Understanding 3D Point Cloud via Unsupervised Learning, Diffusion-based and Frequency-guided Generative Deep Learning Architectures

Jiaxu Liu

A Thesis presented for the degree of
Doctor of Philosophy

Department of Computer Science
Durham University
United Kingdom
Nov 2025

Abstract

With the rapid development of 3D sensing technologies, point clouds have emerged as a fundamental representation for numerous 3D understanding tasks, including classification, segmentation, and generative modeling. Despite their potential, point clouds remain challenging to process due to their unstructured nature and the high computational cost involved in extracting meaningful features. This thesis explores novel methods for both understanding and generating 3D point cloud data, with a particular focus on unsupervised learning and diffusion-based generative modeling. To tackle the challenge of semantic segmentation without manual labels, we introduce a novel unsupervised segmentation framework that combines deep clustering with traditional k-means and superpoint-based methods. This approach enables the model to discover meaningful semantic structures directly from raw point clouds data, eliminating the need for costly human annotations. The framework effectively groups points into semantically coherent regions, demonstrating strong performance across diverse datasets. On the generative modeling front, we propose a new class of diffusion models tailored specifically for point clouds. The first method introduces a one-step, time-variant, frequency-aware diffusion approach. By leveraging the Laplacian operator, we extract frequency-domain features from point clouds and enhance high-frequency components throughout the diffusion process. This frequency-aware strategy, when combined with a powerful latent representation learned using Mamba, enables the synthesis of high-fidelity, semantically rich point cloud samples. Building upon this, we also develop a two-stage generative framework that integrates a variational autoencoder with a latent diffusion model, inspired by stable diffusion techniques. This method features a frequency-aware module that enriches the VAE's latent space with detailed spectral information, which is then further refined during the latent diffusion stage. A specialized architecture for the latent space ensures that fine-grained geometric details are preserved and that the overall generative process remains robust and expressive. Extensive experimental evaluations demonstrate the effectiveness and versatility of our proposed approaches. Both the segmentation and generation techniques achieve state-of-the-art results on widely used benchmark

datasets. These contributions significantly advance the field of 3D point cloud understanding and synthesis, offering scalable and annotation-efficient solutions with practical applications in areas such as computer vision, robotics, and graphics.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

Summoning anew the courage that had faded.
再次鼓起丧失的勇气。

At the beginning of my doctoral journey in Durham, I was like a machine without an operating system. The hardware was complete, yet it could not start, nor did it know which program to run. My supervisor, Toby, was the essential kernel—initializing my direction and teaching me how the algorithms should be called. For his guidance and steady watch, I am deeply grateful.

My co-supervisor, Hubert, was like a cursor blinking in a dark terminal: small, yet never extinguished. Because of that signal, even in the most chaotic code, I could still locate the correct path.

My companions, postdocs Neelanjan Bhowmik, Yona Falinie Binti-Abd-Gaus, and Brian Isaac-Medina, as well as fellow students Ghada Alosaimi, Wenke E, Li Li, Honghao Pan, Yixin Sun, Zhengdi Yu, Zikai Zhang, Ruochen Li, Shuang Chen, Lupeng Zhang, and Ziyi Chang, were like nodes in a distributed system. At times they sent small packets that pushed me forward; at times they connected directly and ran in parallel with me. Through their exchanges and collaborations, isolation dissolved, and the process grew steadier and longer.

My parents are my deepest power source and foundation. Without them, no computation could have begun. Their support and their lifting hands are the sole reason I have been able to keep running until today.

During these years, my partner, Yuetong Guo, has shared the life with me. The world of computation is never free of noise or deadlocks, yet with her presence, every error could be debugged into a story worth telling. The strength of the spirit is more precious than any kernel or cursor.

And to my best friend, Zhongjie Zhang, I owe my gratitude. You are the safe port to which I can always connect. I am also grateful to other friends, Yuanbo Wang,

Weitong Li, Mengjin Zhu, Hao Huang, GaMa, Chenyu, and Hanxuan, among others. Because of you, this vast and intricate network has never been cold or empty.

Dedication

To My Parents.

Contents

List of Figures

List of Tables

List of Symbols

LIDAR Light Detection and Ranging, p. 1, 11 , 29, 41, 90

RGB-D Red-Green-Blue plus Depth, p. 1, 11, 41

MLP Multi Layer Perceptron, p. 1, 11, 14, 77, 86

VAE Variational Autoencoder, p. 3, 5, 9, 17, 18, 21, 47, 71, 76, 77

GAN Generative Adversarial Network, p. 3, 5, 9, 17, 47, 71

CNN Convolutional Neural Network, p. 12, 23

DGCNN Dynamic Graph Convolutional Neural Network, p. 12

SOTA State of the Art, p. 14, 47

DDPM Denoising Diffusion Probabilistic Model, p. 18, 47,, 77, 82

LDM Latent Diffusion Model, p. 18, 71

Mamba State Space Model, p. 8, 21, 23 , 50, 51, 53, 60, 63, 90

SSM State Space Model, p. 23, 51, 60

NLP Natural Language Processing, p. 23

oAcc Overall Accuracy, p. 25, 28, 28, 39

mAcc Mean Accuracy, p. 25, 28, 28, 39

mIoU Mean Intersection over Union, p. 25 , 28 , 28, 39

1-NNA 1-Nearest Neighbor Accuracy, p. 26, 28, 28, 47, 63, 67, 83, 84, 86

Abs50 Absolute 50-Shifted , p. 26, 28, 28, 47, 63, 67, 84

COV Coverage, p. 8, 26, 28, 28, 29, 63, 66, 67, 83, 86, 90

EMD Earth Mover's Distance, p. 8, 26, 28, 28, 29, 63, 66, 67, 83, 86

MMD Minimum Matching Distance, p. 26, 28, 28

CD Chamfer Distance, p. 26, 28, 28, 63, 66, 67, 83, 86

CHAPTER 1

Introduction

With the widespread adoption of 3D sensors such as Light Detection and Ranging (LiDAR), Red-Green-Blue plus Depth (RGB-D) cameras, and laser scanners [6], geometric data can now be captured at unprecedented scale and resolution. Among various 3D representations, such as depth maps, meshes, and voxels, point clouds have become particularly prominent due to their high fidelity, ease of acquisition, and flexibility for downstream processing.

However, unlike 2D images, point clouds are inherently unordered, irregularly sampled, and permutation-invariant. In real-world scans, they also suffer from varying point density, occlusion, and sensor noise, making the direct use of conventional 2D convolutional paradigms suboptimal.

To address these challenges, existing methods for point cloud processing generally fall into three paradigms: (i) **point-based** methods that learn from raw point sets with shared Multi Layer Perceptrons (MLPs) and local neighborhood aggregation [7,8]; (ii) **voxel-based** approaches that discretize 3D space into regular grids and apply 3D convolutions [9, 10]; and (iii) **graph-based** or relational approaches that build a graph over points, voxels, or superpoints to capture spatial relationships [11, 12]. Importantly, the graph paradigm should not be seen as an alternative to point-

or voxel-based representations, but rather as a complementary relational module layered on top of them. In typical architectures, one first processes geometric features via point- or voxel-based backbones, then applies graph layers over points, voxels, or superpoints to model spatial or topological relationships. This hybrid design combines the strengths of both geometric embedding and relational reasoning. Such integrated paradigms have driven significant advances in applications including virtual reality. [13], robotics [14], 3D scene understanding [15], and shape completion [16, 17].

Yet despite remarkable progress, two fundamental challenges persist. First, contemporary segmentation approaches rely heavily on labor-intensive 3D annotations, limiting scalability and applicability in new domains. Second, generative models for point clouds incur high computational overhead and often fail to preserve fine-grained geometric details, particularly over large or complex structures. These limitations motivate this thesis's dual focus: (i) developing fully unsupervised semantic segmentation, and (ii) designing efficient, high-fidelity point cloud generative models. In combination, our work seeks to establish annotation-free semantic understanding and computationally tractable, geometry-aware generation in the 3D point cloud domain, thereby bridging the gap between practical deployment and expressive modeling.

1.1 Motivations

Recent progress in 3D deep learning has produced powerful systems for point cloud segmentation, classification, and generation. However, these systems still face two persistent and interrelated challenges: (i) achieving high learning efficiency, especially under label scarcity or limited computational resources; and (ii) obtaining expressive representations to capture rich geometric structure and fine detail in 3D data.

Our research is driven by these twin goals: **efficiency** and **representation quality**, and is organized into two complementary directions: (1) *annotation-free semantic segmentation*, and (2) *efficient, high-fidelity point cloud generation*. In both directions, we aim to reduce reliance on supervision or heavy computation, while preserving and exploiting the underlying spatial and structural cues inherent to point clouds.

1.1.1 Motivation for Annotation-Free and Structure-Aware Semantic Segmentation

While 3D semantic segmentation is foundational for scene understanding, it remains heavily dependent on large-scale, dense, and accurate annotations. Creating such annotations is labor-intensive and error-prone: it requires annotators to interpret sparse, noisy, and unstructured point clouds in 3D space—often with significant viewpoint ambiguity and occlusions.

Various efforts have aimed to reduce annotation cost, including semi-supervised [18], weakly-supervised [19], and self-supervised [20] methods. However, these still rely on either a small set of human-labeled data or indirect proxy signals. In contrast, **unsupervised learning** directly exploits geometric and topological regularities without any manual labels or pretext assumptions.

To this end, we propose $U3DS^3$, a fully unsupervised segmentation framework for holistic 3D scenes. Our method constructs geometry-based superpoints, learns robust features via voxel-based encodings, and iteratively refines semantic clusters using transformation-consistent pseudo-labels. By leveraging invariance and equivariance principles, we enable the model to learn discriminative and structure-aware representations entirely from raw point clouds.

1.1.2 Motivation for Efficient and Geometrically Faithful Point Cloud Generation

Point cloud generation is a key task for 3D modeling, simulation, scene completion, and data augmentation. However, it remains challenging to simultaneously ensure high output fidelity and tractable training/inference costs. Generative frameworks such as Variational Autoencoder (VAE) [21], Generative Adversarial Network (GAN) [22], and normalizing flows [23] often struggle with instability, blurry results, or limited expressiveness. Recently, diffusion models [24, 25] have shown state-of-the-art performance in fidelity and diversity, but at the cost of significant computational demands due to iterative denoising steps.

To balance these trade-offs, we introduce two generative frameworks that enhance

both *efficiency* and *geometric representation learning*. First, our model TFDM integrates a time-variant frequency-aware encoder with a Mamba-based state space module in the latent space, enabling precise modeling of fine geometric details while significantly reducing computation. Second, we extend this idea in FLDCG, which combines frequency decomposition with a multi-band Transformer VAE to generate high-fidelity shapes from a compressed latent space. These designs directly incorporate frequency-domain priors into the representation learning process, improving edge preservation, structural diversity, and sampling efficiency.

Together, these two research directions reflect a unified motivation: to develop point cloud learning models that are both annotation-efficient and representation-rich, enabling scalable and high-quality 3D understanding and generation.

1.2 Problem Definitions

This thesis addresses two fundamental tasks in 3D point cloud understanding: **semantic segmentation** and **object generation**. These problems serve as both evaluation targets and driving forces for developing more efficient and expressive 3D learning models.

1.2.1 3D Point Cloud Semantic Segmentation

Semantic segmentation in 3D point clouds aims to assign a semantic label to each point in a given scene, enabling machines to understand and interact with their surrounding environment. This task is essential for various applications such as autonomous driving [26, 27], robotics [28], and augmented reality [13].

Let a point cloud be represented as a set $P = \{p_1, p_2, \ldots, p_N\}$, where each p_i denotes a 3D point with coordinates (x_i, y_i, z_i), and optionally, additional features such as RGB color or surface normals. The goal is to assign each point p_i a label l_i from a predefined set of C semantic classes $L = \{l_1, l_2, \ldots, l_C\}$. Formally, the semantic segmentation task is defined as learning a function:

$$f : P \to L, \quad \text{where } f(p_i) = l_i.$$

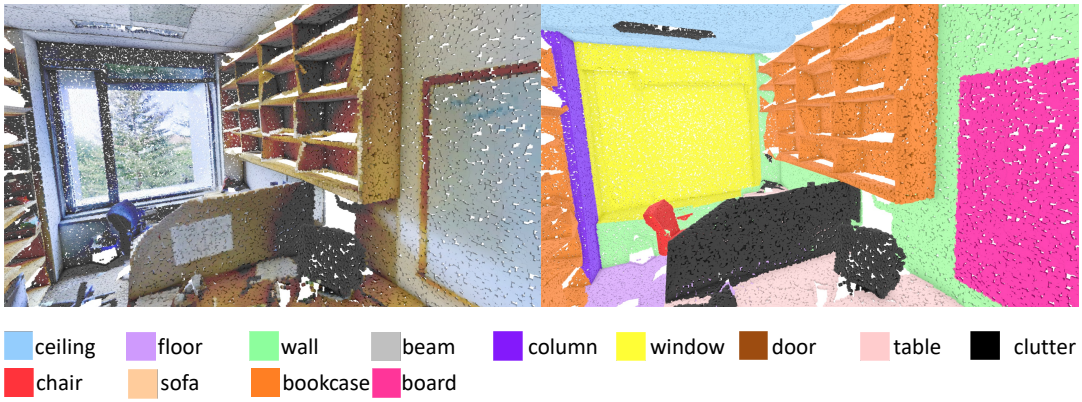

Figure 1.1: Illustration of 3D point cloud semantic segmentation. Left: raw point cloud. Right: per-point semantic labels colored by class.

This function f is typically realized through deep neural networks tailored to 3D data, such as PointNet [7], PointNet++ [8], and more recent architectures like Point Transformer [29,30]. These models must respect the unique properties of point clouds, including permutation invariance, irregular sampling, and geometric locality.

As illustrated in Fig. 1.1, the objective is to map each raw point to a semantic category such as *road, vehicle, building*, or *tree*, yielding dense per-point predictions that are spatially coherent and semantically meaningful.

1.2.2 3D Point Cloud Object Generation

In contrast to segmentation, point cloud generation focuses on synthesizing novel 3D shapes that resemble real-world objects or scenes. This task plays a critical role in simulation, virtual content creation, and 3D data augmentation [13,15]. It also supports downstream applications such as scene completion, domain adaptation, and robustness training for discriminative models.

Formally, let a training set of object shapes be denoted as $\mathcal{S} = \{P^{(n)}\}_{n=1}^{M}$, where each $P^{(n)} = \{p_i\}_{i=1}^{N_n}$ is a set of 3D points $p_i \in \mathbb{R}^3$. The generative goal is to learn a distribution $p_\theta(P \mid c)$ over point clouds, optionally conditioned on an external input c (e.g., class label, partial scan, or image). A generator g_θ then produces samples via:

$$\widehat{P} = g_\theta(z, c), \quad z \sim p(z),$$

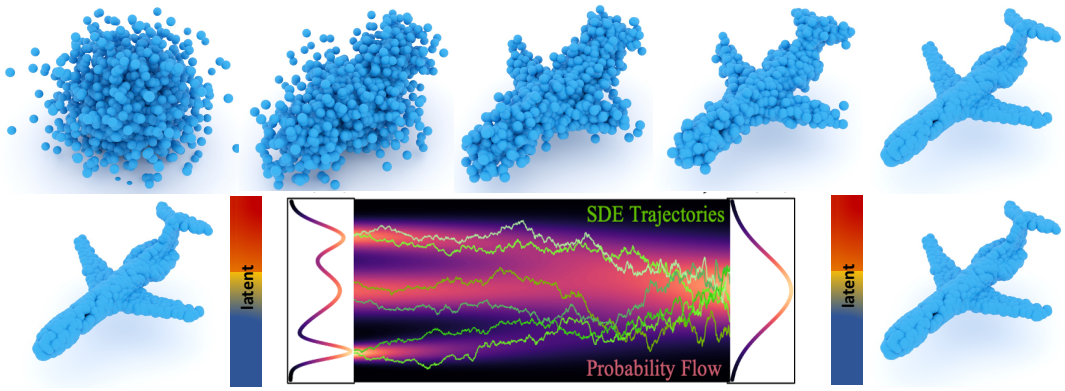

Figure 1.2: Illustration of point cloud diffusion generation. Top: end-to-end diffusion training on points. Bottom: latent diffusion model with training on latent.

where z is a random latent code and c is the conditioning variable. When $c = \varnothing$, the task is *unconditional* generation; otherwise, it is *conditional*.

Modern generative models for point clouds include VAEs [21], GANs [31], normalizing flows [23], and diffusion models [25, 32]. Among these, diffusion-based models have demonstrated superior performance in terms of fidelity and diversity, particularly when enhanced by geometric priors such as frequency-aware or latent-structure-based designs.

As shown in Fig. 1.2, the generator transforms random latent codes into dense point sets, producing shapes that align with target semantics and exhibit high-frequency detail such as edges, corners, and thin parts. This task evaluates both the expressiveness of latent representations and the fidelity of geometric reconstruction.

