3D diffusion based generation model for point cloud annotation and generation

by

Tingting Li

National Centre for Computer Animation

Faculty of Media & Communication

Bournemouth University

A thesis submitted in partial fulfilment of the
requirements of Bournemouth University for the degree of
Doctor of Philosophy

Jul. 2023

Copyright Statement

Abstract

Concurrent with the rapid advancement of applications and 3D scanning sensors, the demand for 3D deep learning based technology and data has increased dramatically. Especially 3D shape with semantic labels plays a significant role in 3D vision problems, such as auto-driven, 3D object detection and 3D scene segmentation, etc. As the deep learning era arrives, automatic, high-quality, and large-scale solutions in annotation 3D shape to the 3D vision problem are desired. Point cloud, as one of the most popular representations of 3D, is facing the same desire.

Point cloud generative model is one type of model that can be used to synthesize a new point cloud. The characteristic of the point cloud generative model indicates that it contains the semantic structure of a point cloud. The interrelationships of point cloud attract many researchers to explore to use of point cloud generative to solve annotated point cloud acquire problems. However, it is still challenging to acquire expressive and accurate annotated point clouds.

This thesis addresses the aforementioned challenge by exploring three aspects: the synthesis of high-quality 3D point cloud objects, the point-label pairs generation, and the evolution of their enhancement strategies.

- This work introduces a point cloud diffusion generation model combining stochastic differential equations and

Markov Chain Mento Carlo samplers. This method can synthesise high-quality 3D point cloud objects and allows a more flexible sampling method to point cloud generation.

- Furthermore, the thesis presents a point-label pairs generation method to alleviate the cost of large-scale point cloud annotation. This method investigates the characteristics of diffusion-based point cloud generation model and exploits a feature interpreter to generate a point cloud with corresponding semantic labels for each point.
- Last, a filter approach for generated point-label pairs is employed to improve the quality of the generated point cloud dataset. As a result, the proposed method resolves the point cloud generation and annotation effectively.

To demonstrate the effectiveness of the proposed method, various experiments were conducted across different scenarios. These experiments not only validated the reliability of the generated point cloud and point-label pairs but also illustrated their superior performance in comparison to GAN-based point-label generation methods. This research represents a substantial contribution to the enhancement of the quality and applicability of 3D point cloud data and understanding.

Acknowledgements

I am profoundly thankful to my mentors, Prof.Jian Chang, Prof.Hui Liang, and Prof.Jian Jun Zhang, for their guidance and support during my PhD studies. Specifically, I appreciate Prof.Hui Liang for his trust in me and the scholarship support. I sincerely thank Prof.Jian Jun Zhang for his support throughout my journey. I am deeply grateful to Prof.Jian Chang for his ongoing encouragement, advice, and faith in me. I can not achieve this without his valuable feedback and involvement.

I also extend my gratitude to Prof.Xiaoguang Han from the University of Hong Kong (Shenzhen) for his invaluable support and generous help. His insightful views and research attitude motivated me at all stages.

My heartfelt thanks go to Zizheng Yan, Dr. Yunbi Liu, and Dr. Zhangcan Ding. Their selfless help before the deadline and their warm friendship have been a source of strength. I am also grateful for the companionship and support of Anran Lin, Mutian Xu, Lingteng Qiu, Chenghong Li, and Chaoda Zheng. Their kindness and motivation have made the lab a warm and engaging workplace.

I want to thank Yunfei Fu for being the rock in my life. His unwavering encouragement and understanding helped me navigate through the challenging moments during my study.

I would like to express my gratitude to my parents as well. They have always supported my choices and believed me.

Last but not least, I have the deepest appreciation for Bournemouth University. Without its supportive academic environment, my doctoral research would not have been possible.

Declaration

This thesis is submitted in fulfilment of the requirement for transfer from Master of Philosophy (MPhil) to the Doctor of Philosophy (PhD) at Bournemouth University.

Contents

Copyright ii

Abstract iii

Acknowledgements v

Declaration vii

List of Figures xii

List of Tables xv

1 Introduction 1

1.1 Background and Motivation 1

1.2 Research Questions 8

1.3 Research Objectives 10

1.4 Contributions 11

1.5 List of Publications 12

1.6 Outline of Thesis 13

2 Literature Review 15

2.1 Point Cloud Feature Descriptor Learning Methods 15

2.1.1 Traditional Methods for Point Cloud Feature Descriptor 16

2.1.2 Deep Learning Methods for Point Cloud Feature Descriptor Learning 16

2.1.2.1 Multi-view Projection-based Methods . . 17

2.1.2.2 Voxel-based Methods 19
2.1.2.3 Domain Point Convolution-based Methods 21
2.1.2.4 Graph-based Methods 22
2.1.2.5 Point-Set-based Methods 23
2.1.2.6 Transformer Structured based Methods . 24
2.2 Point Cloud Generation Methods 25
2.2.1 Traditional Data-driven Methods 26
2.2.2 Autoregressive based Methods 27
2.2.3 VAE based Methods 28
2.2.4 GAN based Methods 32
2.2.5 Diffusion based Methods 36
2.3 Point Cloud Annotation Without Strong Supervision . . 37
2.3.1 Traditional Methods for 3D shape Annotation . . 37
2.3.2 Data-driven Methods for 3D shape Annotation . . 39
2.3.3 Point Cloud Part Segmentation based Deep Learning Methods 40

3 Point Cloud Generative Model based on Stochastic Differential Equations 48
3.1 Introduction 49
3.2 Point Cloud Generation based on Stochastic Differential Equations 51
3.2.1 Preliminaries of Stochastic Process 52
3.2.2 Preliminaries of Stochastic Differential Equations 57
3.2.3 Application of Stochastic Differential Equations on Point Cloud Generation 58
3.3 Point Cloud Sampling based on Markov Chain Monte Carlo Sampler 62
3.3.1 Preliminaries of Markov Chain Monte Carlo . . . 62
3.3.2 Application of Markov Chain Monte Carlo Sampling on Point Cloud Generation 64

3.4 Method 65

3.4.1 Latent Variable Reparameterization 66

3.4.2 Time Encoding 69

3.4.3 Training Objective 70

3.5 Implementation process and experiments 73

3.5.1 Experimental setting 74

3.5.2 Evaluation Metrics 74

3.5.3 Point Cloud Generation Experiments 78

3.5.4 Point Cloud Reconstruction Experiments 80

3.5.5 Point Cloud Upsampling Experiments 82

3.6 Summary 84

4 Generative Approach to 3D Point Cloud Annotation Method 85

4.1 Introduction 86

4.2 Intermediate Feature Analysis and Feature Interpreter . 88

4.2.1 Intermediate Feature Analysis 88

4.2.1.1 Intermediate Feature Analysis Preliminary 88

4.2.1.2 Intermediate Feature Analysis Approach 90

4.2.2 Feature Interpreter 93

4.2.2.1 Method Architecture 94

4.2.2.2 Feature Interpreter Details 95

4.3 Evaluation 96

4.3.1 Experimental setting 96

4.3.2 Evaluation Metrics 97

4.3.3 The effectiveness of Feature Interpreter 99

4.3.4 Validation of Representation Effectiveness 99

4.3.5 Ablation Study 102

4.4 Summary 103

5 Improvement for Point Cloud Dataset Generation 104

5.1 Introduction 105

5.2 Uncertainty Measurement 106

5.2.1 Query Strategy Preliminary 106

5.2.2 Applying Query-By-Committee in Point-Label Pairs Filter 108

5.3 Implementation Process and Experiment Evaluation . . . 110

5.3.1 Experimental Setup 110

5.3.2 Validation of Generated Datasets 111

5.3.3 Comparison with GAN-based Method 113

5.3.4 Application in a Few-shot Scenario 115

5.4 Summary 116

6 Conclusion and Future Work 117

6.1 Conclusion 117

6.2 Limitations and Future Work 120

Bibliography 123

List of Figures

1.1 Example applications related to 3D point cloud technical. The images are cited from (Mo et al. (2019), Hong et al. (2022), Aksoy et al. (2020), Matrone et al. (2020)) 2

1.2 An illustration of different data representations on the Stanford Bunny benchmark. 3

1.3 Challenges for 3D point cloud learning (Bello et al. (2020)). (a) Irregular: dense and sparse areas; (b) No grid/Unstructured: each point is independent and the distance between adjacent points is not fixed; (c) Unordered: point cloud consists of a set of points and is permutation invariant. . . . 4

2.1 Illustration of multi-view projection-based approach of point cloud (Su et al. 2015) 18

2.2 Illustration of the VoxNet Architecture (Maturana and Scherer 2015) 20

2.3 Illustration of point cloud VAE-based generation method 28

2.4 FoldingNet Architecture (Yang et al. 2018). The decoder is a two-cascade structure represented by a green square. Both decoders are folding and concatenating with a codeword, which is a replicated global latent variable. 30

2.5 Illustration of point cloud GANbased generation method 32

2.6 Unsupervised 3D point clouds generated by tree-GAN for multiple classes (Shu et al. 2019a). 34

2.7 3D point cloud part segmentation examples (Kalogerakis et al. 2017) 37

3.1 Overview of point cloud generative modeling through SDE. The desired shape is synthesised from a noise point cloud with prior distribution via a continuous-time SDE-Net. . 52

3.2 The illustration of the directed graphical model of the stochastic process for point cloud 66

3.3 Illustration of normalising flow 67

3.4 The illustration of one layer of normalising flow. 68

3.5 Illustration of normalising flow architecture 69

3.6 The illustration of the training phase of the proposed model. The z is noise with a prior distribution. The Global Latent Variable p_{data} is the distribution of data. The affine layers map the data distribution to the latent variable p_{latent} with Gaussian distribution. 72

3.7 Illustration of PointNet architecture (Qi et al. 2017a). . . 75

3.8 The illustration of the inference phase of the proposed model. The dashed line represents the loop process. . . . 78

3.9 Examples of point clouds synthesised by the proposed model. 79

3.9 Examples of point clouds synthesised by the proposed model. 80

3.10 The illustration of the reconstruction phase of the proposed model. The dashed line represents the loop process. 81

3.11 Examples of point cloud reconstruction. 82

3.12 Global latent variable interpolation. 83

3.13 Visualised experimental results of point cloud completion. 83

4.1 Visualization of the intermediate feature in an image generative model (Zeiler and Fergus 2014). For each layer of visualization, the nine images with the largest reconstructed values are displayed, representing distinct features the layer is attentive to. 90

4.2 Visualization of the diffusion process and corresponding K-means clustering of intermediate features of the point cloud generative model. The top row represents the states of the point cloud in the diffusion process in the time variable. The bottom row represents the results of the corresponding K-means clustering features of the intermediate features. 91
4.3 Calinski-Harabasz Index of intermediate features clustering results. The score represents the quality of the cluster. Different colors represent different layers of the θ_G 93
4.4 Architecture of proposed method. 94
4.5 Illustration of the point-label pairs generation method. . 96
4.6 Illustration of IoU metrics for semantic segmentation (Kukil 2013). 98
4.7 Visualisation examples of segmentation results. 100
4.8 Qualitative comparison of intermediate features based on a different baseline using K-means cluster. 101

5.1 The illustration of point-label pairs generation with filter mechanism. 109
5.2 Diagram of inference phase. 111
5.3 Examples of generated point-label pairs. The feature interpreter is trained with the full ground-truth dataset. . 112
5.4 Examples of generated point-label pairs (from left to right are airplane, chair, table, guitar, car, lamp, bag). The feature interpreter is trained with the few-shot sample of the ground-truth dataset. 113
5.5 Comparison between the generated datasets and ground-truth ones. Different colours denote different filter ratios. 114
5.6 Visualized results of Shu et al. (2019b), Yang et al. (2021) and ours. 115

List of Tables

3.1 Comparison of point cloud generation performance. . . . 81

3.2 Comparison of point cloud reconstruction performance. 82

4.1 Few-shot segmentation on the ground truth dataset. k are the number of samples that are used in Training. 101

4.2 Evaluation of the different intermediate feature extraction variations for part segmentation. 103

5.1 Comparison of point cloud and label generation performance.114

5.2 Comparative Analysis of mIoU Scores for Segmentation Tasks Using a Small Set of Ground Truth (GT) Samples and an Augmented Dataset. 'Samples' indicates the total number of the point cloud in the GT set. 'w/' denotes training results when including the augmented dataset; 'w/o' reflects training exclusively on the GT set. 116

Chapter 1

Introduction

1.1 Background and Motivation

The applications of 3D visualisation have emerged in all aspects of our digital lives. From industrial manufacturing to individual consumer products, 3D vision technology has become more prevalent and sophisticated. As shown in Figure 1.1, with the aid of 3D vision technology, ordinary users can create their 3D content (Mo et al. 2019); Advances in Virtual Reality (VR) products have contributed to the development of novel experiences in tourism; autonomous driving technology has witnessed breakthrough advancements that augment safe navigation (Aksoy et al. 2020); the domain of automated animation generation has seen considerably relieve manual labour costs (Tevet et al. 2023). The realisation of these applications is inseparable from the support of advanced 3D vision technologies and massive amounts of 3D data. Therefore, cutting-edge 3D understanding and analysis techniques, along with the production of corresponding data, are of significant importance.

3D data has various representation forms, as shown in Figure 1.2. Generally speaking, 3D data can be represented by mesh, point cloud, Non-Uniform Rational B-Spline (NURBS), voxel, multi-view projections, implicit field, primitives, etc. Among them, **Mesh** is a geometric representation of triangles or quadrilaterals, which is widely used in modelling and rendering three-dimensional objects (Mortenson 1997). **NURBS**

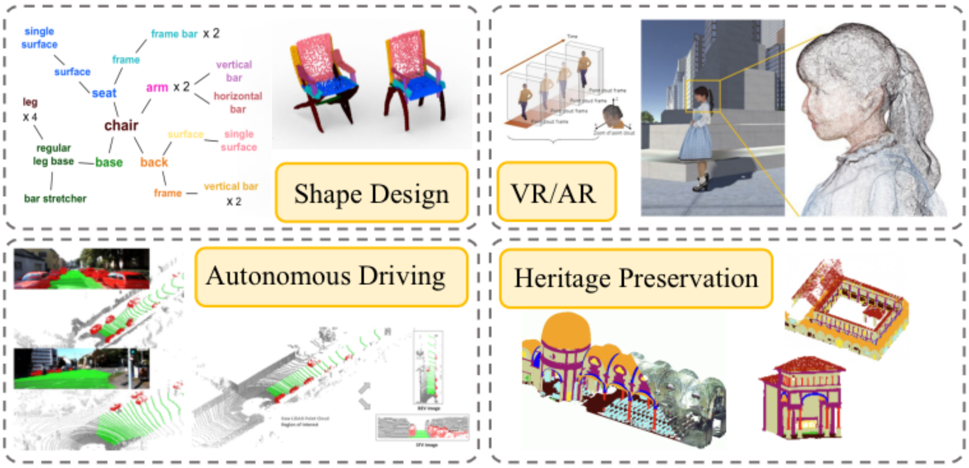


Figure 1.1: Example applications related to 3D point cloud technical. The images are cited from (Mo et al. (2019), Hong et al. (2022), Aksoy et al. (2020), Matrone et al. (2020))

represents standard geometric figures (straight lines, circles, ellipses, spheres, rings) and free-form geometric figures (car bodies, human bodies), making it an ideal choice for computer-aided modelling. **Voxel** represents a value on a regular grid in three-dimensional space (Laine and Karras 2010), which can be memory-intensive and has driven recent research towards exploring more memory-efficient 3D representations (Yang 2020). **Multi-view image projections** are typically used for visualisation in 2D settings (Carlbom and Paciorek 1978, Su et al. 2015, Madsen and Madsen 2016). It has found extensive applications in neural rendering (Mildenhall et al. 2021), robotics (Sünderhauf et al. 2018), etc. **Implicit field** is a scalar field defined by an implicit function, such as a field value represents the distance from a point to the surface of a 3D shape (Park et al. 2019, Mescheder et al. 2019). Implicit fields are often used in physics-based simulations and shape modelling, where it's important to capture the properties of a 3D shape without explicitly defining its boundaries. **Primitive** uses a collection of simple parts or components to represent 3D shape, and often used to approximate the geometry (Zou et al. 2017). 3D **point cloud** is an unordered set of point coordinates in space, sampled from the surface of an object. These 3D data types apply to different research scenarios and application requirements (Qi et al. 2017a, Berger et al. 2017).

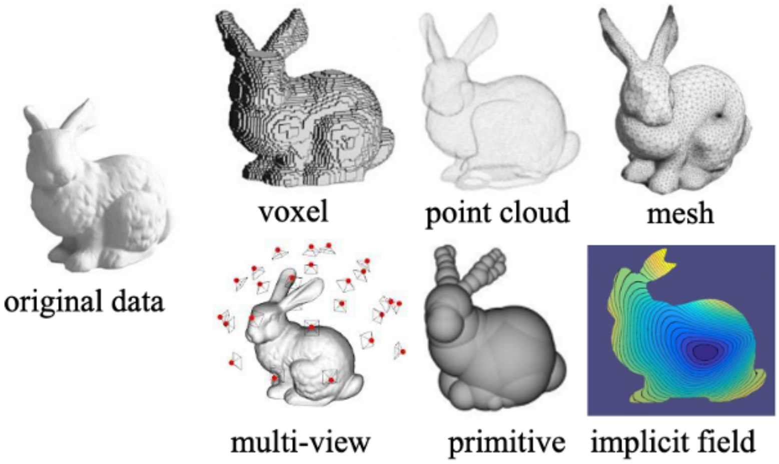


Figure 1.2: An illustration of different data representations on the Stanford Bunny benchmark.

Among these representation formats, point clouds are the closest representation of raw sensor data without any quantization loss (as in volumetric representations) or projection loss (as in multi-view representations) (Qingyong 2022). Besides, point clouds offer a straightforward representation, which is capable of vividly visualizing 3D shapes and facilitating direct editing operation (as in implicit field and primitive) (Arikan et al. 2013). A point cloud is a collection of spatial coordinates of points, which avoids the complexity of meshes (such as polygon definition and connectivity). Given these advantages, point cloud has become increasingly popular for 3D applications due to its inherent advantages. The rise in point cloud-based applications has created a pressing demand for advanced 3D understanding methodologies and a large-scale supply of 3D dataset.

Generally speaking, point cloud data understanding and analysis involves various tasks such as segmentation, classification, detection (Wu et al. 2016a, Xiao et al. 2023). The application of deep learning has revolutionized the field of 2D vision, leading to significant advancements in tasks such as image segmentation, classification, object detection and

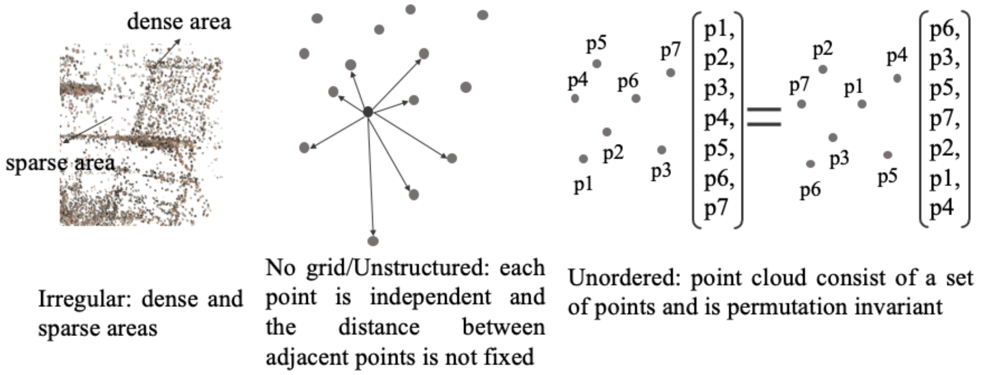

dense area
sparse area
Irregular: dense and sparse areas
No grid/Unstructured: each point is independent and the distance between adjacent points is not fixed
p5 p7 p4 p6 p2 p1 p3
p1, p2, p3, p4, p5, p6, p7
=
p2 p4 p7 p1 p3 p6 p5
p6, p3, p5, p7, p2, p1, p4
Unordered: point cloud consist of a set of points and is permutation invariant

Figure 1.3: Challenges for 3D point cloud learning (Bello et al. (2020)). (a) Irregular: dense and sparse areas; (b) No grid/Unstructured: each point is independent and the distance between adjacent points is not fixed; (c) Unordered: point cloud consists of a set of points and is permutation invariant.

image generation. Inspired by these successes, researchers have sought to leverage deep learning technology to similarly transform the understanding and analysis of 3D point cloud data (Xiao et al. 2023). These tasks are fundamental in deriving actionable insights and extracting valuable information from point cloud data for various applications, including segmentation, classification, registration, and generative modelling. However, the extension of deep learning methods from 2D to 3D data is not straightforward due to inherent differences in 3D point cloud structure. As illustrated in Figure 1.3, the primary challenge lies in the irregular and sparse nature of point cloud data, making it difficult to develop data representations and processing techniques that can robustly handle the attributions. Furthermore, deep learning is generally accomplished by deep neural networks, which rely heavily on substantial labelled data to optimise parameters (Shi et al. 2021). Unfortunately, obtaining annotations for 3D point cloud data, especially for point-level annotated, also presents significant difficulty (Shi et al. 2021). This difficulty in annotation is primarily due to expensive and resource-intensive (Monica et al. 2017). Given these considerations, it is beneficial to solve the issues of reducing the acquisition and annotation cost of 3D point cloud dataset.

The 3D deep generative model is a probabilistic model that has garnered significant interest among researchers since its inception (Carlson 1982, Tangelder and Veltkamp 2008, Van Kaick et al. 2011b, Chaudhuri et al. 2011, Xue et al. 2012, Kalogerakis et al. 2012, Kar et al. 2015, Bansal et al. 2016, Blanz and Vetter 2023). This model is regarded as capable of synthesising the structure of 3D shapes by capturing the flexible probability distribution that defines high-dimensional data (Kalogerakis et al. 2012). The anticipation of this model is due to its potential to provide a solution to the challenges associated with acquiring complex 3D shapes. Point cloud generative model based on deep learning is a type of probabilistic model that can synthesise new data based on what it has learned from existing dataset. This work focuses on 3D point cloud generation, specifically in the context of generating new realistic and point-level labelled samples of point cloud. Point cloud generative models crucially facilitate 3D data generation by learning from and reconstructing original data in existing data repositories to produce new synthetic instances. This is of particular importance as it addresses the challenge of scarcity and diversity in point cloud dataset. Moreover, point cloud generative models enable the exploration of new samples of point cloud data. By learning to generate new instances, these models capture key characteristics and patterns in the data, which can be leveraged for point cloud understanding.

Nevertheless, there are several limitations to extending current techniques for generating point cloud and corresponding point-wise semantic labels.

Limited generation ability of existing generative models.

Due to methodological limitations, many prior techniques have employed VAE-based (variational autoencoder) or GAN-based (generative adversarial network) generative models to tackle the task of 3D point cloud generation. VAE-based models typically need affinity architecture, that is the autoencoder model, and auxiliary objectives to ensure accurate likelihood computation (Song and Ermon 2019). GAN-based models, on the other hand, achieve good results on various evaluation criteria, but it requires adversarial training, which can result in an unstable training process and mode collapse (Ho et al. 2020). In practice, VAE-based and GAN-based point cloud generation models are generally supervised by either the Chamfer Distance (CD) or Earth Mover's Distance (EMD) to optimise the parameters of the network. However, CD loss prioritises accuracy instead of the uniform distribution of shape and does not calculate the matching distance between ground truth and synthesis. EMD loss considers the overall distribution of the point cloud but is computationally expensive. While recent point cloud diffusion-based generation models (Zhou et al. 2021, Luo and Hu 2021) can address these issues, the point cloud diffusion generation models are still in their early stages and have not been fully developed yet. These methods realise high-quality point cloud generation with a stale training process and can be easily applied to various tasks, such as point cloud completion, reconstruction, etc. The initial successes of these methods highlight the need for a more efficient approach to point cloud generation.

Inadequate accuracy of generated point cloud and its label pairs.

On the one hand, the present approach to producing deep point cloud generative models has opened up the possibility of generating 3D point clouds at a reduced initial expense. However, it is crucial to acknowledge that useful point cloud dataset depends on the accurate annotated point clouds and shapes. On the other hand, although related technologies such as point cloud part segmentation (Xu et al. 2010, Wang et al. 2020, Zhao et al. 2021b), domain adaptation (Qin et al. 2019), have been well studied in annotate point cloud fields and endeavoured to solve the shortage of annotate point cloud, their potential in actual point cloud generation remains unknown, as there is no evident validation of these methodologies on synthetic dataset. Furthermore, most existing approaches focus on the generation of point cloud dataset emphasise semantic part-based control. The most typical guideline is to divide the generation process of GAN-based point cloud generative models into two stages (Kol et al. 2022, Yang et al. 2021, Shu et al. 2019a). The structural key points or structural trees of the three-dimensional shape are generated in the first stage, and then the annotation of each point is inherited from these structural features. Due to the irregular shape distribution of the point cloud, the accuracy of the annotation of each point generated by this method is insufficient. Additionally, these methods require a substantial amount of annotated data to facilitate point cloud dataset generation, potentially hindering their practical application. At present, there is relatively little work on the direct construction of annotated point clouds.

Lack of generated point cloud and its label pairs evaluation method.

In addition, the quality of the generated point cloud and its corresponding label pairs is not evaluated in the current generation methods, so it is difficult to directly construct a large dataset and make it available for downstream tasks. It is crucial to enhance and assess the quality of the produced point-label pairs and demonstrate their validity in order to form them as the point cloud dataset. Furthermore, given the expensive nature of annotations, the evaluation method should be capable of ensuring usable point cloud dataset with annotated labels, even with only a limited number of annotated samples available.

This research aims to address the aforementioned challenges and limitations in order to contribute to the development of point cloud generative models and dataset generation techniques. Furthermore, addressing the limitations of existing methods could lead to more effective solutions for real-world applications, benefiting both academia and industry.

1.2 Research Questions

Although related technologies such as point cloud part segmentation (Xu et al. 2010, Wang et al. 2020, Zhao et al. 2021b), domain adaptation (Qin et al. 2019), and part-aware point cloud generation (Shu et al. 2019b, Yang et al. 2021, Li et al. 2021c) made great efforts to address the problem of annotated point cloud data lacking from various perspective. All indicators suggest that the automatic generation of dataset is still in its early stages. This presents a challenge in obtaining precise part labels and realistic point clouds for high-quality point-label pairs. This thesis proposes a solution to the challenge of point cloud and its corresponding point-level semantic label generation using a diffusion-based generative model. The research question is about how the diffusion generation model can be adopted to enhance the point cloud synthesis, with a specific focus on the effectiveness of the generation and automatic annotation. The development of this research will help to reduce the production cost of the point cloud dataset and open new views for related methodologies. This thesis will develop experimental approaches to validate the proposed methods and verify them in different setups. This will prove the effectiveness of the proposed approaches. There are sub-questions related to this that will be studied:

- How to develop a diffusion-based generative model to enhance the quality of the generated point cloud? Despite the significant advancements in point cloud generative models, there remain inherent limitations when it comes to synthesising high-quality point clouds. As mentioned in the last section, emerging diffusion-based generative models have a theoretical foundation that may surpass the other prevalent paradigms of generative models. However, the domain of point cloud generation, specifically regarding to diffusion-based models, remains a novel field. Therefore, further exploration into the fundamental principles of point cloud diffusion models, and enhancing their capability to generate high-quality data, presents a significant challenge.

- Once the point cloud diffusion generative model is established, the subsequent question pertains to how to leverage it for point-label pair generation by innovating annotation techniques. The point cloud dataset comprises two components: the point cloud shape and the point-label correspondence. Because the generative model inherently captures the structure, pattern, and concept of the 3D model, the interwoven relationship between the point cloud shape and point-label correspondence illuminates a promising direction for exploiting the generative model to create point-label pairs. This exploration goes beyond solving the difficulties in acquiring annotated point clouds, it also helps in understanding of the implicit features of point cloud generative models. Thus, it enhances the value and meaning of developing such methods across different levels.

- How to develop an evaluation approach to assess the quality of the generated point cloud and its corresponding labels, thereby expanding their utility in downstream tasks? As a probabilistic model, a generative model can generate a large number of samples. However, it inherently has the risk of generating point clouds with indistinct shapes or establishing incorrect point-label correspondences. A dataset consisting of these unqualified results will fail to be used for downstream tasks. These characteristics bring another challenge into focus: how to design an automated approach to assess the generated point-label pairs, and be able to integrate seamlessly with the point-label pair generation pipeline? Furthermore, this approach should provide a mechanism for filtering the unqualified generated point-label pairs, thereby ensuring their effectiveness and applicability as augment datasets in downstream tasks.

1.3 Research Objectives

Following the research questions discussed above, this thesis aims to achieve three major objectives:

Improved Point Cloud Generative Model: The primary aim of this research is to develop an advanced point cloud generative model capable of generating higher-quality point clouds. The present study hypothesises that the point cloud generation can be treated as a transition of point cloud and prior distribution, which can be simulated by a stochastic process. The hypothesised that an improved generative model in which leverages Stochastic Differential Equations (SDEs) to solve the stochastic process. Thus one objective here is to exploit a point cloud generative model combined with SDEs. This model indicates the ability that effectively addresses the limitations of existing VAE or GAN-based point cloud generation methods.

Point-label Pairs Generation based on Diffusion-based Point Cloud Generation Model. The second objective is to address the challenges of generating accurate and valid annotated point clouds. To achieve this goal, this study proposes a novel method for creating point-label correspondences. The proposed approach leverages the intermediate features of point cloud generative models to train a semantic predictor. This objective builds upon the first objective to create point-wise semantic labels based on obtained high-quality point cloud shapes. This involves constructing a pipeline for point-label pairs produced.

Evaluation and Filtering of Generated Point-label Pairs. The third objective of this research is the evaluation and enhancement of generated point cloud dataset. This objective is met by introducing a novel approach based on the query strategy of active learning. Specifically, this approach computes an uncertainty score for annotated point clouds, serving as a quantitative metric for evaluating their quality. This uncertainty score is then utilised to filter out unsatisfactory point-label pairs, guaranteeing the quality and validity of their application in downstream tasks. The designed module can be integrated with the point-label pairs generation pipeline.

1.4 Contributions

The thesis proposes an advanced technique for generating and annotating point clouds. It is launched around the diffusion-based point cloud generative model and its application. The contributions of the approach of this thesis are:

- **A Point Cloud Diffusion Generative Model.** This research proposes to regard point cloud generation as a continuous time stochastic process. By leveraging Stochastic Differential Equations (SDEs), this research presents a promising generative model for point clouds, overcoming the limitations of current VAE-based and GAN-based methods. Compared to other point cloud diffusion generative models, the proposed approach put forward a flexible and refined generation process, coupling with techniques such as Markov Chain Monte Carlo sampling and time encoding. Experimental results have demonstrated that the proposed model not only generates expressive point clouds but also achieves competitive results when benchmarked against other methods. This lays the foundation that produces fidelity point cloud for point cloud dataset construction. A detailed discussion of the methodology will be provided in subsequent chapters.

- **Generating annotated point cloud based on point cloud generation model.** To address the annotated point cloud generation, this study offers an advanced approach for creating point-label correspondences. Instead of struggling with network architecture or structural representation of shape, this method observes the generation pattern and implicit feature of the diffusion-based point cloud generation model. The proposed method enhances the efficiency and accuracy of point cloud data generation. Furthermore, this method provides an investigation approach for the characteristic of point cloud generative model. This approach can be applied in prevailing baselines of point cloud generative models and visually reveal whether exploring the features inside the generative model can generate point-label pairs. The comparison experiments further illustrate the advantages of the diffusion-based point cloud generative model. Particularly, experimental results have indicated the effectiveness of the proposed methods and the usability of the generated point-label pairs. The generated point cloud dataset has been made publicly accessible, thus providing a valuable resource to the wider research community.

- **Evaluation and validation of the generated point-label pairs.** This method proposes a filter technique aimed at improving the accuracy and quality of generated point cloud and its corresponding labels. With the support of this filtering mechanism, the validation of the produced point-label pairs has been comprehensively discussed and examined. Specifically, the experimental setup for few-shot learning provides evidence of the efficacy of the proposed method. Experimental results verify the validity of the generated point cloud dataset which can argument segmentation task performance.

1.5 List of Publications

The research of this thesis has led to the following publications in peer reviewed journals and conferences:

- Tingting Li, Meili Wang, Xiaoxiao Liu, Hui Liang, Jian Chang, Jian Jun Zhang. 2023. Point cloud synthesis with stochastic differential equations. Computer Animation and Virtual Worlds, p.e2140.

- Tingting Li, Yunfei Fu, Xiaoguang Han, Hui Liang, Jian Jun Zhang, Jian Chang. 2022, October. DiffusionPointLabel: Annotated Point Cloud Generation with Diffusion Model. In Computer Graphics Forum (Vol. 41, No. 7, pp. 131-139).

Other publication out of the scope of this thesis:

- Jiahui Mao, Tingting Li, Feiyu Zhang, Meili Wang, Jian Chang, and Xuequan Lu. 2021. Bas-relief layout arrangement via automatic method optimisation. Computer Animation and Virtual Worlds, 32(3-4), p.e2012.

1.6 Outline of Thesis

This chapter presents the background and motivation of the point cloud annotation method based on the diffusion generative model and lists the contribution of this work. The following chapters of this thesis are organised as follows:

- Chapter 2 reviews the related works and technology for point cloud descriptor learning, point cloud generative models and point cloud segmentation without strong supervision.

- Chapter 3 introduces the preliminaries of stochastic process, stochastic differential equations and Markov Chain Mento Carlo sampling. This chapter shows how stochastic differential equations and Markov Chain Mento Carlo sampling is applied to point cloud generation.

- Chapter 4 presents an investigation approach based on the point cloud generative model and demonstrates the discriminability of the intermediate feature. This chapter introduces the implementation details of point-label pairs generation and shows the experimental results.

- Chapter 5 proposes a filter approach based on uncertainty measurement and how the proposed point cloud pairs generation method is improved. It illustrates the effectiveness of the proposed method and verifies the validity of generated point-label pairs as dataset.

- Chapter 6 concludes this thesis and suggests some possible future directions of the current work.

Chapter 2

Literature Review

In this thesis, the research of point cloud generation is conducted using the deep learning method. To begin with, this chapter reviews the current methods for extracting point cloud feature description operators and analyzes the characteristics of these methods in Section 2.1.

Then this chapter surveys the current development of point cloud generation models and makes a comprehensive comparison and analysis according to different training paradigms (Section 2.2).

Finally, to follow the course of this research, which is to generate point-level semantic labels for point cloud effectively, this chapter revisits the development of point cloud semantic segmentation without strong supervision (Section 2.3).

