. Conf. Comp. Vis.*

Zheng, S., S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr (2015). “Conditional Random Fields as Recurrent Neural Networks”. In: *Proc. IEEE Int. Conf. Comp. Vis.*