1.3 Research Aims

The overarching aim of this thesis is to advance **efficient and representation-rich learning** for 3D point clouds. Our research focuses on two complementary directions: (i) unsupervised semantic segmentation, and (ii) efficient, high-fidelity point cloud generation. In both directions, we emphasize not only reducing annotation or computational costs, but also improving the expressiveness of learned representations.

Aim for Annotation-Free and Structure-Aware Segmentation We aim to establish a segmentation framework that learns semantic structure directly from raw

3D scans without requiring human annotations. This involves:

- **Efficiency:** eliminating costly labels and pretraining by deriving supervision directly from geometric and topological cues, ensuring scalability to large unlabeled datasets.
- **Representation Learning:** leveraging invariance and equivariance principles to learn robust, structure-aware features that generalize across domains and sensing conditions.
- **Holistic Scene Understanding:** extending beyond object-centric segmentation to parse both foreground and background in diverse indoor and outdoor scenes.

Aim for Efficient and Geometrically Faithful Generation We aim to design generative models that achieve high fidelity and diversity while reducing the heavy computational burden of diffusion-based methods. This involves:

- **Efficiency:** introducing latent-space designs and lightweight modules to accelerate training and sampling, reducing both inference time and memory footprint.
- **Representation Learning:** incorporating frequency decomposition and advanced modules such as state space models and transformers to preserve fine-grained geometric details (e.g., edges and thin parts).
- **Flexible Paradigms:** supporting both end-to-end frameworks and two-stage latent designs, offering trade-offs between computational efficiency and generative expressiveness.

In summary, the research aims of this thesis are twofold: (1) to establish an annotation-free and structure-aware framework for point cloud semantic segmentation, and (2) to design efficient and geometrically faithful generative models for point cloud synthesis. Both aims are unified by a focus on learning robust and expressive representations under practical resource constraints.

1.4 Contributions

This thesis makes three main contributions to the field of 3D point cloud learning, spanning unsupervised semantic segmentation and diffusion-based generative modeling. All contributions are unified by the goal of improving both **efficiency** and **representation quality** in 3D understanding.

- **Unsupervised semantic segmentation without labels.** We propose $U3DS^3$, the second fully unsupervised method for holistic 3D scene segmentation that requires no human annotations or model pretraining. By leveraging intrinsic geometric cues through superpoint construction, clustering with pseudo-label refinement, and invariance/equivariance-based feature learning, our framework achieves competitive or state-of-the-art performance on major benchmarks (ScanNet, SemanticKITTI) and robust generalization to diverse environments.

- **An efficient frequency-aware diffusion model with state space model.** We introduce TFDM, an end-to-end point cloud generative framework that integrates a time-variant frequency encoder with dual Mamba blocks in the latent space. This design enables coarse-to-fine recovery of geometric detail while significantly reducing model parameters and inference time (up to 10× and 9× reductions respectively compared with current state-of-the-art methods), achieving state-of-the-art fidelity on ShapeNet-v2 across several classes.

- **A frequency-aware latent diffusion model with multi-band transformers.** We propose FLDCG, a novel two-stage generative model that enriches VAE latent representations with multi-frequency decomposition and transformer modules. By explicitly modeling different spectral bands, our method captures both global shape and high-frequency details, leading to a 0.47% improvement in Coverage Earth Mover's Distance (COV-EMD) and more than 20× reduction in parameters compared to strong baselines.

Together, these contributions advance point cloud learning by: (1) eliminating the dependence on costly 3D annotations, (2) designing diffusion-based generative

models that are both computationally efficient and geometrically expressive, and (3) demonstrating that frequency- and structure-aware representations provide a powerful foundation for generative 3D tasks.

1.5 Publications

The research related to this thesis has been previously published or under review in the following:

- **Liu, J**., Yu, Z., Breckon, T. P., & Shum, H. P. (2024). U3ds3: Unsupervised 3d semantic scene segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 3759-3768).
- **Liu, J.**, Li, L., Shum, H. P., & Breckon, T. P. (2025). TFDM: Time-Variant Frequency-Based Point Cloud Diffusion with Mamba. arXiv preprint arXiv:2503.13004. Under review at Neurocomputing.
- **Liu, J.**, Wenke E., Shum, H. P., & Breckon, T. P. (2025). FLDCG: Frequency-Aware Latent Diffusion for 3D Point Cloud Generation. Under review at International Conference on 3D Vision (3DV).

1.6 Thesis Structure

Having outlined the motivations, defined the problems, and summarized our contributions, we now present the structure of this thesis. The organization follows a logical progression from background and related work to our three main research contributions, concluding with a summary and outlook.

- **Chapter 1 (Introduction):** Introduces the research background, highlights the challenges of efficiency and representation in 3D point cloud learning, and outlines the research aims and contributions of this thesis.
- **Chapter 2 (Literature Review):** Reviews prior studies in point cloud representation learning, including supervised and annotation-efficient segmentation methods, as well as generative models ranging from VAEs and GANs

to diffusion-based approaches. We emphasize the limitations in efficiency and representation quality that motivate our work.

- **Chapter 3 (Unsupervised 3D Semantic Segmentation):** Presents $U3DS^3$, our fully unsupervised segmentation framework. We detail its geometric superpoint construction, clustering with pseudo-labels, and invariance/equivariance-based representation learning, followed by experimental validation on large-scale benchmarks.

- **Chapter 4 (Time-Variant Frequency Diffusion with Mamba):** Introduces TFDM, our end-to-end frequency-aware diffusion model enhanced with state space (Mamba) modules. We describe its architecture, efficiency improvements, and fidelity gains, supported by experiments on ShapeNet-v2.

- **Chapter 5 (Frequency-Aware Latent Diffusion with Multi-Band Transformers):** Describes FLDCG, a two-stage latent diffusion framework with frequency decomposition and multi-band transformer modules. We show how this design enriches latent representations, balances efficiency and fidelity, and outperforms existing baselines.

- **Chapter 6 (Conclusion and Outlook):** Summarizes the key findings of the thesis across segmentation and generation, highlights the broader impact of efficiency and representation improvements, and discusses future research directions in 3D point cloud learning.

CHAPTER 2

Literature Review

2.1 Point Cloud Representations and Architectures

Point clouds are one of the most common and versatile 3D data representations, widely used in autonomous driving, robotics, and virtual reality [13, 26, 28]. They directly encode geometry through discrete 3D points, making them efficient to capture with sensors such as LiDAR, structured light, or RGB-D cameras. Compared to volumetric or mesh representations, point clouds are lightweight and preserve geometric fidelity. However, their unique properties, including irregular sampling, permutation invariance, and density variation, make them challenging for standard convolutional neural networks originally designed for regular 2D grids. This has driven the development of specialized neural architectures for point cloud understanding.

2.1.1 Point-Based Methods

Point-based methods directly consume raw point sets without discretization. The seminal PointNet [7] established the foundation by introducing shared MLPs combined

with a symmetric pooling operator, ensuring permutation invariance. While efficient, PointNet captured only global features and struggled with fine-grained local structures. PointNet++ [8] addressed this by hierarchically grouping neighborhoods and learning local-to-global features, enabling more accurate segmentation and recognition.

Subsequent works further enhanced local feature modeling. For example, KPConv [33] introduced kernel point convolutions, which learn spatial kernels directly in 3D space, achieving strong results in segmentation benchmarks. RandLA-Net [34] proposed random point sampling with lightweight attention, enabling large-scale outdoor scene processing while maintaining efficiency. Point-based methods excel at preserving geometric fidelity since they operate directly on points, but they typically suffer from limited scalability: neighborhood search and local aggregation can be computationally expensive for millions of points.

2.1.2 Voxel-Based Methods

Voxel-based methods discretize 3D space into regular grids, enabling the use of 3D convolutions. Early works such as VoxelNet [9] extended 2D Convolutional Neural Network (CNN) into 3D by applying dense 3D convolutions on voxelized point clouds. However, dense voxelization quickly becomes intractable due to cubic growth in memory and computation with resolution.

To overcome this, sparse convolutional methods such as MinkowskiNet [35] leverage the sparsity of real-world point clouds. Sparse 3D convolutions allow networks to scale to large indoor or outdoor scenes while maintaining high resolution in occupied regions. These models have achieved state-of-the-art results on benchmarks like ScanNet [2] and SemanticKITTI [3]. Despite their efficiency, voxelization introduces quantization artifacts, and high-resolution voxels remain memory-intensive, limiting their ability to capture very fine structures such as thin edges.

2.1.3 Graph-Based Methods

Graph-based methods [11, 36, 37] treat point clouds as irregular graphs, where points are nodes and edges connect spatial neighbors. Dynamic Graph CNN (DGCNN) [38]

pioneered edge convolution (EdgeConv), dynamically constructing kNN graphs to capture local geometric relationships. Graph attention mechanisms further enhanced representation learning by assigning learnable weights to neighbor contributions, allowing more adaptive feature aggregation.

Graph-based methods excel at modeling relational structures and non-Euclidean neighborhoods, capturing local geometry more effectively than vanilla point-based approaches. However, constructing and updating neighborhood graphs is computationally expensive, especially for large-scale scenes. As a result, graph-based models often struggle with scalability, limiting their practicality in real-time applications.

2.1.4 Towards Hybrid and Transformer-Based Architectures

Recognizing the strengths and weaknesses of each paradigm, hybrid architectures combine multiple strategies. For example, point-voxel hybrid networks such as PVCNN [39] process local geometry at the point level while leveraging voxel grids for global context, balancing fidelity and scalability. SparseConv-based [40] backbones are often integrated with point-based refinement modules to achieve better accuracy with manageable costs.

Transformers have recently been adapted to 3D point clouds. Point Transformer [29] introduced self-attention mechanisms that respect permutation invariance and model long-range dependencies. Later versions (Point Transformer v2 [30], v3 [41]) improved efficiency and scalability to larger scenes. Other works exploit sparse attention or hierarchical transformer designs to reduce computation. These transformer-based models represent a major shift toward expressive representation learning in 3D, but they come with significant computational overhead, motivating research into alternatives that retain expressiveness while improving efficiency.

Discussion

The evolution of point cloud architectures highlights a central trade-off: point-based methods preserve fidelity but struggle with scalability; voxel-based methods scale well but suffer from quantization; graph-based methods capture relations but are

computationally heavy; and transformer-based methods offer expressiveness at high cost. Hybrid designs attempt to strike a balance, but efficiency versus representation quality remains an open challenge. This trade-off motivates our exploration of new architectures that are both efficient and representation-rich, as detailed in subsequent chapters on segmentation (Chapter. 3) and generative modeling (Chapter. 4– Chapter. 5).

2.2 Semantic Segmentation of Point Clouds

Semantic segmentation is one of the most fundamental tasks in point cloud understanding, aiming to assign semantic categories to each point in a 3D scene. It is crucial for downstream applications such as autonomous driving, indoor scene analysis, and robotics. Over the past years, the field has evolved from fully supervised methods to annotation-efficient approaches, and more recently to unsupervised paradigms. Below we review these developments in detail.

2.2.1 Supervised Approaches

Fully supervised methods have been the dominant paradigm for 3D point cloud segmentation. Early breakthroughs such as PointNet [7] introduced a simple yet powerful architecture that directly consumed unordered point sets by applying shared MLPs followed by symmetric max-pooling. However, PointNet lacked the ability to capture local structures, which are essential for fine-grained segmentation. PointNet++ [8] addressed this by introducing a hierarchical architecture with neighborhood grouping, capturing local-to-global context.

Subsequent work aimed to improve local feature extraction. Kernel Point Convolution (KPConv) [33] replaced MLPs with learnable kernel points, achieving strong results on large-scale benchmarks. At the same time, voxel-based architectures such as MinkowskiNet [35] applied sparse 3D convolutions to scale up to large indoor and outdoor datasets. More recently, transformer-based architectures like Point Transformer [29, 30] and its successors introduced self-attention mechanisms, enabling explicit modeling of long-range dependencies in irregular point sets.

These supervised methods achieve state-of-the-art (SOTA) performance on standard benchmarks such as S3DIS [1], ScanNet [2], and SemanticKITTI [3]. However, they all rely on dense and accurate point-wise annotations, which are extremely costly to obtain. Unlike 2D images, annotating 3D point clouds requires reasoning in unstructured and often noisy environments, with occlusions and varying density, making the process slow and error-prone. This annotation bottleneck has motivated the exploration of annotation-efficient learning.

2.2.2 Annotation-Efficient Segmentation

Several strategies have been proposed to reduce annotation requirements while maintaining segmentation performance:

Semi-supervised learning. These methods combine a small labeled set with a large pool of unlabeled data. Consistency-based methods enforce prediction stability under perturbations, while pseudo-labeling approaches iteratively refine labels for unlabeled points [18]. Such approaches achieve significant improvements but still require at least some manual labels.

Weakly-supervised learning. Instead of full labels, weak supervision uses inexpensive annotations such as scene-level tags, bounding boxes, superpoint-level hints, or sparse scribbles [19, 42]. These methods greatly reduce annotation cost but often suffer from label noise and lack fine-grained boundary accuracy.

Self-supervised learning. Self-supervised approaches leverage pretext tasks that require no human labels, such as contrastive learning [20], clustering, or point cloud reconstruction. For example, PointContrast and DepthContrast learn transferable representations by contrasting positive and negative pairs across views. While these methods improve feature learning, they often face pretext–task mismatch, where features learned for proxy tasks may not fully transfer to downstream segmentation.

Overall, annotation-efficient methods reduce but do not eliminate dependence on supervision. This raises the question: can we achieve semantic segmentation entirely without labels?

2.2.3 Unsupervised Segmentation

Unsupervised segmentation represents the most extreme form of annotation efficiency, seeking to discover semantic partitions directly from raw point clouds with no supervision at all. Prior works explored clustering-based methods [4, 43], prototype learning, or spectral graph approaches [44, 45]. However, most focus on object-level or part-level segmentation [46, 47], and few address holistic scene segmentation. Moreover, methods often require pretraining or external cues, limiting scalability.

In this thesis, we propose $U3DS^3$, a fully unsupervised semantic segmentation framework for holistic 3D scenes. Unlike previous approaches [34, 48], $U3DS^3$does not rely on pretraining or proxy tasks. Instead, it begins by constructing geometric superpoints, then applies clustering and iterative pseudo-label refinement. To enhance representation learning, we incorporate invariance and equivariance principles [49] in voxelized space, enabling robust feature extraction across transformations.

This design allows us to segment both foreground objects and background regions in indoor and outdoor environments. Extensive evaluations on ScanNet, SemanticKITTI, and S3DIS demonstrate that $U3DS^3$achieves competitive or even state-of-the-art results among unsupervised approaches, establishing a new baseline for annotation-free scene segmentation.

2.3 Generative Modeling for Point Clouds

Generative modeling aims to synthesize plausible 3D shapes represented as point clouds. Such models not only benefit content creation and simulation, but also provide strong priors for discriminative tasks such as completion, registration, and segmentation. For instance, synthetic point clouds can augment training data, balance rare categories, or serve as shape priors for semantic understanding. Standard benchmarks such as ShapeNet [50] have been widely used to evaluate generative methods in terms of fidelity, diversity, and scalability.

Research in this field has progressed through several stages: variational autoencoders, generative adversarial networks, normalizing flows, and most recently, diffusion models. Each paradigm offers distinct advantages and limitations, as

reviewed below.

2.3.1 Variational Autoencoders (VAEs)

VAEs learn latent-variable models by maximizing a variational lower bound on the data likelihood. In point cloud generation, VAEs encode a shape into a latent Gaussian distribution and reconstruct it through a decoder. Works such as setVAE [21] demonstrated that VAEs could generate valid shapes and interpolate smoothly between latent codes. Extensions like hierarchical VAEs and structure-aware VAEs sought to capture part-level relations. However, VAEs often suffer from blurry reconstructions due to the Gaussian prior assumption, which is insufficient for modeling the complex multimodal distribution of 3D shapes. Fidelity is thus limited, especially for sharp edges and thin structures.

2.3.2 Generative Adversarial Networks (GANs)

GANs introduced adversarial training to address the fidelity issue. Point cloud GANs such as r-GAN [22] and ShapeGF [31] trained a generator to produce shapes that fool a discriminator, encouraging sharp and realistic outputs. Part-aware GANs further exploited semantic decomposition to generate structured point clouds. Despite these improvements, GANs are notoriously unstable: they require careful balancing of generator and discriminator, and often suffer from mode collapse, where the generator fails to capture the full diversity of the data distribution. Moreover, scaling GANs to high-resolution shapes or large scenes is computationally challenging, limiting their practicality.

2.3.3 Normalizing Flows

Flow-based models such as PointFlow [23] directly model point distributions using invertible transformations. By maximizing exact log-likelihood, these models provide principled training and latent-variable inference. Variants like DPF-Net [51] introduced hierarchical flows to better capture local structures. While flows excel in likelihood estimation and structured latent space learning, they require expensive

Jacobian computations and deep architectures, making them less efficient. They also struggle to represent very fine-grained geometry without incurring significant computational overhead.