2.1 Point Cloud Feature Descriptor Learning Methods

This section analyses and compares point cloud feature operator extraction methods according to the evolution progress. The discussion begins with a summary of traditional methods, which are centred around the use of computational geometric operators for point cloud feature extraction. Then this section shifts to a review of methods that utilise deep neural networks for point cloud feature descriptor learning. Specifically, this section categorises the current methods according to the operator types that are used in their methods. The existing methods fall into five broad categories: multi-views projection-based methods, voxel-based methods, graph-based methods, domain point convolution-based methods, and point set-based methods.

2.1.1 Traditional Methods for Point Cloud Feature Descriptor

In traditional point cloud feature descriptor extraction algorithms, most methods are designed to adapt to the particular needs of specific tasks, such as 3D shape segmentation, retrieval, and registration (Qi et al. 2017a, Tangelder and Veltkamp 2008).

These feature descriptors are crafted in alignment with the unique requisites of visual tasks, and the inherent statistical and geometric properties of point cloud data, such as distinctive variations. They can generally be classified into intrinsic descriptors (Wang and Solomon 2019) (e.g. WKS (Aubry et al. 2011)), HKS (Bronstein and Kokkinos 2010a) and extrinsic descriptors (Wang and Solomon 2019) (e.g. PFH (Rusu et al. 2008)), D2 (Osada et al. 2002), Inner Distance (Ling and Jacobs 2007), Spin Image (Johnson and Hebert 1999), etc.). Depending on their encoding range, point cloud descriptors can also be classified as local descriptors (e.g. PS (Chua and Jarvis 1997), USC (Tombari et al. 2010), FPFH (Rusu et al. 2009)) or global descriptors (e.g. VFH (Rusu et al. 2010), CVFH (Aldoma et al. 2011)). Local descriptors encode the geometric properties of a point's local neighborhood, while global descriptors encode the overall shape of the entire point cloud. Local descriptors are more compact but less distinctive, while global descriptors have a wider encoding range but are more time-consuming to compute.

2.1.2 Deep Learning Methods for Point Cloud Feature Descriptor Learning

Currently, deep neural networks emerged and become a popular direction for this research field. For a specific task, the design and selection of the best feature descriptors for 3D point cloud data and feature selection using deep learning techniques are at the forefront topic of 3D point cloud processing research (Fang et al. 2015).

However, in contrast to the regular arrangement of 2D images, 3D point cloud data is characterised by disorder and irregularity in space. Therefore, the operators of point cloud objects and point cloud scenes should be rotationally invariant and scale invariant. These two characteristics make it difficult to directly apply the network structure and experience of 2D images to the field of point cloud processing. For this reason, researchers within the field have attempted to organise and represent point cloud data in different ways in order to extend deep learning techniques to feature extraction from point cloud data. The existing methods fall into five broad categories: multi-view projection-based methods, voxel-based methods, graph-based methods, domain point convolution-based methods and point set-based methods.

2.1.2.1 Multi-view Projection-based Methods

While 2D images have a regular pattern and positions, the main character of 3D point cloud data is disorder and irregularity in space. Besides, point cloud objects do not have a fixed orientation or scale. These two fundamental characteristics cause difficulty to directly apply the network structures and techniques used for 2D images to point cloud processing. Many researchers in the field have attempted to organise and model point cloud data in different ways to extend deep learning techniques for extracting features of point cloud data.

The multi-views projection-based approach first projects the 3D point cloud in multiple views and estimates a set of corresponding depth images, as shown in Figure 2.1. Then the approach performs feature extraction directly on each projected image using a Deep Convolution Neural Network (CNN) designed for 2D images, and subsequently fuses the extracted feature maps as the feature descriptor of point cloud. Therefore,

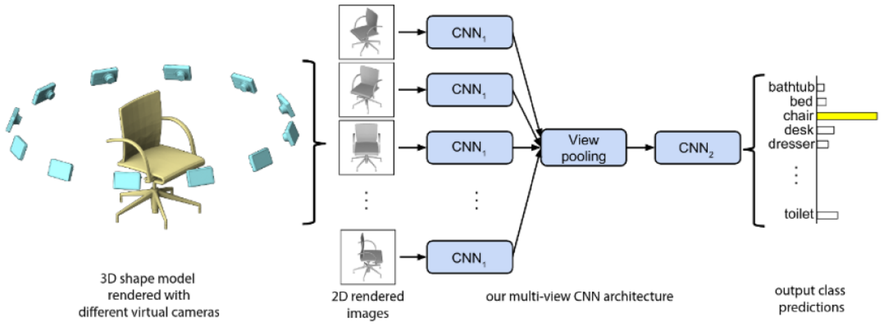


Figure 2.1: Illustration of multi-view projection-based approach of point cloud (Su et al. 2015)

the main challenges of these methods are how to determine the number and view angle of the projected images and how to fuse the features of the depth images to obtain the most expressive feature.

Su et al. (2015) proposed a method that projects 3D objects at fixed view angles and takes the 2D projected images for feature extraction. The global feature of the point cloud is obtained by maximum pooling the set of features of 2D projected images. They apply the method to classification tasks and verify the effectiveness. Yu et al. (2018) emphasised the importance of relationships between projection images from different view angles and proposed the use of relational networks to connect corresponding features between different projection images, such as region-region and view-view. This method aggregates the features of these views to obtain a global feature of the 3D shape. Lawin et al. (2017) believed that the features 3D shapes is in alignment with the feature of 2D projected images. They propose to project the 3D point cloud as a 2D image from multiple virtual views and then use a 2D convolutional network to predict the semantic labels for each projected image. Finally, the predicted results are mapped back to 3D space to obtain the segmentation labels of the point cloud. Tatarchenko et al. (2018) proposed a method based on tangent convolution for the feature extraction method of 3D point clouds. In their method, the multi-view images are the tangent planes of each point in the point cloud, and feature extraction of the tangent plane is used as a feature of the point cloud, i.e. by tangent convolutions.

The main goal of the above methods is to solve the problem of how to apply 2D convolution to 3D point clouds, which has been relatively well developed. The key designs of these methods are to reduce occlusion caused by 2D projections or to increase the linkage of projections and refine point cloud structure and geometric features.

2.1.2.2 Voxel-based Methods

The voxel-based method leverages the voxel representation of 3D shape to extract the feature descriptor. Voxels divide space into 3D grids. A voxel shape is defined by the grids that contain the shape. The main advantage of voxel representation is that it can directly be used by a 2D convolution operator to a 3D vision field.

Maturana and Scherer (2015) first proposed transforming 3D point clouds into regular 3D grids and applying 3D neural convolution to learn point cloud features. Wu et al. (2015) represented 3D point clouds as probability distributions of binary variables on a 3D voxel grid. Although this representation approach has achieved good performance for various small-scale point cloud recognition tasks, using voxel to represent the 3D point cloud space requires high computational cost. This is because the 3D point cloud is very sparse and contain surface information of the object. Many methods have proposed various ways to improve storage costs and reduce computing costs. Riegler et al. (2017) used a hybrid grid octree structure to hierarchically partition point cloud and encodes the octree structure using a bitstring representation. The indexes of feature vector for each voxel grid can be obtained. This method significantly reduces the amount of memory and computation required for training 3D

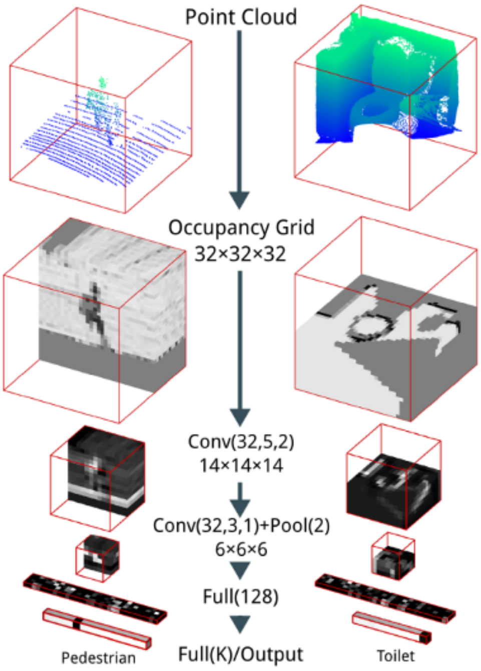


Figure 2.2: Illustration of the VoxNet Architecture (Maturana and Scherer 2015)

convolutional networks. Le and Duan (2018) proposed a hybrid neural network, PointGrid, that efficiently processes point cloud data. PointGrid integrates point and voxel representations. Huang and You (2016) proposed using 3D convolutional networks to predict voxel-level labels for 3D segmentation tasks. The semantic label of each voxel determines the label of the point within that voxel grid. However, the accuracy of this method is not high enough. To obtain a more accurate segmentation effect, Tchapmi et al. (2017) proposed SegCloud, which adopts trilinear interpolation and conditional random field strategies to improve Huang and You (2016)'s method. In order to achieve more efficient network training, Graham et al. (2018) proposed the Submanifold Sparse Convolutional Network (SSCN). This method extracts 3D point cloud features on voxels containing points while excluding empty space. This reduces computations and speeds up training. SSCN requires fewer resources and less time to train a network for 3D point cloud feature learning.

In summary, voxel-based methods allow 3D convolutions to be directly applied to point cloud data. However, this type of approach needs to make trade-off between fidelity and computer efficiency. Because coarse voxel will lead to loss of detail, while fine voxel grids require substantial computing resource.

2.1.2.3 Domain Point Convolution-based Methods

The method based on domain point convolution defines the convolution operation according to the spatial distribution (generally continuous distribution or discrete distribution) of the local adjacent points of the point cloud. The output of the convolution kernel is a weighted combination of adjacent point features. The weight of adjacent points is determined according to their spatial position relative to the current point. Li et al. (2018b) put forward PointCNN, which converts point cloud data into latent features and reorders points using an X-transformation matrix to capture the local feature of each point. This method then applies typical convolutional operators to the transformed features. This method samples a random point and its k-nearest neighbour (k-NN) points from local neighbourhoods. These neighbourhoods undergo an X-transformation block before passing through a PointNet-like multi-layer perceptron (MLP). The X-transformation matrix facilitates the reordering of points within local neighborhoods before applying typical convolutional operators. Xu et al. (2018b) defined the convolution of the point cloud as the product of a step function and a Taylor polynomial on k nearest neighbours. The step function obtains geometric information by encoding the local geodesic distance, and the Taylor polynomial is used to capture complex geometric changes in the 3D point cloud. Esteves et al. (2018) represented 3D objects as multivalued spherical functions and constructed spherical convolution networks.

Liu et al. (2019b) proposed RS-CNN, which uses multi-layer perceptrons (MLP) to learn the mapping from low-level features to high-level features between local neighboring points and determine the weights of neighboring points. PointConv, a method proposed by (Wu et al. 2019), transforms a convolution kernel as a nonlinear function of the coordinates of neighbouring points. The kernel includes weight and density functions that define the influence of the points on each other during the convolution operation. KPConv (Thomas et al. 2019) approximated continuous convolution kernels by using discrete kernel points.

2.1.2.4 Graph-based Methods

Many approaches also suggest reconstruct the 3D point cloud based on the structure of the graph, and then defining a convolution operation on the graph to learn features. Te et al. (2018) came up with an RGCNN for building graphs for point clouds. The graph's edge is the connection between each point and all the other points, which construct a set. During training, the graphs for point clouds get updated at each layer by adjusting the Laplacian matrix. Simonovsky and Komodakis (2017) regarded each point as a vertex of the graph, and the directed edges of the graph are the connections between each vertex and all its neighbours in the graph. These approaches are relatively straightforward but demand significant computing resources. Wang et al. (2018) proposed LocalSpecGCN, a spectral convolution network that operates on local graphs. The local graph is constructed from the k nearest neighbours. This method can exploit local structural information and require less computational resources than previous approaches. Zhang and Rabbat (2018) proposed an alternative graph construction strategy. They built the graph based on the k nearest neighbours of a point cloud, weighting each edge using a Gaussian kernel. They defined convolutional filters as Chebyshev polynomials in the graph spectral domain. Global pooling and multi-resolution pooling captured global and local features of the point cloud. Wang et al. (2019) proposed the DGCNN, which constructs a graph using the latent features of each point. Similar to EdgeConv (Wang et al. 2019), this approach first used a multi-layer perceptron

(MLP) to extract the features of each edge in the graph. It aggregates these features across channels to serve as the features of neighbouring points. Liu et al. (2019a) put forth a Dynamic Points Agglomeration Module or DPAM, which relies on graph convolution. This approach simplifies sampling, clustering, and pooling points into a single step by multiplying an agglomeration matrix with a points feature matrix.

2.1.2.5 Point-Set-based Methods

Point-set-based methods aim to process 3D point clouds as point sets. The key intuition of this type of method is to use multiple shared multi-layer perceptrons (MLPs) to extract the feature of each point independently and acquire the feature descriptor of the point by aggregation functions. Point-set-based methods are simple and effective and have gradually become the mainstream method of 3D point cloud representation learning.

Qi et al. (2017a) proposed PointNet, which is the baseline of point-set based method. This method uses a multi-layer perceptron to extract the features of all points of the point cloud to a high-dimensional feature space and then aggregates the features through maximum pooling to obtain a global description of the point cloud. PointNet pioneered a simple and powerful approach to 3D point cloud representative learning, which inspired many subsequent methods. The follow-up work PointNet++ (Qi et al. (2017b)) improves upon PointNet by addressing its limitation of ignoring local features of point clouds. This method extracts features of the local neighbourhood points and uses a hierarchical network structure to aggregate the extracted features.

Along with the same intention, Hu et al. (2020) used point features and positions as input to assist local feature aggregation. In addition, this method gradually increases the scope of the receptive field for each point, thereby it can effectively preserving the geometric details of the point cloud and the integrity of the position information. Lin et al. (2019) proposed to build an index table for point cloud and features spaces learned by PointNet and this index table can accelerated the inference process. Moreover, many specialised networks have been presented to capture the global feature for each point and local structures in the meanwhile, including methods based on neighbouring feature pooling (Zhao et al. 2019a), attention-based aggregation (Yan et al. 2020), and local-global feature concatenation (Duan et al. 2019, Sun et al. 2019).

2.1.2.6 Transformer Structured based Methods

As the Transformer-structured networks achieve great effect in the natural language processing (NLP) area, many studies in the 3D vision community have been devoted to applying Transformer to point cloud feature learning. Since the self-attention mechanism is the primary factor of Transformer's success, Point Attention (P-A) (Feng et al. 2020), PCT (Guo et al. 2021), and ShapeContextNet (Xie et al. 2018) all proposed point cloud feature learning frameworks that combine point-based and self-attention mechanisms. These methods achieved excellent performance compared with the state-of-the-art convolutional networks. Building on their works, subsequent research proposed combining Transformer models and point cloud feature learning in a more thorough manner.

Zhao et al. (2021a) introduced a vector subtraction attention operator in the point cloud Transformer network. Compared with the commonly used scalar attention, vector attention supports adaptive modulation of individual feature channels, which has better representative learning ability. This representation appears to be very beneficial in 3D data processing. In 3DCTN, Lu et al. (2022) investigated and compared different self-attention operators in 3D Transformer, including scalar attention and different forms of vector attention.

In summary, this section has provided a comprehensive review of point cloud feature descriptor learning methods, tracing their evolution path.

As these methods have progressed, their capabilities have become increasingly powerful, algorithms more robust, and accuracy higher. The introduction of deep learning has had a significant impact on the field of point cloud feature learning. The evolution reflects technological advances and a deepening understanding of the point cloud data, opening up new views for research in this field.

2.2 Point Cloud Generation Methods

3D shape generation is a crucial problem in computer graphics and computer vision, with numerous applications in fields such as virtual reality, video games, and architecture. In order to generate accurate and realistic 3D shapes, researchers have developed a variety of algorithms, ranging from traditional methods based on computational geometry principles to more recent data-driven approaches that leverage the power of deep learning.

Traditional computer graphics usually start with computational geometry-based principles to solve 3D shape generation problems, such as implicit surfaces, parametric surfaces and triangle meshes (Bloomenthal and Wyvill 1990, Velho et al. 1999, Tavakkoli and Dhande 1991, Shum et al. 1996, Stal and Turkiyyah 1996, Wong et al. 1989). Their common algorithm for this problem is to use implicit modelling for creating and editing 3D models in an interactive and intuitive way and then generating shapes with desired properties according to specific design or engineering requirements.

Generally, the algorithms follow a set of processing steps that include defining a shape representation, such as a set of implicit surfaces, and then using optimisation techniques to explicitly optimise the 3D shape to meet specific criteria. These methods lay a solid foundation for the basic concepts and techniques used in traditional 3D shape generation algorithms and serve as a starting point for understanding this challenge.

2.2.1 Traditional Data-driven Methods

With the enrichment of 3D digital assets, numerous pioneering approaches have proposed the utilisation of data-driven methods to address the synthesis of 3D shapes. Velho et al. (1999) utilised a facial database to construct an average face deformation model. Given a new facial image, they aligned and combined this image with the model, adjusting relevant model parameters and deforming the model until the discrepancy between the model and the facial image was minimised. During the generation, texture optimisation adjustments were performed to finalise the facial modelling process. Kalogerakis et al. (2012), Huang et al. (2015), and Chaudhuri et al. (2011) proposed the assembly-based methods to reconstruct part-oriented object generation and extend the generation of large-scale 3D databases. Their methods generate new 3D digital models by retrieving and merging parts from existing databases. However, these methods for object synthesis are non-parametric. These methods can only generate objects that have a similar geometry structure and make minor morphological modifications. They are also limited by external lighting conditions of source images and the characteristics of the object itself. The robustness of this type of method is insufficient, and adjustments to the reconstruction algorithm are needed to accommodate different reconstruction objects. Building on the work of Stal and Turkiyyah (1996), Sarkar and Chakrabarti (2014) put forward to explore the design space to develop potential design solutions. They employed twelve distinct search methods to identify the target domain of problems, create solutions and assess solutions.

To further exploit the potential of the database, several approaches (Kar et al. 2015, Kanazawa et al. 2018) made an attempt to combine deep learning and 3D shape synthesis. This method extracts the features of the 2D images and then takes them as priors to estimate 3D shapes. But their method is limited to a specific category and cannot handle complex geometries.

The methods mentioned above are limited by the generalisation of the algorithms, so it is difficult to use them directly to further develop 3D content creation and thus contribute to the development of the 3D visual community.

As the importance of creative support in 3D modelling has become more widely recognised, numerous improved techniques have been proposed to assist users in freely exploring the modelling space through 3D generation. In recent years, generative models have gained significant attention as a promising approach for this purpose. The potential of neural network-based generative models is particularly noteworthy, as they demonstrate the ability of neural networks to understand and generate plausible 3D objects. These models aim to generate novel point cloud representations of 3D objects, which can be used for various applications, including shape synthesis, data augmentation, and completion (Achlioptas et al. 2018). On the other hand, the growth of generative models has brought new opportunities for creative exploration in 3D modelling, while the development of more sophisticated generative models is likely to lead to further advances in this field.

Broadly, these techniques can be further distinguished by the training methodologies they employ, such as **Variational Autoencoder (VAE)-based, Generative Adversarial Network (GAN)-based, and autoregressive-based 3D generation, etc.**. The rest of this section focuses on the recent advancements in point cloud generative models, which are categorised based on their training methods.

2.2.2 Autoregressive based Methods

PointGrow (Sun et al. (2020)) is one of the notable works of Autoregressive-based methods. It estimated the probability of samples autoregressively based on previously generated points. However, this method is restricted to generating a fixed-dimension point cloud because it assumes a determinate order of point cloud. Cheng et al. (2022) also leveraged an autoregressive-based network to generate a point cloud. To improve the limitation of PointGrow, they propose to resort disorder point cloud by canonical mapping.

2.2.3 VAE based Methods

Generally speaking, VAE-based point cloud generation networks consist of encoders and decoders, as illustrated in Figure. 2.3 (Kramer 1991).

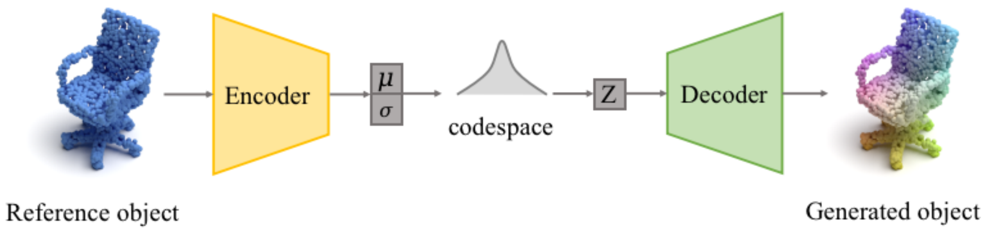


Figure 2.3: Illustration of point cloud VAE-based generation method

The VAE-based point cloud generation method assumes that the encoder can learn the distribution of feature descriptors of all point clouds in the dataset and align them to a prior distribution. The methods discussed in Section 2.1 show that a neural network can be trained to learn the feature descriptor of a point cloud. Building upon foundational research of feature learning, VAE-based point cloud generation networks leverage the rich informational content of feature descriptors to facilitate the synthesis of new point clouds. Meanwhile, a generator (the decoder) can be trained to produce points based on the feature descriptor. point clouds not present in the dataset are synthesized by randomly sampling from a prior distribution. The model is optimised by the variational inference and maximizing ELBO to ensure the final output is identical to the input. Usually, the VAE-based point cloud generative models are optimised by two loss functions:

$$\begin{aligned} L_{\text{rec}} &= \|x - \hat{x}\|_2 = \|x - G(z)\|_2 \\ L_{\text{KL}} &= D_{KL}(\mathcal{N}(\mu, \sigma)||\mathcal{N}(0, \mathbf{I})) \\ L_{VAE} &= L_{\text{rec}} + L_{\text{KL}} \end{aligned} \tag{2.1}$$

where L_{rec} denotes the reconstruction loss; L_{KL} denotes the similarity loss; $\|x - \hat{x}\|_2$ denotes the Euclidean (L2) norm of the difference between x and its reconstruction from decoder $G(z)$; $D_{KL}(\mathcal{N}(\mu, \sigma)||\mathcal{N}(0, \mathbf{I}))$ represents the Kullback-Leibler Divergence between a normal distribution with mean μ and standard deviation σ and the standard normal distribution; The overall loss is the sum of the reconstruction and similarity losses.

Girdhar et al. (2016) presented TL-Net, a method for learning a low-dimensional vector representation of 3D objects. Although TL-Net does not directly address point cloud generation, it focuses on generating objects from an image. The vector representation learned by TL-Net could potentially be utilised as input for point cloud generation methods for 3D data. Toward generation task, Achlioptas et al. (2018) proposed a method using a deep autoencoder (AE) architecture with adversarial training and demonstrated the effectiveness of the proposed method on various benchmark datasets.

Yang et al. (2018) proposed FoldingNet, which uses a graph-based encoder and a two-level cascade decoder to fold a grid within the global feature. It provides a baseline paradigm for point cloud generation that introduces folding a 3D grid to generate. The frame of this method is shown in Figure 2.4. Gadelha et al. (2018) proposed a hierarchical tree structure to capture both local and global features of a point cloud to process large-scale point clouds. The hierarchical structure was obtained from the downsample and upsample of the point cloud. In contrast to previous methods that predict the distribution of each point in a point cloud and the overall shape distribution, Yang et al. (2019) proposed PointFlow, which leverages continuous normalising flows (Papamakarios

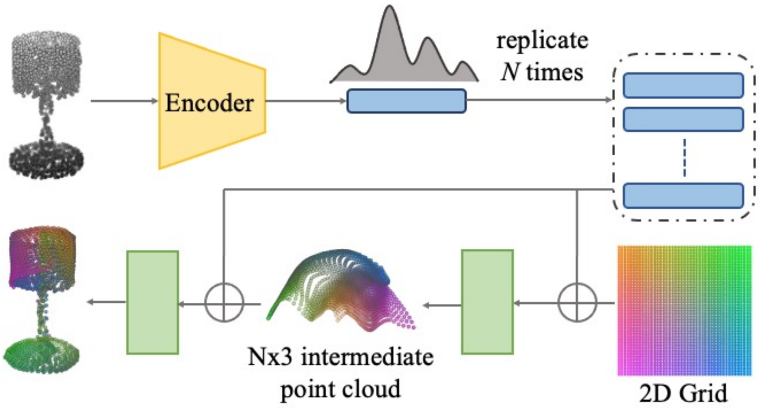


Figure 2.4: FoldingNet Architecture (Yang et al. 2018). The decoder is a two-cascade structure represented by a green square. Both decoders are folding and concatenating with a codeword, which is a replicated global latent variable.

et al. (2021)) to approximate complex distributions of a point cloud. This method can generate high-quality and diverse point clouds. It serves as one of the baselines of point cloud generation. Kim et al. (2020) proposed to use efficient "discrete" affine coupling layers instead of computationally expensive ODE solvers for training and generation in continuous normalising flows. This method designs two-level hierarchical latent space to approximate the distribution of shape and use softflow framework to optimise normalising flow. Their model can generate point cloud with fine-grained geometry and diverse shape. However, the training and generation efficiency of ODE solvers is very low. To address this, Klokov et al. (2020) proposed to use the discrete affine coupling layer to speed up both the training and generation of point cloud models. Nguyen et al. (2021) proposed using sliced Wasserstein distance instead of Chamfer distance to optimise the generator's parameters. Although this modification can lead to faster training and reduced computational cost, the quality of the generated point clouds may not surpass that of previous methods. Li et al. (2022a) proposed EditVAE, a framework for parts-aware point cloud generation that disentangles latent space into parts, enabling controllable parts editing while preserving inter-part dependencies. Likewise, Postels et al. (2021) adopted a mixture of normalising flows to generate a part-ware point cloud. However, the boundaries between the generated parts are not clearly defined, and the meaning is ambiguous. The motivation of Postels et al. (2021) is to fit the distribution of parts, while from the perspective of improving shape distribution fitting, PointOT (Zhang et al. (2022b)) proposed to use SCOT (semi-continuous optimal transportation) to optimise the latent space of point cloud. In 2022, Li et al. (2022c) proposed Primitive3D, a point cloud generation method with semantic labels. Unlike EditVAE (Li et al. (2022a)), which involves a two-stage training process, the training of this model prioritises learning the semantics of the generated parts. This is achieved through the acquisition of a semantic tree via data preprocessing.

With the proposal of the Transformer network structure, the ability of the network to process high-precision information within the module has increased significantly. The successful application of Transformers (Vaswani et al. 2017) in vision tasks has significantly enhanced the network's capability to process high-precision information within the module. Drawing on (Vaswani et al. 2017), Kim et al. (2021) presented SetVAE that incorporates a self-attention mechanism and a hierarchical latent space. This combination enables the model to attend to different points of disorder set and capture long-range dependencies, thereby enhancing the network's ability to process high-precision information. The proposed architecture overcomes the challenge of processing disordered point clouds, offering improved performance. It improves the generation results by improving the problem of processing disordered point clouds. Zhang et al. (2022a) introduced a two-level cascaded generation network structure with an attention mechanism. This architecture is based on the FoldingNet framework. The first network generates coarse point clouds, and the second network refines the output to produce a high-quality 3D point cloud model.

2.2.4 GAN based Methods

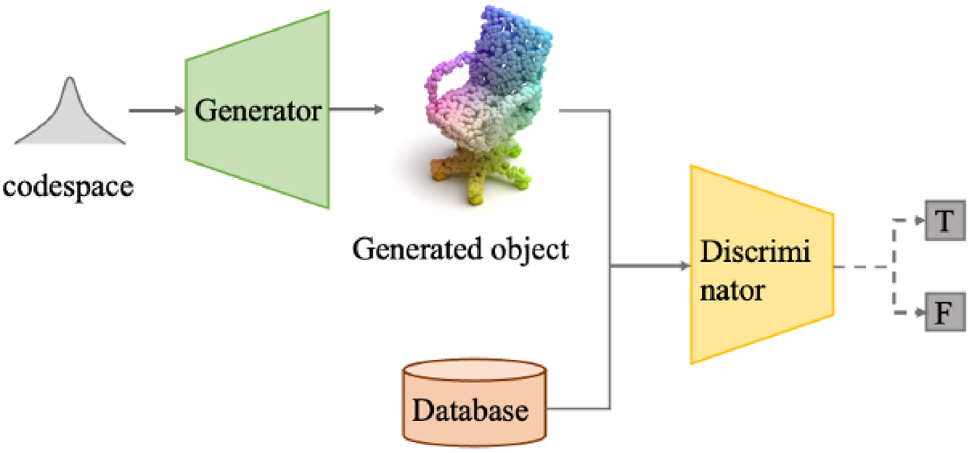


Figure 2.5: Illustration of point cloud GANbased generation method

As shown in Figure 2.5, a GAN-based generative model (Goodfellow et al. 2020) is a deep learning model consisting of a generator network and a discriminator network. The generator network generates new data samples resembling the training data, while the discriminator network distinguishes between generated and real data. During training, the generator tries to deceive the discriminator by producing realistic samples, leading to improvements in generating realistic data. This method of adversarial generation can be represented by the following equation:

$$\min_G \max_D L_{GAN} = \log D(x) + \log(1 - D(G(z))) \tag{2.2}$$

Wu et al. (2016b) first applied GAN to 3D generation; however, due to the limitations of the convolution kernel operator, they utilised voxels as the representation of generated data, which requires substantial computing resources. Subsequent research aimed to reduce the amount of computation necessary by building upon their approach. Li et al. (2018a) proposed the first GAN-based 3D point cloud generation method. Valsesia et al. (2018) designed a GAN model that combined multi-layer dynamic graph convolutions network (GCN) to extract local features from point clouds. The experimental results demonstrate that their generators could be transferred to many downstream tasks, such as segmentation and upsampling. However, the dynamic matrix of their method resulted in high computational costs.

Shu et al. (2019a) proposed Tree-GAN, which forms the latent variable of point cloud to a tree-structure representation and utilises graph to perform convolution on point clouds. While it is able to edit point clouds on the semantic level without prior knowledge, the accuracy of the label fell short of the expectation. To address this, Gal et al. (2020) extend it into a multi-root version, where each node could generate and control different parts of the point cloud. However, there was no clear classification boundary between different parts, meaning they lacked a clear semantic definition. Ramasinghe et al. (2020) introduced Spectral-GANs, which utilized spherical harmonics to represent point clouds and improve the quality of generated point clouds. Mo et al. (2020) proposed PT2PC that divided each 3D point cloud into a top-down tree structure and used an encoder to learn the semantic and structural information of each node. However, this method required preprocessing data and prior knowledge, limiting its applications. Li and Baciu (2020) proposed Self-Attention GAN for Point Clouds (SAPCGAN), based on Tree-GAN, which integrated self-attention-based graph learning with the tree structure of a point cloud to generate informative 3D point clouds. This method suffered from non-convergence and mode-collapse.

Hui et al. (2020) refined the graph network into a cascade architecture, progressively generating point clouds. Since their discriminators also adopt a cascade structure, the network parameters are significantly increased and heighten the complexity of optimisation. Arshad and Beksi (2020) proposed a method for conditional progressive point cloud generation. This method is designed to generate colored point clouds within a specific category. During each training iteration, a point transformer, which builds on graph convolutions, is used to progressively evolve the

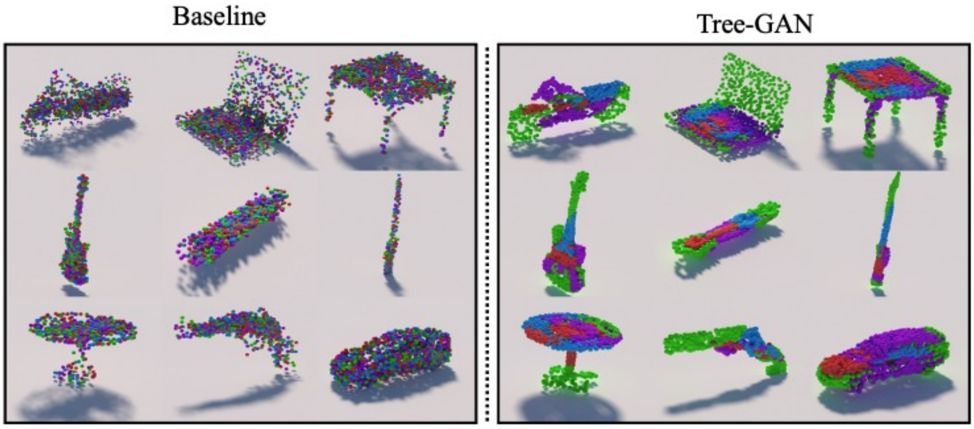


Figure 2.6: Unsupervised 3D point clouds generated by tree-GAN for multiple classes (Shu et al. 2019a).

coarse point vector into an increasing resolution point cloud while conditioning on class labels for multiclass point cloud creation. HSGAN (Li and Baciu 2021a) used Hierarchical Self-Attention GAN that hierarchically converts the prior distribution into a graph representation, and then transforms it into a 3D shape. SG-GAN (Li and Baciu 2021b) made attempt to generate a point cloud within aware topology representation. It designs a network with a hierarchical mixture model that combines self-attention with an inference tree structure to generate 3D shapes with accurate and compact geometric structures. Like wisely, Yang et al. (2021) proposed a method for generating a point cloud and corresponding part semantic labels from a different perspective. They use a two-level cascade generation approach, where the first cascade network generates key structural points, and the second network completes the entire point cloud based on those key structural points. The difference from SG-GAN is that it adopts two-stage generation and the key structure points generated in the first stage can be used to explicitly edit the shape of the point cloud. Li et al. (2021c) proposed SPGAN, which significantly improves the quality and accuracy of generated point clouds. They introduced an adaptive instance normalisation layer in the generator, based on the design of StyleGAN (Karras et al. 2020). Their network structure has become the mainstream backbone of point cloud work. Wen et al. (2021) proposed a framework using dual-network as a generator to improve the quality of point cloud generation. This dual network is constructed by two generators, where the first generator constructs a dense point cloud to sketch the fundamental geometric structure of the point cloud, and the second generator enhances this point cloud to produce a refined output. WarpingGAN (Tang et al. 2022) proposed a novel approach to 3D shape generation by formulating the process as learning a function that warps multiple 3D priors into different local regions of a 3D shape, guided by local structure-aware semantics. This method further utilises a stitching loss to eliminate gaps between different partitions of a generated shape corresponding to different priors, thus improving the quality of the generated shapes. Yang et al. (2022) proposed PC-cGAN, a method for synthesising high-quality point clouds with conditional information. It adopts the BranchGCN, an improved tree-structured graph network, to aggregate the features of the root of a tree and its neighboring points. This method can generate objects with specific desired categories and avoid intra-category hybridisation, that is where generated objects do not belong to a specific category and data imbalance problems. Instead of improving controllable generation and generation quality, Wang et al. (2023) proposed MSG-Point-GAN that uses a multi-scale progressive method to improve the stability of the training process.