2.3.4 Diffusion-Based Generative Models

Diffusion Model Diffusion models have recently emerged as the most powerful paradigm for 3D point cloud generation. Inspired by denoising diffusion probabilistic models (DDPMs) in images, 3D diffusion models [24, 25, 32, 52–54] progressively add Gaussian noise to point clouds and learn to reverse the process through iterative denoising. These models have shown superior fidelity and diversity compared to VAEs, GANs, and flows. Conditional diffusion [52] frameworks enable text-guided or image-guided point cloud generation, opening possibilities for cross-modal synthesis.

Recent work has also explored accelerating diffusion inference by reducing the number of sampling steps. Consistency Models [55] learn a consistency relation across noise levels, enabling high-quality generation with one or a few denoising steps, and Phased Consistency Models [56] further refine this idea with staged generation. These advances suggest that diffusion-based generators can improve inference speed not only through architectural design (e.g., latent diffusion), but also via improved sampling objectives.

Latent Diffusion Model Latent Diffusion Models (LDMs) [57] build upon this framework by applying the diffusion process in the learned latent space instead of pixel or point-level space. This dramatically reduces computational costs and memory requirements, while still preserving high-quality generation. Originally introduced in 2D vision tasks such as image synthesis and inpainting, LDMs have since been extended to 3D domains. Works like Point-E [58], LION [52], and others leverage latent diffusion to model point cloud generation, where the VAE serves as a backbone to encode complex geometric data into tractable latent representations.

Point Cloud Diffusion Point Cloud Denoising Diffusion Model [32] is a pioneering work that applies diffusion processes directly to point clouds, effectively capturing

their complex distributions. Point-Voxel Diffusion [25] considers merging point and voxel representations to achieve more reliable generation results. LION [52] proposes a Variational Autoencoder (VAE) framework with a hierarchical latent space modeled by two diffusion processes. FrePolad [54] further improves LION by introducing a frequency-based loss function to obtain frequency-rectified latent representations, enhancing detail preservation. Additionally, an adapted Diffusion Transformers for point clouds [53, 59] have been proposed to operate on voxelized point clouds. In contrast, TIGER [24] applies transformers to latent point cloud features, leveraging the transformer architectural capability of long-range dependency capture.

Despite their advantages, diffusion models face two key limitations. First, they are computationally expensive: generating one sample requires hundreds of iterative denoising steps, leading to slow inference. Second, while they capture global structures well, preserving fine-grained details such as edges and corners remains challenging. To address these issues, researchers have begun exploring latent diffusion models, which first compress point clouds into a latent space using autoencoders before applying diffusion. This reduces computational cost while retaining fidelity, but the design of effective latent spaces remains an open challenge.

2.3.5 Positioning of Our Work

While prior diffusion-based methods achieve strong results, they often prioritize fidelity at the expense of efficiency, or vice versa [60]. Moreover, most methods treat point clouds uniformly, without explicitly considering frequency components or geometric structures that could enhance representation.

This thesis addresses these gaps with two contributions. First, in Chapter. 4, we introduce **TFDM**, an end-to-end diffusion model that integrates a time-variant frequency encoder and state space modules (Mamba). TFDM captures the coarse-to-fine generation process more effectively while significantly reducing computational cost. Second, in Chapter. 5, we propose **FLDCG**, a latent diffusion model enriched with frequency decomposition and multi-band transformer modules. By explicitly modeling different frequency bands in the latent space, FLDCG achieves strong fidelity with far greater efficiency.

Together, these models advance point cloud generative modeling by simultaneously improving efficiency and representation quality, moving beyond the limitations of existing diffusion frameworks.

2.4 Frequency-Aware and Geometric Representation Learning

Representation learning in 3D point clouds not only requires efficiency but also the ability to capture geometric structures at different scales. Frequency analysis offers a natural perspective: low-frequency components describe coarse global structures, while high-frequency components encode fine geometric details such as edges, corners, and thin parts. Frequency-aware modeling has been extensively explored in 2D image processing and is increasingly being applied to 3D point clouds.

2.4.1 Frequency Analysis in 2D Images

In computer vision, Fourier [61] and spectral methods [45] have long been used to analyze image structures. The Fourier transform decomposes images into different frequency components, with low frequencies corresponding to smooth variations and high frequencies capturing sharp edges. Laplacian [62] and wavelet transforms [63] extend this to multiscale decompositions, enabling edge-preserving filtering, denoising, and compression. Recent neural architectures also exploit frequency decomposition to improve image super-resolution, style transfer, and generative modeling [64]. These developments highlight the value of explicitly incorporating frequency information into deep learning pipelines.

2.4.2 Frequency and Spectral Methods in 3D Point Clouds

Extending frequency analysis to irregular raw point sets requires graph-based formulations [65]. Point clouds can be represented as graphs, where the Laplacian matrix encodes local neighborhood structure. Spectral graph theory enables decomposition into eigenvalues and eigenvectors, forming a basis for filtering or separating frequency

bands. Low-frequency eigencomponents correspond to smooth, global structures, while high-frequency components highlight local variations and fine geometry [65, 66].

Several works have applied spectral methods to point clouds for segmentation and classification [67, 68]. For example, Laplacian eigenbasis has been used to detect shape boundaries, and spectral convolutions extend standard convolutions to non-Euclidean domains [66]. However, explicit frequency modeling remains under-explored compared to direct spatial-domain learning.

2.4.3 Frequency in Point Cloud Generation

Frequency-domain analysis has been extensively applied in 2D vision tasks [36, 64, 69] such as super-resolution, style transfer, and generative modeling. Its core advantage lies in decomposing signals into low- and high-frequency components: low frequencies describe global structures, while high frequencies encode local details and sharp transitions. Despite its success in images, the use of frequency information in 3D point cloud generation is still relatively limited.

The coarse-to-fine property of diffusion models makes frequency particularly relevant for point clouds. In early denoising steps, low-frequency components capture the overall shape, while in later steps, high-frequency signals refine fine structures such as edges, corners, and thin parts. Recent works have begun to explore this perspective. For example, [70] transform point clouds into signed distance fields and apply wavelet decomposition, separating the diffusion process into low- and high-coefficient parts. [54] prioritize high-frequency regions during training and adopt edge-aware objectives to improve geometric fidelity. These studies provide preliminary evidence that frequency-aware design can enhance robustness and detail preservation. Nevertheless, explicit integration of frequency modeling into point cloud generative frameworks remains unexplored.

2.4.4 Relevance to Our Work

This thesis builds upon these early attempts and develops dedicated frequency-aware generative architectures for point clouds. We propose two complementary approaches:

- **TFDM (Chapter. 4):** introduces a *time-variant frequency encoder* that progressively emphasizes different frequency bands across denoising steps. In particular, high-frequency components are amplified in later stages to refine fine structures. Combined with dual latent state space (Mamba) modules, TFDM captures the coarse-to-fine generative trajectory efficiently, reducing computational cost without sacrificing fidelity.

- **FLDCG (Chapter Chapter. 5):** designs a *multi-band transformer* within a VAE encoder. Using spectral graph decomposition, point clouds are separated into multiple frequency bands, each processed by a dedicated transformer branch. This architecture enables latent diffusion to preserve both global low-frequency structure and high-frequency details, achieving superior fidelity with substantial efficiency gains.

Together, these contributions demonstrate that frequency-aware representations are a powerful tool for bridging the gap between **geometric detail preservation** and **scalable, efficient point cloud generation**.

2.5 Advanced Architectures in 3D Deep Learning

As point cloud learning has matured, the community has moved beyond early point-, voxel-, and graph-based methods to explore more advanced neural architectures. Two major directions have emerged: transformer-based architectures and state space models. In addition, hybrid frameworks that combine multiple paradigms seek to balance fidelity and efficiency.

2.5.1 Transformers for Point Clouds

Transformers have revolutionized sequence modeling in natural language processing and computer vision, and have recently been adapted to 3D point clouds. Point Transformer [29] introduced self-attention mechanisms that are permutation-invariant and capable of modeling long-range dependencies across point sets. Follow-up work

such as Point Transformer v2 [30] and v3 [41] improved scalability through more efficient attention mechanisms and hierarchical designs.

Transformer-based models for point clouds could adopt sparse attention, local-global feature hierarchies, or hybrid designs combining convolutional layers with self-attention. These architectures have demonstrated strong results in segmentation [34], classification [41], and generative tasks [53], as they effectively capture both local and global context. However, transformers are computationally demanding, requiring quadratic complexity in sequence length, which is prohibitive for large-scale 3D scenes or high-resolution point clouds.

Although standard self-attention incurs quadratic complexity in the number of points, recent work has shown that this limitation can be alleviated through approximate attention mechanisms with linear complexity. Linformer [71] reduces the cost of self-attention by projecting the key and value sequences into a low-rank subspace, achieving $O(N)$ complexity while preserving competitive performance. Performer adopts a kernel-based perspective and approximates softmax attention using random feature mappings [72], enabling linear-time and linear-memory attention without assuming sparsity or low-rank structure.

While these approaches [29, 30, 41, 71, 72] demonstrate that efficient transformers are feasible, their application to point cloud learning remains limited. The irregular structure and strong locality of point clouds often favor local or hierarchical attention designs, and fully global linear attention has yet to be widely adopted in large-scale 3D scenarios. This motivates the exploration of alternative architectures, such as state space models, which naturally scale linearly with sequence length.

2.5.2 State Space Models

State space models (SSMs) offer an alternative to attention. Mamba is a selective SSM that integrates gating and convolution-like operations by modeling long-range dependencies with linear recurrent structures. The S4 architecture [73] demonstrated that carefully parameterized state space representations could capture long-sequence dynamics with sub-quadratic complexity. More recently, Mamba [74] introduced a selective SSM that integrates gating and convolution-like operations, achieving

efficiency comparable to CNNs while retaining the expressive power of transformers. Mamba has been successfully applied in Natural Language Processing (NLP) and vision, but its potential for 3D point cloud generation remains underexplored.

For point clouds, SSMs are particularly appealing because they can model long-range geometric dependencies without incurring the quadratic cost of attention. This makes them a promising direction for efficient large-scale point cloud learning. In this thesis, we investigate Mamba in the context of point cloud generative modeling (Chapter Chapter. 4), demonstrating that it offers a favorable balance between computational efficiency and representation quality.

2.5.3 Hybrid Architectures

Given the trade-offs of different paradigms, hybrid architectures have become increasingly popular. Some methods combine voxel-based backbones for scalability with point-based refinement for fidelity, while others integrate sparse convolution with transformer layers to capture both local and global features. Multi-branch architectures allow task-specific specialization, for example using convolution for local geometry and self-attention for semantic context.

Hybrid approaches embody a broader trend in 3D deep learning: no single paradigm is sufficient on its own, and combining complementary designs often yields better performance. However, hybrids can also introduce significant complexity and computational overhead, making efficiency a continuing concern.

Discussion

The evolution from transformers to state space models and hybrids illustrates the ongoing search for architectures that balance **efficiency** and **representation power**. Transformers provide strong global reasoning but are expensive; SSMs like Mamba offer efficiency but are less explored in 3D; and hybrids attempt to combine the best of multiple worlds. These observations directly motivate our design choices in later chapters: incorporating Mamba into frequency-aware diffusion (TFDM, Chapter. 4) and leveraging multi-band transformers in latent diffusion (FLDCG, Chapter. 5).

2.6 Evaluation and Metrics in Point Cloud Learning

Evaluating point cloud learning models requires metrics that reflect both semantic correctness for discriminative tasks and geometric fidelity for generative tasks. In addition, efficiency measures are crucial given the computational demands of large-scale 3D models. This section reviews the standard metrics used in the literature and highlights those adopted in this thesis.

2.6.1 Segmentation Metrics

Semantic segmentation requires assigning per-point semantic labels, making both accuracy and boundary quality important. The most commonly used metrics include:

- **Overall Accuracy (oAcc):** The fraction of correctly classified points across the dataset:

$$\text{oAcc} = \frac{\sum_{i=1}^{N} \mathbf{1}(\hat{y}_i = y_i)}{N},$$

  where N is the total number of points, y_i is the ground-truth label of point i, $\hat{y}_i$ is the predicted label, and $\mathbf{1}(\cdot)$ is the indicator function. oAcc provides a global measure but may be biased toward majority classes.

- **Mean Accuracy (mAcc):** The average per-class accuracy across all classes:

$$\text{mAcc} = \frac{1}{C} \sum_{c=1}^{C} \frac{TP_c}{N_c},$$

  where C is the number of classes, TP_c is the number of correctly predicted points in class c, and N_c is the total number of ground-truth points in class c. mAcc balances performance across classes, including rare ones.

- **Mean Intersection over Union (mIoU):** The average IoU across all classes:

$$\text{IoU}_c = \frac{TP_c}{TP_c + FP_c + FN_c}, \quad \text{mIoU} = \frac{1}{C} \sum_{c=1}^{C} \text{IoU}_c,$$

where TP_c, FP_c, and FN_c denote true positives, false positives, and false negatives for class c. mIoU is widely regarded as the most representative metric for segmentation benchmarks such as S3DIS [1], ScanNet [2], and SemanticKITTI [3].

In this thesis, we report oAcc, mAcc, and mIoU for all segmentation experiments.

2.6.2 Generative Modeling Metrics

For generative models, evaluation must capture both **fidelity** (how realistic samples are) and **diversity** (how well they cover the target distribution). Standard metrics include:

- **1-Nearest Neighbor Accuracy (1-NNA):** Given a set of real samples $\mathcal{R}$ and generated samples $\mathcal{G}$, the 1-NNA metric is computed as the leave-one-out classification accuracy of a 1-nearest neighbor classifier:

$$\text{1-NNA} = \frac{1}{|\mathcal{R}| + |\mathcal{G}|} \sum_{x \in \mathcal{R} \cup \mathcal{G}} \mathbf{1}\big(y(x) = \hat{y}(x)\big),$$

  where $y(x)$ is the true domain label (*real* or *generated*), and $\hat{y}(x)$ is the label predicted by a 1-NN classifier. A score close to 50% indicates that generated samples are indistinguishable from real data, i.e., an ideal balance of fidelity and diversity.

- **Absolute 50-Shifted 1-NNA (1-NNA-Abs50):** To make the interpretation clearer, we shift 1-NNA relative to the ideal 50%:

$$\text{1-NNA-Abs50} = \big|\text{1-NNA} - 50\big|.$$

  A lower score indicates the generated distribution is closer to the real data distribution.

- **Coverage (COV):** Coverage measures how many reference shapes are matched

by at least one generated sample:

$$\mathrm{COV}(\mathcal{R}, \mathcal{G}) = \frac{\left|\left\{r \in \mathcal{R} \;\middle|\; \exists g \in \mathcal{G},\ d(r,g) = \min_{g' \in \mathcal{G}} d(r,g')\right\}\right|}{|\mathcal{R}|},$$

where $d(\cdot,\cdot)$ is a point-set distance (CD or EMD). Higher COV indicates greater diversity.

- **Minimum Matching Distance (MMD):** MMD measures fidelity by computing the average distance between each reference shape and its closest generated sample:

$$\mathrm{MMD}(\mathcal{R}, \mathcal{G}) = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \min_{g \in \mathcal{G}} d(r,g).$$

  Common distances $d(\cdot,\cdot)$ include:

  - *Chamfer Distance (CD):*

$$d_{\mathrm{CD}}(P,Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p-q\|_2^2 + \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \|p-q\|_2^2,$$

    where P, Q are point clouds.

  - *Earth Mover's Distance (EMD):*

$$d_{\mathrm{EMD}}(P,Q) = \min_{\phi: P \to Q} \frac{1}{|P|} \sum_{p \in P} \|p - \phi(p)\|_2,$$

    where ϕ is a bijection between point sets P and Q.

This thesis adopts COV, 1-NNA-Abs50 with CD and EMD as the main generative evaluation metrics. For TFDM and FLDCG, these metrics quantify improvements in both fidelity (lower 1-NNA-Abs50) and diversity (higher COV) compared to baselines.

2.6.3 Efficiency Metrics

Efficiency is a central concern especially in diffusion-based generation. Beyond accuracy or fidelity, models are evaluated by:

- **Parameter Count:** The number of trainable parameters, reflecting model size.
- **Training Time:** The training hours required for diffusion model convergence.
- **Inference Time:** The wall-clock time to generate outputs, measured per sample or per scene. Diffusion models in particular are sensitive to the number of denoising steps, making inference time a critical factor.

In this thesis, we highlight efficiency improvements of up to 10× parameter reduction and 9× faster inference in TFDM, and over 20× reduction in FLDCG compared to strong diffusion baselines.

2.6.4 Dataset-Level Evaluation Conventions

Different benchmarks standardize different subsets of metrics:

- **Segmentation Benchmarks:** - *S3DIS*, *SemanticKITTI* and *ScanNet* report oAcc, mAcc and mIoU
- **Generative Benchmarks:** - *ShapeNet-v2* is the most widely used, with COV, MMD (CD/EMD), and 1-NNA as standard metrics.

Following these conventions ensures that our results are directly comparable to prior work across segmentation and generation tasks.

Discussion

In summary, point cloud learning evaluation combines **semantic accuracy** (OA, mIoU, per-class IoU, boundary metrics), **generative quality** (COV, MMD, CD, EMD, 1-NNA), and **efficiency measures** (parameters, training time, inference time). By reporting across all these dimensions, this thesis provides a comprehensive assessment of methods that emphasizes both annotation-free learning and efficient, high-fidelity generation.

CHAPTER 3

USDS^3: Unsupervised Point Cloud Scene Semantic Segmentation

3.1 Introduction

As a crucial task in 3D computer vision, there has been increasing attention paid to point cloud segmentation in recent years due to its broad applicability to many real-world applications such as autonomous driving, virtual reality, robotics, and human-computer interaction. However, owing to the unordered and unstructured nature of point clouds, it is a non-trivial exercise to undertake segmentation upon them. In recent years, supervised point cloud segmentation approaches have made significant progress [7, 8, 20, 33, 34, 38, 75] against several benchmark datasets [1–3, 50]. However, these approaches rely heavily on copious fully-annotated training data, in the form of labeled 3D point clouds. It is both time-consuming and labour-intensive to obtain such annotations accurately and consistently - especially for dense and complex 3D scenes. An alternative body of work leverages semi-supervised [18] and weakly-supervised [19, 48, 76, 77] approaches to mitigate the labelled data requirements, but still require labour-intensive annotation at some level and lack of being readily scalable and adaptable to new datasets. Our work aims to characterize 3D features

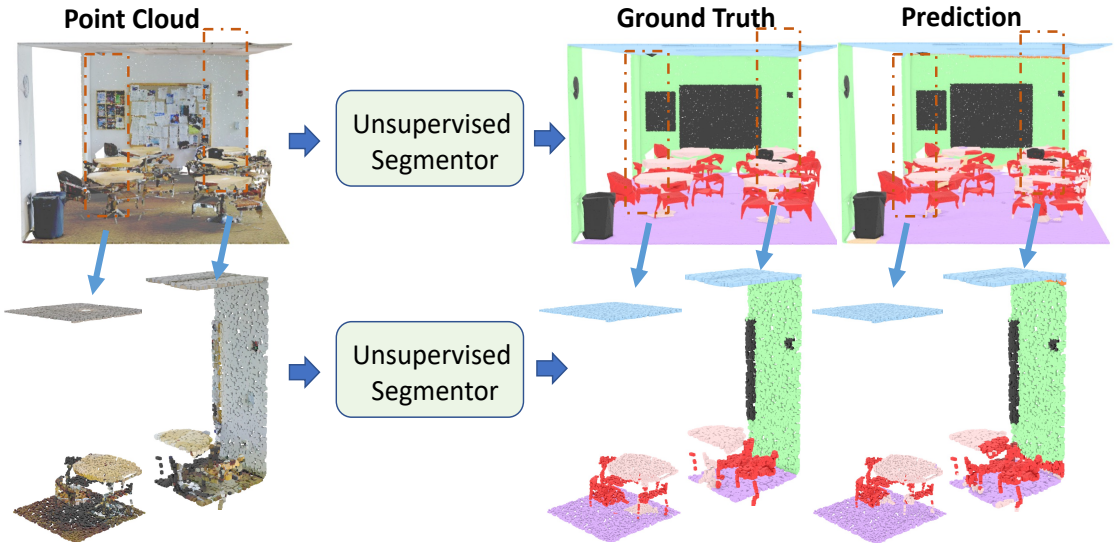

Figure 3.1: Illustration of proposed U3DS^3 on S3DIS dataset [1] Left to right denotes real scene, ground truth and $U3DS^3$ segmentation results respectively for the full scene (upper), for a single point cloud block input (lower).

without any explicit guidance allowing it to learn from the intrinsic structure of the data, and offer independence from erroneous, bias or inconsistent annotations, which significantly differ from prior weakly-supervised methods. To date, there are only a handful of prior works trying to address fully unsupervised segmentation for point clouds [47, 78–80]. However, these approaches essentially focus on object-level segmentation or co-segmentation and cannot recover the full 3D scene labels without extra scene priors [47, 79, 80] and only a recent work [78] has attempted to address fully unsupervised semantic segmentation for 3D scenes. Our proposed new $U3DS^3$ approach performs full holistic segmentation for the entire 3D scene in a scene-agnostic manner, spanning both indoor and outdoor scenarios across differing metric scales and achieving superior results on ScanNet [2] and SemanticKITTI [3] when compared to [78].

Despite the growth of unsupervised learning on 2D image segmentation [49, 81–84], there is a lack of in-depth investigation into any 3D point cloud equivalent. Although some achievements in unsupervised segmentation learning have addressed 3D point cloud data via domain adaptation [85, 86], our work does not rely upon transfer learning. OGC [47] leverages the dynamic motion pattern of a (LiDAR derived) point

cloud sequence to acquire dynamic tracks and achieve competitive results for object-level segmentation. Similarly, Yang et al. [79] successfully apply unsupervised learning for object co-segmentation in point clouds. [78] made the first attempt towards unsupervised 3D semantic segmentation via region growing to generate high-quality over-segmentation, but their method does not fully leverage the intrinsic geometric information of the point clouds and tends to predict over-smooth segmentations with more background (e.g. floor, wall) and overlook detailed object categories of the scene.

Traditional clustering methods, like k-means [5] and DBSCAN [4], can be beneficial in establishing unsupervised semantic segmentation baselines. However, these methods still exhibit notable drawbacks. k-means [5], for instance, struggles to converge effectively with non-convex datasets, exhibits weaknesses in handling uneven data distributions, and struggles to form coherent clusters in the presence of outliers and data noise. Interestingly, some existing unsupervised approaches [49, 78] also incorporate k-means as a component of their algorithms. On the other hand, DBSCAN [4] encounters challenges when dealing with categorical features, often fails to identify clusters with varying densities, requires a drop in density to identify boundaries, and experiences decreased performance in high-dimensional scenarios.

The goal of our approach is to enable a generalized method that is able to perform semantic segmentation for large-scale indoor and outdoor 3D scenes without utilizing any human labels or dynamic information between LiDAR frames. This chapter takes a new step towards scene-level unsupervised semantic segmentation with a novel strategy. Specifically, we first apply voxel cloud connectivity segmentation (VCCS) [87] to generate the initial superpoint and merge them according to the distance and normals of the superpoints. Following this, we propose the baseline method by applying mini-batch k-means [88] on the features of a 3D point cloud to generate and update the clustering centroids, and subsequently calculate the distance between features and clustering centroids to assign labels for each point as pseudo-labels under the guidance of the superpoint. After that, we train the network with the pseudo-labels to provide new network parameters for the next iteration of clustering. Subsequently, we apply a non-parametric classifier that

operates solely on the feature space distance. Finally, by leveraging the invariance and equivariance of the volumetric representations, we are able to apply differing volumetric transformations on the point cloud input and a subsequent voxelized reverse geometric transformation on these feature representations.

In this manner, our network is capable of producing several variant feature representations from the same data source. This transformation operation is derived from a very intuitive sense that the same inputs should result in similar predictions even under geometric transformation due to the principle of invariance. Fundamentally, we learn a feature representation that maximizes the effective semantic class separation. We provide two pathways to enforce color invariance and geometric equivariance that each provide our underlying inductive bias for semantic consistency and geometric structure by way of consistent clustering assignment across the two pathways. This is performed via iterative optimization of the clustering loss, which enforces a discriminative feature space capable of high-level visual similarity disambiguation. Finally, we train our voxel-based method in an end-to-end manner. Furthermore, our evaluation illustrates promising results across both indoor and outdoor datasets, S3DIS [1], ScanNet [2] and SemanticKITTI [3], demonstrating the effectiveness and practicality of our method and providing an initial reference performance for completely unsupervised 3D semantic scene segmentation. Overall, we propose a simple yet effective framework that makes the new approach towards the task of unsupervised point cloud segmentation for holistic 3D scenes, named $U3DS^3$. Fig. 3.1 illustrates an initial qualitative result of our approach. Our key contributions are summarized as:

- We propose a novel unsupervised semantic segmentation method to leverage the invariance and equivariance through geometric transformation for both 3D indoor and outdoor holistic scenes.
- We analyze and compare existing clustering approaches and the concurrent state-of-the-art, demonstrating the advantages and superiority of our method for efficient unsupervised learning on large-scale point clouds of holistic 3D scenes with faster convergence.
- We conduct extensive experiments and ablation studies to demonstrate signifi-

cant improvement over standard baselines, across the S3DIS [1] ScanNet [2] and SemanticKITTI [3] benchmark datasets, and illustrate both the practicability of the proposed framework and justify the intuition behind our design.

3.2 U3DS^3 Methodology

This work formulates the task of unsupervised point cloud semantic segmentation as point-level segmentation, where every point within the point cloud needs to be assigned a label of a fixed number of semantic class labels.

To state formally, given a point cloud set $\boldsymbol{P}$ without labels, let $\boldsymbol{c} = \{c_i\}$ and $\boldsymbol{f} = \{f_i\}$ denote the point coordinates and the corresponding features from $\boldsymbol{P} \in \mathbb{R}^{N\times 3}$, $\boldsymbol{F} \in \mathbb{R}^{N\times d}$, where N is the number of points of the input point cloud, and d denotes the feature size, which contains coordinates, colours, and normalized positional information. Hence, the goal of this work is to learn a semantic segmentation function $\boldsymbol{g}_\theta$, which is able to predict per-point labels in an unsupervised way for $\boldsymbol{P}$ using only $\boldsymbol{c}$ and $\boldsymbol{f}$.

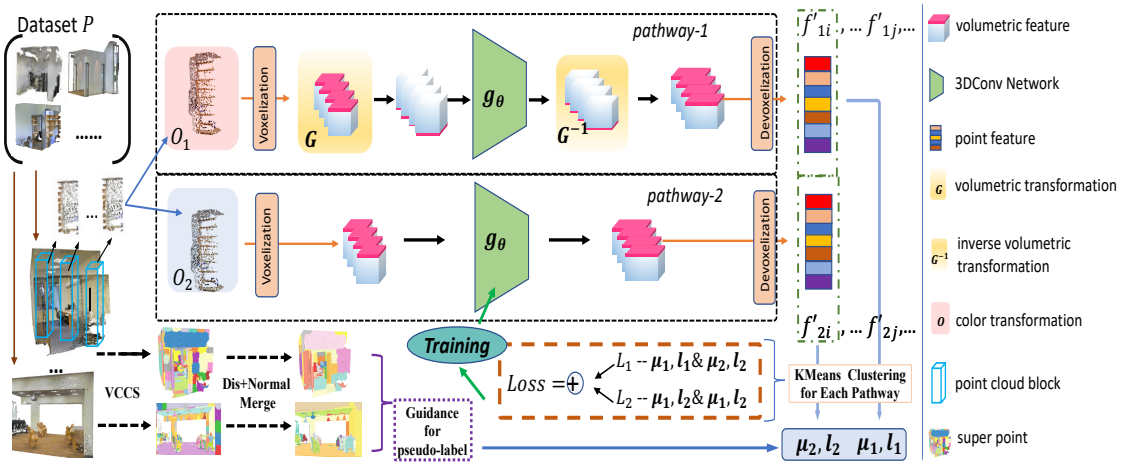

Figure 3.2: Overview of the proposed U3DS^3 each input point cloud is assigned to two pathways and gives two groups of clustering centroids and labels for training. The point cloud is initially calculated to form a superpoint. This superpoint is then merged to produce a refined superpoint, which guides the generation of pseudo-labels. The pink part of the volumetric feature here denotes reverse the input tensor along the z axis.

As shown in Fig. 3.2, for each input point region, we first apply two different colour transformations and afterwards convert them to the volumetric domain. Two different random color transformations are applied to expose the two pathways to complementary color variations, encouraging the learned representations and final predictions to be invariant to color transformations. For *pathway-1* in the top row, we implement a geometric transformation before the voxelized features are fed into the model. After the forward pass, we operate a corresponding inverse geometric transformation to the output features to ensure this representation shares the same properties with the non-transformed *pathway-2*. Subsequently, we cluster features from the different point cloud blocks and produce two groups of clustering centroids and labels, which can be used for further training and loss assembled from different pathways.

3.2.1 Superpoint

For all point clouds $P_1, P_2, P_3, \ldots$ within a point cloud set $\boldsymbol{P}$, we adhere to the VCCS [87] method to obtain initial superpoints for each point cloud. These can be denoted as $\{\{SP_1^1, SP_1^2, SP_1^3, \ldots\}, \{SP_2^1, SP_2^2, SP_2^3, \ldots\}, \ldots\}$, where SP_j^i represents the i-th superpoint in the j-th point cloud. The initial superpoints may vary across different point clouds. Subsequently, we employ a straightforward strategy to merge the superpoints within each scene: 1) Identify the smallest superpoint SP^i along with its two closest neighboring superpoints SP^{k1}, SP^{k2}; 2) Compute the vector addition of points within each superpoint and calculate the cosine similarity, here simply noted as $cos[SP^i, SP^{k1}]$; 3) Merge the smallest superpoint with the one that exhibits higher cosine similarity; 4) Repeatedly execute the above three steps until the superpoints reach a predetermined number. This simplistic approach is based on the principle that similar semantic objects possess comparable normals. Ultimately, the updated superpoints become $\{\{SP_1^{n1}, SP_1^{n2}, \ldots\}, \{SP_2^{n1}, SP_2^{n2}, \ldots\}, \ldots\}$, ensuring that the points within the same superpoint are assigned identical labels. We define the final superpoint count as a parameter, represented by γ_{sp}. For all datasets, the optimal value is empirically found as $\gamma_{sp} = 40$.

3.2.2 Voxelization and Devoxelization

We produce different representations for the input point cloud via the geometric transformation on the volumetric domain, where a voxel-based architecture is naturally adopted for such representation. Here, using voxelization and devoxelization in the pipeline, we present a simple yet effective network which contains only 3D convolutional layers with batch normalization without any additional component (details in Sec. 3.2.3).

Voxelization inevitably introduces quantization error, and its magnitude depends on the grid resolution r. A higher r preserves finer geometric details by reducing discretization artifacts, but increases memory and computation cost. Conversely, a lower r improves efficiency but may oversmooth thin structures and boundaries due to coarser discretization. In practice, r is chosen to balance detail preservation and tractability, and we fix $r = 32$ as a compromise that provides stable performance under our computational budget.

Given the input points coordinate $\boldsymbol{c}$ with corresponding features $\boldsymbol{f}$ in the input blocks, we normalize the coordinates $\boldsymbol{c}$ before voxelizing the original points to gain scale invariance. Specifically, we normalize the coordinate $\boldsymbol{c}$ into [0,1] and denoted by $\boldsymbol{c}^* = \{c_i^*\}$. In this process, the point features (including the coordinates) do not change, and the normalized coordinates are only used for converting the feature to the proper volumetric space.

When transferring the features $\boldsymbol{f}$ with normalized coordinates $\boldsymbol{c}^* = \{\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*\}$ into the voxel grids $\{\boldsymbol{V}_{m,p,q}\}$, the interpolated feature f_i for the voxel grid is calculated as the mean value of the features of points located in the grid.

$$\begin{aligned} \boldsymbol{V}_{m,p,q} = \frac{1}{K_{m,p,q}} \sum_{i=1}^{n} \boldsymbol{I}[floor(x_i^* \times r) = m, \\ floor(y_i^* \times r) = p, floor(z_i^* \times r) = q] \times f_i \end{aligned} \tag{3.1}$$

where r denotes the voxel resolution and $\boldsymbol{I}$ is an indicator function that indicates whether coordinates c_i belong to the voxel grid $\{m, p, q\}$. $K_{m,p,q}$ represents the count of points falling within the grid $\{m, p, q\}$, and $floor(\cdot)$ is floor function that outputs the greatest integer less than or equal to the input.

In terms of the per-point clustering, we need to devoxelize the voxel-based features output from the model g_θ to point-based features. We follow the trilinear interpolation of PVCNN [39] instead of the traditional nearest neighbor interpolation to ensure that nearby points are not assigned identical features.