Inspired by SP-GAN, Kim et al. (2023) proposed a point cloud editing method based on GAN inversion. It combines canonical mapping to retrain the encoder of SP-GAN and inputs the target point cloud to encoder to obtain the style feature vector. Their method improves the editing effect while retaining the high-quality generation results of SP-GAN.

2.2.5 Diffusion based Methods

Contrary to the GAN-based methods, which attempt to improve the point cloud generation model from the perspectives of delicate network structure or representation design, the diffusion models propose to realise the point cloud generation model by originating from the perspectives of training paradigm and generation theory. Denoising Diffusion Probabilistic Model (DDPM) (Sohl-Dickstein et al. 2015) proposed a bi-direction process. The forward process systematically and gradually destroys data distribution by injecting prior noise. The network learns the reverse diffusion process, yielding a flexible and tractable generative model of the data.

The success of the diffusion probabilistic distribution (Yang and Wang 2019, Nichol and Dhariwal 2021) approach has inspired many follow-up works that extend the diffusion probabilistic approach to 3D point cloud generation. They have viewed 3D point cloud generation as a probabilistic distribution transform. For example, Cai et al. (2020) learned the gradient of the log probability density with respect to point clouds and samples point clouds using Langevin dynamics. Luo and Hu (2021) exploited the Denoising Diffusion Probabilistic Models (DDPM) with fixed time-steps for point cloud generation. Following their work, many research works applied DDPM to various point cloud related-fields, such as point-voxel generation (Zhou et al. 2021), point cloud completion (Lyu et al. 2021a) and achieved remarkable results. LION (Vahdat et al. 2022) and 3D-LDM (Nam et al. 2022) both use the diffusion process in latent space, similar to the continuous normalising flow. This further explores the development of 3D point cloud diffusion generative models. Diffusion has an advantage over GAN in that the generalisation of the generation effect is higher. Lee et al. (2023) took advantage of this and explored the use of diffusion models to generate scene-level 3D point clouds, which consist of multiple objects and their spatial relationships. It successfully uses 3D generation models for the first time on scene-level 3D data.

2.3 Point Cloud Annotation Without Strong Supervision

Part segmentation refers to the process of subdividing a 3D shape into its constituent parts and assigning corresponding labels to these segments based on their semantics (as depicted in Figure. 2.7). The difficulty for part segmentation of 3D shapes is twofold. First, shape parts with the same semantic label have a large geometric variation and ambiguity. Second, the method should be robust to noise and sampling (Guo et al. 2020).

This section mainly reviews the methods that used a small number of point clouds with ground-truth labels to obtain a large number of annotation samples. There are many solutions for solving this task, such as semi/weakly supervised semantic segmentation, few-sample 3D point cloud segmentation, point cloud abstract representation learning, etc.

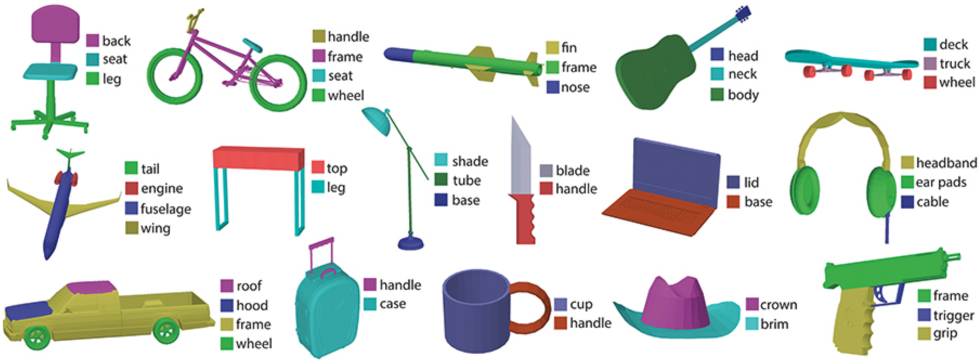


Figure 2.7: 3D point cloud part segmentation examples (Kalogerakis et al. 2017)

2.3.1 Traditional Methods for 3D shape Annotation

This section reviews early techniques for 3D shape semantic annotation, which generally used geometric features or other hand-crafted descriptors.

Traditionally, part segmentation has focused on extracting low-level features from shape over-segmentation, such as from shape features using edges (Rabbani et al. 2006) or surface attributes, such as normals, curvatures, and orientations (Rabbani et al. 2006, Jagannathan and Miller 2007), concavity-aware fields (Au et al. 2011), then groups similar features through a clustering algorithm, thereby splitting the original blocks corresponding to the features. The extracted features include scale-invariant heat kernel signatures (SIHKS) (Bronstein and Kokkinos 2010b), shape-diameter function (SDF) (Shapira et al. 2008), Gaussian curvature (GC) (Gal and Cohen-Or 2006), etc. However, 3D shape segmentation effect strongly depends on human high-dimensional semantic cognition, so it is very difficult to accurately segment 3D shapes using a single surface geometric feature (Chen et al. 2009).

To further improve the generalisation performance of the segmentation algorithms and segmentation accuracy, some methods proposed to use the potential correlation to segment a set of models, also known as the co-segmentation problem. Such methods usually use data-driven as the core algorithm and can achieve more accurate segmentation of 3D semantic parts (Wang et al. 2012). Both Golovinskiy and Funkhouser (2009) and Chen et al. (2009) used a top-down approach to construct a corresponding structure map for each type of shape to segment the 3D model. Simari et al. (2009) first proposed to introduce prior knowledge to solve co-segmentation. Their approach requires users to provide semantic knowledge specific to each shape or part of shapes and formulate this knowledge to fit the optimisation framework. Their method can only be applied to objects that can be aligned spatially. Xu et al. (2010) proposed to use anisotropy to split the 3D model into style and content and use deform-to-fit to segment the 3D model. According to their framework, 'style' refers to the anisotropic scales of shape components, characterized by individual and relative scales, while 'content' encompasses the geometric and functional attributes of these components, including their functions and spatial arrangements within the object. These methods are all unsupervised data-driven methods.

2.3.2 Data-driven Methods for 3D shape Annotation

Kalogerakis et al. (2010) proposed for the first time the use of supervised data-driven methods to segment 3D mesh models. They proved that segmentation accuracy and adaptive ability of the algorithm can be significantly improved by introducing supervision information. As per Van Kaick et al. (2011a), it is possible to enhance prior knowledge by utilising segmented examples to improve the accuracy of segmentation. Huang et al. (2011) proposed a method to globally optimise the segmentation of an entire 3D model dataset, aiming for an effect akin to fully supervised. However, the stability of the proposed method is significantly compromised by the initialisation patches, which leads to inconsistent performance across different times. The initial patches are produced through the collection of distinct segments resulting from randomized patch clustering. In response to this issue, Sidi et al. (2011) introduced a method that clusters and segments based on implicit features, rather than relying on geometric characteristics. However, this approach tends to categorise geometrically distant parts as the same semantic label. This limitation hinders their method's performance on benchmarks evaluating Intersection over Union (IoU). Meng et al. (2013) and Hu et al. (2012) both addressed the problem of 3D model segmentation by proposing an approach that begins with an overly segmented initialisation patch, followed by a clustering step based on corresponding shape descriptors. However, the effectiveness of their methods is heavily dependent on the precision of the manual-designed shape descriptors. Building upon the structure and part descriptors introduced by (Kalogerakis et al. 2012), such as curvature histograms, shape diameter, and silhouette features, Huang et al. (2015) proposed a method for joint analysis and synthesis of 3D shape features. This method combined and analysed these 3D shape features, then matched similar parts using a probability model to achieve semantic segmentation. Kim et al. (2013) extend the work of Kalogerakis et al. (2010) by using geometric pattern templates. Likewise, Zheng et al. (2014) devised a probabilistic model that captures spatial relationships between different parts. This model facilitates the identification of common configurations across various shapes. Xie et al. (2014) introduced the application of the Extreme Learning Machine (ELM) technique to further enhance segmentation accuracy. Despite the advantageous performance of such learning-based approaches over methods dependent on manually crafted feature descriptors, they bear inherent challenges. Particularly, when input 3D models are subjected to extensive rigid transformations, the precision of segmentation would decline and easily be trapped in unsatisfactory local minima.

With the rapid progression of deep learning, a series of methods have been introduced to leverage deep learning for addressing the weakly supervised segmentation of 3D models. Guo et al. (2015) put forward an approach that involves learning from stacked feature descriptors of individual triangular patches. Shu et al. (2016) proposed a method that initially generates random patch feature descriptors, and then utilises a deep encoder to cluster feature vectors with similar semantic characteristics. When compared to previous unsupervised learning methods, these techniques demonstrate conspicuous improvement in segmentation accuracy.

2.3.3 Point Cloud Part Segmentation based Deep Learning Methods

This section investigates the most recent and advanced methods for point cloud annotation, which are predominantly based on deep learning.

While the aforementioned methods in the last section are designed specifically for mesh-form 3D models, their performance significantly hinges on the construction of the 3D model mesh. Recently, deep learning methods based on point features have been employed for point cloud segmentation. The point cloud feature operators in networks such as PointNet (Qi et al. 2017a)) and PointNet++(Qi et al. 2017b), which discussed in Section.2.1, have proposed a paradigm of connecting learned global descriptors with point features, then classifying each point into a part category through a multi-layer perceptron (MLP). These models offer an oriented toward point solutions. DGCNN (Wang et al. 2019) uses graph convolutions for point clouds. PointConv (Wu et al. 2019) reconstruct the point cloud to establish neighborhoods for the convolution operator, and GDANet (Xu et al. 2021) utilised attention alongside MLP. Capsule Networks (Zhao et al. 2019b) propose architectural changes that implicitly model parts for tasks like object classification and segmentation. However, their performance in weakly supervised and semi-supervised segmentation tasks considerably falls behind that in fully supervised tasks.

Yi et al. (2016a) proposed a semi-supervised machine learning approach for 3D model annotation. Their method utilises active learning to identify the most uncertain shape parts in each model and requires user intervention for manual annotations. This approach achieves high precision in 3D model annotation, but it is labor-intensive. To address the challenges of annotating articulated 3D models, Yi et al. (2018) proposed a method that jointly extracts shape correspondences and parts. Unlike the strategies proposed by Hu et al. (2012) and Sidi et al. (2011), they extracted feature descriptors through convolution operations. Their approach involves manipulating these feature descriptors in the feature space. This allows for a more efficient label transfer from a select set of template shapes to unseen shapes, guided by a process of deformation-driven reconstruction.

To discover and segment semantically meaningful parts in 3D shapes using tags associated with the shapes, Muralikrishnan et al. (2018) developed a hierarchical network structure named WU-NET, trained using weakly supervised datasets. Their approach segments 3D shapes by learning a correspondence between high-level semantic tags and the geometric parts of the shapes. In the attempt to further reduce the demand for labeled data, Sharma et al. (2019) proposed a few-shot learning approach to 3D shape segmentation. They trained a cascading network where the first network abstracts high-dimensional features of each point, and the second network classifies the semantic label of each high-dimensional feature. However, their method's reliance on building tree-structure hierarchies and suboptimal segmentation precision limit its efficacy. Similar to the deformation-reconstruction-alignment method proposed by Yi et al. (2018), methods proposed by Yuan and Fang (2020) and Wang et al. (2020) deform the to-be-annotated point cloud into a shape similar to the annotated point cloud. Then it transfers labels based on changes during the deformation process. Notably, ROSS (Yuan and Fang 2020) is specifically for one-shot semantic segmentation tasks, while Wang et al. (2020) proposed improving label transfer accuracy by estimating semantic labels for each point using a continuous probability distribution function.

Xu and Lee (2020) put forward an approach for handling incomplete label data, capable of providing a complete semantic label annotation even when only 10% of the supervised information is available for each 3D point cloud. Xie et al. (2020) presented a view that a shortcoming in previous methods, which the representations learning via these methods are the main barrier for assisting downstream segmentation tasks. They proposed a contrastive learning-based representation of point clouds and demonstrated the pre-trained representation of point cloud can support part segmentation tasks when applied with semi-supervised fine-tuning. Similarly employing contrastive learning, Jiang et al. (2021) introduced pseudo-label guidance for contrastive loss computation and a category-balanced sampling strategy to mitigate the class imbalance in point clouds. Inspired by PointContrast (Xie et al. 2020), Hou et al. (2021) proposed to leverage point-level correspondence and spatial context in the scene into the contrastive learning framework. The proposed method improves the performance on semi-supervised part segmentation. Loizou et al. (2020) designed BoundaryNet that learns semantic part boundaries from geometric features. Hassani and Haley (2019) and Gadelha et al. (2020) both utilised clustering features in point-based architectures to achieve semantic part segmentation. Their methods depend on highly structured point cloud representation, typically requiring a pre-trained 3D model convex decomposition model. Deng et al. (2021) proposed a semi-supervised pre-training system that includes user queries and assistance for dataset labelling. This system provides point clouds with pseudo labels that come from a pre-trained segmentation model.

Kawana et al. (2021) presented a multi-view unsupervised part decomposition method to explicitly target man-made articulated objects with mechanical joints. Sharma et al. (2022) introduced PRIFIT which is designed for the extraction of semantic shapes in 3D shape abstraction. Their approach can be viewed as a semantic segmentation technique that employs a point-based network to learn point-wise features. These features are then utilised as input for concurrent training of a primitive extraction network via clustering and a semantic segmentation network. This method enables semantic segmentation with a limited number of labeled samples. Although these two methods are proposed for shape abstraction, but they can also be used to annotate a point cloud with semantic label (Sharma et al. 2022).

Li et al. (2021b) developed a semantic part segmentation network framework combined with self-supervised learning, which is a GAN-based model. The generator in the traditional GAN is replaced by a segmentation network in this approach, with the discriminator assessing the accuracy of the segmentation. While this method can segment the semantic part of the 3D model with few-shot annotated a point cloud, it also inherits the high training cost associated with GANs. Following a similar solution, Cheng et al. (2021) proposed Sspc-net, a method integrating self-supervision into semantic segmentation tasks. This method builds a graph of a small region based on labelled points and uses this graph to generate pseudo labels by dynamic propagation. Deviant from these two methods, Sun et al. (2022) designed a self-supervised network learning to separate and reconstruct mixed point cloud shapes. In this manner, the trained self-supervised network can segment the semantic part of the point cloud with only a small number of labelled points.

Towards semi-supervised segmentation tasks for scene-level point clouds, Wei et al. (2020) put forward a hierarchical network that employs a pre-trained classification network to generate pseudo labels for the objects within the scene. Then the semantic segmentation network is trained in a fully supervised manner by leveraging these pseudo labels.

Zhang et al. (2021a) designed a transfer learning-based strategy aimed at enhancing the performance of weakly supervised point cloud segmentation. Their proposed method consists of two components: a self-supervised pre-trained task and sparse label propagation algorithm. The self-supervised pre-training task learns the prior distribution through point cloud colouring and transfers it to the weakly supervised network to improve its representation ability; sparse label propagation can spread the label to unlabeled points and expand the supervision information. and reduce computational complexity. This method requires an additional dataset for learning prior knowledge and transferring this knowledge to the weakly supervised segmentation task. Likewise, Zhang et al. (2021c) presented an enhanced self-supervised learning framework by introducing a perturbed branch. This perturbed branch ensures predictive consistency between the perturbed and original branches, subsequently improving prediction accuracy. SQN (Hu et al. 2022) demonstrated a significant decline in current semi-supervised segmentation methods when the label percentage drops below 0.1% in large-scale point cloud semantic segmentation. This method constructed a compact representation based on the high-dimensional features of sampling points and neighborhood points. They used semantic segmentation to prove the effectiveness of the learned representation. Their method is very stable and significantly improves the segmentation results in this scenario.

Jones et al. (2022) proposed SHRED for more granular semantic segmentation of 3D models. SHRED comprises three modules: 'split', which oversegments a point cloud into small mutually exclusive sets; 'fix', which clusters these small sets into individual regions; and 'merge', which fuses these small regions into finely-grained parts. Each module is trained separately, resulting in segmentation accuracy that outperforms the baseline. Drawing inspiration from the prototypical network by Snell et al. (2017), Su et al. (2022) proposed adding a multi-prototype classifier before the segmentation network, which enhances the recognition ability of cross-category semantic labels with identical functions by introducing a multi-prototype classifier. This classifier is capable of representing the subclasses within each semantic category by maintaining multiple prototypes for each class. Building upon this, Zhao et al. (2021b) designed computing geometric dependencies and semantic correlations point-wise using attention mechanisms. Their method showed considerable improvements in segmentation accuracy compared to the baseline.

1T1C (Liu et al. 2021), SegGroup Tao et al. (2022), and LESS (Liu et al. 2022) all proposed leveraging a small amount of user-provided annotations as ground-truth to aid segmentation. In these methods, the user-provided annotation information can be used to generate semantic labels for sub-regions, which are then propagated to other points within the same region. However, there are some differences: SegGroup requires more granular annotations, while LESS targets LIDAR point clouds and pre-processes the data using heuristic algorithms.

Adopting a strategy similar to active learning, Chen et al. (2023) presented a class-level confidence-based 3D semi-supervised learning method to address the data imbalance issue in semi-supervised point cloud semantic segmentation. Their proposed method distinguishes the learning status of point clouds based on confidence, and then samples for learning from point clouds with a lower learning status. The learning status, that is measure of how well a point cloud has learned to classify each semantic part within a dataset, is computed by class-level confidence samples by dynamic thresholding and a re-sampling strategy. Sun et al. (2023) proposed a semi-supervised 3D shape segmentation method, using multi-level consistency and part replacement to enhance network training, thereby facilitating 3D segmentation learning from a small amount of labeled data and a large amount of unlabeled data. The multi-level consistency loss is employed to enforce the consistency of network predictions at multiple levels, while the part replacement scheme is used to enhance the structural variations of labeled 3D shapes. Gkanatsios et al. (2023) put forward a semi-parametric learning framework named Analogical Networks for 3D scene parsing. The model employs analogical reasoning to map input scenes to modified and combined past labeled visual experiences, rather than directly mapping input scenes to part segmentation.

With the rise of multimodality, more and more works have been devoted to multimodal-based few-sample point cloud segmentation. Chen et al. (2021) proposed a framework based on multimodal semi-supervised learning, which aims to utilise 3D data from different modalities to improve data efficiency for 3D classification and retrieval tasks. The modal information mainly used by the proposed method includes different representations of 3D data, including point clouds, images, and meshes. The framework introduces instance-level consistency constraints and a novel Multimodal Contrastive Prototype (M2CP) loss. Liu et al. (2021) used semi-supervised learning to infer the 3D structure of generic objects. This approach decomposes real 2D images into latent representations such as category, shape, albedo, lighting, and camera projection matrices using a federated learning fitting module to obtain segmented 3D shapes.

This chapter has undertaken a review of the methodologies for advancing point cloud feature learning, generation and segmentation without strong segmentation methods. It reviews the current methods for extracting point cloud feature descriptors and explores the popular point cloud generation models. It also investigates the innovative annotation techniques that mitigate the cost for extensive labelled datasets. By showcasing the potential of these techniques, this thesis aims to harness deep learning's transformative power for enhancing 3D point cloud annotated data generation.

Chapter 3

Point Cloud Generative Model based on Stochastic Differential Equations

In line with the aforementioned in Section 1.1 Chapter 1, an expressive point cloud is necessary for building a high-quality point cloud dataset. Therefore, to address the issue of generating datasets using point cloud generative models, the first step is to generate high-quality point clouds. In this chapter, a point cloud generative model based on stochastic differential equations is proposed. More specifically, this model integrates normalizing flows and time encoding to create the framework, and the model parameters are optimized by leveraging the training objective of SDE. To further improve the quality of the generated point cloud, and to fully exploit the characteristics of diffusion-based models, a sampler drawing upon the Markov Chain Monte Carlo sampling method, referred to as the Corrector Sampler, is introduced. To illustrate the method and implement details of the proposed point cloud generation model, the organisation of this chapter is shown as follows:

- Section 3.1 overviews the proposed method and motivation of each module.
- Section 3.2 begins with a review of stochastic processes, leading to a comprehensive understanding of Stochastic Differential Equations

(SDE) and their reverse counterparts. Then it formulates how to employ these principles to simulate a point cloud generation process.

- Section 3.3 discusses the Markov Chain Monte Carlo sampler, investigating how to apply it in the point cloud generation process and formulating the approach of a point cloud sampling generation algorithm that leverages the MCMC sampler.
- Section 3.4 details the network design of the point cloud generation model. It includes the incorporation of the normalising flow model, practical details of the diffusion model-based point cloud generation with a flexible time process, and the definition of the loss function.
- Section 3.5 outlines the experimental design and the evaluation methods, and presents a comprehensive analysis of the experimental results.
- Section 3.6 summarises this chapter.

3.1 Introduction

The point cloud is one of the most popular 3D shape representations that can represent diverse shapes with a set of sparse and discrete 3D points. Recent advancements in 3D sensors have catalysed a multitude of 3D processing and understanding tasks based on point clouds, including object classification and semantic segmentation (Zhao et al. 2019c). The past decade has witnessed great advances in deep neural learning on 3D point cloud tasks (Qi et al. 2017a b). The main goal of these methods is to design a suitable structure of the deep neural network to learn representations and then apply them to a variety of 3D tasks, which can demonstrate the effectiveness of the methods.

As one of the most important members of the unsupervised learning family, self-supervised learning fundamentally does not rely on manually labeled data for the acquisition of high-quality representation learning. Generative models, as a subset of self-supervised learning, can effectively learn the representations, and the ability of the generative models is demonstrated by the quality of yielded objects. A robust point cloud generation model can provide expressive 3D data that is different from the existing 3D data repository, which can further be used to augment 3D understanding tasks. This highlights the importance of point cloud generation in the field of 3D analysis.

Recently the 3D vision community has contributed significant development for point cloud generative models. This development in generative models provides an alternative solution for acquiring 3D data, such as variational auto-encoders (VAEs) (Gadelha et al. 2018, Zamorski et al. 2020), generative adversarial networks (GANs) (Wu et al. 2016b, Ramasinghe et al. 2020, Shu et al. 2019a), auto-regressive (Sun et al. 2020), etc. All aforementioned point cloud synthesis methods are supervised by either the Chamfer Distance (CD) or Earth Mover‘s Distance (EMD) to optimise the network to generate expressive point clouds (Lyu et al. 2021b). However, CD loss does not compute the matching distance between ground truth and synthesis and emphasises accuracy rather than the uniform distribution of shape. EMD loss can be computed as the minimal value of a linear program, which is sensitive to the overall distribution of the point cloud, but its computational expense is high.

In light of these challenges, Diffusion Probabilistic Generative Models (Ho et al. 2020) have emerged as a promising solution, demonstrating state-of-the-art performance on multiple tasks. These models employ a simple mean squared error (MSE) loss function, circumventing the weaknesses of previous methods. Therefore, point cloud synthesis based on the diffusion probabilistic model can avoid the weakness of previous methods.

Existing point cloud synthesis approaches based on Diffusion Probabilistic Models (DDPMs) (Luo and Hu 2021, Zhou et al. 2021, Cai et al.

2020) follow a process of gradually introducing noise into the ground truth point cloud, subsequently employing a neural network to reverse this process. Nevertheless, these techniques often approximate the diffusion process with a fixed-step Markov chain. This leads to limitations in the expressiveness of the synthesised point clouds.

Drawing inspiration from Song et al. (2020), this thesis proposes a method for point cloud synthesis based on Stochastic Differential Equations (SDE). Different from the aforementioned methods, this method proposes to formulate the point cloud generation as a continuous time stochastic process. Our goal is to learn a transition kernel that can synthesise plausible point clouds based on SDE. Time embedding is employed to exploit the arbitrary sample of time. It allows the reverse process can be applied with arbitrary time length. Point cloud synthesis can benefit from this in two aspects: a) the network can better understand the time variable; b) the transform process is smoother and more flexible. Additionally, Langevin MCMC (Parisi 1981) samplers with SDE-based approaches are introduced and applied to improve over simple progression sampling methods. The proposed model is implemented on point cloud generation and unsupervised representation learning. Experimental results demonstrate that our model achieves competitive performance on three learning tasks: point cloud synthesis, auto-encoding generation, and point cloud completion. The content in this section has been derived from our previously published paper Li et al. (2023).

3.2 Point Cloud Generation based on Stochastic Differential Equations

This section reviews preliminaries of stochastic process, and stochastic differential equations. Then this section analyses how to apply the stochastic differential equations to point cloud generation. The overview of the key idea is given in Figure 3.1.

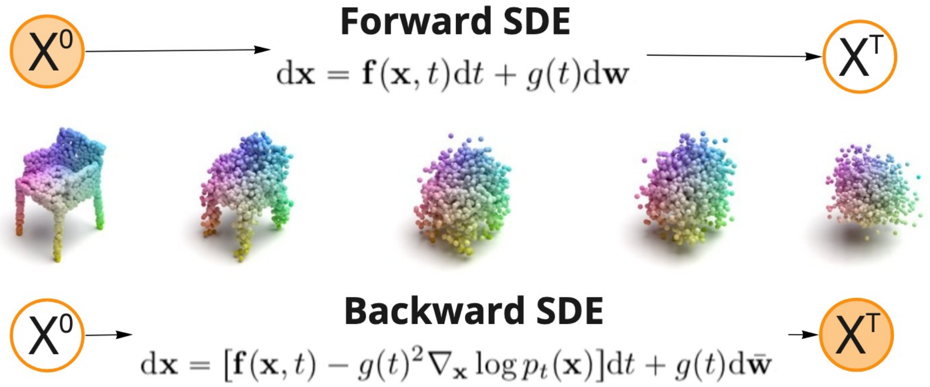

Forward SDE
$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$
X^0
X^T
Backward SDE
$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$
X^0
X^T

Figure 3.1: Overview of point cloud generative modeling through SDE. The desired shape is synthesised from a noise point cloud with prior distribution via a continuous-time SDE-Net.

Section 3.2.1 reviews the fundamentals of stochastic processes, including their definition, classification, and key conceptions. Section 3.2.2 revisits stochastic differential equations, examining their definition, formulation, and interrelationships. Lastly, section 3.2.3 explains the application of stochastic differential equations to point cloud generation, detailing the derivation process.

3.2.1 Preliminaries of Stochastic Process

A stochastic process, central to probability theory and related fields, represents a collection of random variables indexed by time. A stochastic process can be defined as a mathematical object regarded as a collection of random variables (Parzen 1999). These variables are usually associated with or indexed by a set of numbers, typically viewed as points in time, giving the interpretation of a sequence of random events (Doob 1942). Fundamentally, the stochastic process represents the evolution of a system of random values over time or space, thus providing a mathematical framework for modelling and understanding systems that evolve in a way that may be probabilistically determined but not precisely predictable (Parzen 1999, Krylov and Vladimirovich 2002). The general definition of stochastic process is as follows:

Definition: *Let* $(\omega, \mathcal{F}, P)$ *be a probability space, if there is a family of random variables* $X = \{X_t\}_{t\in[0,\infty)}$ *(that is, for each* t*,* X_t *is a measurable function of* $(\omega, \mathcal{F}) \to (R, \mathcal{B})$*), then* X *is called a probability space, and* $(R, \mathcal{B})$ *is called the state space of the stochastic process.*

In this definition, $\mathcal{F}$ is a sigma-algebra on ω (a collection of events for which probabilities can be assigned); and P is a probability measure that assigns probabilities to the events in $\mathcal{F}$; ω can be regarded as every possible pollen grain or every possibility of pollen grain movement; and $X_t(\omega)$ is a function at every moment t, which describes the position of this pollen. Although possibility is an abstract concept, each possibility has some specific values to describe it. For example, the momentum, acceleration, and velocity of a particle are all specific values that describe this abstract ω, so for each ω, corresponding to position information $X_t(\omega)$, which is a function of ω.

Stochastic processes have many significant characteristics, such as separability (Itō 2006), indistinguishable (Rogers and Williams 2000), independence (Lapidoth 2017), regularity (Khoshnevisan 2002). The properties used in this research are as follows:

- Independence: If the random variables in the process are mutually independent at different times or spatial points, then the process exhibits independence, which enlightens the assumption in Chapter 4.
- Stationarity: A stationary stochastic process refers to its statistical properties remaining unchanged with time or space shifts, including first-order stationarity (mean is constant) and second-order stationarity (correlation only depends on the time interval and is independent of specific time).

Stochastic processes can be classified by their state space, index sets, or inter-variable correlations. A common approach is based on the cardinality of the index set and state space, which helps distinguish between discrete and continuous time stochastic processes.

These processes, despite their contrasting temporal structures, often share connections. For instance, continuous-time Markov chains and random walks can be viewed as generalised versions of their discrete-time counterparts.

- Discrete-time Processes: Since time points are countable, discrete-time stochastic processes can be represented with sequences or matrices. They are usually easier to analyse and compute mathematically. Some common methods include Markov chains and stochastic difference equations.

- Continuous-time Processes: For continuous-time stochastic processes, time points are uncountable, and continuous functions, differential equations, or stochastic differential equations are used to represent them. They often require more advanced mathematical tools to analyse and solve problems. Common methods include stochastic integral equations, Monte Carlo methods, and particle filters.

The stochastic process has different representation approaches, including the Bernoulli process, Wiener process, Markov process, martingale, Levy process, etc., the following are some common methods:

- **Bernoulli Process:** is a type of discrete-time stochastic process with two possible outcomes. It is defined by a series of independent random variables $\{X_t, t \in \{0, 1, 2, \ldots\}\}$, where each X_t follows a Bernoulli distribution with parameter p. This process is a Bernoulli process. Bernoulli process can be formulated as:

$$P(X_t = 1) = p \ , \ P(X_t = 0) = 1 - p, \tag{3.1}$$

where the probability p is expressed as the probability of occurrence of $X_t = 1$.

Bernoulli process is often used to describe two possible outcomes of experiments, such as flipping a coin, where the value of the probability p is 1, and the probability $1 - p$ is 0. However, it is noteworthy that the Bernoulli process can only be used for the motion process of a single particle in low dimension, and the variable state of Bernoulli is binary, so it cannot be directly used to simulate the point cloud generation process.

- **Markov process:** If the random process X(n) satisfies at any time, the distribution of all the distances experienced in the past is the same as that of the nearest point, that is

$$F(x, t|x_n, x_{n-1}, \cdots, x_2, x_1, t_n, t_{n-1}, \cdots, t_2, t_1) = F(x, t|x_n, t_n), \tag{3.2}$$

which implies

$$\begin{aligned} &P\{X(t) \leq x|X(t_n) = x_n, \cdots, X(t_1) = x_1\} \\ &\qquad = P\{X(t) \leq x|X(t_n) = x_n\}. \end{aligned} \tag{3.3}$$

All random processes with Markov property can become Markov process. Markov property is the so-called finite memory. Markov process is a special type of stochastic process in which only the current value of a variable is relevant for future predictions, while the historical values of the variable and how the variable has evolved from past to present are not relevant for future predictions.

- **Random Walk:** Consider a simple, symmetric random walk of particles on a straight line. In each time increment Δt, a particle is allowed to move right with a probability $p = \frac{1}{2}$, covering a distance $\Delta x > 0$, or with an equivalent probability $p = \frac{1}{2}$, it can move left by Δx. Each movement event is independent of the others. If X_t

designates the position of the particle at time t, and $\Delta x = c\sqrt{\Delta t}$ as $\Delta t \to 0$, then

$$X_t \sim \mathcal{N}(0, c^2 t) \tag{3.4}$$

is representative of a **random walk**. If this process is extended to a point cloud, whereby X symbolises a cluster of points states in a three-dimensional space, the diffusion motion of the point cloud can be effectively described.

- **Wiener Process/Brownian motion:** Assume that the increment Δx in the random walk process is stable and follows a normal distribution with an expectation of 0 and a variance of $c^2 t$, then

  $$X_{s+t} - X_s \sim N\left(0, c^2 t\right) \tag{3.5}$$

  At this point, X_t becomes a continuous function of t. $\{X_t : t \geqslant 0\}$ is referred to as Brownian motion or Wiener process, with $c = 1$ denoting standard Brownian motion.

  Assuming $X_0 = 0$, it is called standard Brownian motion with zero initial value. At this time

  $$X_t \sim \mathcal{N}(0, t) \tag{3.6}$$

  By infinitely dividing the discrete random walk, Brownian motion in the continuous case can be attained. While Brownian motion is continuous, it is not differentiable everywhere, which prevents it from being solved by universal differential equations. However, the Brownian motion exhibits Markov property, and the independent increment satisfies the normal distribution. In the process of point cloud generation, each point can be seen as moving in continuous Brownian motion over time.

The primary focus of this thesis is the application of stochastic process to simulate the point cloud generation process.

3.2.2 Preliminaries of Stochastic Differential Equations

Stochastic Differential Equations (SDE) are a tool for describing the evolution of a stochastic process (Stochastic Process) over time or space. SDE typically consist of two components: a deterministic part and a stochastic part. The deterministic component represents the behaviour of the system in the absence of random perturbations, while the stochastic component represents the impact of random perturbations on the system's evolution. In this way, SDE can be used to model the dynamic behaviour of a system under the combined influence of deterministic and random factors. The stochastic component is typically comprised of one or more random processes, such as the Wiener process or Brownian motion.

Stochastic Differential Equations (SDE): A typical It$\hat{o}$ stochastic differential equation has the following form:

$$\mathrm{d}X(t) = f(X(t), t)\mathrm{d}t + g(X(t), t)\mathrm{d}W(t), \tag{3.7}$$

where $X(t)$ is the stochastic process (such as stock price, pollution concentration, etc.), t is time, f and g are known functions, and $W(t)$ is one or more stochastic processes, usually Brownian motion. $\mathrm{d}X(t)$, $\mathrm{d}t$ and $\mathrm{d}W(t)$ represent small changes in X(t), t and W(t), respectively.

SDE simulates a stochastic process with a given initial state. SDE provides a functional tool to model the stochastic process that supports point cloud generation.

Backward Stochastic Differential Equations (BSDE): Like SDE, BSDE models the behaviour of stochastic processes under the influence of both deterministic and random factors. The primary difference is that BSDE works in the reverse direction, starting from the final state and seeking to determine the initial conditions of the process.

The general form of a backward SDE can be formulated as follows:

$$Y(t) = \xi + \int_t^T f\left(s, Y_s, Z_s\right) ds - \int_t^T Z_s dW_s, 0 \leq t \leq T, \tag{3.8}$$

where $Y(t)$ is an anti-stochastic process, $W(t)$ is a Brownian motion, $Z(t)$ is a function of $Y(t)$ and t, $f(t, Y(t), Z(t))$ is a related function, and ξ represents the process value at time T.