3.2.3 Baseline: Clustering and Iteration

$U3DS^3$ applies a clustering-based method iteratively to generate pseudo-labels and train our baseline method, as inspired by DeepCluster [43]. Adapting [43] to the 3D domain is non-trivial due to the irregular nature and varying sparsity of point clouds. We present a simple yet effective strategy: switching between generating pseudo-labels via clustering with the current feature representations, and training new feature representations with the generated pseudo-labels. Different from [43, 49], the segmentation function g_θ should be able to produce per-point features, and we replace the parametric classifier with a non-parametric distance metric. Specifically, we denote the voxelization and devoxelization operations as $\boldsymbol{Z}$ and $\boldsymbol{Z}^{-1}$. The voxelized feature is $\boldsymbol{v} = \{v_i\} = \{\boldsymbol{Z}(f_i, c_i^*)\}$, and the output voxelized feature of the 3D convolutional function is $\boldsymbol{v}^{out} = g_\theta(\boldsymbol{v})$. Finally, the features for clustering can be denoted as $\boldsymbol{f}^{'} = \left\{ f_i^{'} \right\} = \left\{ \boldsymbol{Z}^{-1}(v_i^{out}, c_i^*) \right\}$. The main procedure can be separated as two parts:

(1) Using the current embeddings and k-means to cluster the points with superpoints guidance in the point cloud:

$$\min_{\boldsymbol{l},\boldsymbol{\mu}} \sum_i \left\| f_i^{'} - \mu_{l_i^{sp}} \right\|^2 \tag{3.2}$$

where l_i^{sp} denotes the cluster label of point c_i with the constraint of superpoint, and μ_k denotes the k-th cluster centroid. Note the features $f_i^{'}$ and the centroids μ_k have the same dimension.

(2) Using the class labels as pseudo-labels, we train a classifier via cross-entropy loss, which is shown in the point cloud setting as:

$$\min_{\theta,W} \sum_i L_{CE} \left(g_W(f_i^{'}), l_i^{sp}, \boldsymbol{\mu} \right) \tag{3.3}$$

where g_W denotes the parametric classifier. Under the unsupervised setting, it will be very challenging to train a classifier jointly with constantly changing pseudo-labels. We therefore choose to label points only based on their cosine distance from to the clustering centroids in feature space. Specifically, the loss function shows the following format:

$$\min_{\theta} \sum_{i} L_{cluster}\left(f_i^{'}, l_i^{sp}, \boldsymbol{\mu}\right) \tag{3.4}$$

$$L_{cluster}\left(f_i^{'}, l_i^{sp}, \boldsymbol{\mu}\right) = -log\left(\frac{e^{-d(f_i^{'}, \mu_{l_i^{sp}})}}{\sum_t e^{-d(f_i^{'}, \mu_t)}}\right) \tag{3.5}$$

where $d(\cdot, \cdot)$ denotes the cosine distance.

3.2.4 Volumetric Transformations

To improve robustness in the unsupervised setting under different scenarios, we leverage the invariance and equivariance of volumetric representations of point clouds. Invariance means that the labelling should not change after applying different transformations such as colour jittering. Equivariance in the volumetric domain means when we apply a geometric transformation to the point cloud, the corresponding 3D convolutional feature should be similarly transformed, and the corresponding labels are also wrapped according to this transformation.

For simplicity, we name the two pipelines processing the two representations as *pathway-1* and *pathway-2*. To produce two different representations for an individual input block, we apply a geometric transformation before volumetric feature extraction and then perform a corresponding inverse transformation on the final voxelized features.

Specifically, let $\boldsymbol{G}$ and $\boldsymbol{G}^{-1}$ denote the voxelized feature geometric transformation and its reverse transformation respectively, and $\boldsymbol{O}$ is the colour transformation. For point $\boldsymbol{c}$ with its feature $\boldsymbol{f}$, we apply different colour transformations for original features $\boldsymbol{f}$:

$$\boldsymbol{f}_1 = \boldsymbol{O}_1(\boldsymbol{f}), \boldsymbol{f}_2 = \boldsymbol{O}_2(\boldsymbol{f}) \tag{3.6}$$

Next, we transform these two features into the voxel grid, noting that $\boldsymbol{c}_1^*$ is actually equal to $\boldsymbol{c}_2^*$:

$$\boldsymbol{v}_1 = \boldsymbol{Z}(\boldsymbol{f}_1, \boldsymbol{c}_1^*), \boldsymbol{v}_2 = \boldsymbol{Z}(\boldsymbol{f}_2, \boldsymbol{c}_2^*) \tag{3.7}$$

After that, the voxelized feature transformations are applied to the volumetric domain: only the features of *pathway-1* are transformed whilst the other remains unchanged. The geometric transformations operate on the voxelized feature $\boldsymbol{v}$ and the corresponding reverse geometric transformations operate on the output voxel feature $\boldsymbol{v}^{out}$:

$$\boldsymbol{v}_1^{out} = \boldsymbol{G}^{-1}\left\{g_\theta[\boldsymbol{G}(\boldsymbol{v}_1)]\right\}, \boldsymbol{v}_2^{out} = g_\theta(\boldsymbol{v}_2) \tag{3.8}$$

Subsequently, we perform de-voxelization to get the features for clustering:

$$\boldsymbol{f}_1^{'} = \boldsymbol{Z}^{-1}(\boldsymbol{v}_1^{out}, \boldsymbol{c}_1^*), \boldsymbol{f}_2^{'} = \boldsymbol{Z}^{-1}(\boldsymbol{v}_2^{out}, \boldsymbol{c}_2^*) \tag{3.9}$$

3.2.5 Losses and Labelling Scheme

Given input clouds $\boldsymbol{c}$ with features $\boldsymbol{f}$, according to the colour and geometric transformations introduced in Section 3.3.3, two different feature representations, $\boldsymbol{f}_1^{'}, \boldsymbol{f}_2^{'}$, can be produced. By leveraging these two features, we cluster the two representations separately to get two groups of centroids and pseudo-labels:

$$\boldsymbol{l^{(1)}}, \boldsymbol{\mu^{(1)}} = arg\min_{\boldsymbol{l},\boldsymbol{\mu}} \sum_i \left\| f_{1i}^{'} - \mu_{l_i^{sp}} \right\|^2 \tag{3.10}$$

$$\boldsymbol{l^{(2)}}, \boldsymbol{\mu^{(2)}} = arg\min_{\boldsymbol{l},\boldsymbol{\mu}} \sum_i \left\| f_{2i}^{'} - \mu_{l_i^{sp}} \right\|^2 \tag{3.11}$$

We then set two loss functions. Firstly, the feature representation should match

the pseudo-labels produced by the same pathway:

$$L_1 = \sum_i L_{cluster}\left(f'_{1i}, l_i^{sp(1)}, \boldsymbol{\mu^{(1)}}\right) + \sum_i L_{cluster}\left(f'_{2i}, l_i^{sp(2)}, \boldsymbol{\mu^{(2)}}\right) \tag{3.12}$$

Similarly, the feature representation should whilst match the pseudo-labels produced by the different pathway:

$$L_2 = \sum_i L_{cluster}\left(f'_{1i}, l_i^{sp(2)}, \boldsymbol{\mu^{(2)}}\right) + \sum_i L_{cluster}\left(f'_{2i}, l_i^{sp(1)}, \boldsymbol{\mu^{(1)}}\right) \tag{3.13}$$

The final training objective is their summation:

$$L_{final} = L_1 + L_2 \tag{3.14}$$

The loss encourages the feature from one pathway to adhere to labels generated by another pathway, which encourages the network to label similarly to feature representations from different pathways.

Hungarian Algorithm: To match the clustering labels with the real labels, we utilize the Hungarian algorithm [89] accross the whole dataset every epoch. Specifically, where C is categories, P is the predicted set and G is the ground truth (GT) set. $S^{C\times C}$ is the matching matrix, where S_{ij} denotes the matching degree between i^{th} predicted category and j^{th} GT category. Criterion: finding bijection $\mathbf{f}: i \to j$ to maximize $\sum_{i=1}^{C} S_{i,f(i)}$.

3.3 Experiments

Implementation Details: We implement a simple yet effective framework with 8 layers 3D convolution, where each layer employs a 3D batch normalization and leaky rectified linear activation function (ReLU). The input point cloud contains 12D features, i.e., the point coordinates (x, y, z) in the normalized block coordinate system,

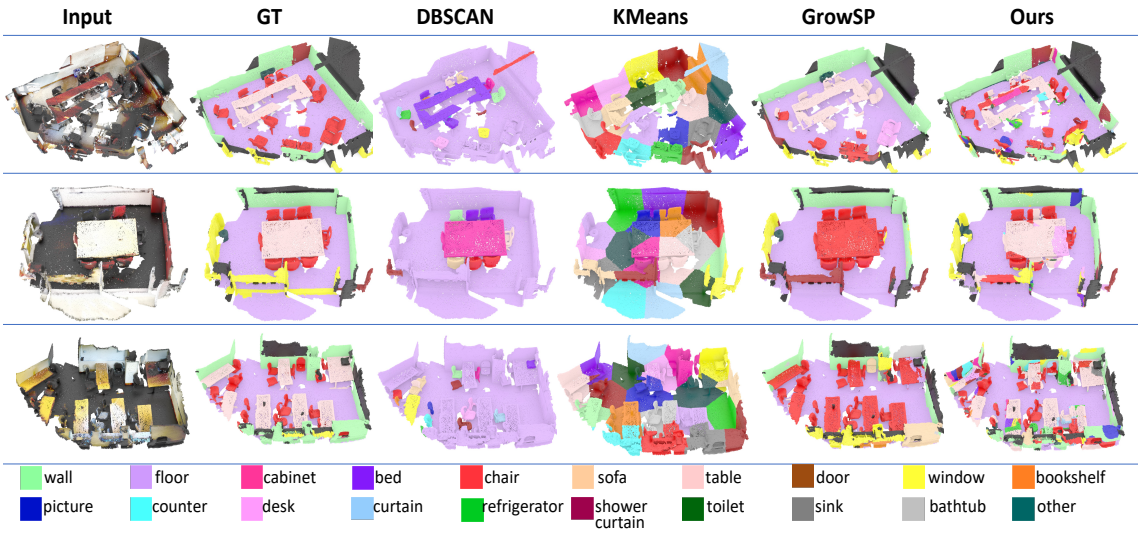

Figure 3.3: Qualitative results on ScanNet [2]. Each class label is assigned a colour (as per legend, right). This illustration shows superior segmentation performance compared to the baselines.

colour information (R, G, B), per-point normals and normalized raw coordinates in the original scene coordinate system. Note that no colour information is provided in SemanticKITTI [3].

Training: We use a batch size of 4 with 4096 points per batch for all datasets. The chosen optimizer is stochastic gradient descent (SGD) with a learning rate of $1e-4$ and a weight decay of $1e-5$. We train our network for 10 epochs. For the geometric transformation in the volumetric domain, we reverse the order of tensors along the given x, y and z-axis respectively. The colour transformation comprises random contrast and random brightness adjustment. The output feature dimension from the model and the clustering feature dimension is set to 128. The resolution of the voxel grid is set to 32. Besides, we use the FAISS library [90] on GPU to compute the cluster centroids via employing a mini-batch k-means approach [88].

Evaluation: For evaluation and comparison with other methods, we choose two classical unsupervised clustering methods, k-means [5], DBSCAN [4], and the only unsupervised semantic segmentation method GrowSP [78] as baselines. Our method is evaluated with three metrics: overall accuracy (oAcc), mean accuracy (mAcc) and the mean intersection of union (mIoU) on all datasets. All experiments are

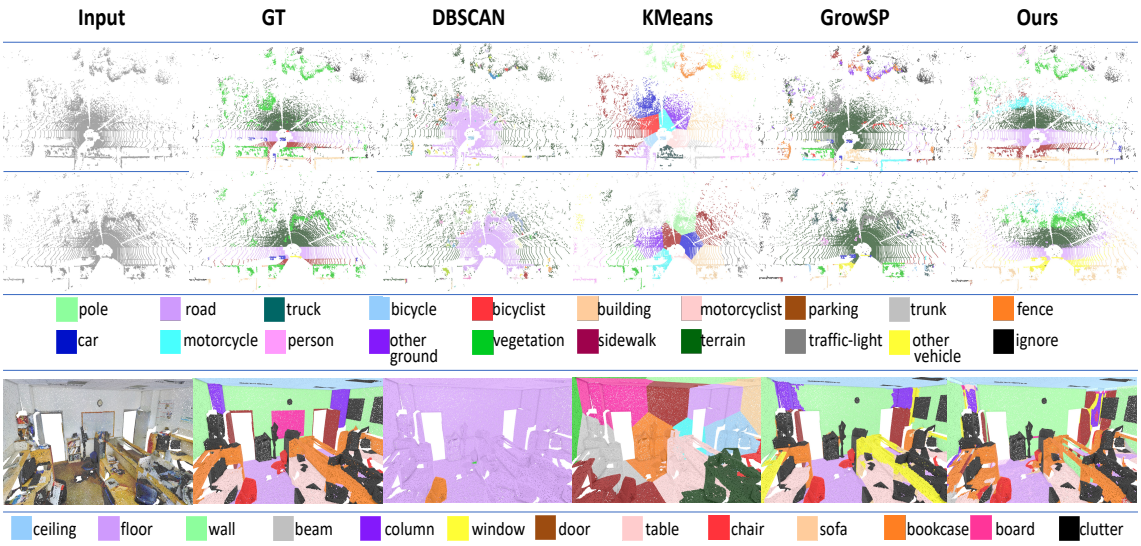

Figure 3.4: Qualitative results on SemanticKITTI [3] (Top 2 rows) and S3DIS [1] (bottom row). Our method draws more versatile results compared with DBSCAN [4] and is more stable than k-means [5], which shows promising segmentation results.

Method	Level of Supervision	mIoU	mAcc	oAcc
KMeans [5]	unsupervised	3.4	10.4	10.2
DBSCAN [4]	unsupervised	6.1	10.1	15.3
GrowSP [78]	unsupervised	25.4	44.2	57.3
$U3DS^3$**(ours)**	unsupervised	**27.3**	**46.8**	**60.1**

Table 3.1: Semantic segmentation results on ScanNet dataset. We evaluate 20 categories on validation set

performed on a single NVIDIA RTX 2080Ti GPU.

3.3.1 Datasets

We evaluate $U3DS^3$ on two indoor and one outdoor benchmark: S3DIS [1], ScanNet [2] and SemanticKITTI [3].

S3DIS [1] is a large-scale indoor scenes dataset which consists of 271 point cloud rooms in six areas. The annotations of each point in the point cloud scene belong to 13 semantic categories. We train the model in areas 1, 2, 3, 4, 6 and test it in area 5 following [7, 33, 91]. We exclude clutter and test with 12 classes for a fair comparison with GrowSP [78], nevertheless, we also test with 13 categories to compare with the

Method	Level of Supervision	mIoU	mAcc	oAcc
KMeans [5]	unsupervised	2.5	8.1	8.2
DBSCAN [4]	unsupervised	6.8	7.5	17.8
GrowSP [78]	unsupervised	13.2	19.7	**38.3**
$U3DS^3$	unsupervised	**14.2**	**23.1**	34.8

Table 3.2: Semantic segmentation results on SemanticKITTI dataset. We evaluate 19 categories on validation set

existing supervised, weakly, and semi-supervised methods.

ScanNet-v2 [2] is an RGB-D real-world indoor dataset. It contains 1201 scenes for training, 312 for validation, and 100 for online testing. For scene semantic segmentation, it has 40 classes and one unlabelled class for training and 20 classes and for testing. We compare with existing clustering and unsupervised methods on the validation set.

SemanticKITTI [3]: is a large-scale outdoor dataset that is based on the KITTI Vision Odometry Benchmark. For the semantic segmentation task, it provides 22 sequences with point-wise annotation of 19 classes. Each sequence contains a number of scene scans collected by the complete 360 field-of-view of the employed automotive LIDAR, where sequences 11-21 are used for online testing, 08 is the validation set and the others are training sets.

Data Preparation: For all datasets, we choose $\gamma_{sp} = 40$ as the superpoint number for each scene. We first apply uniform downsampling to S3DIS [1] and ScanNet [2] with the sub-grid size 0.03 and subsequently follow PointCNN [91] to sample point clouds into blocks to ensure that each data sample in the batch has the same number of points. For S3DIS [1] and ScanNet [2], the block size is 1.5×1.5 on xy plane, and each block contains 4096 points. For SemanticKITTI [3], we set each block size as 5×5 on xy plane with 4096 points. For each point cloud, we utilize VCCS [87] to derive the initial superpoint. This is then merged for enhanced segmentation, as detailed in Sec. 3.2.1. Furthermore, due to the characteristics and predominance of roads in outdoor SemanticKITTI [3] datasets, we apply RANSAC [92] to fit a plane as the road for improved generation of superpoints. Note this process will not be utilized elsewhere.

Method	Level of Supervision	mIoU	mAcc	oAcc
PTv2 [93]	fully supervised	72.6	78.0	91.6
KPConv [33]	fully supervised	67.1	72.8	-
SSP+SPG [94]	fully supervised	61.7	68.2	87.9
PointNet [7]	fully supervised	41.4	-	-
Jiang et al. [18]	semi-supervised (10%)	57.7	-	69.1
MT [95]	weakly supervised (1pt)	44.4	-	-
Zhang et al. [19]	weakly supervised (1pt)	48.2	-	-
KMeans [5]	unsupervised	9.4	21.2	22.1
DBSCAN [4]	unsupervised	9.2	19.8	17.5
GrowSP(12) [78]	unsupervised	**44.6**	**57.2**	**78.4**
$U3DS^3$(ours)(12)	unsupervised	42.8	55.8	75.5
$U3DS^3$(ours)	unsupervised	40.1	52.9	72.3

Table 3.3: Semantic segmentation results on S3DIS Area-5 Evaluations are compared using mIoU, mAcc and oAcc across various methods. Where (12) indicates the exclusion of clutter, while the results without (12) are tested with 13 classes.

3.3.2 Results and Comparison on Benchmarks

To thoroughly evaluate our $U3DS^3$, we test our methods on the indoor S3DIS [1], ScanNet [2] and outdoor SemanticKITTI [3] benchmarks. Tab. 3.1 to Tab. 3.3 respectively shows the semantic segmentation results on the ScanNet, SemanticKITTI and S3DIS dataset. Not surprisingly, fully supervised methods provide the best performance. From Tab. 3.3, our method significantly outperforms the existing clustering methods, where it achieves 75.5% overall accuracy and 42.8 mIoU on the S3DIS dataset. Moreover, our method is even close to the performance reported by the fully supervised method [7] and some up-to-date weakly supervised methods [42, 95], which is a big step forward for unsupervised semantic 3D scene segmentation.

Moreover, we outperform GrowSP [78] on both the ScanNet and SemanticKITTI datasets. Specifically, as displayed in Tab. 3.1, our method achieves a superiority of +1.9 mIoU and +2.6 mAcc over their results. Additionally, Tab. 3.2 demonstrates that our method achieves 1 mIoU and 3.4 mAcc higher than GrowSP [78], despite having a slightly lower oAcc. Fig. 3.3 shows the qualitative comparison on S3DIS, which further demonstrates the superiority of our method.

Baseline	Eqv	Inv	γ_{sp}	mIoU	mAcc	oAcc
✓				29.8	42.5	55.3
✓		✓		30.7	43.5	57.2
✓	✓			33.9	45.9	61.4
✓	✓	✓		34.8	46.3	63.2
✓	✓	✓	80	38.8	49.7	68.7
✓	✓	✓	60	41.0	52.6	72.4
✓	✓	✓	40	**42.8**	**55.8**	**75.5**
✓	✓	✓	20	41.9	53.9	74.3

Table 3.4: Ablation study on S3DIS Area-5 Eqv denotes equivariant voxelized feature transformation; Inv denotes invariant colour transformation. γ_{sp} denotes the final superpoint number.

3.3.3 Ablation Study

To showcase the effectiveness of each module and the different volumetric transformations. We conduct eight groups of experiments on the S3DIS [1] dataset: (1) the baseline approach proposed in Sec. 3.3.3, (2) adding colour transformation on the basis of the control group (1), (3) adding voxelized feature transformation on the basis of the control group (1), and (4) full model without prior superpoint, (5)-(8) different final prior superpoints as guidance. As shown in Tab. 5.2, our full model clearly outperforms the baseline on all of the evaluation metrics, benefiting from the delicate volumetric transformation design and superpoint prior. Groups (3) and (4) outperform by **+5** mIoU and **8** oAcc compared to the baseline. More interestingly, the improvement of adding the geometric transformation for equivariance is more significant than that of the invariance transformations, which is different from prior unsupervised learning work in the 2D domain [46, 83, 84]. It is known that point clouds essentially present much stronger geometric priors than 2D images with explicit 3D structures, which we believe can significantly help the 3D representations to be more robust and consistent cross-view and less sensitive to light changes and jittering. Moreover, the employment of superpoints can significantly enhance the overall performance. This enhancement is a result of the more abundant information of superpoints, which facilitates the pre-segmentation of the scene into higher-level semantic classes. Additional results are available in the supplementary material.

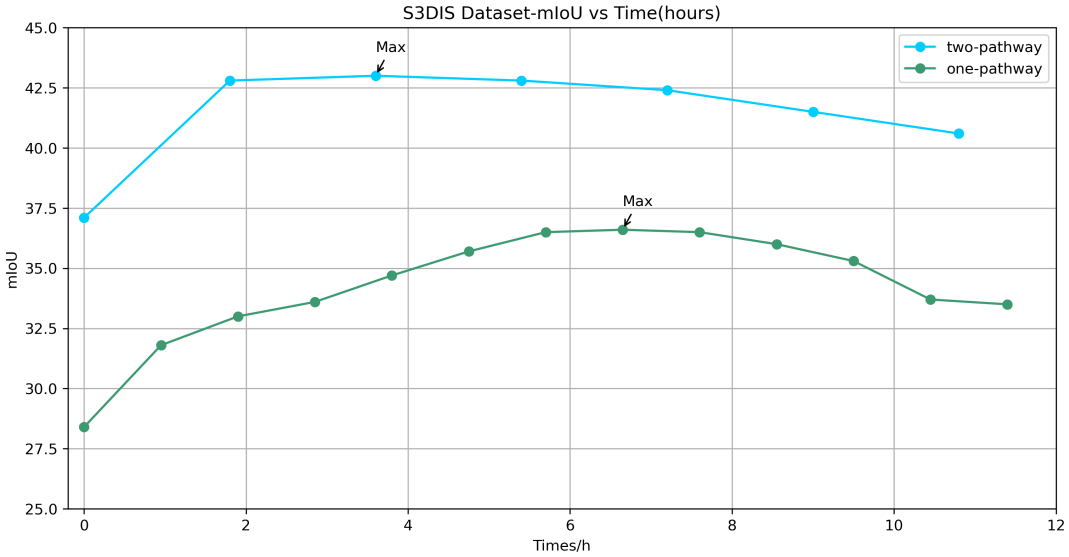

Figure 3.5: Convergence figure Demonstrates that the two-pathway method can accelerate the convergence.

3.3.4 Analysis

Our $U3DS^3$ approach demonstrates a promising level of performance on both indoor and outdoor datasets when compared to existing baselines. In contrast to GrowSP [78], our method achieves superior results on ScanNet [78] and SemanticKITTI [3]. As the scene complexity increases, the quality of GrowSP [78] superpoints tends to degrade. In contrast, our approach not only incorporates pre-segmentation but also employs a two-pathways training algorithm, leveraging the concepts of invariance and equivariance.

Nonetheless, slight performance degradation can occur in practical scenarios. To address this, we have implemented three strategies: (i) splitting the largest cluster when another cluster in the set reaches zero entities; (ii) applying mild centroid perturbation during updates; and (iii) re-weighting for loss balancing using per-class pseudo-label ratios at each epoch. Additionally, our two-pathways approach expedites the convergence time during training. For instance, while training with only one pathway necessitates around 8 epochs to achieve convergence, the two-pathways approach accomplishes convergence in just 2-3 epochs.

3.4 Conclusion and Discussion

We propose a novel generalized unsupervised semantic segmentation method for both indoor and outdoor 3D scenes with objects and the background. Our method leverages a simple yet effective framework via clustering and iterative generation leveraging the invariance and equivariance of the volumetric representations with the assistance of superpoint. Experiments show promising performance on S3DIS, ScanNet and SemanticKITTI datasets which proves the superiority of our approach beyond all the existing baselines. This work aims to provide more insight for 3D unsupervised learning. Future work will explore improved point sampling strategies and an extension to point- or graph-based representations, benefiting other areas related to unsupervised learning, metric learning and 3D representation learning.

CHAPTER 4

Time-Variant Frequency-Based Point Cloud Generation with Mamba

4.1 Introduction

Point clouds are favored in 3D tasks for their fidelity, ease of acquisition, and simple structure. Their generation is increasingly vital in applications such as VR, robotics [28], mesh modeling, and scene reconstruction [16, 17, 96]. However, unlike continuous 2D images [60, 97], point clouds are inherently discrete and unordered, posing unique challenges for generative modeling.

Existing generative models targeting point clouds span a wide range of methods, including variational autoencoders (VAE) [21], generative adversarial networks (GAN) [31], and normalizing flows [23, 98]. However, these methods often face challenges in achieving stable and high-fidelity generation, limiting their effectiveness in complex 3D point cloud tasks. Recently, denoising diffusion models [32] have demonstrated superior 3D point cloud generation performance, by defining a forward process that gradually perturbs the point cloud into standard Gaussian noise, and then learns to recover that original cloud through a reverse denoising process. Once trained, new point clouds can be generated by directly sampling from the Gaussian

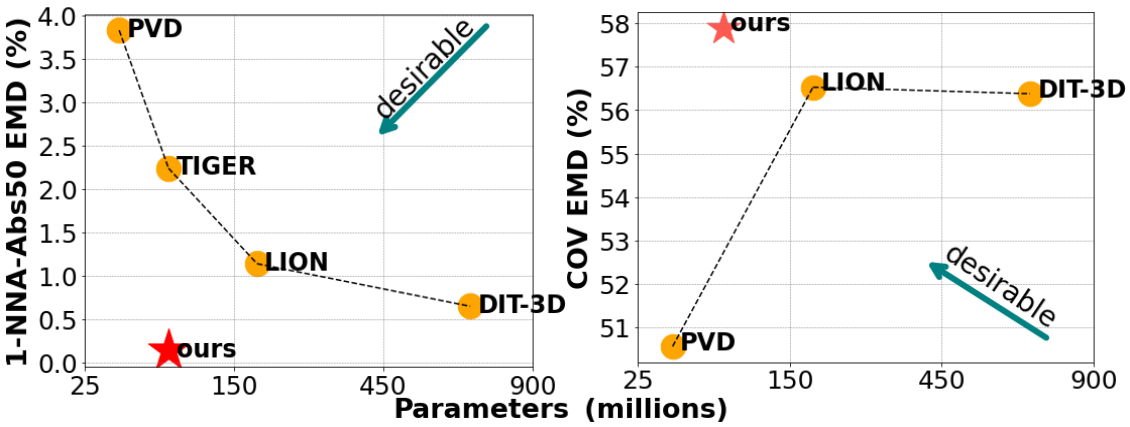

Figure 4.1: 1-NNA-Abs50 EMD & COV EMD (Sec. 4.3) performance (%) *vs.* parameter size (millions) on ShapeNet-v2 Car category. For 1-NNA-Abs50 EMD (left), lower value indicates better generation quality and fidelity. For COV EMD (right), higher is better diversity. In both plots, moving left along the horizontal axis denotes smaller model.

distribution and using the same progressive denoising process, offering a more robust and accurate approach to 3D point cloud generation. Despite these advantages, the complexity of diffusion models imposes high computational demands, making scalability challenging for efficient point cloud generation.

Recent advances in sequence modeling—particularly the Mamba architecture [74] leveraging state space models for effective sequence handling, have demonstrated significant potential. Studies [99, 100] demonstrate Mamba's effectiveness in point cloud tasks. Meanwhile, diffusion models continue to show promise for 3D generation [25, 32], with additional work exploring transformer-based networks [24, 53] and latent space representations [52, 54] to further enhance model expressiveness.

Beyond architectural design, we examine the sequential nature inherent in diffusion modeling. Because each time step depends on the previous one, this process naturally imposes a sequential structure across time steps; moreover, within each denoising step, the model must capture local and global dependencies to maintain shape coherence. This recursive temporal dependency structure aligns well with the sequence modeling capabilities of state space models such as Mamba. Furthermore, point clouds can be viewed as discrete samples from continuous geometric manifolds. State space models, especially Mamba, approximate continuous-time dynamical systems through

structured recurrence, offering a mathematically grounded approach to modeling such geometry-aware sequences. These properties make Mamba not merely an efficient alternative to transformers, but a theoretically sound modeling choice for 3D point cloud diffusion. Moreover, diffusion models require substantial computational resources, particularly due to the multiple iterations involved in the reverse process, which makes both training and inference highly time-consuming. These demands are especially prohibitive in real-time or resource-constrained environments. Whilst Mamba is more computationally lightweight than a transformer architecture, its integration with diffusion processes for 3D point clouds remains underexplored due to these challenges. To address these limitations, we integrate a state space model into point cloud diffusion. Mamba captures long-range dependencies with high efficiency, which is essential for iterative refinement over large 3D spaces. Compared to transformers, it achieves lower computational cost while maintaining strong modeling capacity [73]. To handle the unordered nature of point clouds, we apply space-filling curve serialization [99], preserving local geometry. This enables Mamba to model both global geometry and fine-grained details throughout the diffusion process.

Recent work has explored frequency analysis in both 2D and 3D [63, 65, 66, 101], including its integration with diffusion models [63, 64] and with Mamba architectures for 2D images [102]. Some studies [103] have combined frequency analysis and Mamba in 2D diffusion. However, extending this to 3D point clouds remains challenging: unlike 2D data where Fourier or spectral methods apply directly, point clouds are sparse and discrete, making frequency decomposition within state-space models considerably harder.

To address these aforementioned research gaps, we propose TFDM, a novel point cloud diffusion architecture integrated with the Mamba framework. Given the high computational and memory demands of modeling long sequences, we improve efficiency by introducing dual latent Mamba blocks (DM-Block) in the latent space. This novel design reduces model size while preserving performance, offering a more compact yet effective approach compared to conventional Mamba-based architectures. Furthermore, we draw inspiration from 2D image diffusion, where coarse structures (low-frequency

components) are recovered at an early stage with refined details (high-frequency components) recovered later. Such pattern also extends to 3D point clouds: the generation process initially forms a blurred overall shape before refining contours, where high-frequency components are related to edges and corners, and flat regions to low-frequency features. Based on this observation, we emphasize high-frequency areas in later time steps. Specifically, by employing a U-Net architecture with multiple downsampling layers, we propose a time-variant frequency-based encoder (TF-Encoder), replacing traditional farthest point sampling with our frequency-based method to better select keypoints in later time steps, thereby capturing more details during the final recovery.

Overall, our contributions can be summarized as follows:

- The first joint use of frequency-based analysis for Denoising Diffusion Probabilistic Models (DDPM) combined with the use of Mamba architecture, to address the computational demands of 3D point cloud diffusion modeling.
- A novel end-to-end architecture (**TFDM**) that integrates a promising **T**ime-variant **F**requency-based Encoder (TF-Encoder) with **D**ual latent **M**amba Block (DM-Block) to enhance high-frequency point cloud details. It adapts to the diffusion timestep within the Mamba latent space of a point cloud DDPM, ensuring precise detail refinement.
- Extensive experiments on the established ShapeNet-v2 [104] benchmark dataset that demonstrates both state-of-the-art (SoTA) performance (ShapeNet-v2: 0.14% on 1-NNA-Abs50 EMD and 57.90% on COV EMD) and the efficacy (reducing up to 10× and 9× on parameters and inference time) of our approach across multiple reference categories.

4.2 Methodology

We begin by introducing corresponding background Sec. 4.2.1, then formulating the generative diffusion objective in Sec. 4.2.2. Building on the robust 3D modeling capacity of Mamba blocks, we propose a novel diffusion framework for point clouds

(see Fig. 5.2). Specifically, in Sec. 4.2.3 we introduce a frequency-based point cloud filter to extract key frequency components. In Sec. 4.2.4, we describe a time-variant frequency encoder that uses these components for key-point sampling. Finally, in Sec. 4.2.5, we present our two-stream latent Mamba architecture, which integrates state space modeling, frequency analysis, and diffusion to generate high-fidelity point clouds efficiently.

4.2.1 Background

Denoising Diffusion Probabilistic Model For given samples $x_0 \sim q(x_0)$, the diffusion model gradually reverses a Markovian fixed forward diffusion process:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \tag{4.1}$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}), \tag{4.2}$$

where T denotes the time step, $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the transition kernel progressively perturbs the input with a sequence of pre-defined variance schedule $(1-\alpha_1), ..., (1-\alpha_T)$.

The reverse process is parameterized as a Markovian chain $p_\theta(\mathbf{x}_{0:T})$ which is equal to $p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$,

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}), \tag{4.3}$$

where $p(\mathbf{x}_T)$ is standard Gaussian and $\mu_\theta(\mathbf{x}_t, t)$ is the learnable object, with setting σ_t^2 as a fixed variance schedule. This object is optimized by matching the ground truth denoising step, which can be interpreted as learning the source noise ϵ_0 by minimizing $w(t) \parallel \epsilon_\theta(x_t, t) - \epsilon_0 \parallel_2^2$, where $w(t)$ is a parameter only depends on timestep. The optimization objective thus becomes:

$$\mathcal{L} = \mathbb{E}_{t\sim[1,T]} w(t) \parallel \epsilon_\theta(\mathbf{x}_t, t) - \epsilon_0 \parallel_2^2 \tag{4.4}$$

After training, generation can be achieved via the inverse chain by sampling from a standard Gaussian distribution.