BSDE incorporates information about the system's state at the end of the time interval through the terminal condition ξ. This allows BSDE to be used in situations where specific terminal conditions need to be met.

Ma and Yong (1999) put forward that solving forward-backwards SDE can be formulated by an optimal system:

$$\begin{cases} dX(t) = \left[aX(t) - b^2 Y(t)\right] dt + dW(t), \\ dY(t) = -[aY(t) + X(t)]dt + Z(t)dW(t), \quad t[0, T] \\ X(0) = x, \quad Y(T) = X(T). \end{cases} \tag{3.9}$$

where the equation for $X(\cdot)$ is forward (since it is given the initial datum) and the equation for $Y(\cdot)$ is backward (since it is given the final datum).

3.2.3 Application of Stochastic Differential Equations on Point Cloud Generation

The last sections revisit stochastic process, SDE, and BSDE. These approaches are particularly useful in cases where the object or scene being modeled exhibits a behaviour that can be described by stochastic process. The point cloud generation is similar to a stochastic process, moving the noisy and disordered points slowly to form a shaped point cloud. Therefore, a solution can be sought from the stochastic process theory.

By simulating the stochastic process, the point cloud generation can be regarded as the transition between the prior distribution and realistic point cloud with varying noise and uncertainty. To address this challenge, this section elaborates on how to utilise SDE to model the movement and behaviour of this transition, so that simulates the stochastic process in point cloud generation. Furthermore, the solution of SDE is proposed, which is based on Bayes' theorem.

Forward stochastic process. This diffusion process can be represented as an Itô SDE (Itô 1973). It is the rule for differentiating a function of a stochastic process. In this thesis's setting, it is the process that gradually transforms a 3D shape into 3D Gaussian noise. It can be formulated as:

$$dx = \mathbf{f}(x, t)dt + g(t)d\mathbf{w}, \tag{3.10}$$

where $\mathbf{w}$ is the standard Brownian motion (Wiener process); $\mathbf{f}(\cdot, t) : R^d \to R^d$ is a drift coefficient of x_t, and $g(\cdot) : R \to R$ the diffusion coefficient of x_t. In this thesis, the diffusion coefficient is a $d \times d$ scalar matrix.

Therefore, the forward process of diffusion process can be discretised as:

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t + \mathbf{f}_t(x_t)\Delta t + \boldsymbol{G}_t \epsilon_t, t = 0, 1, \ldots, T-1 \tag{3.11}$$

where $\epsilon_i \in \mathcal{N}(0, \mathbf{I})$, which conforms with Gaussian distribution. In this case, when the $\Delta t \to 0$, Equation 3.11 can be transformed to:

$$\boldsymbol{x}_{t+\Delta t} - \boldsymbol{x}_t = f_t(\boldsymbol{x}_t)\Delta t + g_t\sqrt{\Delta t}\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{3.12}$$

The diffusion process itself does not depend on the starting distribution of the point cloud. Formally, let $p_{latent} = \mathcal{N}(0, \boldsymbol{I})$ denotes the distribution of point cloud at $t = T$, where $\mathcal{N}$ is the Gaussian distribution. Given N points in a point cloud $X = \{\ x^i | i = 1, .., N\} \in R^{N \times 3}$, we assume p_{data} to be the distribution of the each point cloud X in the dataset. For each point x^i in point cloud X, the status in diffusion process x_t^i can be indexed by a continuous time variable $t \in [0, T]$.

Then, the stochastic process can be used to model the transition process of the distribution of the point cloud, which effectively induces a desired shape of points by transforming a prior distribution p_{data}. The synthesis of the point cloud can thus be represented by the reverse diffusion process.

It's important to note that x_T is an unstructured prior distribution (3D Gaussian Distribution), and x_0 and p_{data} share the same distribution. Therefore, solving Equation 3.12 can be transformed into a probability problem, more specifically, estimating the likelihood of $p(\boldsymbol{x}_{t+\Delta t})$.

Backward stochastic process. Equation 3.12 allows the conditional probability to be given by:

$$\begin{aligned} p\left(\boldsymbol{x}_{\boldsymbol{t}+\Delta t} \mid \boldsymbol{x}_{\boldsymbol{t}}\right) &= \mathcal{N}\left(\boldsymbol{x}_{\boldsymbol{t}+\Delta t}; \boldsymbol{x}_{\boldsymbol{t}} + \boldsymbol{f}_t\left(\boldsymbol{x}_{\boldsymbol{t}}\right)\Delta t, g_t^2 \Delta t \boldsymbol{I}\right) \\ &\propto \exp\left(-\frac{\left|\boldsymbol{x}_{\boldsymbol{t}+\Delta t} - \boldsymbol{x}_{\boldsymbol{t}} - \boldsymbol{f}_t\left(\boldsymbol{x}_{\boldsymbol{t}}\right)\Delta t\right|^2}{2 g_t^2 \Delta t}\right) \end{aligned} \tag{3.13}$$

Applying Bayes' theorem directly gives:

$$p\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_{\boldsymbol{t}}\right) = \frac{p\left(\boldsymbol{x}_{\boldsymbol{t}} \mid \boldsymbol{x}_{\boldsymbol{t-1}}\right) p\left(\boldsymbol{x}_{t-1}\right)}{p\left(\boldsymbol{x}_{\boldsymbol{t}}\right)} \tag{3.14}$$

and conditional Bayes' theorem

$$p\left(\boldsymbol{x}_{\boldsymbol{t-1}} | \boldsymbol{x}_{\boldsymbol{t}}, \boldsymbol{x}_{\boldsymbol{0}}\right) = \frac{p\left(\boldsymbol{x}_{\boldsymbol{t}} | \boldsymbol{x}_{\boldsymbol{t-1}}\right) p\left(\boldsymbol{x}_{\boldsymbol{t-1}} | \boldsymbol{x}_{\boldsymbol{0}}\right)}{p\left(\boldsymbol{x}_{\boldsymbol{t}} | \boldsymbol{x}_{\boldsymbol{0}}\right)} \tag{3.15}$$

It can be substituted into Equation 3.13 to get

$$p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t+\Delta t}\right) \propto \exp\left(-\frac{\left\|\boldsymbol{x}_{t+\Delta t} - \boldsymbol{x}_t - \boldsymbol{f}_t\left(\boldsymbol{x}_t\right)\Delta t\right\|^2}{2 g_t^2 \Delta t} + \log p\left(\boldsymbol{x}_t\right) - \log p\left(\boldsymbol{x}_{t+\Delta t}\right)\right) \tag{3.16}$$

When $\Delta t \to 0$, $p\left(\boldsymbol{x}_{\boldsymbol{t}+\boldsymbol{\Delta t}} \mid \boldsymbol{x}_{\boldsymbol{t}}\right)$ will not equal to 0. Therefore, we can use Taylor expansion to analyse Equation 3.16 when the $\Delta t \to 0$:

$$\begin{aligned} \log p\left(\boldsymbol{x}_{\boldsymbol{t}+\boldsymbol{\Delta t}}\right) \approx \log p\left(\boldsymbol{x}_{\boldsymbol{t}}\right) + \left(\boldsymbol{x}_{\boldsymbol{t}+\boldsymbol{\Delta t}} - \boldsymbol{x}_{\boldsymbol{t}}\right) \cdot \nabla_{\boldsymbol{x}_{\boldsymbol{t}}} \log p\left(\boldsymbol{x}_{\boldsymbol{t}}\right) + \\ \Delta t \frac{\partial}{\partial t} \log p\left(\boldsymbol{x}_{\boldsymbol{t}}\right) \end{aligned} \tag{3.17}$$

where the $p(x_t)$ denotes the probability density of a random variable equal to x_t at time t. After substituting the Equation 3.17 into Equation 3.16:

$$\begin{aligned} p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t+\Delta t}\right) &\propto \exp\left(-\frac{\left\|\boldsymbol{x}_{t+\Delta t}-\boldsymbol{x}_t-\left[\boldsymbol{f}_t(\boldsymbol{x}_t)-g_t^2\nabla_{\boldsymbol{x}_{\mathrm{f}}}\log p(\boldsymbol{x}_t)\right]\Delta t\right\|^2}{2g_t^2\Delta t}+\mathcal{O}(\Delta t)\right) \\ &\approx \exp\left(-\frac{\left\|\boldsymbol{x}_t-\boldsymbol{x}_{l+\Delta t}+\left[\boldsymbol{f}_{t+\Delta t}(\boldsymbol{x}_{t+\Delta t})-g_{i+\Delta t}^2\nabla_{x_{t+\Delta t}}\log p(\boldsymbol{x}_{l+\Delta t})\right]\Delta t\right\|^2}{2g_{l+\Delta t}^2\Delta t}\right) \end{aligned} \tag{3.18}$$

In this case, $p\left(\boldsymbol{x_t}|\boldsymbol{x_t}+\Delta t\right)$ can be regarded as a normal distribution with mean equals to $\boldsymbol{x_t}+\Delta t-\left[f_{t+\Delta t}\left(\boldsymbol{x_t}+\Delta t\right)-g_{t+\Delta t}^2\nabla_{x_{t+\Delta t}}\log p\left(\boldsymbol{x_{t+\Delta t}}\right)\right]\Delta t$ and normal distribution equals to $g_{t+\Delta t}^2\Delta t\boldsymbol{I})$. When taking the limit of $\Delta t \to 0$, then corresponding to SDE:

$$dx=\left[f_t\left(\boldsymbol{x}\right)-g_t^2\nabla_x\log p_t\left(\boldsymbol{x}\right)\right]dt+g_t d\mathbf{w} \tag{3.19}$$

Solve stochastic differential equations. To solve for $\nabla x\log p_t(x)$, the forward stochastic process can be presented discretely:

$$\begin{aligned} \boldsymbol{x}_t-\boldsymbol{x}_{t+\Delta t}= &-\left[\boldsymbol{f}_{t+\Delta t}\left(\boldsymbol{x}_{t+\Delta t}\right)-g_{t+\Delta t}^2\nabla_{x_{t+\Delta t}}\log p\left(\boldsymbol{x}_{t+\Delta t}\right)\right]\Delta t \\ &+g_{t+\Delta t}\sqrt{\Delta t}\boldsymbol{\varepsilon} \end{aligned} \tag{3.20}$$

This formulation simulates the point cloud generation process. And we can obtain

$$\begin{aligned} p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right)=\lim_{\Delta t\to 0}\int\cdots\iint p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-\Delta t}\right)p\left(\boldsymbol{x}_{t-\Delta t} \mid \boldsymbol{x}_{t-2\Delta t}\right) \\ \cdots p\left(\boldsymbol{x}_{\Delta t} \mid \boldsymbol{x}_0\right)d\boldsymbol{x}_{t-\Delta t}\boldsymbol{x}_{t-2\Delta t}\cdots\boldsymbol{x}_{\Delta t} \end{aligned} \tag{3.21}$$

When the $f_t(x) \propto x$, $p(x_t|x_0)$ can be solved analytically

$$p\left(\boldsymbol{x_t}\right)=\int p\left(\boldsymbol{x_t} \mid \boldsymbol{x_0}\right)\widetilde{p}\left(\boldsymbol{x_0}\right)d\boldsymbol{x_0}=E_{\boldsymbol{x_0}}\left[p\left(\boldsymbol{x_{t|}}\boldsymbol{x_0}\right)\right], \tag{3.22}$$

where $\widetilde{p}(\boldsymbol{x_0})$ denotes the distribution of data p_{data}. Then,

$$\nabla_{x_t}\log p\left(\boldsymbol{x}_t\right)=\frac{\mathbb{E}_{x_0}\left[\nabla_{x_t}p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right)\right]}{\mathbb{E}_{x_0}\left[p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right)\right]}=\frac{\mathbb{E}_{\boldsymbol{x_0}}\left[p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right)\nabla_{\boldsymbol{x_t}}\log p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right)\right]}{\mathbb{E}_{\boldsymbol{x}_0}\left[p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right)\right]} \tag{3.23}$$

Since the solution of $p(x_t|x_0)$ involves the average of all training samples x_0, the amount of calculation is large, and the generalisation ability is not good enough. Therefore, a neural network can be used to learn a function $s_\theta(x_t, t)$, so that it can directly calculate $\nabla_{x_t} \log p(x_t)$.

To make $s_\theta(x_t, t)$ equal to the weighted average of $\nabla_{x_t} \log p(x_t|x_0)$, the value of $\|s_\theta(x_t, t) - \nabla_{x_t} \log p(x_t|x_0)\|^2$ should be minimised, which is

$$\frac{\mathbb{E}_{x_0}\left[p\left(\boldsymbol{x_t}|\boldsymbol{x_0}\right)\|\mathrm{s}_\theta\left(\boldsymbol{x_t}, t\right) - \nabla_{\boldsymbol{x_t}} \log p\left(\boldsymbol{x_t}|\boldsymbol{x}_0\right)\|^2\right]}{\mathbb{E}_{x_0}\left[p\left(\boldsymbol{x_t}|\boldsymbol{x_0}\right)\right]} \tag{3.24}$$

When the samples are sufficiently diverse, the value of $\mathbb{E}_{x_0}\left[p\left(x_t|x_0\right)\right]$ does not change. For simplicity, Equation 3.24, we can remove it directly, and obtain:

$$\begin{aligned} &\int \mathbb{E}_{x_0}\left[p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right)\left\|\boldsymbol{s}_\theta\left(\boldsymbol{x}_t, t\right) - \nabla_{\boldsymbol{x}_t} \log p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right)\right\|^2\right] d\boldsymbol{x}_t \\ =&\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t \sim p(\boldsymbol{x}_t|\boldsymbol{x}_0)\tilde{p}(\boldsymbol{x}_0)}\left[\left\|\boldsymbol{s}_\theta\left(\boldsymbol{x}_t, t\right) - \nabla_{\boldsymbol{x}_t} \log p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right)\right\|^2\right] \end{aligned} \tag{3.25}$$

3.3 Point Cloud Sampling based on Markov Chain Monte Carlo Sampler

The major difference between the diffusion-based model and other types of generative models is the flexible inference approach. To further take advantage of this characteristic, a Markov Chain Monte Carlo based sampling method is proposed in this section.

In this section, Markov Chain Monte Carlo and its definition are briefly reviewed in Section 3.3.1. Section 3.3.2 designs a sampling method based on Markov Chain Monte Carlo, which can be employed by the point cloud generative model.

3.3.1 Preliminaries of Markov Chain Monte Carlo

One attribution of diffusion models is their objective of calculating the gradients of log probability density functions, which makes them apart from others (Song and Ermon 2019). Consequently, once a diffusion model has been trained, point cloud generation is realised by literate sampling like a backward stochastic process. This inference process offers a more flexible method of generation. Hence, this section introduces the concept of employing Markov Chain Monte Carlo (MCMC) sampling methods to enhance the quality of the generated point clouds.

The Markov Chain Monte Carlo (MCMC) method is a variant of the Monte Carlo algorithm, which is mainly used for sampling from complex probability distributions. The MCMC method combines the characteristics of the Markov chain (Markov Chain) and the Monte Carlo method and approximates the target distribution by constructing a Markov chain, so as to realise the sampling of the distribution. Markov chain Monte Carlo mainly has the following two characteristics:

- Markov chain: represents a series of variables wherein each subsequent state's occurrence relies exclusively upon its immediate predecessor, with no dependency on any preceding states. In such sequences, the future is conditionally independent of the past, given the present state.
- Monte Carlo method: This is a method of performing numerical calculations with random sampling. In statistical physics, for example, Monte Carlo methods can be used to simulate the behaviour of a system.

Markov chain Monte Carlo methods use Markov chains to perform random walks in the state space to generate samples of the target probability distribution. If the Markov chain is properly designed, then as the walk progresses, the samples generated by the Markov chain will get closer and closer to the target probability distribution. Therefore, the state of the Markov chain can be used as a sample drawn from the target probability distribution. The basic steps of the MCMC method are as follows:

- Initialisation: Select an initial state x_0, and a suitable Markov chain transition matrix.

- Transition: Transition from the current state x_t to the next state x_{t+1} according to the transition matrix of the Markov chain. This process may require adjusting the transition probabilities using an accept-reject criterion so that the stationary distribution is equal to the target distribution.

- Convergence Judgment: Check whether the Markov chain is converged. If not converged, return to step 2. If converged, continue to the next step.

- Sampling: Sample from the converged Markov chain. Since these samples come from the stationary distribution of the target distribution, they can be used to estimate the expectation, variance, and other statistics of the target distribution.

Common algorithms of MCMC include the Metropolis-Hastings algorithm, Gibbs Sampling, Langevin sampling method, etc.

3.3.2 Application of Markov Chain Monte Carlo Sampling on Point Cloud Generation

Langevin MCMC (Parisi 1981) uses gradient information to guide the stochastic process, letting the stochastic process converge to the target distribution faster. This is very helpful for sampling in high-dimensional spaces, especially if the shape of the target distribution is complex (e.g, multimodal or highly skewed). Langevin MCMC sampler combines the characteristics of stochastic gradient descent and Markov chain to sample more efficiently when exploring the target distribution. Specifically, starting from the current sample $x(t)$, an update is conducted in the gradient direction, and a random disturbance is added to obtain a new sample $x(t+1)$.

To illustrate the Langevin MCMC sampler in point cloud generation, the inference phase of the proposed model is initially scrutinised. By starting from noise with the prior distribution p_{latent} and reversing the diffusion process, a point cloud shape can be obtained from the data distribution p_{data}. During the inference phase, a sampling approach that combines the reverse stochastic process and the Langevin MCMC approach is designed and applied. Specifically, $X_{t-\Delta t}$ for X_t is estimated via the SDE-Net model s_θ and global latent variable $\mathbf{z}$ on each time step, and then Langevin MCMC sampler can be used to refine X_t. This simplified corrector conditional sampler is as follows Algorithm.1:

Algorithm 1 Corrector Sampler

```
**Initialisation:**$\{x_T, \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \sigma, r\}$
**for** $i \leftarrow T$ *to* $1$ **do**
  **for** $j \leftarrow 1$ *to* $n$ **do**
    $\omega \sim \mathcal{N}(0,1)$# Sample random noise
    $g \leftarrow s_\theta(X_i^{j-1}, \sigma^i, \mathbf{z})$# Predict gradient
    $\epsilon \leftarrow 2(r\|\omega\|_2/\|g\|_2)^2$# Refine gradient
    $X_i^j \leftarrow X_i^{j-1} + \epsilon g + \sqrt{2\epsilon}\omega$
  $X_{i-1}^0 \leftarrow X_i^n$
```

As shown in Algorithm.1, r denotes the signal-to-noise ratio of the Langevin MCMC sampler. $\|\omega\|_2$ is a random noise with Gaussian distribution. ω, r and output of the SDE-Net s_θ jointly determine the step size ϵ. This additional step, which is referred to as the corrector step, helps us to obtain a more accurate point cloud. This section incorporates material from the following publication Li et al. (2023).

3.4 Method

This section introduces the design details of the point cloud generative model. The overview of the point cloud generation based on the stochastic process is shown in Figure 3.2.

The point cloud generative model utilises a point-based network to extract each shape's latent variable and employs a normalising flow to

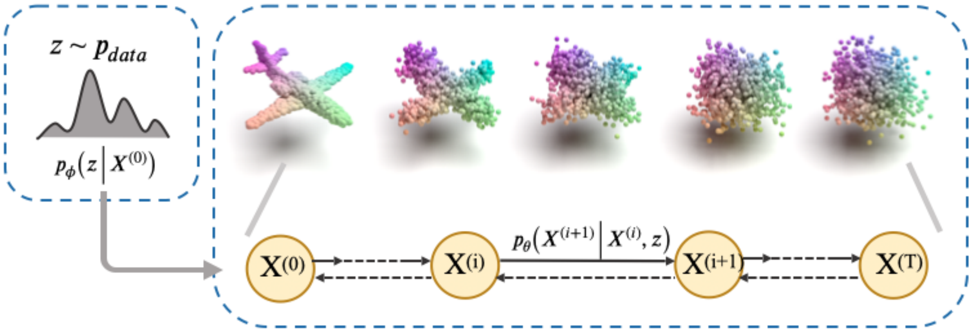


Figure 3.2: The illustration of the directed graphical model of the stochastic process for point cloud

transform the latent variable distribution into a Gaussian prior distribution. During the training stage, forward stochastic differential equations and feature distribution are leveraged to train the point cloud generator.

The two main modules in the point cloud generative model - normalising flow and time encoding - are introduced. Normalising flow is a tuple-stack model that learns complex data representations, and its definitions, functions, and implementation details are discussed in Section 3.4.1. Time encoding simulates the continuous time of stochastic differential equations, facilitating a more flexible sampling of the stochastic process. The effects and formulations of time encoding are explained in Section 3.4.2. The adopted research framework partitioned the training objective into three components. Section 3.4.3 details the training objective and implementation.

3.4.1 Latent Variable Reparameterization

This section details the implementation of the normalising flows technique (Rezende and Mohamed 2015, Dinh et al. 2016) within the point cloud generation model based on stochastic differential equations. Normalising flows are a class of deep-learning models used to perform density estimation and probabilistic modelling. The core idea behind normalising flows is to transform a simple probability distribution (such as a Gaussian or Uniform distribution) into a more complex one by applying a series

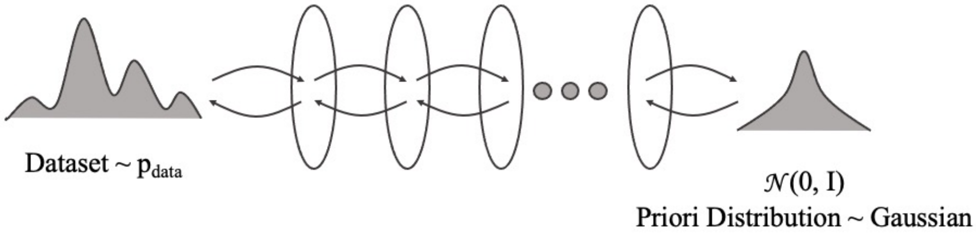


Figure 3.3: Illustration of normalising flow

of invertible, differentiable transformations. These transformations are devised to adjust the probability density of the initial distribution to resemble the target data distribution (Figure 3.3).

A normalising flow constitutes a stack of affine coupling layers $f = \{f_1, .., f_n\}$ as a reversible transform between a prior distribution and a complicated distribution. In particular, $p_{data} = f_n \circ f_{n-1} \circ \dots f_1 (z)$ is the output variable and $\mathbf{z}$ can be estimated from p_{data} via the inverse mapping:

$$z = f_1^{-1} \circ \dots f_n^{-1} (p_{data}) , \tag{3.26}$$

where $\circ$ denotes the Hadamard product. Normalising flows are particularly useful for modeling complex, high-dimensional probability distributions, and have applications in generative modeling, Bayesian inference, and other areas of machine learning.

The transformations used in normalising flows are reversible, enabling their application in both forward and reverse directions. This attribute facilitates efficient sampling from the learned distribution (via inverse transformations) and computation of the probability density function of a data point using the change of variables formula. Additionally, normalising flows can learn highly expressive and complex probability distributions by concatenating a series of straightforward transformations, as shown in Figure 3.4. By adjusting the architecture and depth of the model, normalising flows can capture intricate dependencies and correlations in the data.

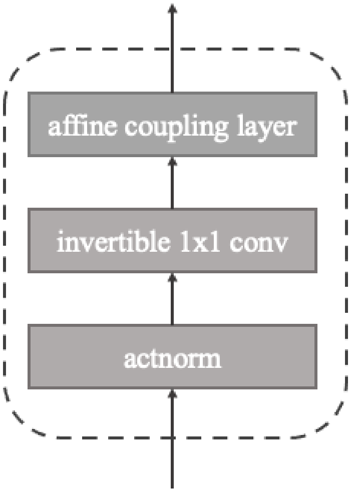


Figure 3.4: The illustration of one layer of normalising flow.

Given these properties and challenges, a specific application is proposed for the modelling of point clouds. The permutation invariance property of point clouds necessitates careful handling, which can be computationally demanding and complex. To tackle this, the distribution of distribution framework (Yang et al. 2019) is employed, modelling the generation process as a distribution of point cloud distributions framework. The distribution of distribution framework refers to the framework that generates a point cloud, which is the distribution of output based on a latent code, that is, the simulated distribution of the point cloud dataset. This framework is a hierarchical approach for modelling 3D point clouds, where the first level of distribution represents the overall shapes, and the second level models the distribution of points within a given shape. This framework decouples the generation process into two stages. In the first stage, this network samples a latent code from the base distribution, which captures the global shape properties and variations across different point clouds. In the second stage, the network models the local structure and fine-grained details of individual point clouds.

The chosen form of Normalising Flows for this task is the Finite Flows,

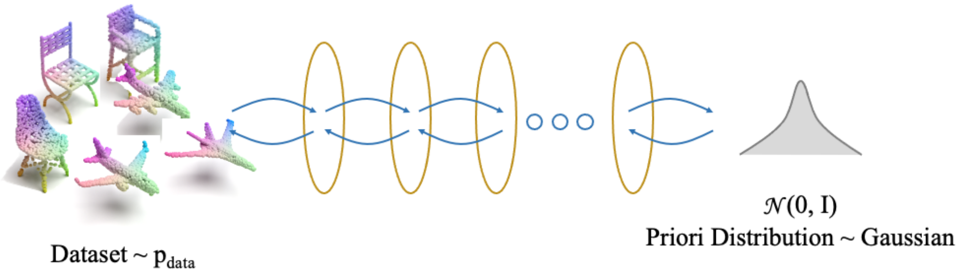


Figure 3.5: Illustration of normalising flow architecture

comprising a finite number of reversible transformations. Figure 3.5 illustrates the architecture of normalising flow that is adopted in the experiment. Formally, given the latent variable p_{data_X} with distribution p_{data}, let network φ denote instantiated affine coupling layers that map p_{data} to the output variable z with prior distribution $\mathrm{P}\left(z\right)$. The exact probability of the output variable is estimated by the change of variables formula:

$$\mathrm{P}\left(p_{data}\right) = \mathrm{P}(z)\left(\frac{\partial\phi}{\partial z}\right) \tag{3.27}$$

,where $z = \varphi^{-1}\left(p_{data}\right)$.

3.4.2 Time Encoding

This section introduces the concept of Time Encoding, which facilitates the accurate capture of temporal sequences within the point cloud generation. It elaborates on the theory behind Time Encoding and its specific implementation in the experimental framework.

Time encoding (Vaswani et al. 2017, Devlin et al. 2018, Lazar and Pnevmatikakis 2011) is a technique used in neural networks to incorporate information about the order or position of elements in a sequence, such as tokens in a sentence or time steps in a time series. Time encoding refers to methods that embed temporal information into the input features, helping the model to understand the order of events in a time series or the progression of tokens in a sequence. Time encoding can be implemented using various techniques, such as adding timestamps or time-based features to the input data, or using time-aware attention mechanisms in the model architecture. Time encoding is particularly useful for time series analysis, forecasting, and sequence-to-sequence tasks.

In this point cloud generation diffusion model with a stochastic differential equation, the additional input time step t allows a single model to use a common set of parameters to handle different noise levels. However, the experiment shows the network can ignore the time step t when attaching it with input directly. Besides, it is suboptimal to increase the parameter of the network to handle this parameter, which increases the learning burden. Therefore, the proposed method adopts time encoding with Gaussian random features to encode time step t (Tancik et al. 2020).

In particular, the time embedding TE is defined as:

$$TE = [\sin(2\pi wt); \cos(2\pi wt)] \tag{3.28}$$

where operator $[a, b]$ denotes the concatenation; $w \sim \mathcal{N}(0, I)$ is a frozen random matrix. This time embedding plays a crucial role in ensuring the time-aware capability of our model, allowing it to adequately account for temporal dynamics present in the point cloud generation process. The use of Gaussian random features for time encoding is justified by their flexibility and expressiveness, making them suitable for representing complex time-dependent dynamics. Moreover, since w is frozen (i.e., it does not change during training), this approach does not add any learnable parameters to the model, thereby reducing the computational cost and the risk of overfitting.

3.4.3 Training Objective

It can be seen from Equation 3.25 that the network can simulate the denoising of point cloud generation process by minimising $\|s_\theta(x_t, t) - \nabla_{x_t} \log p(x_t|x_0)\|^2$. In this section a forward stochastic process is constructed and shows how to solve $\nabla_{x_t} \log p(x_t|x_0)$.

However, it is not easy to solve $p(x_t|x_0)$. Therefore, without loss of generality, it can be assumed that the data set confirms the standard normal distribution, that is $\tilde{p}\left(x_0\right) \sim \mathcal{N}\left(x_0; \mathbf{0}, \mathbf{I}\right)$ and $p\left(x_{t-1}|x_t, x_0\right)$ conforming to Gaussian distribution.

Defined in DDPM (Ho et al. 2020),

$$p(x_t|x_0) \sim \mathcal{N}(x_t; \bar{\alpha}_t x_0, \bar{\beta}_t^2 \mathbf{I}), \tag{3.29}$$

where $\alpha > 0$, $\beta > 0$, $\alpha^2 + \beta^2 = 1$.

In this case,

$$\begin{aligned} x_t &= \bar{\alpha}_t x_0 + \bar{\beta}_t \epsilon \\ x_{t+\Delta t} &= \bar{\alpha}_{t+\Delta t} x_0 + \bar{\beta}_t \epsilon \end{aligned} \tag{3.30}$$

However, unlike DDPM, which fits a Markov chain with a fixed number of steps, the proposed method tries to simulate a continuous stochastic process. Therefore, this research assumes $t \in [0, 1]$, $\bar{\alpha}_0 = 0$, $\bar{\alpha}_1 = 1$, $\bar{\beta}_0 = 0$, $\bar{\beta}_1 = 1$.

Combined with Equation 3.13, we can obtain:

$$\begin{aligned} x_{t+\Delta t} &= x_t + f_t \Delta t x_t + g_t \Delta t \epsilon \\ &= (1 + f_t \Delta t) x_t + g_t \sqrt{\Delta t} \epsilon \end{aligned} \tag{3.31}$$

From this we can get,

$$\begin{aligned} \bar{\alpha}_{t+\Delta t} &= \left(1 + f_t \Delta t\right) \bar{\alpha}_t \\ \bar{\beta}_{t+\Delta t}^2 &= \left(1 + f_t \Delta t\right)^2 \bar{\beta}_t^2 + g_t^2 \Delta t \end{aligned} \tag{3.32}$$

Let $\Delta t \rightarrow 0$, respectively solve,

$$f_t = \frac{d}{dt}\left(\ln \bar{\alpha}_t\right) = \frac{1}{\bar{\alpha}_t} \frac{d\bar{\alpha}_t}{dt}, g_t^2 = \bar{\alpha}_t^2 \frac{d}{dt}\left(\frac{\bar{\beta}_t^2}{\bar{\alpha}_t^2}\right) = 2\bar{\alpha}_t \bar{\beta}_t \frac{d}{dt}\left(\frac{\bar{\beta}_t}{\bar{\alpha}_t}\right) \tag{3.33}$$

In this case, the optimize objective in Equation 3.25 can be denoted as

$$\nabla_{\boldsymbol{x}_{\mathrm{t}}} \log p\left(\boldsymbol{x_t} \mid \boldsymbol{x_0}\right) = -\frac{\boldsymbol{x_t} - \bar{\alpha}_t \boldsymbol{x_0}}{\bar{\beta}_t^2} = -\frac{\boldsymbol{\varepsilon}}{\bar{\beta}_t} \tag{3.34}$$

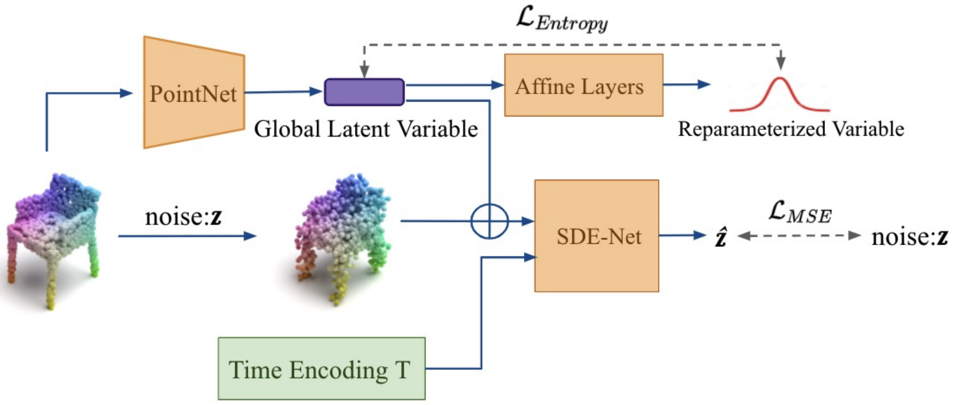


Figure 3.6: The illustration of the training phase of the proposed model. The z is noise with a prior distribution. The Global Latent Variable p_{data} is the distribution of data. The affine layers map the data distribution to the latent variable p_{latent} with Gaussian distribution.

According to Ho et al. (2020), we can set $\boldsymbol{s}_\theta(x,t) = -\frac{\epsilon_\theta(x,t)}{\bar{\beta}_t}$ based on Bayesian definition, and substituting the above formula into the Equation 3.25 can get

$$\frac{1}{\bar{\beta}_t^2}\mathbb{E}_{x_0\sim\bar{p}(x_0),\boldsymbol{N}\sim(\mathbb{N},I)}\left[\left\|\varepsilon_\theta\left(\bar{\alpha}_t\boldsymbol{x}_0+\bar{\beta}_t\boldsymbol{\varepsilon},t\right)-\boldsymbol{\varepsilon}\right\|^2\right] \tag{3.35}$$

Our model is trained in an end-to-end fashion by minimising the above objective of all point sets in the dataset. The training phase is shown in Figure 3.6. The θ in the figure denotes the diffusion model to approximate stochastic differential equations, and network ϕ learns the distribution of p_{data}.

The proposed method implements a point cloud auto-encoder based on stochastic differential equations in experiments. It is possible to directly apply KL loss over the latent variable outputted by ϕ, but it has been proved that it unavoidably restricts the performance of network (Chen et al. 2016). We employ normalising flow to enhance the representation of the network instead of using KL loss to parameterise the latent variable. Formally, leveraging on Equation 3.25, we rewrite the objective of our network:

$$
\begin{aligned}
\mathcal{L}(\theta, \phi, \varphi) = &\left(\frac{\varepsilon_\theta\left(x, t\right)}{\bar{\beta}_t}\right)^2 \\
&+ D_{KL}\left(\phi(p_{data}|X^0)\|\mathrm{P}(\mathbf{z})\left|\det\left(\frac{\partial\varphi}{\partial\mathbf{z}}\right)\right|^{-1}\right) \\
&+ H\left[\phi(p_{data}|X^0)\right]
\end{aligned}
\tag{3.36}
$$

where the first optimise objective $\left(\frac{\varepsilon_\theta(x,t)}{\bar{\beta}_t}\right)^2$ is the training objective of the SDE-Net θ; the second training object $\det\left(\frac{\partial\varphi}{\partial\mathbf{z}}\right)$ is for optimize the affine layers φ; KL loss and entropy loss are used to optimize the encoder ϕ and the affine layers. The training adopts end-to-end fashion. Portions of this section are based on the published paper Li et al. (2023).

3.5 Implementation process and experiments

This section details the process of implementation and analyses the experimental results to evaluate the performance of our proposed point cloud generation model. The experiments are designed to explore the model's capacity in various tasks, such as point cloud generation, reconstruction, and completion. This section incorporates material from the following our paper Li et al. (2023).

First, in Section 3.5.1, the experimental setup is introduced, including the dataset, the configuration of the point cloud generation network, and the specific parameters adopted during the training phase. Next, Section 3.5.2 explains the Evaluation Metrics in detail, which are crucial for evaluating tasks related to point cloud generation. The section also includes the respective definitions of each metric. The selection of these metrics was based on their ability to provide a comprehensive evaluation of the point cloud generation model's performance. Then, the experimental results of the point cloud generation network on the generation task are shown and comprehensively discussed the experimental results in Section 3.5.3. Because the network framework adopts the Autoencoder structure, the results of the point cloud reconstruction experiment are shown and compared as well in Section 3.5.4. The extended application of the generation network - the results of point cloud completion are discussed in Section 3.5.5.

3.5.1 Experimental setting

The experiment of this section is carried out on the ShapeNet datasets (Chang et al. 2015) and ModelNet40 (Uy et al. 2019). ShapeNet consists of 51,127 point clouds for the training set and 1,184 for testing from 55 object categories. The proportion of training, testing and validation sets respectively are 80%, 15% and 5%. ModelNet40 contains 12311 mesh CAD models from 40 categories, of which 9843 CAD models are used for training, and 2468 CAD models are used for testing. The main experiments of this chapter are conducted on chair, airplane, car, and guitar models to demonstrate the proposed method's effectiveness. For each shape for training, we randomly sample 2048 points from it. It should be emphasised that the implementation of our method is not constrained by the number of sample points.

The proposed method adopts PointNet for the architecture of encoder ϕ. The architecture of PointNet is shown in Figure 3.7. As for latent variable parameterisation φ, the proposed method uses 3 layers with 256 hidden units and a ReLU activation function. An MLP architecture is employed for modelling network θ with stochastic differential equations. We apply 6 layers of full linear for the model θ. We use 1 layer of 64 hidden units for the Time encoding.

3.5.2 Evaluation Metrics

The experiment uses 128 batches (batch size) to train on different categories of the two datasets and uses 128 batches for direct training on all categories. The Adam optimiser with a momentum of 0.9 is used for optimisation, the initial learning rate is set to 0.001, and the decay rate is 0.5. For the classification network, it needs to train 1000 rounds

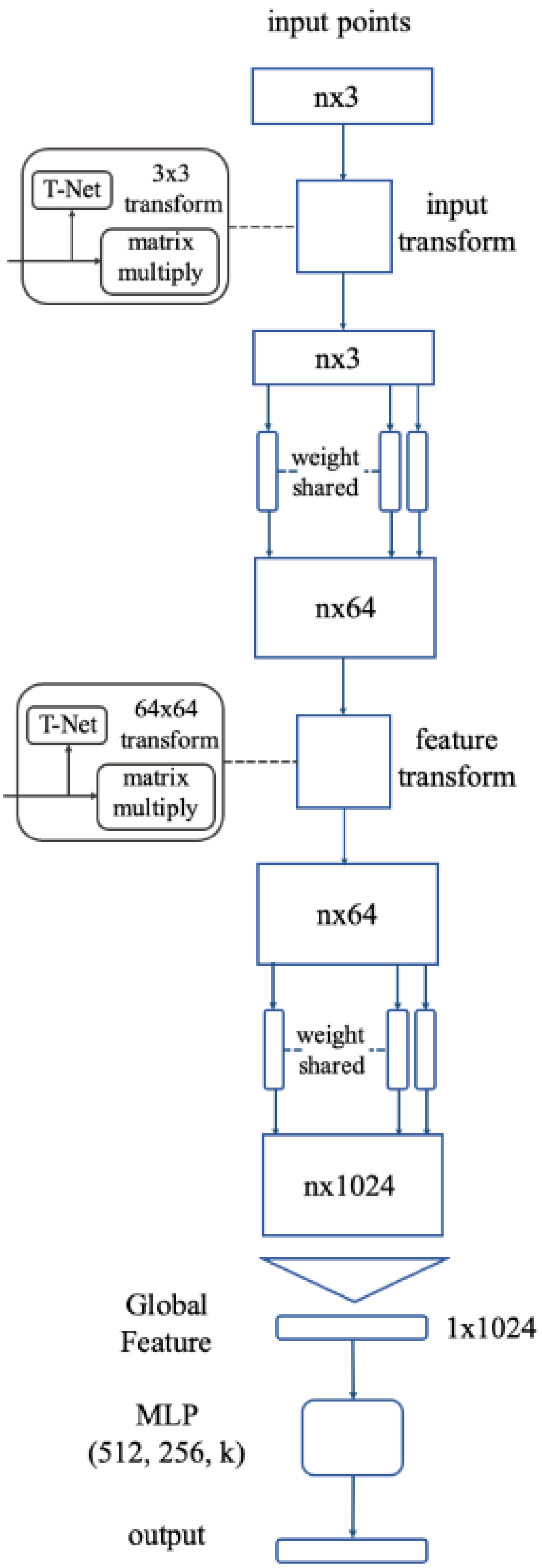


Figure 3.7: Illustration of PointNet architecture (Qi et al. 2017a).

(Epoch). It takes about 12 hours for training point cloud generation to converge. All experiments are performed on NVIDIA RTX 2080Ti GPU. During the inference phase, we set T=1000 to generate each point cloud.

In the context of deep learning, compared with traditional computer systems, models trained from large amounts of data have higher diversity and complexity. Therefore, benchmark evaluation technology will cover a wide range of applications and provide various evaluation standards. This section summarises the benchmarks in the field of point cloud generation.

Coverage (COV) (Achlioptas et al. 2018) is a metric to assess how well the model can capture the diversity and characteristics of the data distribution of the ground-truth dataset. For a set of high dimensional data, for example, a set of N-dimensional data vectors, each of them represents a specific sample, then the COV can be used to describe the variation relationship between each pair of dimensions in these samples. COV can reveal which features of the data are correlated, whether those features are positively or negatively correlated, and to what extent they are correlated.

To compute the Converage between the generated dataset $X = \{x\}$ and the ground-truth dataset $Y = \{y\}$

$$\mathrm{COV}\left(X, Y\right) = \frac{|\{\arg\min_{y \in Y} D(x, y) \mid x \in X\}|}{|Y|} \tag{3.37}$$

where $|Y|$ denotes the number of the point cloud in the ground-truth dataset; $D(\cdot, \cdot)$ denotes a distance measurement method that can be CD or EMD; $COV \in [0, 1]$, where $COV = 0$ denotes none of the point clouds in dataset X is covered by the point clouds in dataset Y within the distance threshold ϵ; $COV = 1$ denotes all of the point clouds in dataset X are covered by the point clouds in dataset Y within the distance threshold ϵ. COV alone might not provide a complete assessment of the performance of a generative model. It is often used in conjunction with other metrics to provide a more comprehensive evaluation of the model's performance.

Minimum matching distance (MMD) (Achlioptas et al. 2018) is a similarity metric used to compare two sets of points or objects, typically in Euclidean space. The key idea behind MMD is to find the best possible correspondence between the points in the two sets, such that the overall distance between the matched points is minimised.

To compute the Maximum Mean Discrepancy (MMD) between the generated dataset X and the ground-truth dataset Y, we first need to calculate the mean embeddings of the two distributions X and Y in a Reproducing Kernel Hilbert Space (RKHS) using a kernel function K.

$$MMD(X,Y) = \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in X} D(X,Y), \tag{3.38}$$

where $D(\cdot,\cdot)$ can be replaced by CD or EMD;

1-NN classifier accuracy (1-NNA) (Xu et al. 2018a) is a measure of the performance of the KNN classifier on a given dataset. This metric can be used to measure whether the distributions of two sets are equal.

$$1 - NNA(X,Y) = \frac{\sum_{x \in X} D\left[N_x \in X\right] + \sum_{y \in Y} D\left[N_y \in Y\right]}{|X| + |Y|}, \tag{3.39}$$

where N_x denotes the nearest sample of x in X. For each sample, its 1-NNA classifier classifies it into X or Y according to its nearest neighbour sample. If X and Y are sampled from the same distribution and there are enough samples, then the classifier should be 50% accurate. The closer the accuracy rate is to 50%, the distribution of X and Y are closer.

Jensen-Shannon Divergence (JSD) (Yang et al. (2019)) is a symmetric and bounded measure of similarity between two probability distributions. It is based on the Kullback-Leibler (KL) divergence, which is another measure of divergence between probability distributions. JSD is defined as the average of two KL divergences, one for each distribution

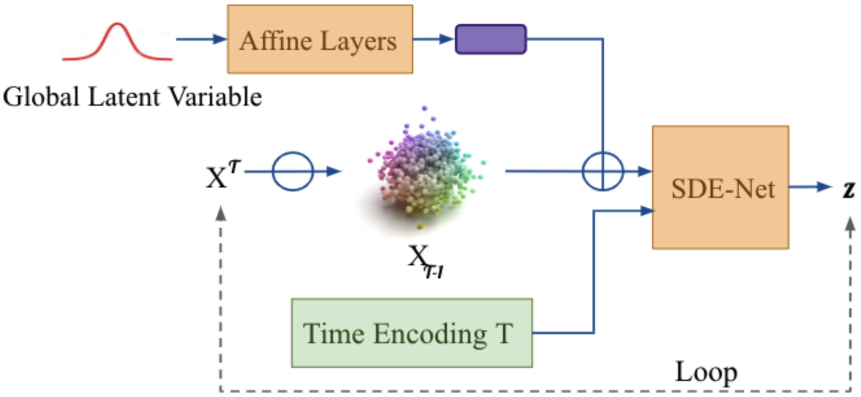


Figure 3.8: The illustration of the inference phase of the proposed model. The dashed line represents the loop process.

with respect to a mixture of the two distributions:

$$\begin{aligned} JSD(X\|Y) &= 0.5 * KL(X||\frac{X+Y}{2}) + 0.5 * KL(Y||\frac{X+Y}{2}) \\ KL(X\|Y) &= \sum P(x) \log \frac{X}{Y} \end{aligned} \tag{3.40}$$

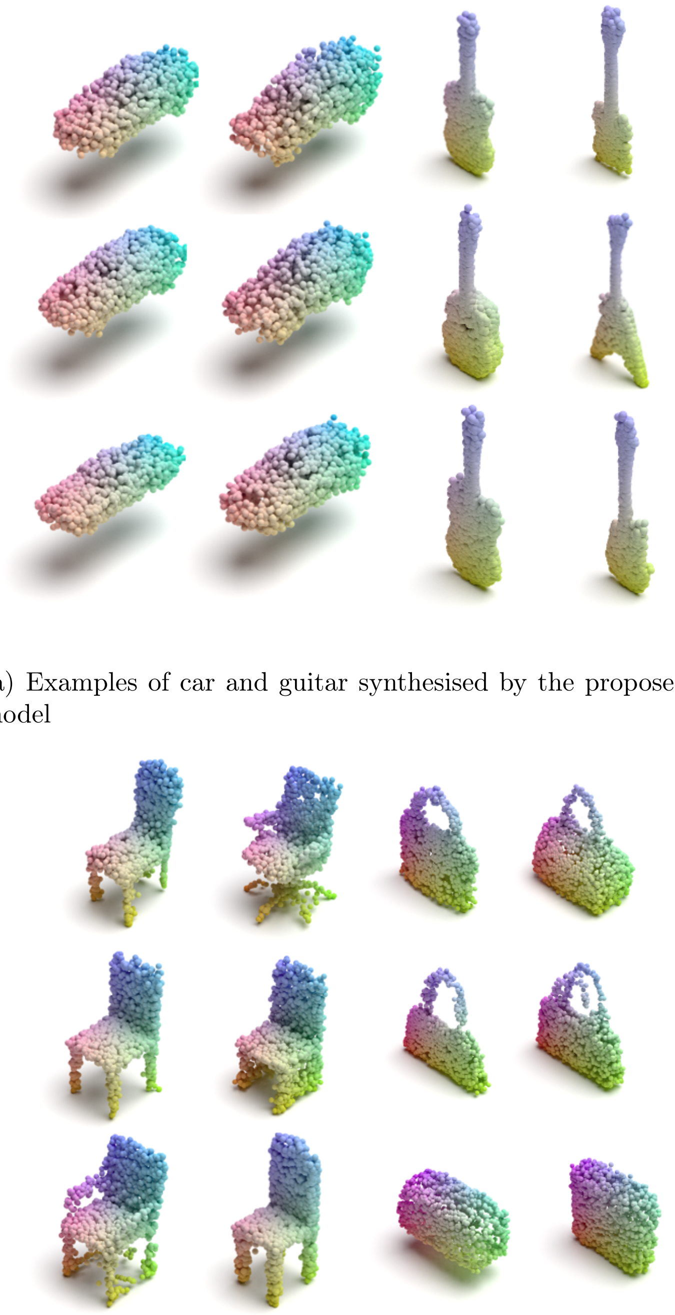
JSD is always non-negative, with $JSD(X\|Y) = 0$ if and only if $X = Y$. JSD is bounded between 0 and log(2), where the upper bound occurs when the two distributions are completely disjoint.

For point cloud synthesis, we follow the evaluation set-up in Yang et al. (2019) and Achlioptas et al. (2018) to compare in terms of all the above metrics.

3.5.3 Point Cloud Generation Experiments

This section shows the results of the generation experiment. In addition, the experiments in this section also performed quantitative comparative experiments to demonstrate the superiority of our method through tables and compared them through visual effects.

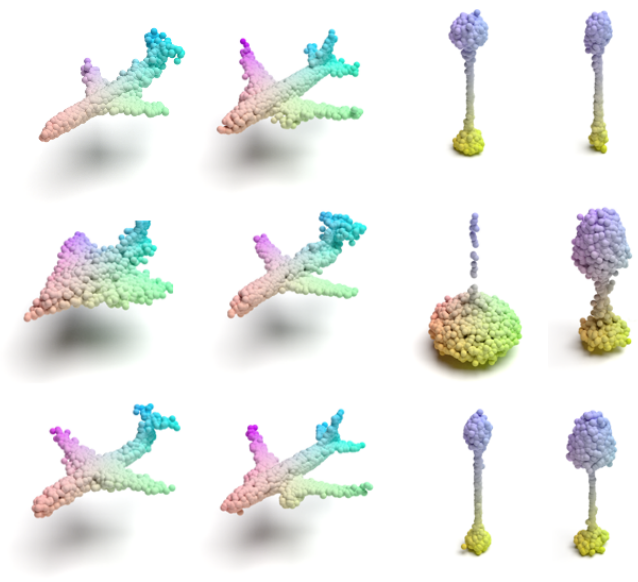
Figure 3.8 shows the inference phase of this process. Figure 3.9 shows some examples of point clouds generated by the proposed model. We normalise each generated shape and evaluate the proposed model generated point clouds by the metrics in Section 3.5.2. The experimental results demonstrate that the proposed method can synthesise a point cloud with a distinguished structure and clear surface.



(a) Examples of car and guitar synthesised by the proposed model

(b) Examples of chair and bag synthesised by the proposed model

Figure 3.9: Examples of point clouds synthesised by the proposed model.



(a) Examples of airplane and lamp synthesised by the proposed model

Figure 3.9: Examples of point clouds synthesised by the proposed model.

In addition, we quantitatively compare the proposed method with the following state-of-the-art generative models: PC-GAN (Achlioptas et al. 2018), GCN-GAN (Valsesia et al. 2018), TreeGAN (Shu et al. 2019a) and PointFlow (Yang et al. 2019). The comparison results are shown in Table 3.1. It can be seen that the proposed method outperforms other types of point cloud generative models and reaches competitive results compared with the diffusion-based point cloud generative model.

3.5.4 Point Cloud Reconstruction Experiments

Because according to the design in Section 3.2, the proposed model adopts the framework of Autoencoder. To further demonstrate the effectiveness of this approach, this section presents the effect of point cloud reconstruction. In the point cloud reconstruction experiment, firstly, the effect of the method is compared and displayed in a visual way. Then the validity of the method is proved by quantitative comparison.

Table 3.1: Comparison of point cloud generation performance.

Category	Model	MMD(↓)		COV(%,↑)		1-NNA(%,↓)		JSD(↓)
		CD	EMD	CD	EMD	CD	EMD	-
Airplane	Wu et al. (2016b)	3.819	1.810	60.17	13.84	97.59	98.52	6.188
	Valsesia et al. (2018)	4.713	1.650	51.04	18.62	89.13	98.60	6.669
	Shu et al. (2019a)	4.323	1.953	51.37	8.40	84.86	99.67	15.646
	Yang et al. (2019)	3.692	1.990	47.98	64.65	82.39	85.36	3.296
	Luo and Hu (2021)	3.588	**1.101**	45.00	**44.65**	**79.20**	83.22	2.921
	Ours	**3.508**	1.132	**45.02**	39.20	78.00	**82.12**	**2.383**
Chair	Wu et al. (2016b)	13.436	3.104	90.23	22.14	69.67	100.00	6.649
	Valsesia et al. (2018)	15.354	2.213	34.84	15.09	81.86	95.80	21.708
	Shu et al. (2019a)	14.936	3.613	32.02	6.77	82.92	100.00	13.282
	Yang et al. (2019)	13.631	1.856	36.86	26.38	76.13	78.40	12.474
	Luo and Hu (2021)	**12.211**	1.900	33.84	**44.22**	**70.20**	**69.44**	**7.821**
	Ours	12.879	**1.819**	**39.53**	41.86	69.46	73.50	9.499

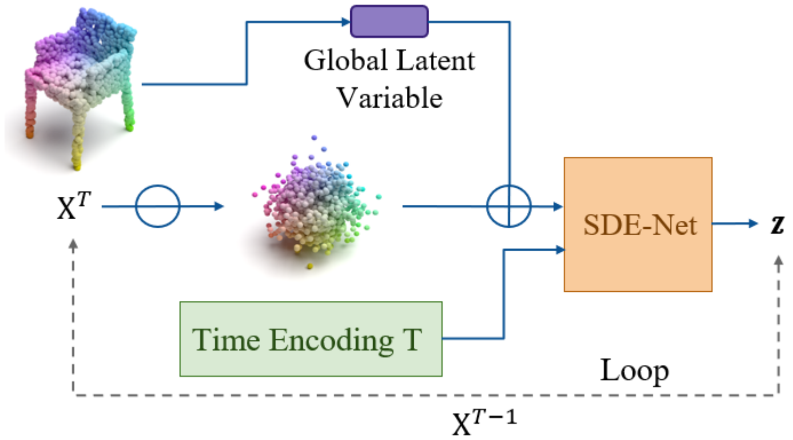


Figure 3.10: The illustration of the reconstruction phase of the proposed model. The dashed line represents the loop process.

We compare with state-of-the-art point cloud auto-encoder: AtlasNet (Groueix et al. 2018), PointFlow (Yang et al. 2019), and ours. We evaluate the quality of the proposed method with three categories: airplane, chair, and all categories, and the comparison results are shown in Table 3.2. As shown in Table 3.2, the proposed method achieves better performance in CD score and competitive results when compared to EMD. All the point cloud in comparison experiments contain the same number of points and are normalised by the same approach. We visualise the point cloud reconstruction as shown in Figure 3.11. It can be seen that the proposed method can reconstruct faithful and clean point clouds.

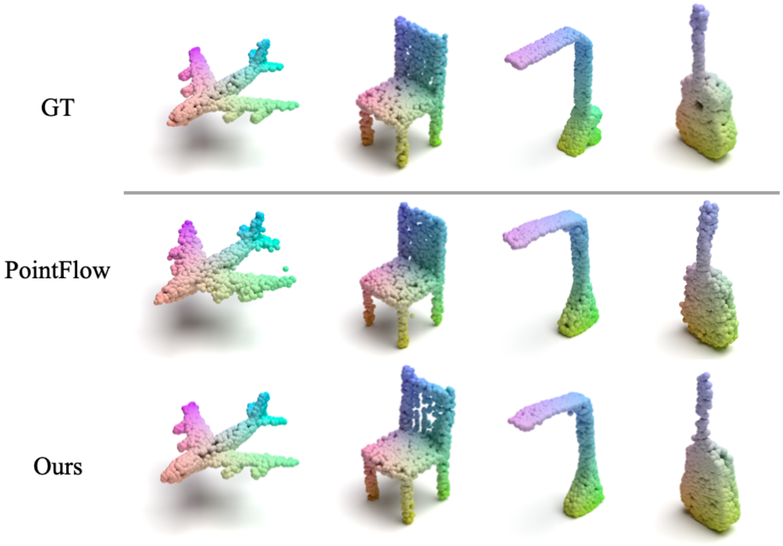


Figure 3.11: Examples of point cloud reconstruction.

Table 3.2: Comparison of point cloud reconstruction performance.

Dataset	Metric	(Groueix et al. 2018)(S1)	(Yang et al. 2019)	Ours
Airplane	CD	**2.000**	2.420	2.921
	EMD	4.311	**3.311**	3.624
Chair	CD	6.979	6.795	**6.631**
	EMD	5.550	5.008	**4.578**
All	CD	6.906	7.550	**6.130**
	EMD	5.617	5.172	**4.456**

We further implement extrapolation and visualise the point cloud in Figure 3.12. In this experiment, we project the global latent variable produced by the auto-encoder encoder and interpolate between them. The interpolation results show the interpolated shapes generated by the proposed method, demonstrating the model's ability to learn informative representations and smoothly transition between different point cloud formations within the latent space.

3.5.5 Point Cloud Upsampling Experiments

This section illustrates the effect of point cloud upsampling. During inference, the proposed model can generate a sequence of point-wise move distance sampling each point cloud, which allows point cloud completion. Motivated by this property, we conduct an additional experiment to point cloud upsampling and completion as the applications of the proposed

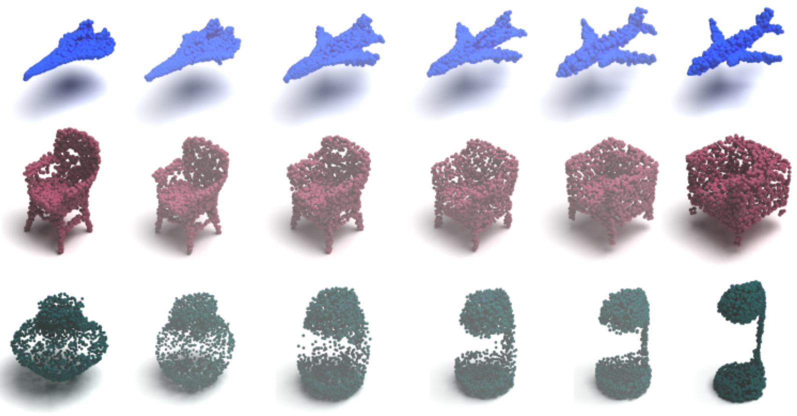


Figure 3.12: Global latent variable interpolation.

method.

Specifically, we use the partial point cloud as the input and use the pre-trained encoder to estimate the global shape variable. Then we employ the global shape variable and the partial point cloud to synthesise the completed point cloud. The visualised qualitative results are shown in Figure 3.13. As shown in the results, the proposed model can complete a precise point cloud when the reference point cloud is sparse.

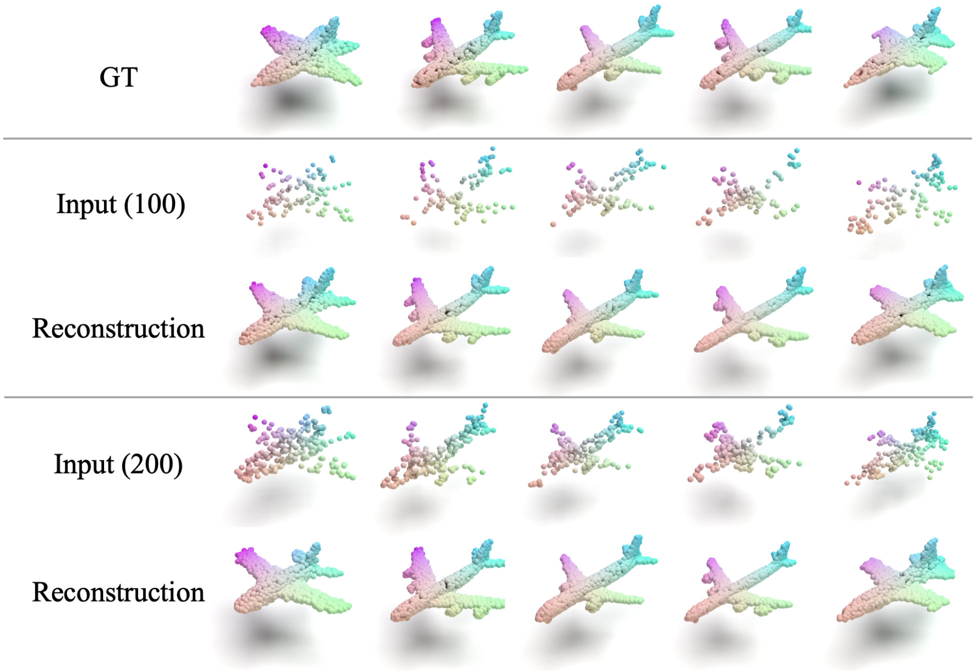


Figure 3.13: Visualised experimental results of point cloud completion.

3.6 Summary

In this chapter, point cloud generation is transformed as a transition process between a distinguished shape and noise with prior distribution, which is the Gaussian distribution in a discussion. Then stochastic process and stochastic differential equations are incorporated to simulate and solve the generation process. A framework for a point cloud generative model based on SDE is detailed and implemented. The proposed work brought a new point cloud conditional generation approach to the family of point cloud generation based on diffusion models. By combining the time encoding and SDE, the proposed method can make the transformation between the noise and point cloud more smooth and more flexible. Additionally, the Lagvien MCMC sample is employed to improve the quality of the generated point cloud. Experimental results demonstrated that the proposed model can generate an expressive point cloud and achieve competitive results compared with other methods.

Chapter 4

Generative Approach to 3D Point Cloud Annotation Method

The preceding chapter focuses on solving the annotated 3D point cloud generation challenge. This chapter proposes a label-efficient point cloud annotation solution based on a point cloud generative model, building upon the foundational understanding developed.

As discussed in Chapter 1, point cloud generative models capture point cloud collections' shape, pattern and semantic information. In light of this, the goal of this method is to reuse a pre-trained point cloud generative model to generate point cloud and its corresponding label. To demonstrate the effectiveness of the proposed approach, experiments have been conducted on the point-wise semantic transformation of a point cloud generator, demonstrating its usability for semantic segmentation tasks. The experimental results indicate that the intermediate features learned through the point cloud diffusion-based model are interpretable compared to the representations obtained by existing supervised techniques, such as Zhao et al. (2019b), Yang et al. (2019). This chapter is organised as follows:

- Section 4.1 overviews the motivation and the overall architecture of the proposed method. Additionally, this section summarises the observation in point cloud diffusion-based generative models and brings up the main assumption, which motivates the proposed approach.

- Section 4.2 explores an approach to generate point-label pairs based on the point cloud diffusion-based method. In this section, the introduction of the intermediate feature of point cloud generation is first presented, and a method to analyse the intermediate feature is provided. Based on the analysis of intermediate features, a feature interpreter is introduced and applied to generate point-label pairs.
- Section 4.3 shows the experimental results of the proposed method in different experimental setups and compares the effectiveness of the proposed method.
- Section 4.4 summarises this chapter.

4.1 Introduction

With the advancements in hardware such as LIDAR and scanners, there has been a corresponding surge in applications reliant on three-dimensional data. These applications have stimulated the demand for advanced three-dimensional technologies, emphasizing the need for refined tools and methods capable of handling, processing, and understanding complex 3D data structures, amongst which those techniques based on deep learning have exhibited the most rapid development, demonstrating impressive capabilities in handling complex 3D data. Despite their remarkable performance, deep neural networks demand a large quantity of labelled and clean data for training, making the process both time-consuming and costly. This limitation hinders the widespread development of these technologies in real-world applications.

Another solution that uses a point cloud generative model to generate point cloud with point-wise semantic label, Mo et al. (2020), Gal et al.

(2021), Yang et al. (2021), Shu et al. (2019b), is to synthesise expressive point clouds while having control of the structure. Point clouds and point-wise semantic labels are bred from key structural points in these methods. However, due to the irregular distribution and high complexity of 3D point clouds, existing generative models often struggle with explicit structural controllability and producing realistic-looking shapes. The approach in this chapter goes beyond existing solutions in terms of explicit point-label pairs generation as it generates point-wise labels without affecting the shape generation results because the semantic information is obtained from the intermediate features of the generator.

By learning from unlabeled data, the generative model produces point clouds with prominent semantic features, demonstrating a robust representation learning ability. This is because the process of generating realistic point clouds inherently involves learning the spatial configuration of various semantic components, therefore, encoding valuable semantic information within the point cloud generation network. This approach is motivated by two observations: First, the coarse structure of the point cloud is primarily recovered at the early stage of the diffusion process by the generator of the diffusion model, with details gradually enriched at the later stages. Second, during the diffusion process, the generator should generate a set of independent and identically distributed random variables, and it can form a plausible shape.

These observations lead us to analyse the intermediate features of the point cloud diffusion generative model, seeking to understand its discriminability and its potential for semantic interpretation. The intermediate features are aggregated and transformed into point-wise semantic labels by the Feature Interpreter. This approach enables the annotation of point-wise labels without affecting the quality of point cloud generation. The above section extends the discussion from our published work Li et al. (2022b).

4.2 Intermediate Feature Analysis and Feature Interpreter

This section presents the method, referred to as DiffusionPointLabel, for generating point cloud datasets along with their corresponding semantic labels. The objective of this method is to extract meaningful information from the pre-trained generative model and create an effective representation of structure and semantics within the generated point cloud datasets. This approach can be divided into two main parts: Intermediate Feature Analysis (Section 4.2.1) and the Feature Interpreter (Section 4.2.2). This section details each of these components and discusses their respective roles in point cloud dataset generation.

4.2.1 Intermediate Feature Analysis

This section introduces the intermediate feature in the point cloud generative model and a general approach for analysing it. Then this section elaborates on how to analyse the intermediate feature of the point cloud diffusion-based generative model. The analysis demonstrates the discriminability of intermediate features that can serve as the foundation for developing the Feature Interpreter, which is essential for generating explicit semantic labels.

4.2.1.1 Intermediate Feature Analysis Preliminary

An ongoing criticism of using neural networks is that they are black-box approaches with little understanding of what the network does in the form of simple human-consumable algorithms. Therefore, some methods have been proposed to gain insight into the learning process of the network by analysing the characteristics of the middle layer of the neural network, improving its design and performance, so enhance the interpretability of deep neural network. One of the widely studied topics is to learn good intermediate representations from a nearly unlimited amount of unlabeled images and videos, which can then be used in various supervised learning tasks such as image classification (Ng and Jordan 2001) and recognition (Ranzato et al. 2011, Hinton 2007).

In a deep neural network, the input data is sequentially transformed by a series of layers. Each layer performs a specific operation (e.g., convolution, pooling, normalization, or activation) and generates a new set of features. As data spans through the network, features become more abstract and high-level, enabling the network to recognise and represent more complex relationships in the data. Intermediate features are the result of applying transformations to the input data as it spans through the network. They capture increasingly complex patterns, structures, and abstractions in the data, enabling networks to learn and generalise efficiently. One of the examples is shown in Figure 4.1. As illustrated in this Figure, different layers of the network focus on different features, showing that the network can distinguish among various data types at different layers. This figure provides a tangible example of how intermediate features work in image generation networks.

Zeiler and Fergus (2014) and Zintgraf et al. (2017) exhibited that each layer in an image classifier network captures an increasingly complex and abstract representation of the input data. Each layer of the network gradually extracts more and more advanced image features until the last layer compares these features to make a classification result. Li et al. (2021a) and Zhang et al. (2021b) demonstrated that the intermediate of the GAN-based image generative model can be used to realise image segmentation.

Different from the above two types of methods to analyse the characteristics of the image classification neural network, this paper mainly analyses the characteristics of the middle layer of the point cloud generation model and whether it has interpretable value. Moreover, this study further proposes how to use these intermediate layer features to generate point cloud semantic label. The following analysis will highlight the characteristics and the learnable value of the middle layer in point cloud generation models.

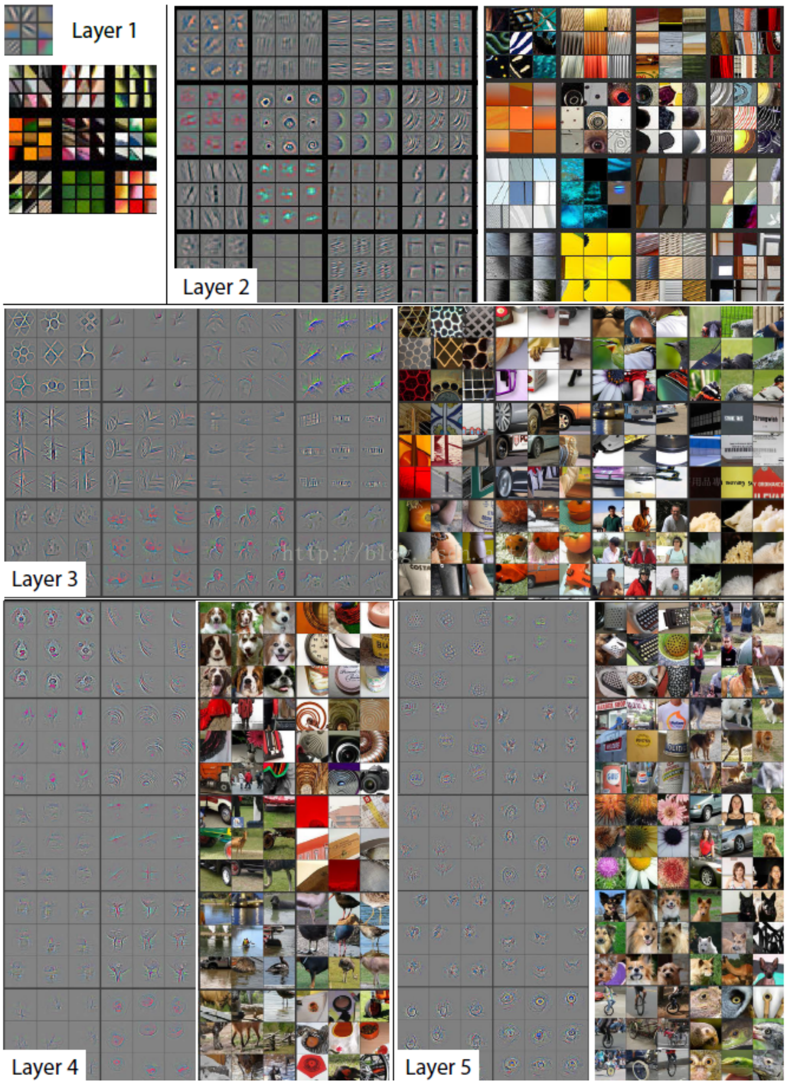


Figure 4.1: Visualization of the intermediate feature in an image generative model (Zeiler and Fergus 2014). For each layer of visualization, the nine images with the largest reconstructed values are displayed, representing distinct features the layer is attentive to.