State Space Model The State Space Model (SSM) [74] can be described as a continuous system that maps a 1-D function or sequence $x(t)$ to $y(t)$, with medilated through a N-D latent state $h(t)$.

$$h^{'}(t) = \mathbf{A}h(t) + \mathbf{B}x(t), y(t) = \mathbf{C}h(t), \tag{4.5}$$

where $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ are parameters can be learned via gradient descent. Mamba [74] improved the SSM by relaxing the time-invariance constraint and discretize the formulation via a timescable transformation parameter Δ. By using zero-order hold techniques, the parameters can be defined as:

$$\overline{\mathbf{A}} = \exp(\Delta\mathbf{A}), \overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}\left(\exp(\Delta\mathbf{A}) - \mathbf{I}\right)\Delta\mathbf{B} \tag{4.6}$$

Subsequently, eq. (4.6) can be discretized and is able to compute the outputs as specific time step:

$$h^{'}(t) = \overline{\mathbf{A}}h(t) + \overline{\mathbf{C}}x(t), y(t) = \mathbf{C}h(t) \tag{4.7}$$

Finally, it employs a structured global convolution to enhance computational efficiency:

$$\overline{\mathbf{K}} = (\mathbf{C}\overline{\mathbf{B}},\ \mathbf{C}\overline{\mathbf{A}\mathbf{B}},\ \ldots,\ \mathbf{C}\overline{\mathbf{A}}^{M-1}\overline{\mathbf{B}}), \quad y = x * \overline{\mathbf{K}}, \tag{4.8}$$

where M and $\overline{\mathbf{K}}$ represent individually the length of sequence x and the kernel of the global convolution.

Graph Filter Given a graph $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{A}^u)$ let $\mathcal{V} = v_1, ..., v_N$ denote a set of N nodes and $\mathbf{A}, \mathbf{A}^u \in \mathbb{R}^{N\times N}$ represent the weight and unweight adjacency matrix. We refer to one-channel features on all nodes to be a graph signal $\mathbf{s} \in \mathbb{R}^N$. $\mathbf{A}$ has eigen decomposition $\mathbf{A} = V\mathbf{\Lambda}V^{-1}$ where the matrix V contains eigenvectors of $\mathbf{A}$ and $\mathbf{\Lambda}$ is diagonal eigenvalue matrix corresponding to ordered eigenvalues $\lambda_1, ..., \lambda_N$.

As stated in [62], the ordered eigenvalues represent frequencies on the graph. Consider $\mathbf{A}$ as a graph shift operator and take a signal $\mathbf{s}$ to produce $\mathbf{y} = \mathbf{A}\mathbf{s}$, which is $V^{-1}\mathbf{y} = \mathbf{\Lambda}V^{-1}\mathbf{s}$. The graph Fourier transformation of graph signal $\mathbf{s}$ and $\mathbf{y}$: $\hat{\mathbf{s}} = V^{-1}\mathbf{s}, \hat{\mathbf{y}} = V^{-1}\mathbf{y}$ could be considered as frequency contents of signal $\mathbf{s}$ and $\mathbf{y}$.

Additionally, a graph filter is a polynomial in the graph shift [61]: $h(\mathbf{A}) = \sum_{l=0}^{L-1} h_l \mathbf{A}^l$, where h_l, L denote filter coefficients and the length of filter respectively. This filter takes signal $\mathbf{s}$ and generate $\mathbf{y} = h(\mathbf{A})\mathbf{s} = Vh(\mathbf{\Lambda} V^{-1}\mathbf{s})$, making $V^{-1}\mathbf{y} = h(\mathbf{\Lambda} V^{-1}\mathbf{s})$ then $\hat{\mathbf{y}} = h(\mathbf{\Lambda}\hat{\mathbf{s}})$. The diagonal matrix $h(\mathbf{\Lambda})$ is the graph frequency response of the filter $h(\mathbf{A})$ can be denoted $\hat{h}(\mathbf{A})$, and the frequency response of λ_i is $\sum_{l=1}^{L-1} h_l \lambda_i^l$.

4.2.2 Generative Modeling of Point Clouds

Given a point cloud $\mathcal{X} \in \mathbb{R}^{N\times 3}$ consisting of N points, our goal is to generate a high-fidelity point cloud from Gaussian noise $p(\mathbf{x}_T)$ by learning the transition probability $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Specifically, we model the mean of the transition distribution while keeping a predetermined variance throughout the diffusion reverse process. Similar to TIGER [24] and PVD [25], we employ a U-Net backbone for $\mu_\theta(\mathbf{x}_t, t)$ to incorporating a newly designed Mamba layer and frequency-based key point selection to enhance its capability. To sample the point cloud, we perform denoising from $p(\mathbf{x}_T)$ over T timesteps by minimizing the MSE discrepancy loss $\mathbb{E}_{t\sim[1,T]} \parallel \epsilon_\theta(\mathbf{x}_t, t) - \epsilon_0 \parallel_2^2$ between the predicted noise $\epsilon_\theta(\mathbf{x}_t, t)$ and the true noise ϵ_0, ensuring accurate denoising performance across different time steps.

4.2.3 Point Cloud Graph Filter

Unlike the 2D domain, where spectral analysis methods such as Fourier and wavelet transforms are straightforwardly applicable [65, 66, 69], the irregular, non-Euclidean nature of point clouds [105, 106] demands the development of alternative approaches for defining frequency components. The absence of a structured grid in point cloud data [29, 107] complicates the direct adoption of traditional spectral techniques, thus motivating a tailored method to effectively capture and process the inherent frequency characteristics of point cloud geometry.

Graph Construction: To capture the geometric structure in point clouds and topological relationships between points, we construct a k-nearest neighbors (k-NN) graph. The graph signals are further leveraged to extract high-frequency points with no trainable parameters.

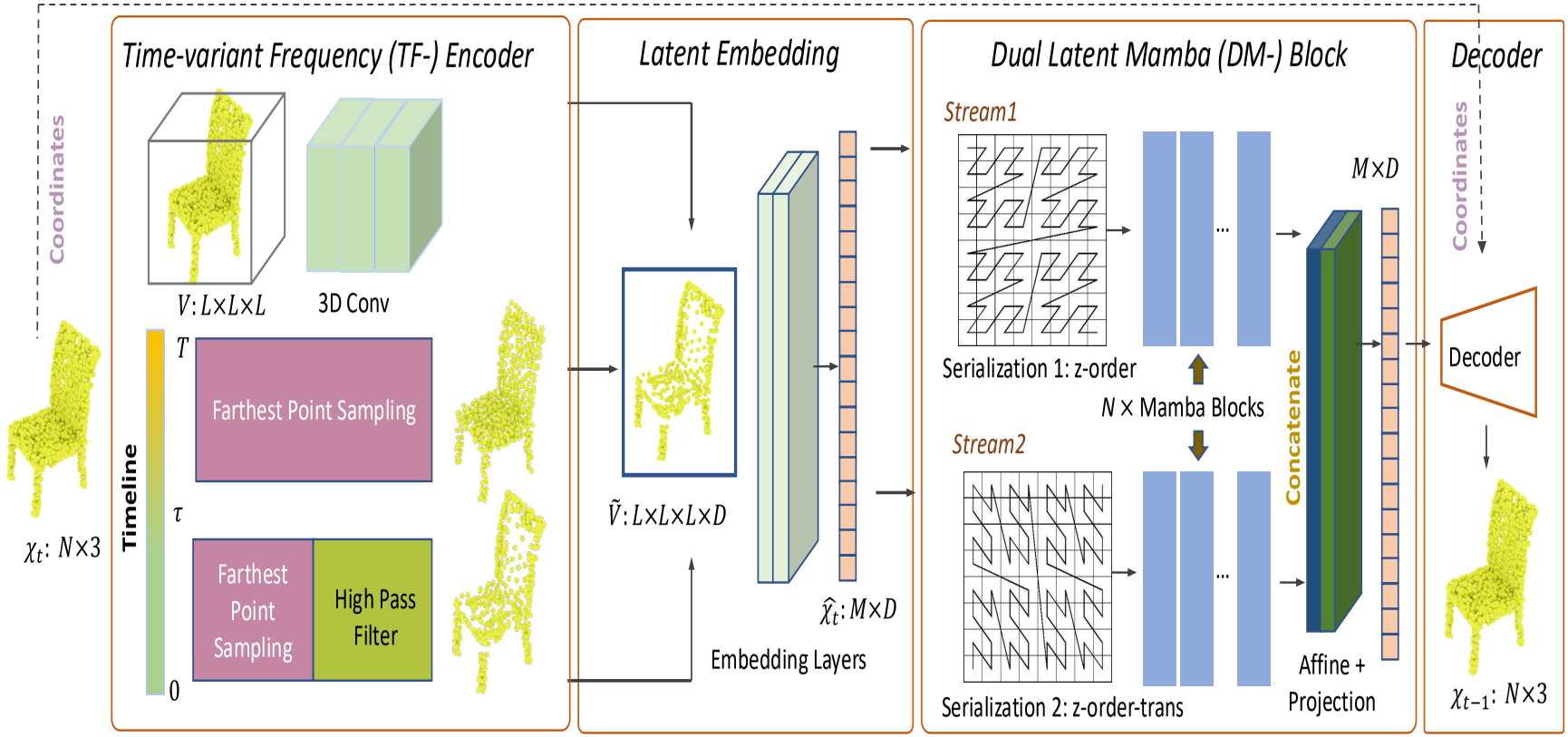

Figure 4.2: The overview architecture of our proposed TFDM. The network takes a point cloud at timestep t as input and aims to predict the noise component in $\mathcal{X}_t$ to obtain the point cloud at timestep $t-1$. Initially, the input point cloud is passed through a time-variant frequency-based encoder. This is followed by a latent embedding module that generates a latent point cloud $\hat{\mathcal{X}}_t$. The latent point cloud is then processed through Two-Streams Mamba blocks, which apply different serialization methods to extract diverse and complementary features. Subsequently, an affine transformation block is employed to align the latent point clouds from the different streams, ensuring consistency and integration of the extracted features. Finally, the aligned latent representation is decoded back into the 3D space.

Given a point cloud $\mathcal{X} = \{x_i, \ldots, x_N\}$ with corresponding d-dimensional features $\mathbf{f}_i \in \mathbb{R}^d$, $i \in \{1, \ldots, N\}$, we construct a k-NN graph $\mathcal{G} = (\mathcal{V}, \tilde{\mathcal{A}}_u, \tilde{\mathcal{A}}_w)$. Each point x_i corresponds to a node $v_i \in \mathcal{V}$, $\tilde{\mathcal{A}}_u$ and $\tilde{\mathcal{A}}_w \in \mathbb{R}^{N \times N}$ are normalized unweighted and weighted adjacency matrices encoding point dependency in feature space. The unweighted $\tilde{\mathcal{A}}^u_{ij}$ and weighted edges $\tilde{\mathcal{A}}^w_{ij}$ connecting two nodes v_i and v_j are defined as:

$$\begin{aligned} \tilde{\mathcal{A}}^u_{ij} &= \mathbb{1}(x_j \in \mathcal{N}(x_i)), \\ \tilde{\mathcal{A}}^w_{ij} &= \kappa(\|x_i - x_j\|^2) \cdot \tilde{\mathcal{A}}^u_{ij}, \end{aligned} \tag{4.9}$$

where $\kappa(\cdot)$ is a non-negative function, e.g., a Gaussian function, to ensure that $\tilde{\mathcal{A}}_w$ is a diagonally dominant matrix; $\mathcal{N}$ represents the neighborhood; $\mathbb{1}(\cdot)$ represents the indicator function which returns 1 if the specified condition (the function input) is true and 0 if it is false.

For efficiency considerations, although the Laplacian is defined over all points, it is constructed from a sparse k-NN graph, such that each node only connects to a small local neighborhood. In our framework, spectral analysis is performed only at a limited number of diffusion timesteps (less than 10%). Specifically, Laplacian eigendecomposition is applied selectively when frequency-based point ordering is required, rather than at every denoising step. As a result, this operation is invoked sparsely during diffusion and its computational overhead remains manageable, without dominating the overall runtime.

When the number of points becomes very large, a natural alternative is to perform spectral analysis on local subgraphs instead of the full point set. For example, one can compute Laplacian eigenmodes on spatially partitioned regions or local k-NN neighborhoods and aggregate the resulting frequency scores. Such localized Laplacian analysis preserves the ability to capture high-frequency geometric variations while further reducing computational cost, and can be readily integrated into our framework if needed.

Local curvature estimation via eigenanalysis of per-point covariance matrices provides an efficient way to capture neighborhood-level geometric variation. However,

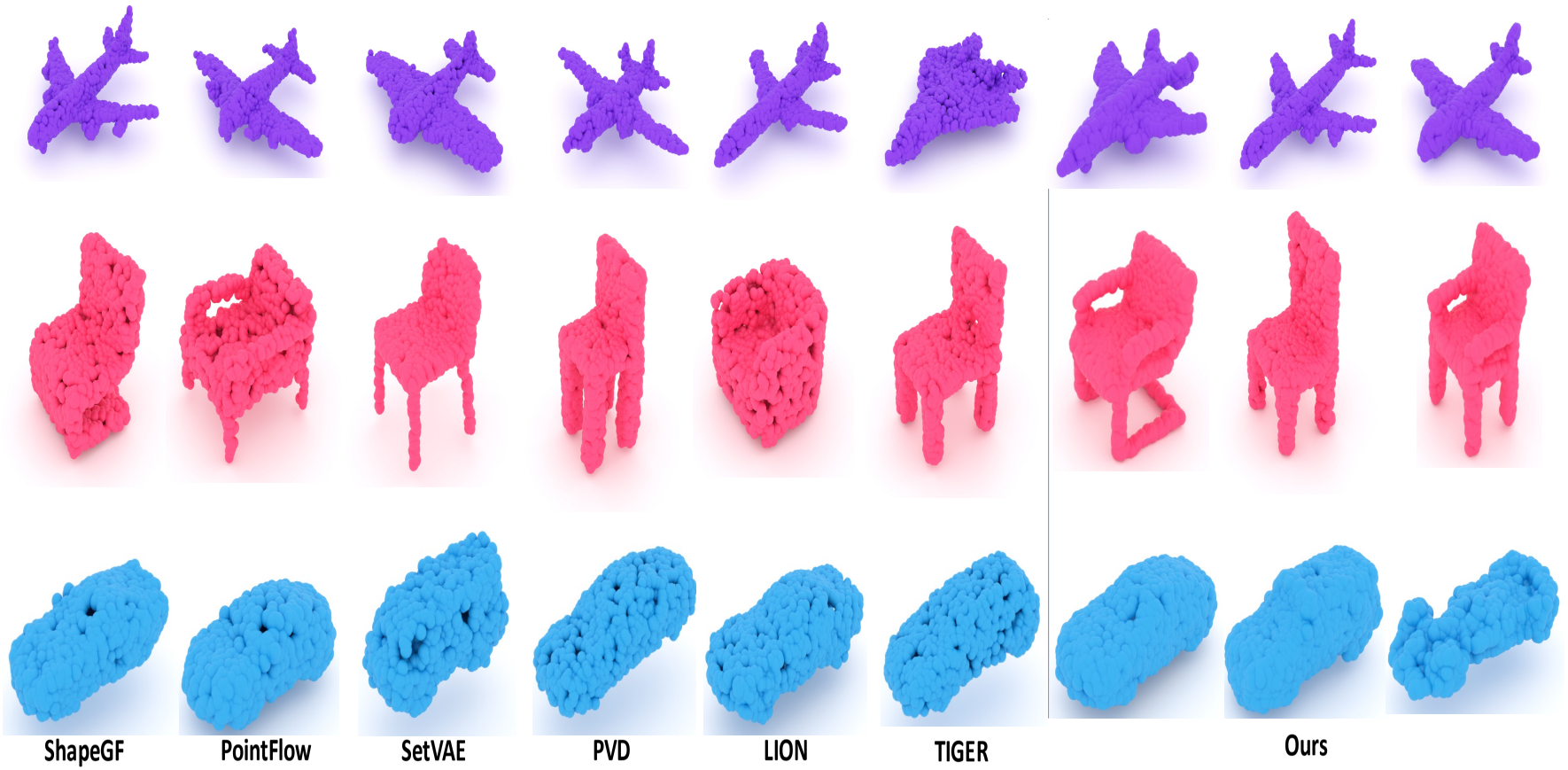

Figure 4.3: Qualitative results comparing our approach (right) with other leading contemporary approaches (left/middle). Our TFDM can generate high-quality and diverse point clouds. Three illustrative object categories $\{airplanes, chairs, cars\}$ are included here only.

such curvature measures are inherently local and depend strongly on neighborhood scale and sampling density. In contrast, Laplacian eigenmodes define a graph-spectral notion of frequency, characterizing how a point varies relative to the entire neighborhood graph. This global frequency perspective is better aligned with our objective of identifying high-frequency structural variations that are most informative for diffusion-based refinement, rather than relying solely on local surface curvature.