4.2.1.2 Intermediate Feature Analysis Approach

This section is based on findings from our publication Li et al. (2022b). The point cloud generation process based on the diffusion generative model is shown in the top row of Figure 4.2. As shown in Equation 3.20, the output of the diffusion generator at each step should be independent and identically distributed random variables. However, in the early and later stages of the diffusion process, the change tendency of the point cloud is different: coarse structure in the former and fine details in the latter. Therefore, we assume that the point representation of the diffusion generator has different discriminability alongside the diffusion process.

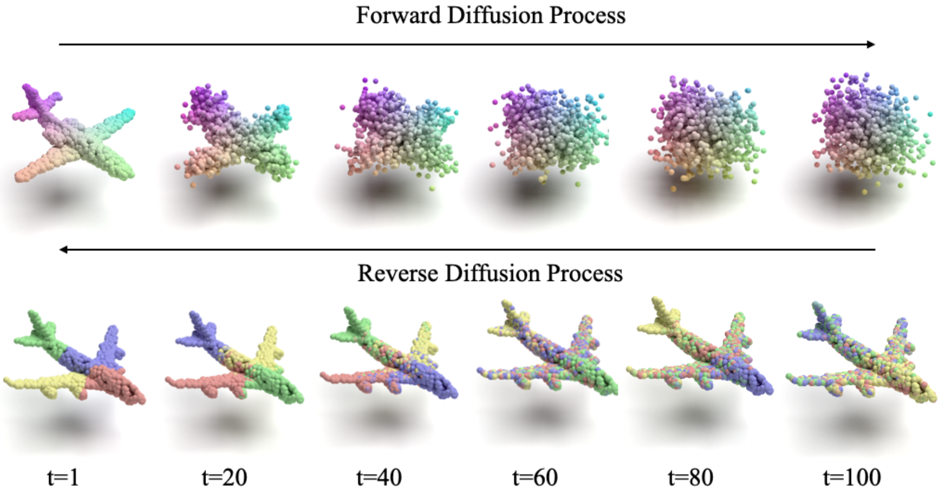


Figure 4.2: Visualization of the diffusion process and corresponding K-means clustering of intermediate features of the point cloud generative model. The top row represents the states of the point cloud in the diffusion process in the time variable. The bottom row represents the results of the corresponding K-means clustering features of the intermediate features.

To prove our assumption in Section 4.1, we use the K-means cluster to analyse the intermediate features of the diffusion generator at different time steps. In practice, we freeze a diffusion model θ and take a point cloud $X \in \mathbb{R}^{N\times 3}$ and a time step t as the inputs of θ. Then we extract the intermediate features of one layer of the generator θ_G. Because the generator we used is an MLP, the intermediate features of every layer at each time step t can be denoted by $C_{i,t} \in \mathbb{R}^{N\times out_i}$. We use the K-means clustering algorithm to estimate the cluster group of each point from the intermediate features of X and visualise the results, as shown in the bottom row of Figure 4.2. In K-means clustering, the number of clusters (K) is informed by the number of semantic parts from the ground truth of the point cloud being analysed. For example, the airplane in Figure 4.2

has 4 semantic parts, and then we set the K-means cluster number to be 4. The results of K-means demonstrate that the discriminability of intermediate layer features gradually increases with decreasing time steps and is interpretable at the semantic level.

To further determine which layer of features we should extract or at which time steps, we quantitatively compute the K-means clustering results. If the K-means clustering effect is good (the cluster groups of the close points are very similar, and the cluster groups of the distant points are not the same), it means that the discriminability of this intermediate feature is very high. Otherwise, it will remain low and cannot be used for further learning. We extract and cluster the intermediate features of each layer of θ_G from time $t = 0$ to T. We compute the clustering results with the Calinski-Harabasz Index algorithm Caliński and Harabasz (1974), and the result is shown in Figure 4.3. The Calinski-Harabasz Index algorithm is used to measure the quality of the cluster model without the ground-truth label. The Calinski-Harabaz score is defined as the ratio of separation and cohesion of clusters. The formulation is shown as follows:

$$s = \left(\frac{SS_B}{k-1}\right) / \left(\frac{SS_W}{N-k}\right) \tag{4.1}$$

where k denotes the number of clusters, N denotes the number of point of point cloud, SS_B denotes the variance between different clusters, and SS_W denotes the variance within one cluster. The higher the score, the better the clustering effect of K-means.

As shown in Figure 4.3, the Calinski-Harabasz Index score starts converging to a high value when $t = T/4$, which means the features are more discriminative and can be well clustered. The experiments utilise the intermediate features at $t < T/4$. It is worth mentioning that this score becomes stable at $t \to T$. The attribution is that the discriminability of intermediate features tends to be consistent at this time, and the point features represent the object's category or overall shape information.

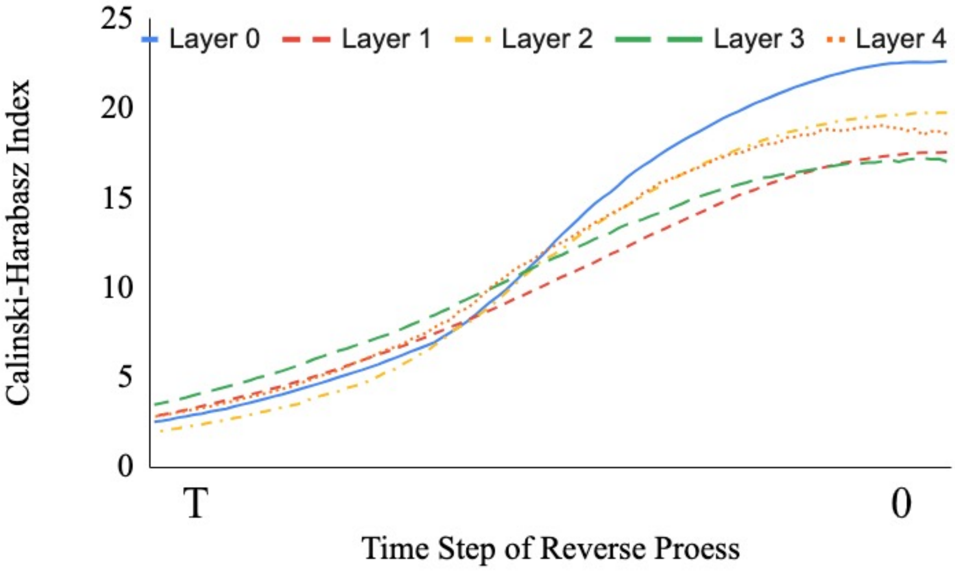


Figure 4.3: Calinski-Harabasz Index of intermediate features clustering results. The score represents the quality of the cluster. Different colors represent different layers of the θ_G

By analysing the intermediate features of the diffusion generator, we have demonstrated their potential to provide valuable information for learning representations of structure and semantics, which is essential for the subsequent development of the Feature Interpreter.

4.2.2 Feature Interpreter

The following section extends the discussion from our published work Li et al. (2022b). Building upon the insights gained from our intermediate feature analysis in section 4.2.1, this section develops the Feature Interpreter to generate explicit semantic labels for the generated point cloud, thereby creating a more comprehensive and meaningful representation of the data.

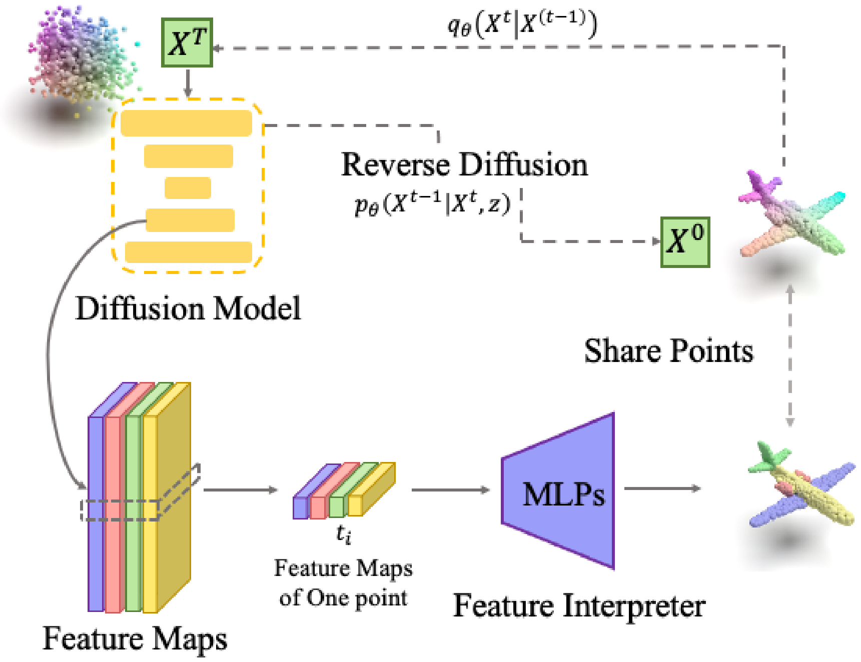


Figure 4.4: Architecture of proposed method.

4.2.2.1 Method Architecture

In our proposed method, the first stage is the extraction of intermediate features from the point cloud. Given a random latent code $z \sim \mathcal{N}(0, I)$ and a random noise of point cloud $X \in \mathbb{R}^{N\times3} \sim \mathcal{N}(0, I)$, our aim is to generate a point cloud $X \in \mathbb{R}^{N\times3}$ along with its corresponding semantic labels $SL \in \mathbb{R}^{N\times b}$. Here, b denotes the number of semantic label categories in the point cloud.

This is achieved by extracting intermediate features from different layers of the diffusion generator θ_G at specific time steps $t = t_i|i = 1, .., T$. Each $C_{i,j} \in \mathbb{R}^{N\times C_{out_i}}$ represents the intermediate feature of the i-th layer at time step $t = j$. We then concatenate multiple $C_{i,j}$ into a composite feature C^*.

In the second stage, the Feature Interpreter takes the composite feature C^* and transforms it into point-label pairs, providing the detailed semantic information associated with the point cloud.

The overall of this method is shown in Figure 4.4. The feature interpreter can serve as a parallel extension as the point cloud generative model, which would not affect the performance of the point cloud generator.

4.2.2.2 Feature Interpreter Details

In the process of generating a point cloud dataset using a diffusion generative model, not only the point cloud but also the corresponding semantic labels should be generated in order to better understand the underlying structure and semantics of the objects in the point cloud. Therefore, this section introduces the concept of feature interpreter to realise the generated point cloud to generate clear point-wise semantic labels.

Feature Interpreter aims to bridge the gap between intermediate features extracted from generative models and semantic labels. By leveraging the multi-layer perceptron (MLP) or a self-attention mechanism, the feature interpreter converts the concatenated intermediate features into point label pairs, providing a way to associate the resulting point cloud with its corresponding semantic label. This process not only enables the generated point cloud to generate explicit point-wise semantic labels but also enhances the interpretability of the generated point cloud.

The Feature Interpreter takes the intermediate features as the input, aiming to generate explicit semantic labels for the generated point cloud. An MLP is implemented to realise the label prediction. Similar to the K-means clustering process, we freeze a diffusion model θ and take a point cloud $X \in \mathbb{R}^{N\times 3}$ and a time step t as the inputs of θ. Based on the analysis in section 3.1, we sample $C_{0,t}$ across different time steps, where $t < T/4$. Then the intermediate features $C_{0,t}$ of the θ_G are upsampled via linear interpolation and concatenated to form $C^* \in \mathbb{R}^{N\times 1024}$. In practice, we use a three-layers MLP to predict the semantic label for each point from the C^*. The Feature Interpreter is optimised by cross-entropy loss. Figure 4.5 shows the structural details of the function of the Feature Interpreter.

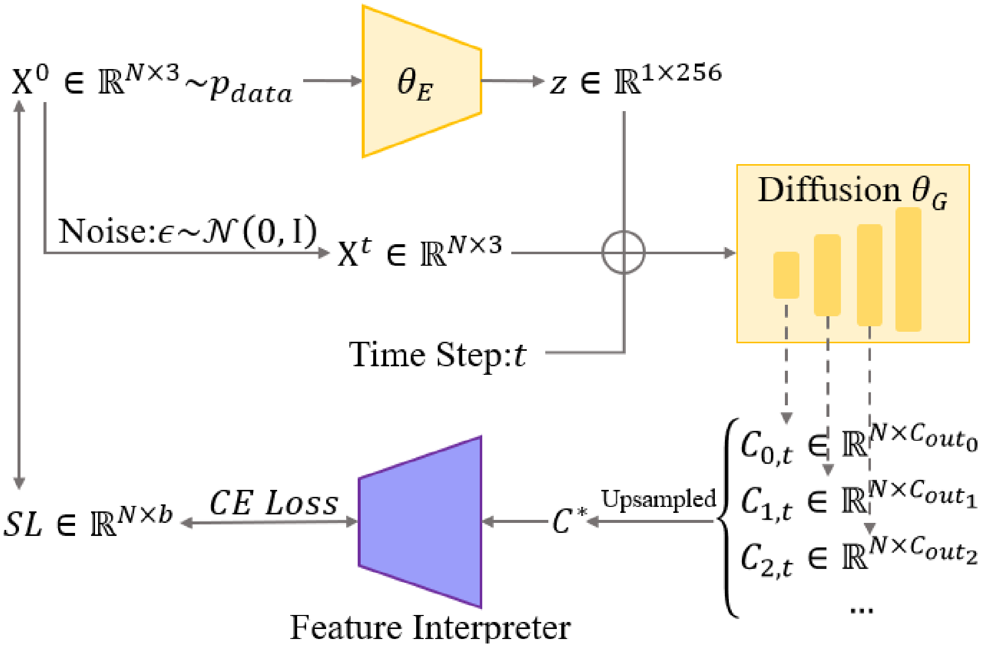

$X^0 \in \mathbb{R}^{N\times3} \sim p_{data}$
θ_E
$z \in \mathbb{R}^{1\times256}$
Diffusion θ_G
Noise:$\epsilon \sim \mathcal{N}(0, I)$
$X^t \in \mathbb{R}^{N\times3}$
Time Step:t
$C_{0,t} \in \mathbb{R}^{N\times Cout_0}$
$C_{1,t} \in \mathbb{R}^{N\times Cout_1}$
$C_{2,t} \in \mathbb{R}^{N\times Cout_2}$
...
Upsampled
C^*
$SL \in \mathbb{R}^{N\times b}$
$CE\ Loss$
Feature Interpreter

Figure 4.5: Illustration of the point-label pairs generation method.

4.3 Evaluation

This section presents the specific implementation details of the experimental setup. Firstly, the data set used in the experiment is introduced, and the evaluation metrics used in the evaluation experiment are introduced. Then it analyses the function of Feature Interpreter and how to design an experiment to evaluate the effect. Then this section conducts ablation and comparative experimental research on the experiments in this section. The experiments and analysis presented here were originally discussed in our previous publication (Li et al. 2022b).

4.3.1 Experimental setting

This section introduces the dataset that used in the experiments and implementation and training details.

Dataset Description. In the 3D point cloud object part segmentation task, training and testing are carried out on ShapeNetPart (Yi et al. 2016b). The dataset contains 16880 CAD models in 16 categories; each model is labelled with 2 to 6 parts, and there are 50 parts for all objects in total. The experiment follows the division of the standard training set/test set, and selects the model of this category for training according to whether the point cloud generation model is a category-specific model, and uses all the models in the test set for testing. Sampling 2048 points for all original 3D objects, using the coordinates xyz as the input of the network for training.

Implementation and Training Details. The model is implemented based on Pytorch and trained a single NVIDIA GeForce GTX 2080ti graphics card. For different categories of objects, the experiment chooses two structures of multi-layer perceptrons (MLP) with ReLu activation function. For objects with only three partial semantic categories, the experiment is chosen to be performed on a three-layer MLP, and the middle hidden layers are 512, 256, respectively. For objects with more than three partial semantic categories, the experiment is chosen to be performed on a five-layer MLP, and the middle hidden layers are 512, 256, 128, and 64, respectively.

The experiment uses batches with 16 samples for split training, and for the case of few-shot, the batch size is the same as the number of samples. The experiment uses the Adam optimiser with a momentum of 0.9 for optimization, the initial learning rate is set to 0.001, and the decay rate is 0.5. The experiment requires training for 500 rounds. It takes about 8 hours for the semantic segmentation task to train to converge.

The experiments include PointNet++ (Qi et al. (2017b)) as the baseline that evaluates the effectiveness of the Feature Interpreter.

4.3.2 Evaluation Metrics

Intersection over Union (IoU) is a widely recognized metric in object detection and segmentation tasks, measuring the degree of overlap between two areas. It is computed by dividing the area of overlap between the predicted bounding box (or segmentation mask) and the ground truth bounding box (or mask) by the area of their union. In semantic segmentation, unlike object detection, which typically involves regular-shaped bounding boxes, the prediction output is a segmentation mask that outlines the shape of an object or region point-wise. IoU in semantic segmentation compares predicted and ground truth masks point-by-point to evaluate label accuracy. This method evaluates IoU by measuring alignment between predicted and ground truth object outlines, ensuring segmentation correctness. The illustration of IoU is shown in Figure 4.6.

$$IoU_i = \frac{TP_i}{TP_i + FN_i + FP_i}, \tag{4.2}$$

where TP_i, FN_i, FP_i is the True Positive, False Negative and False Positive score of class i, respectively. As shown in Figure 4.6, True Positive denotes the intersecting region between the ground truth and the segmentation mask; False Positive denotes the region of the segmentation mask that is beyond the boundaries of the ground truth; False Negative denotes the section of the ground truth that the model did not manage to identify.

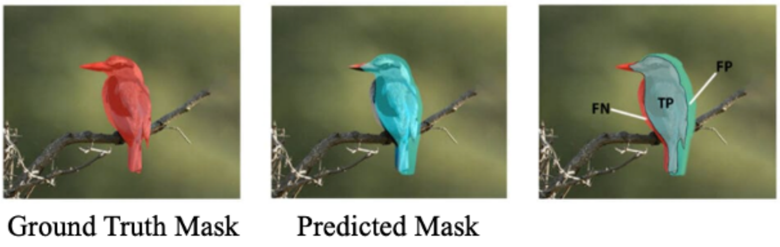


Figure 4.6: Illustration of IoU metrics for semantic segmentation (Kukil 2013).

Its role is not only to determine positive and negative samples but also to evaluate the distance between the output box (predicted box) and ground-truth. IoU not only can reflect the detection effect of the predicted detection frame and the real detection frame, but it also has a characteristic of scale invariance.

This experiment mainly uses Mean Intersection over Union (mIoU) as the evaluation metric. mIoU is the most commonly used evaluation metric in semantic segmentation and object detection.

$$mIoU = \frac{1}{K} \sum_{i=1}^{K} \frac{TP_i}{TP_i + FN_i + FP_i}, \tag{4.3}$$

where K denotes the number of classes number. Overall, the formula indicates that the intersection of the predicted value and the real value of each category is divided by the union, and then the average is taken, that is, the average of the IoU value.

4.3.3 The effectiveness of Feature Interpreter

We believe that one of the future application scenarios of our method is to generate point cloud datasets for a new category. Since the cost of point cloud annotating is too high, we can use few-shot examples of point-label pairs and generate large-scale annotated point cloud datasets. Therefore, it is important to verify whether our method can generate results with high segmentation accuracy when the example samples are scarce. We conduct few-shot segmentation to verify the effectiveness of our method. Figure 4.7 shows the visualised results of the segmentation results. Moreover, we compare the evaluation results with baseline Qi et al. (2017b) as shown in Table 4.1. Our method demonstrates comparable performance to the baseline when trained on a few samples. The comparison results demonstrate that our method is capable of generating compelling point-label pairs in a few-shot setting. In this experiment, we set the training epoch as 20, the learning rate starts from 0.001, which decayed by 0.1 every 2 epochs. This experiment incorporates material from the previous publication Li et al. (2022b).

4.3.4 Validation of Representation Effectiveness

Since the intermediate features of feature analysis and learning are extracted from a diffusion generative network with an autoencoder frame,

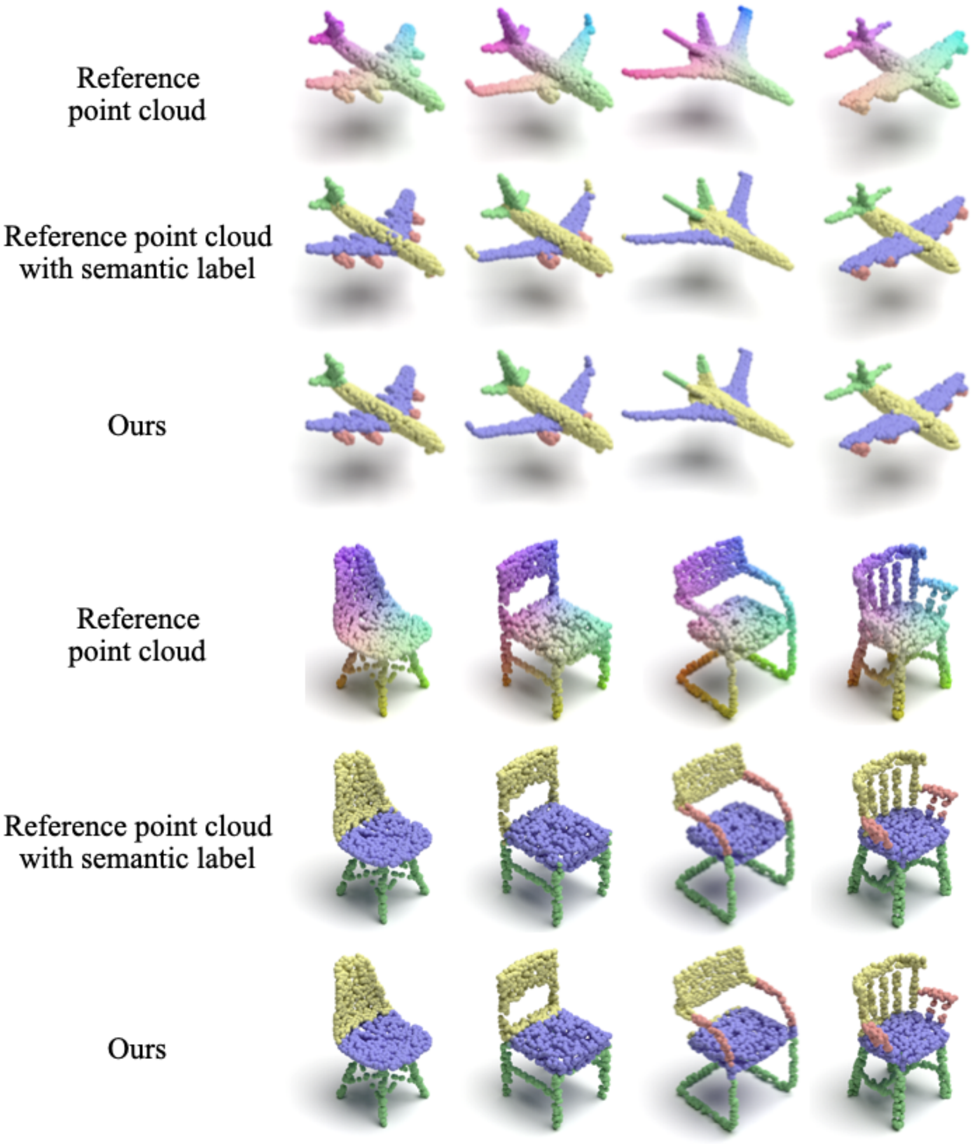

Reference point cloud
Reference point cloud with semantic label
Ours
Reference point cloud
Reference point cloud with semantic label
Ours

Figure 4.7: Visualisation examples of segmentation results.

Category	Model	k=1	k=3	k=5	k=10	k=16	k=32
Airplane	Baseline	20.9	47.2	29.4	43.3	59.6	64.6
	Ours	**58.1**	**62.8**	**63.9**	**64.8**	**66.0**	**67.2**
Chair	Baseline	33.9	63.8	50.0	64.8	**79.5**	**81.6**
	Ours	**66.2**	**67.9**	**72.1**	**74.7**	77.6	78.2

Table 4.1: Few-shot segmentation on the ground truth dataset. k are the number of samples that are used in Training.

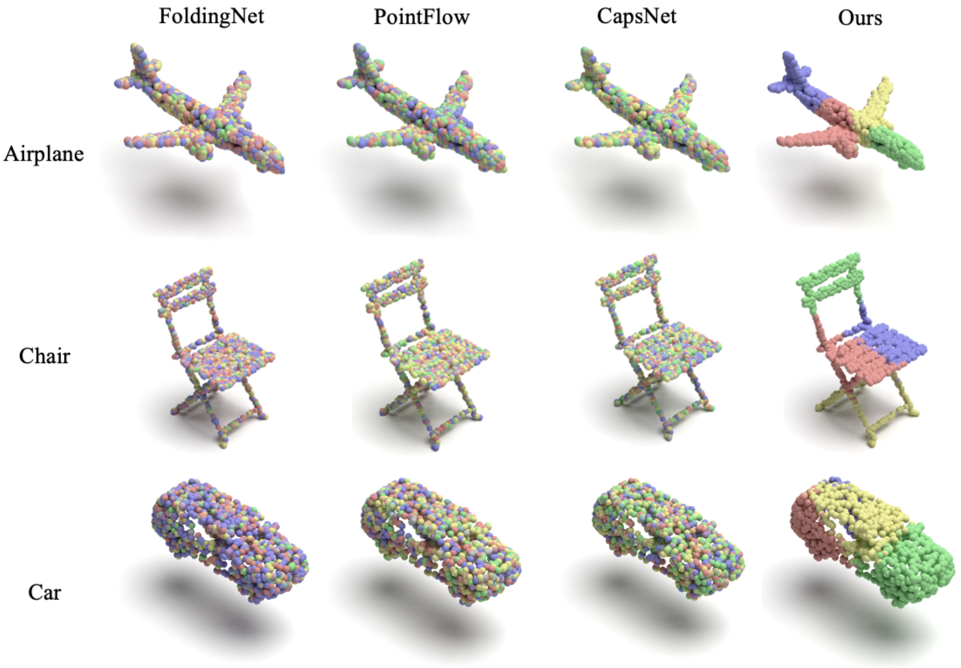


Figure 4.8: Qualitative comparison of intermediate features based on a different baseline using K-means cluster.

we naturally question whether these learnable intermediate features have nothing to do with the diffusion process but only benefit from the autoencoder framework. Therefore, we conduct an experiment to find out whether other methods capable of extracting intermediate features from point clouds can achieve the same effect. To the best of our knowledge, this is the first work to find out the discriminability of intermediate features in a point cloud generative model.

As in our method, we first collect and cluster latent feature spaces of existing generative models: CapsNetwork (Zhao et al. 2019b), PointFlow

(Yang et al. 2019), and FoldingNet (Yang et al. 2018). The cluster effect is shown in Figure 4.8.

The comparison results answer our question: the discriminability of the intermediate features benefited from the diffusion process, and not all point cloud autoencoder networks have similar discriminability. The possible explanations could be that (a) these models use CD-Loss to optimise the parameter of the network, which calculates the overall structural similarity; (b) these models train the network in a one-shot discriminative way.

Therefore, their intermediate features do not contain fine-grained information. This finding also explains why recent works that extract feature blocks to do segmentation tasks choose PointNet++ instead of PointNet. The former can recognise fine-grained local features. The reason we use PointNet as an encoder is to prove that even without a complicated or deep hierarchy, the diffusion model is still able to bring out local features. This analysis builds upon the methodology I developed in our earlier work (Li et al. 2022b).

4.3.5 Ablation Study

Intuitively, there are two deterministic factors of representation discriminability. This section has been developed from concepts and data from our publication (Li et al. 2022b). The first is that intermediate features with the highest dimension have better discriminability because they may contain the most information. The second is that we tend to consider the features of the shallow layer because the feature of the deeper layer is closer to the estimated noise of the diffusion process, while the shallow layers contain abstract information, such as semantics.

Then we compare it to the following settings: a) the features of the shallow layers are upsampled to the highest dimension; b) the features of the layer that has the highest dimension.

	$C_{(0,0)}$	Upsample $C_{(0,0)}$	$C_{(2,0)}$
Airplane	75.7	76.0	72.8

Table 4.2: Evaluation of the different intermediate feature extraction variations for part segmentation.

The results are proved in Table 4.2. The intermediate features within the highest dimension slightly underperform the features of the shallow layers. The experimental results confirmed that the intermediate features of the shallow layer have better discriminability.

4.4 Summary

To conclude, this chapter presents a simple and useful paradigm for the generation of point-label pairs. The intermediate feature analysis focuses on the investigation of the intermediate features of the point cloud generation diffusion model. By analysing these features at different time steps, this section aims to gain insights into the abstract information learned by the generative model, and how it represents the interpretable structure and semantics of point clouds. This analysis can be served as the foundation for developing the Feature Interpreter, which is essential for generating semantically interpretable semantic labels. Feature interpreter builds upon the insights gained from the Intermediate Feature Analysis. Feature Interpreter is employed to transform intermediate features of the point cloud diffusion generative model into the semantic label. By leveraging a Multi-Layer Perceptron (MLP), the Feature Interpreter realises label prediction and optimises it using cross-entropy loss. The effectiveness and efficiency of the proposed method are demonstrated via semantic label prediction without strong supervised samples. This section is summarized from our previous paper (Li et al. 2022b).

Chapter 5

Improvement for Point Cloud Dataset Generation

This chapter elaborates on the approach for improving the quality of the generated point cloud dataset. The main goal of the proposed approach is to facilitate the generated dataset that can be utilized by the downstream semantic segmentation task. To this end, drawing inspiration from the query strategies of active learning, this chapter proposed a point-label pairs filter method based on Query-By-Committee. This chapter is organised as follows:

- Section 5.1 introduces the motivation of the filter module and overview of the proposed method.
- Section 5.2 lays the theoretical groundwork for our approach by exploring the fundamentals of active learning and the specific strategy of Query-By-Committee (QBC). Then it discusses how to use it to filter low-quality point clouds and unqualified corresponding labels.
- Section 5.3 highlights the design of experimental and practical application of this method. The experimental results demonstrate the effectiveness of the proposed method.
- Section 5.4 summarizes this chapter.

5.1 Introduction

In the above chapters, this research has proposed point cloud and point-wise semantic label generation methods. The proposed method realises point cloud dataset generation with few-shot annotated point cloud samples. To ensure that the point cloud and its corresponding point-wise labels can be effectively used for downstream applications, such as semantic segmentation methods, it is important to carefully examine and verify the accuracy and precision of the generated datasets. The method proposed in this chapter addresses how to make the generated point cloud datasets useful.

On the one hand, the point cloud dataset generation pipeline in the previous chapters contains an implicit assumption that every point cloud generated by the point cloud generation model has a distinguishable 3D shape. However, this is not always the case as generative models can occasionally produce results that deviate from the expected distribution or lack a discernable shape (Zhang et al. 2021b). On the other hand, although the pipeline established above has demonstrated significant potential in reducing the cost of point cloud dataset generation, the feature interpreters trained with few-shot learning techniques are not sufficiently developed for practical use in generating point-label pairs for downstream tasks (Li et al. 2022b). Towards to address these challenges, this chapter proposes an approach to elevate the quality of the generated 3D point-label pairs, which can further be used to filter unqualified samples. The proposed solution draws on uncertainty measurement, which is a strategy widely used in active learning to select valuable train samples and enhance the performance of training deep neural networks. By integrating uncertainty measurement into the generation process, the quality of the generated point cloud dataset can be significantly improved.

The experiments show that integrating uncertainty metrics into the proposed method for generating point-label pairs can outperform previous state-of-the-art methods by more than **15%** in terms of mIoU. Additionally, it can provide substantial benefits to downstream semantic segmentation tasks. Thus, this chapter highlights the value of our proposed method in augmenting the quality and scale of the point cloud dataset, thereby paving the way for more reliable and efficient downstream applications.

5.2 Uncertainty Measurement

Section 5.2.1 elaborates on the theoretical underpinnings of uncertainty measurement strategies and provides a comprehensive understanding. Section 5.2.2 details the implementation of how to integrate it into the 3D point-label generation pipeline.

5.2.1 Query Strategy Preliminary

The approach in this section is to take inspiration from active learning. The main idea of active learning is to obtain more valuable labelled data at a lower cost and further improve the effect of the algorithm. The active learning model can be represented by the following formula:

$$A = (C, Q, S, L, U), \tag{5.1}$$

where C is a group or a classifier that predicts labels for the unlabeled samples and L is the samples that have already been labelled and are used for training the classifier; U denotes the pool of samples that have not yet been labelled; Q is a mechanism or strategy that the active learning system uses to identify and select the most informative samples from U. The active learning strategy aims to select the most informative samples from this pool for labelling; S is an expert who provides the correct labels for the samples selected by the query function from the unlabeled pool U (Goudjil et al. 2018).

Active learning operates by initiating the training process with a modestly sized set of labelled samples, L. In each iteration, the system identifies the most informative samples from the pool of unlabeled data, U, using a specifically designed query function, Q. These selected samples are then presented to a supervisor, S, who provides the necessary labels. With these new labels, the classifier, C, is updated, enhancing its performance for the subsequent iteration. Active learning is a cyclic process until a certain stopping criterion is reached (Rubens et al. 2015). For every round, the value of a possible instance is assessed via a scoring metric, and the instance achieving the top rank is subject to questioning (Nguyen et al. 2022). The query function Q is used to estimate the uncertainty score of one or a batch of samples.

Generally speaking, the use of information entropy is the main basis for measuring uncertainty. The greater the information entropy, the greater the uncertainty and the richer the amount of information contained. In fact, some uncertainty-based active learning query functions are designed using information entropy, such as Entropy query-by-bagging (Copa et al. 2010). Therefore, the uncertainty strategy is to try to find samples with high uncertainty, because the rich amount of information contained in these samples is useful for optimizing model parameters.

The main idea of the proposed approach is to filter low-quality point clouds and unqualified corresponding labels through query strategy functions, precisely committee-based queries. Query-By-Committee (QBC) is one of the most common strategies of query function (Vandoni et al. 2019). The main idea behind QBC is to maintain a committee of different models, all trained on the same labelled dataset (Yao et al. 2020). The models are used to make predictions on unlabeled data, and the examples with the greatest disagreement among committee members are considered the most informative and selected for querying. The rationale behind QBC is that if different models in the committee disagree on an example, that example is likely to be difficult or ambiguous, and obtaining its true label will provide valuable information for improving the model. In practice, QBC needs a metric to measure the inconsistency of predictions between models. Commonly used metrics include Vote Entropy (Settles 2009) (the proportion of different classes predicted by the model), Kullback–Leibler (KL) divergence (Muslea et al. 2000) and information entropy (the entropy based on the probability of the model's predictions) (Chen et al. 2017, Kuo et al. 2018), etc.

An intuitive solution for implementation is to employ a committee of pre-trained few-shot semantic segmentation networks as members of QBC. In cases like our application, where the generated point clouds may not exhibit unique shapes and no single baseline stands out as superior within the few-shot annotated setup, it could be more advantageous to duplicate feature interpreters instead of designing new ones. Moreover, a duplicate of the feature interpreter, in this case, provides the requisite information, facilitating the seamless integration of an adapted fusion into the overall process.

5.2.2 Applying Query-By-Committee in Point-Label Pairs Filter

This section builds upon the methodology I developed in the earlier work (Li et al. 2022b). The last section establishes the merits of the Query-By-Committee approach in the previous section, this section further details how this strategy can be applied to the point-label pairs filtering.

Following Zhang et al. (2021b), Gadelha et al. (2020), the Jensen-Shannon (JS) divergence (Kuo et al. 2018) is employed to compute the uncertainty measure for each point-label pair. Specifically, a committee of Feature Interpreters are trained in the same way separately. Then, the label likelihood $LS \in \mathbb{R}^{N \times b}$ for the point cloud $X \in \mathbb{R}^{N \times 3}$ can be obtained and used as the uncertainty score. Here, b represents the number of classification categories for each type of point cloud. For instance,

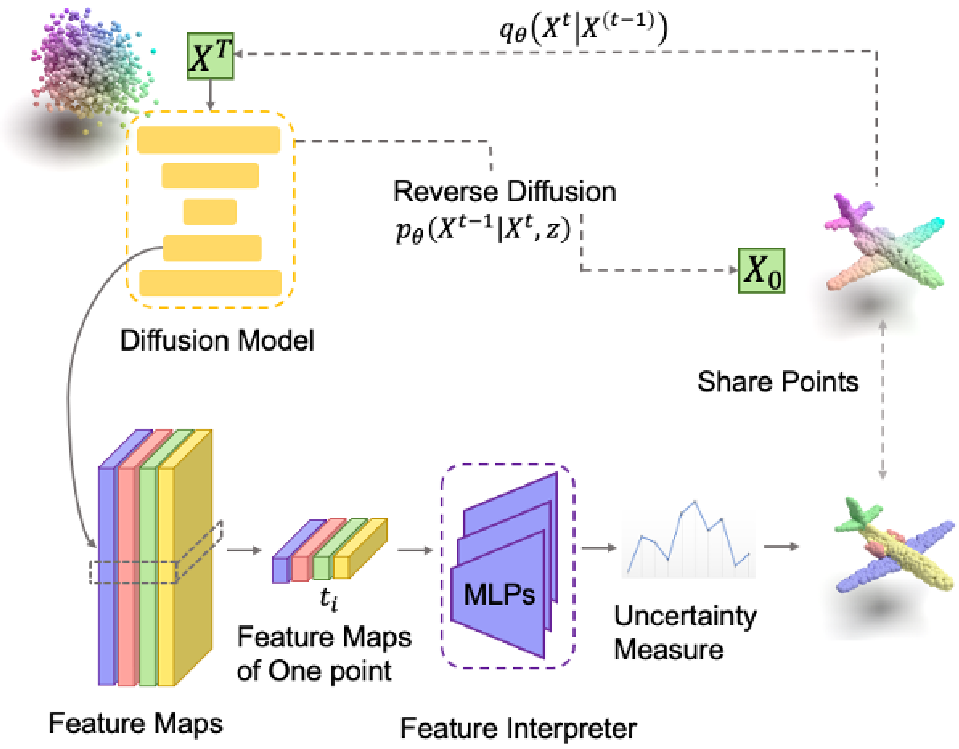


Figure 5.1: The illustration of point-label pairs generation with filter mechanism.

$b = 4$ when segmenting airplane, corresponds to the 4 semantic parts in airplane point cloud. The pipeline of point-label pairs generation with filter mechanism can be represented in Figure 5.1.

Formally, the uncertainty measurement is denoted by $\mathcal{JS} \in \mathbb{R}^N$. The computation can be formulated as:

$$\mathcal{JS} = H(\frac{1}{M}\sum_{i}^{M} LS_i) - \frac{1}{M}H(LS_i), \tag{5.2}$$

where M denotes the number of Feature Interpreters in one committee; LS_i denotes the label likelihood of the i-th Feature Interpreter for point cloud; H denotes the entropy function. We use the score of the lowest-rated 200 points of point cloud $\mathcal{JS}$ as the uncertainty score for each point cloud in the implementation. The uncertainty score can be used to filter unqualified point clouds.

5.3 Implementation Process and Experiment Evaluation

In this section, the implementation of a process of validation and assessment and the experimental results of the proposed method are further delineated. This section has been developed from concepts and design from my publication (Li et al. 2022b).

Section 5.3.1 clarifies the experimental setup, including the evaluation metrics, primarily mean Accuracy (mAcc) and mean Intersection over Union (mIoU).

To verify the validation of the generated datasets and the effectiveness of the proposed method, two experiments with different setup conditions are conducted and examined:

- The first experiment, detailed in Section 5.3.2, evaluates that the dataset generated along with the filtering mechanism is close to the real dataset and thus can be used reliably.
- The second experiment aims to verify that the proposed method can effectively enhance existing semantic segmentation tasks, especially in a few-shot scenario. Section 5.3.4 details the experimental details and analyses experimental results.

Finally, Section 5.3.3 presents a comparative analysis with GAN-based generation methods, showcasing the superior performance of the proposed approach.

5.3.1 Experimental Setup

For 3D semantic segmentation, mean class Accuracy (mAcc) and mean class Intersection over Union (mIoU) are the most frequently used metrics to measure the accuracy of segmentation methods. mAcc is the mean accuracy encompassing all classifications (OAcc), which is per-class OAcc.

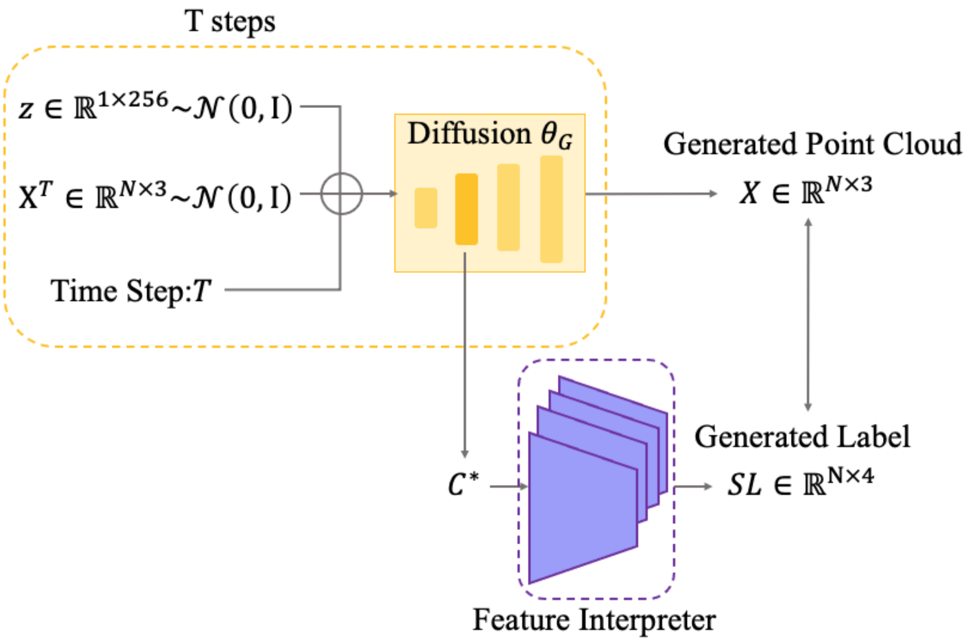

T steps
$z \in \mathbb{R}^{1\times256} \sim \mathcal{N}(0,\mathrm{I})$
$\mathrm{X}^T \in \mathbb{R}^{N\times3} \sim \mathcal{N}(0,\mathrm{I})$
Time Step:T
Diffusion θ_G
Generated Point Cloud
$X \in \mathbb{R}^{N\times3}$
C^*
Generated Label
$SL \in \mathbb{R}^{\mathrm{N}\times4}$
Feature Interpreter

Figure 5.2: Diagram of inference phase.

and then averaging over the total number of classes K. It is a comprehensive definition of the total performance across different categories. It can be formally represented as:

$$mAcc = \frac{1}{K}\sum_{i=0}^{K-1}\frac{TP_i}{TP_i + FN_i}, \tag{5.3}$$

where K denotes the total number of classes. Besides, because experiments in this section are also related to semantic segmentation, the training dataset is the same as in the last section.

5.3.2 Validation of Generated Datasets

Figure 5.2 shows the overview of the inference phase. Figure 5.3 are visualized results of the generated point-label pairs. The content in this section has been derived from my previously published paper (Li et al. 2022b).

To demonstrate the generation of point-label pairs can be used in another part segmentation network, and improve the performance of part

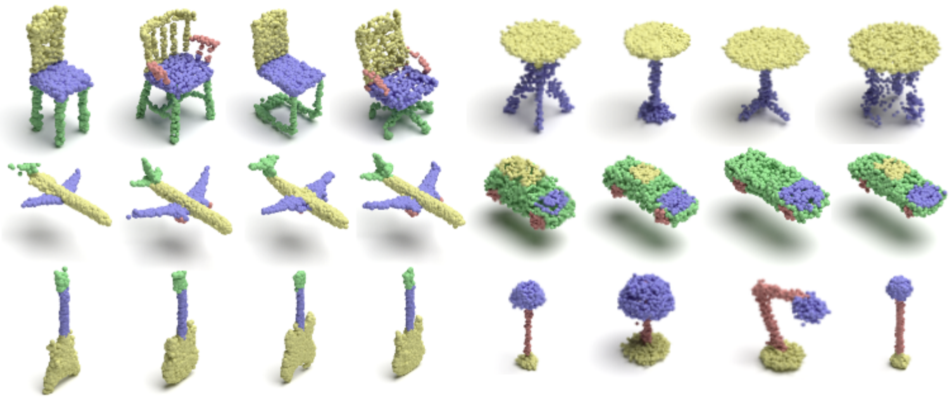


Figure 5.3: Examples of generated point-label pairs. The feature interpreter is trained with the full ground-truth dataset.

segmentation methods, this section first demonstrates the validation of the datasets generated by the feature interpreter. This experiment is conducted with two setups: one with the feature interpreter trained on a full-set dataset, and another with the feature interpreter trained on a few-shot dataset. Figure 5.4 shows some visualized examples of the generated point-label pairs produced by the few-shot trained dataset. Both experiments use the generated point-label pairs generated by the feature interpreter that trained with full-set or few-shot dataset and filtered by the proposed method. The network (Qi et al. (2017b)) is trained for semantic segmentation using a training set of ground truth data. After training, this network is used to validate the generated point-label pairs and the ground truth test set, respectively. When producing the generated dataset, this approach first generated 10,000 point clouds for each category and filtered samples based on their uncertainty scores. Figure 5.5 shows the quantitative comparison of the generated dataset produced by our method.

The generated dataset shows competitive results for most categories compared to the GT dataset. The performance of our generated dataset is much lower than the GT dataset in the category of Lamp. However, the visualized results of the Lamps are plausible. We attribute this re-

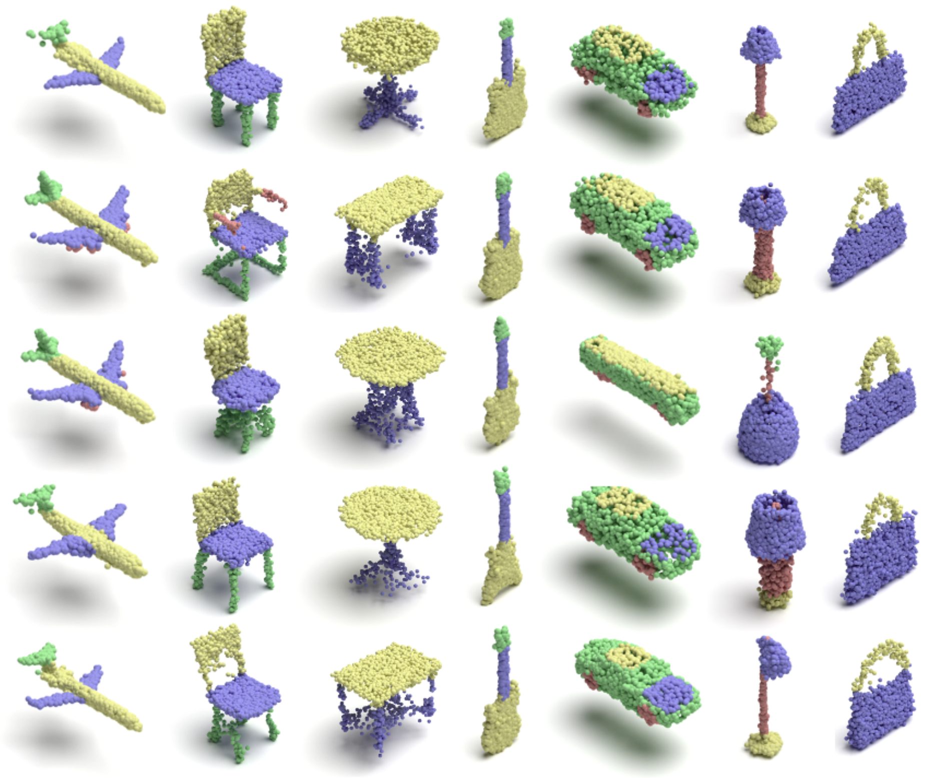


Figure 5.4: Examples of generated point-label pairs (from left to right are airplane, chair, table, guitar, car, lamp, bag). The feature interpreter is trained with the few-shot sample of the ground-truth dataset.

sult to the fact that because the Lamps in the GT dataset are few, the segmentation network has not fully learned the accurate features of the Lamps.

5.3.3 Comparison with GAN-based Method

The work most closely related to the proposed point-label generation method is CPCGAN (Yang et al. (2021)). CPCGAN proposed a two-stage GAN to generate point clouds in a controllable structure manner and is trained on the ShapeNet-Partseg dataset as well. The first stage generator generates the key structural points and corresponding labels. The second stage generator generates the point cloud by expanding the key structural points into a complete point cloud. The semantic labels of the final point cloud are bred from the key structural points of the first

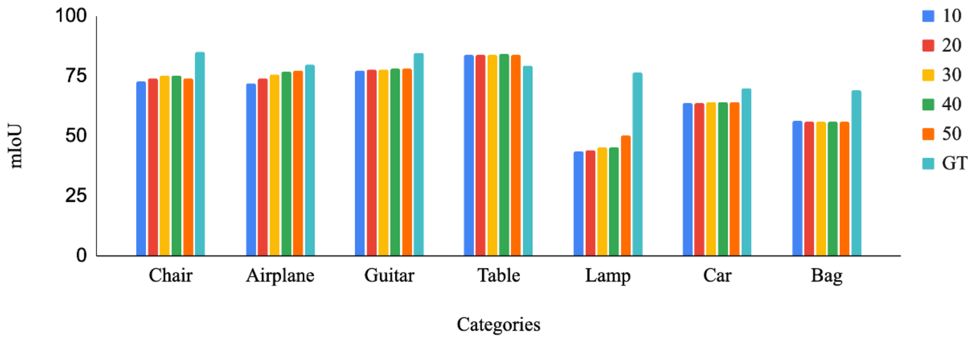


Figure 5.5: Comparison between the generated datasets and ground-truth ones. Different colours denote different filter ratios.

Class	Model	mIoU%(↑)	mAcc%(↑)
Chair	CPCGAN	57.1	83.6
	Ours	**72.0**	**86.3**
Airplane	CPCGAN	67.8	82.6
	Ours	**74.2**	**89.2**

Table 5.1: Comparison of point cloud and label generation performance.

stage. Because it is hard to annotate ground truth labels for generated point clouds, following their experimental setting, a PointNet++ model Qi et al. (2017b) is trained for the semantic segmentation task. Same with the last experimental setup, this pre-trained segmentation network is employed to evaluate the generated point-label pairs. The quantitative comparison is shown in Table 5.1. From the results shown in the table, we can see that our generated point cloud (airplane and chair) outperforms their method consistently on the two evaluation metrics for both mIoU and mAcc by a large margin.

Moreover, the visualized comparison results are shown in Figure 5.6. During the visualised experiment, some point-label pairs generated by both methods are randomly picked and the semantic labels are by colour, using the same colour for the same label across models. From these results, we can see that the point clouds with semantic labels generated by our method exhibit more accurate labels, whereas Shu et al. (2019a)

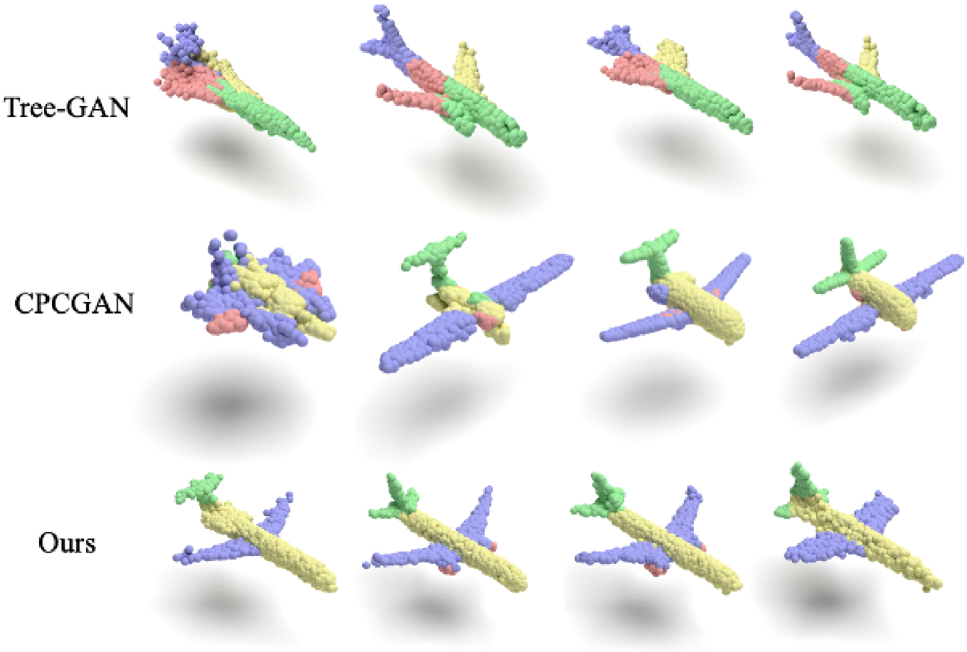


Figure 5.6: Visualized results of Shu et al. (2019b), Yang et al. (2021) and ours.

and Yang et al. (2021) tend to generate noisy semantic labels. This section is adapted from our published paper (Li et al. 2022b).

5.3.4 Application in a Few-shot Scenario

We further conduct an experiment to simulate a few-shot sample scenario. The following section extends the discussion from my published work Li et al. (2022b). First, we sample a small set of GT point-label pairs to train on our method. Then we concatenate the GT samples with the generated point-label pairs as an augmented train set. Then we use the small set of GT point-label pairs and the augment train set to train a segmentation network, respectively. The segmentation results of the mIoU metric are shown in Table 5.2. In this experiment, We generated 2000 point-label pairs and set the filter as 30%. We used a PointNet (Qi et al. 2017a) as the segmentation network.

The segmentation results indicate that the augmented dataset can drastically improve the performance of the segmentation network. Moreover, it is worth noting that the improvement is more significant when

Samples	128		256	
Category	w/o	w/	w/o	w/
Chair	60.87	79.36	74.54	80.59
Airplane	36.71	69.74	41.91	64.80
Guitar	49.98	83.90	69.01	86.82
Table	51.50	72.74	57.95	72.76
Lamp	44.35	65.03	61.15	67.71
Car	22.94	44.90	24.15	47.81

Table 5.2: Comparative Analysis of mIoU Scores for Segmentation Tasks Using a Small Set of Ground Truth (GT) Samples and an Augmented Dataset. 'Samples' indicates the total number of the point cloud in the GT set. 'w/' denotes training results when including the augmented dataset; 'w/o' reflects training exclusively on the GT set.

the sample scale is small, particularly for the Chair and Car. Experimental results suggest that our work can provide a practical solution for annotated point cloud generation.

5.4 Summary

This chapter elaborates on a filter mechanism for generated point cloud point-label pairs. It end-to-end filters unqualified point-label pairs by estimating the uncertainty score via a committee of feature interpreters. These scores are then used to filter out samples deemed unqualified, aiming to improve the dataset quality without extensive manual intervention. Experimental results demonstrate the effectiveness of this filtering mechanism in enhancing the usability of generated datasets and contributing to the reliability and efficiency of point cloud dataset generation. Further improvements are needed, including enhancing the granularity of segmentation for a border application.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The demand for more advanced methods to generate and annotate point cloud data is increasing due to the growth of 3D computer vision, especially when powered by deep neural networks. This thesis developed a series of approaches to solve the challenges of the generation and annotation of point cloud data. This work takes a step towards a more efficient and effective approach in the domain of point cloud dataset generation by addressing the time-consuming and labour-intensive nature of manual annotation, as well as overcoming the limitations of existing automated annotation techniques. Specifically, this research reduces the manual effort involved in the point cloud annotation process. Traditionally, this process has been known for its time-consuming and labour-intensive demands. Additionally, it overcomes the limitations of current automated annotation techniques, which include inadequate accuracy, poor generation quality, and difficulty in capturing the geometrical and semantic intertwined relationship of 3D objects.

To achieve this target, this thesis first proposed a point cloud generation method. Then a point-label pairs generation approach is built upon the point cloud generation model, and an evaluation approach is designed which can integrate with the entire process. The summary of this thesis is as follows:

In this thesis, the point cloud synthesised based on the stochastic differential equations (SDE) technique is proposed to improve the performance of point cloud generation, which is particularly suitable for large-scale applications. This method proposed to model point cloud generation as the transition between noise with prior distribution and a distinguished shape, which can be simulated by a stochastic process. The training objective can be solved by stochastic differential equations. Specifically, the solution for combining point cloud generation with differential stochastic equations is investigated, and a deep neural network framework is developed for the point cloud generative network. Compared to previous methods, which are based on the VAE or GAN training paradigm, point cloud generation based on stochastic differential equations provides an efficient and straightforward definition of point cloud generation. Then, this frame is coupled with the time encoding and Markov chain Monte Carlo to carry out a flexible sampling approach. As a result, the training scheme of the point cloud generative model is stable, and the generation quality of the point cloud can be improved. This thesis implements a series of experiments, such as point cloud generation, point cloud reconstruction, and point cloud completion, to demonstrate the advantages of the proposed point cloud diffusion-based generation method.

Then, based on the proposed point cloud diffusion generative method, a point-label pairs generation method is proposed to reduce the cost of large-scale point cloud annotation. First of all, the characteristics of the point cloud diffusion generative model are investigated, and an assumption is formed. The intermediate feature of the point cloud diffusion-based generative model has discriminability and is explicable at the semantic level, thereby being used to help annotate the point cloud. To verify the assumption, a feature interpreter is employed to transform the intermediate feature into a point-wise semantic label. The feature interpreter is a parallel branch of the point cloud generation model; therefore, the generation of end-to-end point-label pairs is viable. As a result, the proposed approach resolves the point-label pairs generation and preserves the quality of the point cloud yielded by the point cloud generation model. To verify the effectiveness of the proposed approach, the experiment implements semantic segmentation in a few-shot setup. In this experiment, the feature interpreter based on the point cloud diffusion-based model can effectively generate point-label pairs. Moreover, the experiments affirm that the discriminability of the intermediate representation of point cloud generation does not naturally exist, whereas the point cloud diffusion-based generation model is essential to achieve the proposed approach.

To further enhance the quality of generated point-label pairs and ensure their effective application in segmentation tasks, this thesis introduced a filtering approach for point-label pair generation. Inspired by the query strategy of active learning and its uncertainty measurement, the filtering approach utilises an uncertainty score, estimated by a committee of feature interpreters, to filter generated point-label pairs. This approach seamlessly integrates with the point-label pairs generation pipeline without impacting the quality of the generated point cloud. To verify the validation of the generated dataset and the usefulness of the proposed method, experiments with two different settings are conducted. In the first experiment, a segmentation network was trained on the training set of the ground truth (GT) dataset and subsequently tested on both the generated dataset and the test set of the real dataset. The results of this experiment confirmed the close resemblance of our dataset to the real one, indicating its reliability. In the second experiment, a scenario with only a few-shot labels was simulated. A few-shot dataset was used to generate a dataset, which was then utilised as an enhanced set for training a segmentation network. For comparison, a segmentation network was trained independently on the few-shot datasets, and the results were verified using the real test set. The results from both settings affirm that our method is feasible and that the generated datasets are effective. Additionally, a comparison experiment was conducted against the GAN-based point-label generation method. The results quantitatively demonstrate the superior performance of the proposed method.

6.2 Limitations and Future Work

Despite this thesis having made contributions in advancing the point cloud dataset generation based on the point cloud diffusion generative model, it is important to note that this field is rapidly evolving, and there are numerous promising technologies and avenues worthy of further research. The work in this thesis opens up the following potential pathways for future exploration:

Conditional Generation and Multi-modal Generation. While the proposed point cloud generative model has successfully applied diffusion-based 3D generative models for generating and annotating point cloud data, it has primarily focused on exploiting point cloud data alone. Understanding and transferring the intention of the user to create is also important for 3D content generation. Incorporating multimodal fusion could potentially enhance the performance of the model, which presents a promising future direction. 3D conditional generative model-based multiple modalities have been developed over the past year. Following this trend, AutoSDF (Mittal et al. 2022), DreamFusion (Poole et al. 2022), and GET3D (Gao et al. 2022) have begun exploring 3D conditional generative models based on multi-modal inputs. These additional modalities could potentially provide more context and enhance the richness and versatility of the output of the 3D generative model, thus increasing its usefulness for various downstream tasks. Their method stays at the phase of how to align conditional input and the generated 3D shape, so the accuracy of the generated shape is limited. Besides, these methods have not explored how to use the pre-trained model to assist other 3D tasks.

Contextual Understanding. The current methodology could benefit from integrating contextual understanding by utilizing auxiliary data types fusion. The generation ability and representation of the model could be limited when subtle context-specific details are required, particularly in articulated shapes, such as recognizing the relationships and structures between different parts of the shape. Contextual understanding can play a crucial role in generating more accurate and nuanced 3D objects and their annotations.

Fine-grained understanding of 3D point cloud objects. In certain applications, more detailed comprehension of object components in 3D point clouds is necessary. For example, when a robot navigates through an indoor environment, it must be able to precisely identify which shelf of a cabinet to grasp. To achieve this level of precision, it is necessary to perform semantic segmentation of 3D point cloud object parts, followed by instance segmentation at a finer semantic level. In follow-up research, the segmentation task of 3D object refinement still needs more research, including instance-level automatic annotation. In addition, because point cloud data is sparse and unstructured, performing instance segmentation with few samples directly on the point cloud is extremely challenging. One possible solution is to tackle the problem of fine-grained understanding of 3D point cloud objects from the perspective of point cloud generation models.

Transfer learning in point cloud understanding. The research presented in this thesis primarily emphasises the use of point cloud generative models' intermediate features for semantic label transformation in few-shot scenarios. Yet, the challenges in obtaining point cloud datasets extend beyond the high costs of acquiring labelled data. In addition to this, although the proposed methodology is capable of securing a point cloud generative model with substantial expressive power, the model's training robustness against noise remains a concern. Therefore, the transferability of the proposed model to open-world scenarios requires further exploration. Moreover, enhancing the model's capacity to generate diverse, new shapes in open-world contexts with few training examples could significantly mitigate the demand for task-specific labelled data. This represents another potential future research direction that this thesis will explore. Thus, it is expected that developing the transfer learning ability will further improve the performance of the point cloud dataset generation.

Weakly supervised point cloud end-to-end annotation. The proposed method focuses on the challenge of how to transform the intermediate feature of the point cloud generative model to the semantic label in a few-shot. However, it remains a question of how to handle scenarios where only partial or noisy annotations are available or if the annotation clues are indirect cues. For example, a cue might indicate that the point cloud represents a "four-legged object" or a "curved structure" without specifying the exact type of object or structure. This approach could be useful in situations where obtaining detailed annotations for every point in the point cloud would be challenging or time-consuming.

Bibliography

Achlioptas, P., Diamanti, O., Mitliagkas, I. and Guibas, L., 2018. Learning representations and generative models for 3d point clouds. *International conference on machine learning*, PMLR, 40–49.

Aksoy, E. E., Baci, S. and Cavdar, S., 2020. Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving. *2020 IEEE intelligent vehicles symposium (IV)*, IEEE, 926–932.

Aldoma, A., Vincze, M., Blodow, N., Gossow, D., Gedikli, S., Rusu, R. B. and Bradski, G., 2011. Cad-model recognition and 6dof pose estimation using 3d cues. *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, IEEE, 585–592.

Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M. and Maierhofer, S., 2013. O-snap: Optimization-based snapping for modeling architecture. *ACM Transactions on Graphics (TOG)*, 32 (1), 1–15.

Arshad, M. S. and Beksi, W. J., 2020. A progressive conditional generative adversarial network for generating dense and colored 3d point clouds. *2020 International Conference on 3D Vision (3DV)*, IEEE, 712–722.

Au, O. K.-C., Zheng, Y., Chen, M., Xu, P. and Tai, C.-L., 2011. Mesh segmentation with concavity-aware fields. *IEEE Transactions on Visualization and Computer Graphics*, 18 (7), 1125–1134.

Aubry, M., Schlickewei, U. and Cremers, D., 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. *2011 IEEE* *international conference on computer vision workshops (ICCV workshops)*, IEEE, 1626–1633.

Bansal, A., Russell, B. and Gupta, A., 2016. Marr revisited: 2d-3d alignment via surface normal prediction. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5965–5974.

Bello, S. A., Yu, S., Wang, C., Adam, J. M. and Li, J., 2020. Deep learning on 3d point clouds. *Remote Sensing*, 12 (11), 1729.

Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Guennebaud, G., Levine, J. A., Sharf, A. and Silva, C. T., 2017. A survey of surface reconstruction from point clouds. *Computer graphics forum*, Wiley Online Library, volume 36, 301–329.

Blanz, V. and Vetter, T., 2023. A morphable model for the synthesis of 3d faces. *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 157–164.

Bloomenthal, J. and Wyvill, B., 1990. Interactive techniques for implicit modeling. *ACM Siggraph Computer Graphics*, 24 (2), 109–116.

Bronstein, M. M. and Kokkinos, I., 2010a. Scale-invariant heat kernel signatures for non-rigid shape recognition. *2010 IEEE computer society conference on computer vision and pattern recognition*, IEEE, 1704–1711.

Bronstein, M. M. and Kokkinos, I., 2010b. Scale-invariant heat kernel signatures for non-rigid shape recognition. *2010 IEEE computer society conference on computer vision and pattern recognition*, IEEE, 1704–1711.

Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N. and Hariharan, B., 2020. Learning gradient fields for shape generation. *European Conference on Computer Vision*, Springer, 364–381.

Caliński, T. and Harabasz, J., 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3 (1), 1–27.

Carlbom, I. and Paciorek, J., 1978. Planar geometric projections and viewing transformations. *ACM Computing Surveys (CSUR)*, 10 (4), 465–502.

Carlson, W. E., 1982. An algorithm and data structure for 3d object synthesis using surface patch intersections. *Proceedings of the 9th annual conference on Computer graphics and interactive techniques*, 255–263.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H. et al., 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chaudhuri, S., Kalogerakis, E., Guibas, L. and Koltun, V., 2011. Probabilistic reasoning for assembly-based 3d modeling. *ACM SIGGRAPH 2011 papers*, 1–10.

Chen, X., Golovinskiy, A. and Funkhouser, T., 2009. A benchmark for 3d mesh segmentation. *Acm transactions on graphics (tog)*, 28 (3), 1–12.

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I. and Abbeel, P., 2016. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*.

Chen, Y., Lask, T. A., Mei, Q., Chen, Q., Moon, S., Wang, J., Nguyen, K., Dawodu, T., Cohen, T., Denny, J. C. et al., 2017. An active learning-enabled annotation system for clinical named entity recognition. *BMC medical informatics and decision making*, 17 (2), 35–44.

Chen, Z., Jing, L., Liang, Y., Tian, Y. and Li, B., 2021. Multimodal semi-supervised learning for 3d objects. *arXiv preprint arXiv:2110.11601*.

Chen, Z., Jing, L., Yang, L., Li, Y. and Li, B., 2023. Class-level confidence based 3d semi-supervised learning. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 633–642.

Cheng, A.-C., Li, X., Liu, S., Sun, M. and Yang, M.-H., 2022. Autoregressive 3d shape generation via canonical mapping. *arXiv preprint arXiv:2204.01955*.

Cheng, M., Hui, L., Xie, J. and Yang, J., 2021. Sspc-net: Semi-supervised semantic 3d point cloud segmentation network. *arXiv preprint arXiv:2104.07861*.

Chua, C. S. and Jarvis, R., 1997. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25 (1), 63.

Copa, L., Tuia, D., Volpi, M. and Kanevski, M., 2010. Unbiased query-by-bagging active learning for vhr image classification. *Image and Signal Processing for Remote Sensing XVI*, SPIE, volume 7830, 176–183.

Deng, S., Xu, X., Wu, C., Chen, K. and Jia, K., 2021. 3d affordancenet: A benchmark for visual object affordance understanding. *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1778–1787.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dinh, L., Sohl-Dickstein, J. and Bengio, S., 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Doob, J. L., 1942. What is a stochastic process? *The American Mathematical Monthly*, 49 (10), 648–653.

Duan, Y., Zheng, Y., Lu, J., Zhou, J. and Tian, Q., 2019. Structural relational reasoning of point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 949–958.

Esteves, C., Allen-Blanchette, C., Makadia, A. and Daniilidis, K., 2018. Learning so (3) equivariant representations with spherical cnns. *Proceedings of the European Conference on Computer Vision (ECCV)*, 52–68.

Fang, Y., Xie, J., Dai, G., Wang, M., Zhu, F., Xu, T. and Wong, E., 2015. 3d deep shape descriptor. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2319–2328.

Feng, M., Zhang, L., Lin, X., Gilani, S. Z. and Mian, A., 2020. Point attention network for semantic segmentation of 3d point clouds. *Pattern Recognition*, 107, 107446.

Gadelha, M., RoyChowdhury, A., Sharma, G., Kalogerakis, E., Cao, L., Learned-Miller, E., Wang, R. and Maji, S., 2020. Label-efficient learning on point clouds using approximate convex decompositions. *European Conference on Computer Vision*, Springer, 473–491.

Gadelha, M., Wang, R. and Maji, S., 2018. Multiresolution tree networks for 3d point cloud processing. *Proceedings of the European Conference on Computer Vision (ECCV)*, 103–118.

Gal, R., Bermano, A., Zhang, H. and Cohen-Or, D., 2020. Mrgan: Multi-rooted 3d shape generation with unsupervised part disentanglement. *arXiv preprint arXiv:2007.12944*.

Gal, R., Bermano, A., Zhang, H. and Cohen-Or, D., 2021. Mrgan: Multi-rooted 3d shape representation learning with unsupervised part disentanglement. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2039–2048.

Gal, R. and Cohen-Or, D., 2006. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics (TOG)*, 25 (1), 130–150.

Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z. and Fidler, S., 2022. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35, 31841–31854.

Girdhar, R., Fouhey, D. F., Rodriguez, M. and Gupta, A., 2016. Learning a predictable and generative vector representation for objects. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, Springer, 484–499.

Gkanatsios, N., Singh, M., Fang, Z., Tulsiani, S. and Fragkiadaki, K., 2023. Analogy-forming transformers for few-shot 3d parsing. *arXiv preprint arXiv:2304.14382*.

Golovinskiy, A. and Funkhouser, T., 2009. Consistent segmentation of 3d models. *Computers & Graphics*, 33 (3), 262–269.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2020. Generative adversarial networks. *Communications of the ACM*, 63 (11), 139–144.

Goudjil, M., Koudil, M., Bedda, M. and Ghoggali, N., 2018. A novel active learning method using svm for text classification. *International Journal of Automation and Computing*, 15, 290–298.

Graham, B., Engelcke, M. and Van Der Maaten, L., 2018. 3d semantic segmentation with submanifold sparse convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9224–9232.

Groueix, T., Fisher, M., Kim, V. G., Russell, B. C. and Aubry, M., 2018. A papier-mâché approach to learning 3d surface generation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 216–224.

Guo, K., Zou, D. and Chen, X., 2015. 3d mesh labeling via deep convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 35 (1), 1–12.

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R. and Hu, S.-M., 2021. Pct: Point cloud transformer. *Computational Visual Media*, 7 (2), 187–199.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L. and Bennamoun, M., 2020. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43 (12), 4338–4364.

Hassani, K. and Haley, M., 2019. Unsupervised multi-task feature learning on point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8160–8171.

Hinton, G. E., 2007. To recognize shapes, first learn to generate images. *Progress in brain research*, 165, 535–547.

Ho, J., Jain, A. and Abbeel, P., 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33, 6840–6851.

Hong, H., Pavez, E., Ortega, A., Watanabe, R. and Nonaka, K., 2022. Motion estimation and filtered prediction for dynamic point cloud attribute compression. *2022 Picture Coding Symposium (PCS)*, IEEE, 139–143.

Hou, J., Graham, B., Nießner, M. and Xie, S., 2021. Exploring data-efficient 3d scene understanding with contrastive scene contexts. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15587–15597.

Hu, Q., Yang, B., Fang, G., Guo, Y., Leonardis, A., Trigoni, N. and Markham, A., 2022. Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds. *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, Springer, 600–619.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N. and Markham, A., 2020. Randla-net: Efficient semantic segmentation of large-scale point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11108–11117.

Hu, R., Fan, L. and Liu, L., 2012. Co-segmentation of 3d shapes via subspace clustering. *Computer graphics forum*, Wiley Online Library, volume 31, 1703–1713.

Huang, H., Kalogerakis, E. and Marlin, B., 2015. Analysis and synthesis of 3d shape families via deep-learned generative models of surfaces. *Computer Graphics Forum*, Wiley Online Library, volume 34, 25–38.

Huang, J. and You, S., 2016. Point cloud labeling using 3d convolutional neural network. *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, 2670–2675.

Huang, Q., Koltun, V. and Guibas, L., 2011. Joint shape segmentation with linear programming. *Proceedings of the 2011 SIGGRAPH Asia Conference*, 1–12.

Hui, L., Xu, R., Xie, J., Qian, J. and Yang, J., 2020. Progressive point cloud deconvolution generation network. *European Conference on Computer Vision*, Springer, 397–413.

Itô, K., 1973. Stochastic integration. *Vector and Operator Valued Measures and Applications*, Elsevier, 141–148.

Itō, K., 2006. *Essentials of stochastic processes*, volume 231. American Mathematical Soc.

Jagannathan, A. and Miller, E. L., 2007. Three-dimensional surface mesh segmentation using curvedness-based region growing approach. *IEEE Transactions on pattern analysis and machine intelligence*, 29 (12), 2195–2204.

Jiang, L., Shi, S., Tian, Z., Lai, X., Liu, S., Fu, C.-W. and Jia, J., 2021. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6423–6432.

Johnson, A. E. and Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21 (5), 433–449.

Jones, R. K., Habib, A. and Ritchie, D., 2022. Shred: 3d shape region decomposition with learned local operations. *ACM Transactions on Graphics (TOG)*, 41 (6), 1–11.

Kalogerakis, E., Averkiou, M., Maji, S. and Chaudhuri, S., 2017. 3d shape segmentation with projective convolutional networks. *proceedings of the IEEE conference on computer vision and pattern recognition*, 3779–3788.

Kalogerakis, E., Chaudhuri, S., Koller, D. and Koltun, V., 2012. A probabilistic model for component-based shape synthesis. *Acm Transactions on Graphics (TOG)*, 31 (4), 1–11.

Kalogerakis, E., Hertzmann, A. and Singh, K., 2010. Learning 3d mesh segmentation and labeling. *ACM SIGGRAPH 2010 papers*, 1–12.

Kanazawa, A., Tulsiani, S., Efros, A. A. and Malik, J., 2018. Learning category-specific mesh reconstruction from image collections. *Proceedings of the European Conference on Computer Vision (ECCV)*, 371–386.

Kar, A., Tulsiani, S., Carreira, J. and Malik, J., 2015. Category-specific object reconstruction from a single image. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1966–1974.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. and Aila, T., 2020. Analyzing and improving the image quality of stylegan. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8110–8119.

Kawana, Y., Mukuta, Y. and Harada, T., 2021. Unsupervised pose-aware part decomposition for 3d articulated objects. *arXiv preprint arXiv:2110.04411*.

Khoshnevisan, D., 2002. *Multiparameter Processes: an introduction to random fields*. Springer Science & Business Media.

Kim, C., Zimmer, H., Pritch, Y., Sorkine-Hornung, A. and Gross, M. H., 2013. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32 (4), 73–1.

Kim, H., Lee, H., Kang, W. H., Lee, J. Y. and Kim, N. S., 2020. Softflow: Probabilistic framework for normalizing flow on manifolds. *Advances in Neural Information Processing Systems*, 33, 16388–16397.

Kim, J., Hua, B.-S., Nguyen, T. and Yeung, S.-K., 2023. Pointinverter: Point cloud reconstruction and editing via a generative model with shape priors. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 592–601.

Kim, J., Yoo, J., Lee, J. and Hong, S., 2021. Setvae: Learning hierarchical composition for generative modeling of set-structured data. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15059–15068.

Klokov, R., Boyer, E. and Verbeek, J., 2020. Discrete point flow networks for efficient point cloud generation. *European Conference on Computer Vision*, Springer, 694–710.

Kol, W.-J., Chiu, C.-Y., Kuo, Y.-L. and Chiu, W.-C., 2022. Rpg: Learning recursive point cloud generation. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 544–551.

Kramer, M. A., 1991. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37 (2), 233–243.

Krylov, N. and Vladimirovich, 2002. *Introduction to the theory of random processes*, volume 43. American Mathematical Soc.

Kukil, K., 2013. Intersection over union (iou) in object detection & segmentation. https://learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/.

Kuo, W., Häne, C., Yuh, E., Mukherjee, P. and Malik, J., 2018. Cost-sensitive active learning for intracranial hemorrhage detection. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 715–723.

Laine, S. and Karras, T., 2010. Efficient sparse voxel octrees–analysis, extensions, and implementation. *NVIDIA Corporation*, 2 (6).

Lapidoth, A., 2017. *A foundation in digital communication*. Cambridge University Press.

Lawin, F. J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F. S. and Felsberg, M., 2017. Deep projective 3d semantic segmentation. *Computer Analysis of Images and Patterns: 17th International Conference, CAIP 2017, Ystad, Sweden, August 22-24, 2017, Proceedings, Part I 17*, Springer, 95–107.

Lazar, A. A. and Pnevmatikakis, E. A., 2011. Video time encoding machines. *IEEE Transactions on Neural Networks*, 22 (3), 461–473.

Le, T. and Duan, Y., 2018. Pointgrid: A deep network for 3d shape understanding. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9204–9214.

Lee, J., Im, W., Lee, S. and Yoon, S.-E., 2023. Diffusion probabilistic models for scene-scale 3d categorical data. *arXiv preprint arXiv:2301.00527*.

Li, C.-L., Zaheer, M., Zhang, Y., Poczos, B. and Salakhutdinov, R., 2018a. Point cloud gan. *arXiv preprint arXiv:1810.05795*.

Li, D., Yang, J., Kreis, K., Torralba, A. and Fidler, S., 2021a. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8300–8311.

Li, H., Sun, Z., Wu, Y. and Song, Y., 2021b. Semi-supervised point cloud segmentation using self-training with label confidence prediction. *Neurocomputing*, 437, 227–237.

Li, R., Li, X., Hui, K.-H. and Fu, C.-W., 2021c. Sp-gan: Sphere-guided 3d shape generation and manipulation. *ACM Transactions on Graphics (TOG)*, 40 (4), 1–12.

Li, S., Liu, M. and Walder, C., 2022a. Editvae: Unsupervised parts-aware controllable 3d point cloud shape generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 1386–1394.

Li, T., Fu, Y., Han, X., Liang, H., Zhang, J. J. and Chang, J., 2022b. Diffusionpointlabel: Annotated point cloud generation with diffusion model. *Computer Graphics Forum*, Wiley Online Library, volume 41, 131–139.

Li, T., Wang, M., Liu, X., Liang, H., Chang, J. and Zhang, J. J., 2023. Point cloud synthesis with stochastic differential equations. *Computer Animation and Virtual Worlds*, 34 (5), e2140.

Li, X., Ding, H., Tong, Z., Wu, Y. and Chee, Y. M., 2022c. Primitive3d: 3d object dataset synthesis from randomly assembled primitives. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15947–15957.

Li, Y. and Baciu, G., 2020. Sapcgan: Self-attention based generative adversarial network for point clouds. *2020 IEEE 19th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, IEEE, 52–59.

Li, Y. and Baciu, G., 2021a. Hsgan: Hierarchical graph learning for point cloud generation. *IEEE Transactions on Image Processing*, 30, 4540–4554.

Li, Y. and Baciu, G., 2021b. Sg-gan: adversarial self-attention gcn for point cloud topological parts generation. *IEEE Transactions on Visualization and Computer Graphics*, 28 (10), 3499–3512.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X. and Chen, B., 2018b. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31.

Lin, H., Xiao, Z., Tan, Y., Chao, H. and Ding, S., 2019. Justlookup: One millisecond deep feature extraction for point clouds by lookup tables. *2019 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 326–331.

Ling, H. and Jacobs, D. W., 2007. Shape classification using the inner-distance. *IEEE transactions on pattern analysis and machine intelligence*, 29 (2), 286–299.

Liu, J., Ni, B., Li, C., Yang, J. and Tian, Q., 2019a. Dynamic points agglomeration for hierarchical point sets learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7546–7555.

Liu, M., Zhou, Y., Qi, C. R., Gong, B., Su, H. and Anguelov, D., 2022. Less: Label-efficient semantic segmentation for lidar point clouds. *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, Springer, 70–89.

Liu, Y., Fan, B., Xiang, S. and Pan, C., 2019b. Relation-shape convolutional neural network for point cloud analysis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8895–8904.

Liu, Z., Qi, X. and Fu, C.-W., 2021. One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1726–1736.

Loizou, M., Averkiou, M. and Kalogerakis, E., 2020. Learning part boundaries from 3d point clouds. *Computer Graphics Forum*, Wiley Online Library, volume 39, 183–195.

Lu, D., Xie, Q., Gao, K., Xu, L. and Li, J., 2022. 3dctn: 3d convolution-transformer network for point cloud classification. *IEEE Transactions on Intelligent Transportation Systems*, 23 (12), 24854–24865.

Luo, S. and Hu, W., 2021. Diffusion probabilistic models for 3d point cloud generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2837–2845.

Lyu, Z., Kong, Z., Xu, X., Pan, L. and Lin, D., 2021a. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *arXiv preprint arXiv:2112.03530*.

Lyu, Z., Kong, Z., Xu, X., Pan, L. and Lin, D., 2021b. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *arXiv preprint arXiv:2112.03530*.

Ma, J. and Yong, J., 1999. *Forward-backward stochastic differential equations and their applications*. 1702, Springer Science & Business Media.

Madsen, D. A. and Madsen, D. P., 2016. *Engineering drawing and design*. Cengage Learning.

Matrone, F., Lingua, A., Pierdicca, R., Malinverni, E. S., Paolanti, M., Grilli, E., Remondino, F., Murtiyoso, A. and Landes, T., 2020. A benchmark for large-scale heritage point cloud semantic segmentation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 1419–1426.

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

Meng, M., Xia, J., Luo, J. and He, Y., 2013. Unsupervised co-segmentation for 3d shapes using iterative multi-label optimization. *Computer-Aided Design*, 45 (2), 312–320.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S. and Geiger, A., 2019. Occupancy networks: Learning 3d reconstruction in function space. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4460–4470.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R. and Ng, R., 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65 (1), 99–106.

Mittal, P., Cheng, Y.-C., Singh, M. and Tulsiani, S., 2022. Autosdf: Shape priors for 3d completion, reconstruction and generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 306–315.

Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N. and Guibas, L., 2019. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics (TOG), Siggraph Asia 2019*, 38 (6), Article 242.

Mo, K., Wang, H., Yan, X. and Guibas, L., 2020. Pt2pc: Learning to generate 3d point cloud shapes from part tree conditions. *European Conference on Computer Vision*, Springer, 683–701.

Monica, R., Aleotti, J., Zillich, M. and Vincze, M., 2017. Multi-label point cloud annotation by selection of sparse control points. *2017 international conference on 3D vision (3DV)*, IEEE, 301–308.

Mortenson, M. E., 1997. *Geometric modeling*. John Wiley & Sons, Inc.

Muralikrishnan, S., Kim, V. G. and Chaudhuri, S., 2018. Tags2parts: Discovering semantic regions from shape tags. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2926–2935.

Muslea, I., Minton, S. and Knoblock, C. A., 2000. Selective sampling with redundant views. *AAAI/IAAI*, 621–626.

Nam, G., Khlifi, M., Rodriguez, A., Tono, A., Zhou, L. and Guerrero, P., 2022. 3d-ldm: Neural implicit 3d shape generation with latent diffusion models. *arXiv preprint arXiv:2212.00842*.

Ng, A. and Jordan, M., 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14.

Nguyen, T., Pham, Q.-H., Le, T., Pham, T., Ho, N. and Hua, B.-S., 2021. Point-set distances for learning representations of 3d point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10478–10487.

Nguyen, V.-L., Shaker, M. H. and Hüllermeier, E., 2022. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111 (1), 89–122.

Nichol, A. Q. and Dhariwal, P., 2021. Improved denoising diffusion probabilistic models. *International Conference on Machine Learning*, PMLR, 8162–8171.

Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D., 2002. Shape distributions. *ACM Transactions on Graphics (TOG)*, 21 (4), 807–832.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S. and Lakshminarayanan, B., 2021. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22 (1), 2617–2680.

Parisi, G., 1981. Correlation functions and computer simulations. *Nuclear Physics B*, 180 (3), 378–384.

Park, J. J., Florence, P., Straub, J., Newcombe, R. and Lovegrove, S., 2019. Deepsdf: Learning continuous signed distance functions for shape representation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 165–174.

Parzen, E., 1999. *Stochastic processes*. SIAM.

Poole, B., Jain, A., Barron, J. T. and Mildenhall, B., 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

Postels, J., Liu, M., Spezialetti, R., Van Gool, L. and Tombari, F., 2021. Go with the flows: Mixtures of normalizing flows for point cloud generation and reconstruction. *2021 International Conference on 3D Vision (3DV)*, IEEE, 1249–1258.

Qi, C. R., Su, H., Mo, K. and Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R., Yi, L., Su, H. and Guibas, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Qin, C., You, H., Wang, L., Kuo, C.-C. J. and Fu, Y., 2019. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 32.

Qingyong, H., 2022. *Learning to understand large-scale 3D point clouds*. Ph.D. thesis, University of Oxford.

Rabbani, T., Van Den Heuvel, F. and Vosselmann, G., 2006. Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences*, 36 (5), 248–253.

Ramasinghe, S., Khan, S., Barnes, N. and Gould, S., 2020. Spectral-gans for high-resolution 3d point-cloud generation. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 8169–8176.

Ranzato, M., Susskind, J., Mnih, V. and Hinton, G., 2011. On deep generative models with applications to recognition. *CVPR 2011*, IEEE, 2857–2864.

Rezende, D. and Mohamed, S., 2015. Variational inference with normalizing flows. *International conference on machine learning*, PMLR, 1530–1538.

Riegler, G., Osman Ulusoy, A. and Geiger, A., 2017. Octnet: Learning deep 3d representations at high resolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3577–3586.

Rogers, L. C. and Williams, D., 2000. *Diffusions, markov processes, and martingales: Volume 1, foundations*, volume 1. Cambridge university press.

Rubens, N., Elahi, M., Sugiyama, M. and Kaplan, D., 2015. Active learning in recommender systems. *Recommender systems handbook*, 809–846.

Rusu, R. B., Blodow, N. and Beetz, M., 2009. Fast point feature histograms (fpfh) for 3d registration. *2009 IEEE international conference on robotics and automation*, IEEE, 3212–3217.

Rusu, R. B., Blodow, N., Marton, Z. C. and Beetz, M., 2008. Aligning point cloud views using persistent feature histograms. *2008 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 3384–3391.

Rusu, R. B., Bradski, G., Thibaux, R. and Hsu, J., 2010. Fast 3d recognition and pose using the viewpoint feature histogram. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2155–2162.

Sarkar, P. and Chakrabarti, A., 2014. Ideas generated in conceptual design and their effects on creativity. *Research in Engineering Design*, 25, 185–201.

Settles, B., 2009. Active learning literature survey. *Computer Sciences Technical Report 1648*.

Shapira, L., Shamir, A. and Cohen-Or, D., 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24, 249–259.

Sharma, G., Dash, B., RoyChowdhury, A., Gadelha, M., Loizou, M., Cao, L., Wang, R., Learned-Miller, E., Maji, S. and Kalogerakis, E., 2022. Prifit: learning to fit primitives improves few shot point cloud segmentation. *Computer Graphics Forum*, Wiley Online Library, volume 41, 39–50.

Sharma, G., Kalogerakis, E. and Maji, S., 2019. Learning point embeddings from shape repositories for few-shot segmentation. *2019 International Conference on 3D Vision (3DV)*, IEEE, 67–75.

Shi, X., Xu, X., Chen, K., Cai, L., Foo, C. S. and Jia, K., 2021. Label-efficient point cloud semantic segmentation: An active learning approach. *arXiv preprint arXiv:2101.06931*.

Shu, D. W., Park, S. W. and Kwon, J., 2019a. 3d point cloud generative adversarial network based on tree structured graph convolutions. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3859–3868.

Shu, D. W., Park, S. W. and Kwon, J., 2019b. 3d point cloud generative adversarial network based on tree structured graph convolutions. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3859–3868.

Shu, Z., Qi, C., Xin, S., Hu, C., Wang, L., Zhang, Y. and Liu, L., 2016. Unsupervised 3d shape segmentation and co-segmentation via deep learning. *Computer Aided Geometric Design*, 43, 39–52.

Shum, H.-Y., Hebert, M. and Ikeuchi, K., 1996. On 3d shape synthesis. *Object Representation in Computer Vision*, 131–148.

Sidi, O., Van Kaick, O., Kleiman, Y., Zhang, H. and Cohen-Or, D., 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *Proceedings of the 2011 SIGGRAPH Asia Conference*, 1–10.

Simari, P., Nowrouzezahrai, D., Kalogerakis, E. and Singh, K., 2009. Multi-objective shape segmentation and labeling. *Computer Graphics Forum*, Wiley Online Library, volume 28, 1415–1425.

Simonovsky, M. and Komodakis, N., 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3693–3702.

Snell, J., Swersky, K. and Zemel, R., 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. and Ganguli, S., 2015. Deep unsupervised learning using nonequilibrium thermodynamics. *International Conference on Machine Learning*, PMLR, 2256–2265.

Song, Y. and Ermon, S., 2019. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S. and Poole, B., 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Stal, D. M. and Turkiyyah, G. M., 1996. Skeleton-based techniques for the creative synthesis of structural shapes. *Artificial Intelligence in Design'96*, 761–780.

Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. *Proceedings of the IEEE international conference on computer vision*, 945–953.

Su, Y., Xu, X. and Jia, K., 2022. Weakly supervised 3d point cloud segmentation via multi-prototype learning. *arXiv preprint arXiv:2205.03137*.

Sun, C., Zheng, Z., Wang, X., Xu, M. and Yang, Y., 2022. Self-supervised point cloud representation learning via separating mixed shapes. *IEEE Transactions on Multimedia*.

Sun, C.-Y., Yang, Y.-Q., Guo, H.-X., Wang, P.-S., Tong, X., Liu, Y. and Shum, H.-Y., 2023. Semi-supervised 3d shape segmentation with multilevel consistency and part substitution. *Computational Visual Media*, 9 (2), 229–247.

Sun, X., Lian, Z. and Xiao, J., 2019. Srinet: Learning strictly rotation-invariant representations for point cloud classification and segmentation. *Proceedings of the 27th ACM International Conference on Multimedia*, 980–988.

Sun, Y., Wang, Y., Liu, Z., Siegel, J. and Sarma, S., 2020. Pointgrow: Autoregressively learned point cloud generation with self-attention. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 61–70.

Sünderhauf, N., Brock, O., Scheirer, W., Hadsell, R., Fox, D., Leitner, J., Upcroft, B., Abbeel, P., Burgard, W., Milford, M. et al., 2018. The limits and potentials of deep learning for robotics. *The International journal of robotics research*, 37 (4-5), 405–420.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. and Ng, R., 2020. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33, 7537–7547.

Tang, Y., Qian, Y., Zhang, Q., Zeng, Y., Hou, J. and Zhe, X., 2022. Warpinggan: Warping multiple uniform priors for adversarial 3d point cloud generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6397–6405.

Tangelder, J. W. and Veltkamp, R. C., 2008. A survey of content based 3d shape retrieval methods. *Multimedia tools and applications*, 39, 441–471.

Tao, A., Duan, Y., Wei, Y., Lu, J. and Zhou, J., 2022. Seggroup: Seg-level supervision for 3d instance and semantic segmentation. *IEEE Transactions on Image Processing*, 31, 4952–4965.

Tatarchenko, M., Park, J., Koltun, V. and Zhou, Q.-Y., 2018. Tangent convolutions for dense prediction in 3d. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3887–3896.

Tavakkoli, S. and Dhande, S. G., 1991. Shape synthesis and optimization using intrinsic geometry.

Tchapmi, L., Choy, C., Armeni, I., Gwak, J. and Savarese, S., 2017. Segcloud: Semantic segmentation of 3d point clouds. *2017 international conference on 3D vision (3DV)*, IEEE, 537–547.

Te, G., Hu, W., Zheng, A. and Guo, Z., 2018. Rgcnn: Regularized graph cnn for point cloud segmentation. *Proceedings of the 26th ACM international conference on Multimedia*, 746–754.

Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-or, D. and Bermano, A. H., 2023. Human motion diffusion model. *The Eleventh International Conference on Learning Representations*. URL `https://openreview.net/forum?id=SJ1kSyO2jwu`.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F. and Guibas, L. J., 2019. Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Tombari, F., Salti, S. and Di Stefano, L., 2010. Unique shape context for 3d data description. *Proceedings of the ACM workshop on 3D object retrieval*, 57–62.

Uy, M. A., Pham, Q.-H., Hua, B.-S., Nguyen, T. and Yeung, S.-K., 2019. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. *Proceedings of the IEEE/CVF international conference on computer vision*, 1588–1597.

Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K. et al., 2022. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35, 10021–10039.

Valsesia, D., Fracastoro, G. and Magli, E., 2018. Learning localized generative models for 3d point clouds via graph convolution. *International conference on learning representations*.

Van Kaick, O., Tagliasacchi, A., Sidi, O., Zhang, H., Cohen-Or, D., Wolf, L. and Hamarneh, G., 2011a. Prior knowledge for part correspondence. *Computer Graphics Forum*, Wiley Online Library, volume 30, 553–562.

Van Kaick, O., Zhang, H., Hamarneh, G. and Cohen-Or, D., 2011b. A survey on shape correspondence. *Computer graphics forum*, Wiley Online Library, volume 30, 1681–1707.

Vandoni, J., Aldea, E. and Le Hégarat-Mascle, S., 2019. Evidential query-by-committee active learning for pedestrian detection in high-density crowds. *International Journal of Approximate Reasoning*, 104, 166–184.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Velho, L., De Figueiredo, L. H. and Gomes, J., 1999. A unified approach for hierarchical adaptive tesselation of surfaces. *ACM Transactions on Graphics (TOG)*, 18 (4), 329–360.

Wang, B., Lan, J. and Gao, J., 2023. Msg-point-gan: Multi-scale gradient point gan for point cloud generation. *Symmetry*, 15 (3), 730.

Wang, C., Samari, B. and Siddiqi, K., 2018. Local spectral graph convolution for point set feature learning. *Proceedings of the European conference on computer vision (ECCV)*, 52–66.

Wang, L., Li, X. and Fang, Y., 2020. Few-shot learning of part-specific probability space for 3d shape segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4504–4513.

Wang, Y., Asafi, S., Van Kaick, O., Zhang, H., Cohen-Or, D. and Chen, B., 2012. Active co-analysis of a set of shapes. *ACM Transactions on Graphics (TOG)*, 31 (6), 1–10.

Wang, Y. and Solomon, J., 2019. Intrinsic and extrinsic operators for shape analysis. *Handbook of Numerical Analysis*, Elsevier, volume 20, 41–115.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M. and Solomon, J. M., 2019. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38 (5), 1–12.

Wei, J., Lin, G., Yap, K.-H., Hung, T.-Y. and Xie, L., 2020. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4384–4393.

Wen, C., Yu, B. and Tao, D., 2021. Learning progressive point embeddings for 3d point cloud generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10266–10275.

Wong, A. K. C., Lu, S. W. and Rioux, M., 1989. Recognition and shape synthesis of 3-d objects based on attributed hypergraphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 (3), 279–290.

Wu, J., Zhang, C., Xue, T., Freeman, B. and Tenenbaum, J., 2016a. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29.

Wu, J., Zhang, C., Xue, T., Freeman, B. and Tenenbaum, J., 2016b. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29.

Wu, W., Qi, Z. and Fuxin, L., 2019. Pointconv: Deep convolutional networks on 3d point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9621–9630.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.

Xiao, A., Huang, J., Guan, D., Zhang, X., Lu, S. and Shao, L., 2023. Unsupervised point cloud representation learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xie, S., Gu, J., Guo, D., Qi, C. R., Guibas, L. and Litany, O., 2020. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, Springer, 574–591.

Xie, S., Liu, S., Chen, Z. and Tu, Z., 2018. Attentional shapecontextnet for point cloud recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4606–4615.

Xie, Z., Xu, K., Liu, L. and Xiong, Y., 2014. 3d shape segmentation and labeling via extreme learning machine. *Computer graphics forum*, Wiley Online Library, volume 33, 85–95.

Xu, K., Li, H., Zhang, H., Cohen-Or, D., Xiong, Y. and Cheng, Z.-Q., 2010. Style-content separation by anisotropic part scales. *ACM SIGGRAPH Asia 2010 papers*, 1–10.

Xu, M., Zhang, J., Zhou, Z., Xu, M., Qi, X. and Qiao, Y., 2021. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3056–3064.

Xu, Q., Huang, G., Yuan, Y., Guo, C., Sun, Y., Wu, F. and Weinberger, K., 2018a. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755*.

Xu, X. and Lee, G. H., 2020. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13706–13715.

Xu, Y., Fan, T., Xu, M., Zeng, L. and Qiao, Y., 2018b. Spidercnn: Deep learning on point sets with parameterized convolutional filters. *Proceedings of the European conference on computer vision (ECCV)*, 87–102.

Xue, T., Liu, J. and Tang, X., 2012. Example-based 3d object reconstruction from line drawings. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 302–309.

Yan, X., Zheng, C., Li, Z., Wang, S. and Cui, S., 2020. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5589–5598.

Yang, B., 2020. Learning to reconstruct and segment 3d objects. *arXiv preprint arXiv:2010.09582*.

Yang, G., Huang, X., Hao, Z., Liu, M.-Y., Belongie, S. and Hariharan, B., 2019. Pointflow: 3d point cloud generation with continuous normalizing flows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4541–4550.

Yang, X., Wu, Y., Zhang, K. and Jin, C., 2021. Cpcgan: A controllable 3d point cloud generative adversarial network with semantic label generating. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3154–3162.

Yang, Y., Feng, C., Shen, Y. and Tian, D., 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 206–215.

Yang, Z., Chen, Y., Zheng, X., Chang, Y. and Li, X., 2022. Conditional gan for point cloud generation. *Proceedings of the Asian Conference on Computer Vision*, 3189–3205.

Yang, Z. and Wang, L., 2019. Learning relationships for multi-view 3d object recognition. *Proceedings of the IEEE/CVF international conference on computer vision*, 7505–7514.

Yao, J., Dou, Z., Nie, J.-Y. and Wen, J.-R., 2020. Looking back on the past: Active learning with historical evaluation results. *IEEE Transactions on Knowledge and Data Engineering*, 34 (10), 4921–4932.

Yi, L., Huang, H., Liu, D., Kalogerakis, E., Su, H. and Guibas, L., 2018. Deep part induction from articulated object pairs. *arXiv preprint arXiv:1809.07417*.

Yi, L., Kim, V. G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A. and Guibas, L., 2016a. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35 (6), 1–12.

Yi, L., Kim, V. G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A. and Guibas, L., 2016b. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35 (6), 1–12.

Yu, T., Meng, J. and Yuan, J., 2018. Multi-view harmonized bilinear network for 3d object recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 186–194.

Yuan, S. and Fang, Y., 2020. Ross: Robust learning of one-shot 3d shape segmentation. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1961–1969.

Zamorski, M., Zięba, M., Klukowski, P., Nowak, R., Kurach, K., Stokowiec, W. and Trzciński, T., 2020. Adversarial autoencoders for compact representations of 3d point clouds. *Computer Vision and Image Understanding*, 193, 102921.

Zeiler, M. D. and Fergus, R., 2014. Visualizing and understanding convolutional networks. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, Springer, 818–833.

Zhang, K., Yang, X., Wu, Y. and Jin, C., 2022a. Attention-based transformation from latent features to point clouds. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 3291–3299.

Zhang, R., Chen, J., Gao, W., Li, G. and Li, T. H., 2022b. Pointot: Interpretable geometry-inspired point cloud generative model via optimal transport. *IEEE Transactions on Circuits and Systems for Video Technology*, 32 (10), 6792–6806.

Zhang, Y., Li, Z., Xie, Y., Qu, Y., Li, C. and Mei, T., 2021a. Weakly supervised semantic segmentation for large-scale point cloud. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3421–3429.

Zhang, Y., Ling, H., Gao, J., Yin, K., Lafleche, J.-F., Barriuso, A., Torralba, A. and Fidler, S., 2021b. Datasetgan: Efficient labeled data factory with minimal human effort. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10145–10155.

Zhang, Y., Qu, Y., Xie, Y., Li, Z., Zheng, S. and Li, C., 2021c. Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15520–15528.

Zhang, Y. and Rabbat, M., 2018. A graph-cnn for 3d point cloud classification. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 6279–6283.

Zhao, H., Jiang, L., Fu, C.-W. and Jia, J., 2019a. Pointweb: Enhancing local neighborhood features for point cloud processing. *Proceedings of* *the IEEE/CVF conference on computer vision and pattern recognition*, 5565–5573.

Zhao, H., Jiang, L., Jia, J., Torr, P. H. and Koltun, V., 2021a. Point transformer. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16259–16268.

Zhao, N., Chua, T.-S. and Lee, G. H., 2021b. Few-shot 3d point cloud semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8873–8882.

Zhao, Y., Birdal, T., Deng, H. and Tombari, F., 2019b. 3d point capsule networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1009–1018.

Zhao, Y., Birdal, T., Deng, H. and Tombari, F., 2019c. 3d point capsule networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1009–1018.

Zheng, Y., Cohen-Or, D., Averkiou, M. and Mitra, N. J., 2014. Recurring part arrangements in shape collections. *Computer Graphics Forum*, Wiley Online Library, volume 33, 115–124.

Zhou, L., Du, Y. and Wu, J., 2021. 3d shape generation and completion through point-voxel diffusion. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5826–5835.

Zintgraf, L. M., Cohen, T. S., Adel, T. and Welling, M., 2017. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*.

Zou, C., Yumer, E., Yang, J., Ceylan, D. and Hoiem, D., 2017. 3d-prnn: Generating shape primitives with recurrent neural networks. *Proceedings of the IEEE International Conference on Computer Vision*, 900–909.