Point Cloud High-pass Filter: Our design of the point cloud high-pass graph filter draws insights from the 2D case, where high-frequency components, corresponding to sharp pixel variations like edges, elicit strong responses in the spatial domain. Following GDA [65], we construct our graph filter with the commonly adopted filter operator, specifically the Laplacian operator: $h(\tilde{\mathcal{A}}_w) = I - \tilde{\mathcal{A}}_w$. It takes a graph signal $\mathbf{s}_d \in \mathbb{R}^N, \forall d \in \{1, \ldots, D\}$ and produce filtered $\mathbf{y}_d = h(\tilde{\mathcal{A}}_w) \cdot \mathbf{s}_d \in \mathbb{R}^N$, then the frequency response of $h(\tilde{\mathcal{A}}_w)$ with associated λ_i is:

$$\widehat{h}(\tilde{\mathcal{A}}_w) = \text{diag}(1 - \tilde{\lambda}_1, 1 - \tilde{\lambda}_2, \ldots, 1 - \tilde{\lambda}_N), \tag{4.10}$$

where $\text{diag}(\cdot)$ denotes the diagonal matrix operator. The eigenvalues $\tilde{\lambda}_i$ are thus ordered reversely which represent the frequencies descending. As a result of the frequency response $1 - \tilde{\lambda}_i < 1 - \tilde{\lambda}_{i+1}$, the low frequency will be weakened which makes this to be a high-pass filter.

We apply the filter $h(\tilde{\mathcal{A}}_w)$ to the point cloud $\mathcal{X}$ to obtain the filtered point $h(\tilde{\mathcal{A}}_w)\mathcal{X}$, with each point computed as:

$$(h(\tilde{\mathcal{A}}_w)\mathcal{X})_i = x_i - \sum_{j}^{N} (\tilde{\mathcal{A}}_w)_{i,j} x_j. \tag{4.11}$$

It preserves the variation information with neighbors, as the filtered point in eq. (4.11) computes the difference between a point feature and the linear combination of its neighbor features.

Finally, we derive a frequency-based ordering for each point by computing the l_2-norm in Eq. (11). The top M points are then selected to capture the most dominant high-frequency components. This approach effectively integrates frequency

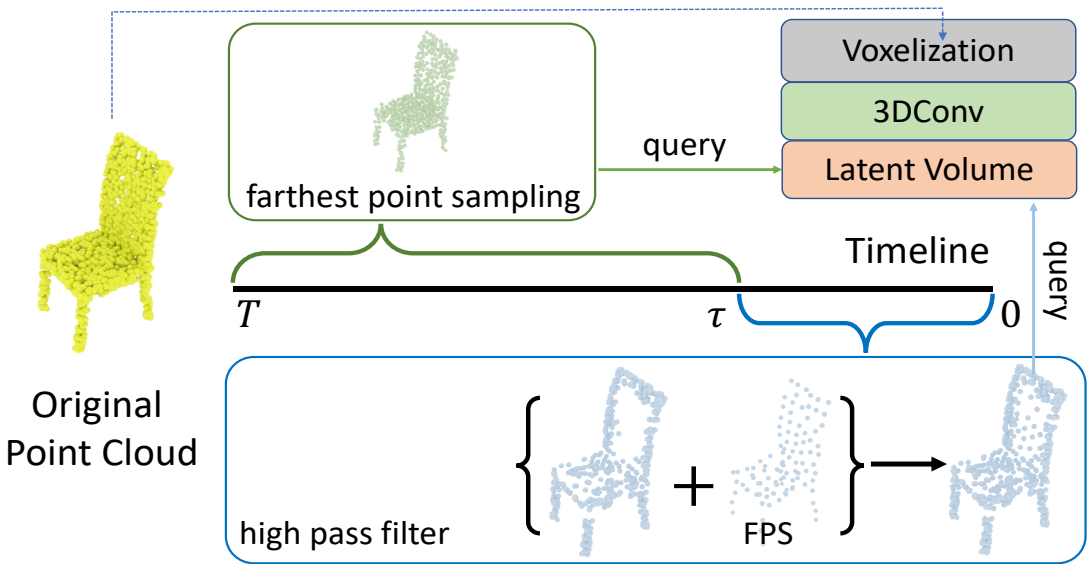

Figure 4.4: Illustration of the frequency key point selection. This process within the encoder to show how different strategies are applied across various timelines to obtain a downsampled point cloud. Subsequently, the downsampled point cloud is used to query the latent volume, resulting in the latent point cloud.

decomposition into the point cloud domain, despite its inherent irregularity.

4.2.4 Time-Variant Frequency Point Cloud Encoder

As shown in Fig. 4.4, we introduce TF-Encoder, a novel encoding mechanism designed to fully leverage the timestep-dependent recovery dynamics of the diffusion process. Unlike static sampling strategies, TF-Encoder adaptively refines point cloud representations by selectively emphasizing frequency information at different timesteps. The core insight is that diffusion first reconstructs coarse, low-frequency structures and progressively refines high-frequency details in later stages. To align with this progression, TF-Encoder dynamically adjusts the sampling process, allocating a greater "high-frequency budget" to later timesteps, where local details become most critical.

Voxel-Based Feature Extraction: Specifically, we utilize the PVCNN [39] backbone, which enables efficient computation by downsampling the point cloud into a voxel grid. For a point cloud at any time step t, denoted as $\mathcal{X}_t \in \mathbb{R}^{N\times 3}$, our TF-E $\mathcal{E}$ transforms the point cloud into a latent space $\hat{\mathcal{X}}_t \in \mathbb{R}^{M\times D}$, where $M < N$, represent-

ing the number of subsampled and original points, respectively. To aggregate the voxelized features, the point cloud $\mathcal{X}_t$ with normalized coordinates $c = (x, y, z)$ need to be voxelized into the voxel grids $\{\boldsymbol{V}_{m,p,q}\}$, where $\boldsymbol{V} \in \mathbb{R}^{L\times L\times L}$ with resolution L. The interpolated latent feature f_i for each voxel grid is computed as the mean of the features of the points within that grid:

$$\begin{aligned}\boldsymbol{V}_{m,p,q} = \frac{1}{K_{m,p,q}} \sum_{i=1}^{n} \boldsymbol{I}[\text{floor}(x_i \times r) = m, \\ \text{floor}(y_i \times r) = p,\ \text{floor}(z_i \times r) = q] \times f_i,\end{aligned} \tag{4.12}$$

where r denotes the voxel resolution and $\boldsymbol{I}$ is an indicator function that indicates whether coordinates c_i belong to the voxel grid $\{m, p, q\}$. $K_{m,p,q}$ represents the count of points falling within the grid $\{m, p, q\}$, and floor$(\cdot)$ is the floor function that outputs the greatest integer less than or equal to the input. After voxelization, multiple 3D convolutional layers with Swish activation [108] and GroupNorm [109] are applied to obtain the latent volume $\tilde{\boldsymbol{V}} \in \mathbb{R}^{L\times L\times L\times D}$ with D channels.

Time-Variant Frequency-Aware Sampling : Unlike standard furthest point sampling (FPS) pipelines in the PVCNN backbone, we jointly incorporate a high-pass graph filter (Sec. 4.2.2) with FPS in a time-variant manner. This design ensures that in the early time-steps $t < \tau$, we maintained a balanced selection of low-frequency structures and a subset of high-frequency regions for better global shape alignment. As the process advances to later time-steps $t \geq \tau$, our approach prioritizes high-frequency points, enabling the precise capture of subtle edges, corners, and intricate contours.

Formally, for M target points, we select ζM points with our graph-based high-pass filter, while the remaining $(1-\zeta)M$ points are sampled via FPS. As diffusion progresses, ζ can be adjusted to emphasize high-frequency details. This adaptive strategy allows TF-Encoder to align with the diffusion trajectory, ensuring time-specific frequency emphasis. The extracted point cloud is given by:

$$\mathcal{X}_t^* = \begin{cases} \zeta h(\tilde{\mathcal{A}}_w)\mathcal{X}_t + (1-\zeta)F(\mathcal{X}_t^{N-M})), & t = 0, 1, ..., \tau \\ F(\mathcal{X}_t), & t = \tau, ..., T \end{cases} \tag{4.13}$$

where $F(\cdot)$ represents the furthest point sampling and $\mathcal{X}_t^{N-M}$ denotes the original point cloud excluding the points that passed the high-pass filter.

Subsequently, we employ trilinear interpolation by querying the latent volume $\tilde{\boldsymbol{V}}$ with the sampled point cloud $\mathcal{X}_t^*$ to obtain the latent features $\hat{\mathcal{X}}_t$. The coordinates of both $\mathcal{X}_t^*$ and $\mathcal{X}_t$ are preserved for upsampling and positional embedding.

Note that under the PVCNN structure, there are four stages of downsampling, gradually reducing the point cloud from 2048 to 1024, 256, 64, and finally 16 points. The subsampling strategy described in eq. (4.13) is simplified after the first downsampling for efficiency. Since the points obtained from the high-pass filter are already in order, we can simply select the top percentage of points, while the rest are sampled using the furthest point sampling method.

4.2.5 Dual Latent Mamba Blocks

Although time-variant frequency emphasis helps refine point selection, directly applying a state space model to raw points in each timestep is computationally expensive, given the high dimensionality and unordered nature of point clouds [105–107]. To address this, we propose Dual Latent Mamba Blocks (DM-Block), which operates in a latent space and serializes the downsampled point set into a 1D sequence conducive to Mamba modeling. It is designed to preserve local neighborhood relationships through diverse space-filling curves and to capitalize on Mamba ability to handle long-range dependencies efficiently.

Space-Filling Curve Serialization: To improve the sequential modeling ability of DM-Block, we reorder the latent points using Hilbert and Z space-filling curves and their transposed versions (Trans-Hilbert and Trans-Z), maintaining spatial proximity in a sequence. This reordering preserves spatial proximity in the sequence, allowing DM-Block to better capture local correlations as neighboring points remain close in the serialized representation. Specifically, space-filling curves are paths that traverse every point within a higher-dimensional discrete space while maintaining spatial proximity to a certain degree and can be mathematically defined as a bijective function $\phi : \mathbb{Z} \to \mathbb{Z}^3$ for the point cloud. Given a space-filling curve $\mathcal{C}$, the latent point cloud $\hat{\mathcal{X}}_t$ is reordered according to its coordinates, resulting in the serialized

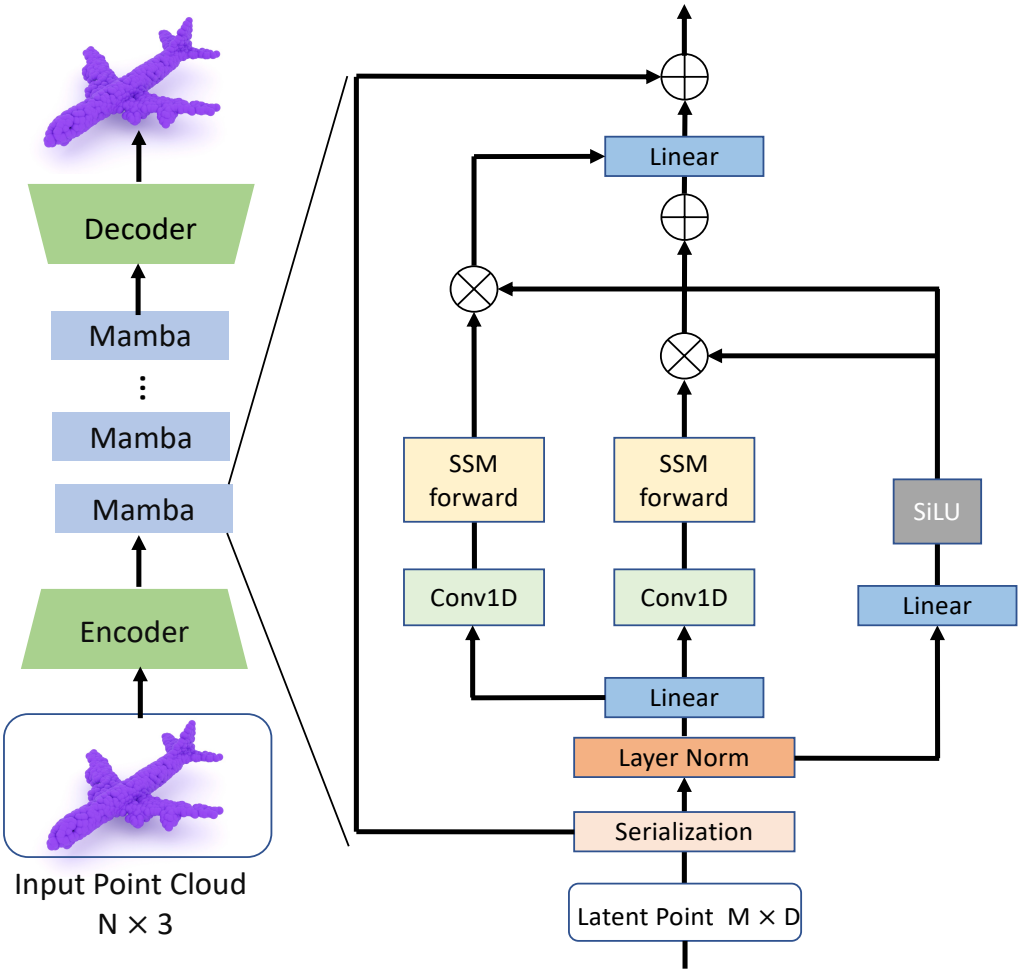

Figure 4.5: Illustration of our proposed Latent mamba block. Including Layer Norm, Linear Layer, forward and backward state space model with its corresponding Conv1D block (N.B. we only perform serialization at the first block).

latent point cloud as follows:

$$\hat{\mathcal{X}}_t^c = \mathcal{C}(\mathcal{X}_t^*)\hat{\mathcal{X}}_t, \quad \text{where } \hat{\mathcal{X}}_t^c \in \mathbb{R}^{M \times D}. \tag{4.14}$$

Bidirectional Latent Mamba: For better efficiency and expressiveness, we employ a bidirectional variant of Mamba to capture forward and backward dependencies along the serialized sequence of the serialized latent point cloud $\hat{\mathcal{X}}_t^c$. Specifically, for each Mamba block, as shown in Fig. 4.5, layer normalization [110], causal one-dimensional convolution, SiLU activation [111], and residual connections are employed. The serialized latent point cloud sequence $\hat{\mathcal{X}}_t^c$ is processed through multiple Mamba blocks.

Given an input $\mathcal{Z}_{l-1}$, the transformation in each block can be expressed as:

$$\begin{aligned}
\mathcal{Z}_{l-1}^{l} &= \mathrm{LN}(\mathcal{Z}_{l-1}), \\
\mathcal{Z}^{'} &= s(\mathrm{Linear}(\mathcal{Z}_{l-1}^{l})), \\
\mathcal{Z}_{l}^{f} &= \mathrm{SSM}_{\mathrm{forward}}(\mathrm{Conv1D}(\mathrm{Linear}(\mathcal{Z}_{l-1}^{l}))), \\
\mathcal{Z}_{l}^{b} &= \mathrm{SSM}_{\mathrm{backward}}(\mathrm{Conv1D}(\mathrm{Linear}(\mathcal{Z}_{l-1}^{l}))), \\
\mathcal{Z}_{l} &= \mathrm{Linear}(\mathcal{Z}^{'} \odot (\mathcal{Z}_{l}^{f} + \mathcal{Z}_{l}^{b})) + \mathcal{Z}_{l-1},
\end{aligned} \tag{4.15}$$

where s represents the SiLU activation function, and $\mathcal{Z}_l$ is the output of the l-th block. The Mamba output $\hat{\mathcal{X}}_{\mathrm{out}} \in \mathbb{R}^{M \times D}$ is obtained after passing stack of Mamba blocks.

Two Streams Affine Fusion: To further enhance representation power, we run two parallel streams with different space-filling orders (e.g., Z vs. Z-Trans), each capturing distinct structural cues. We then propose to fuse them with a simple learnable affine transform, aligning features from both streams. This yields an aggregated representation that retains global shape coherence and local detail sensitivity. Specifically, for the features output from different streams $\hat{\mathcal{X}}_{\mathrm{out}}^{c1}$ and $\hat{\mathcal{X}}_{\mathrm{out}}^{c2}$, we perform affine transformation as follows:

$$\hat{\mathcal{X}}_{\mathrm{out}}^{m} = \mathrm{Proj}((\hat{\mathcal{X}}_{\mathrm{out}}^{c1} \odot \gamma^{c1} + \delta^{c1}) \oplus (\hat{\mathcal{X}}_{\mathrm{out}}^{c2} \odot \gamma^{c2} + \delta^{c2})), \tag{4.16}$$

where $\gamma^{c1}, \gamma^{c2} \in \mathbb{R}^{D}$ and $\delta^{c1}, \delta^{c2} \in \mathbb{R}^{D}$ are scale and shift factors, respectively. The operator $\odot$ denotes element-wise multiplication, and $\oplus$ denotes concatenation. $\mathrm{Proj}(\cdot)$ represents a projection network that projects the concatenated features from $\mathbb{R}^{M \times 2D}$to $\mathbb{R}^{M \times D}$. Subsequently, the final output feature $\hat{\mathcal{X}}_{\mathrm{out}}^{m} \in \mathbb{R}^{M \times D}$ aggregates both global and local information.

Point Cloud Decoder: Finally, a point cloud decoder is applied to upsamples the latent point cloud to predict the noise ϵ_θ, thus completing our diffusion pipeline. As shown in Fig. 4.6, we employ trilinear interpolation to convert the latent point cloud $\hat{\mathcal{X}}_{out}^{m} \in \mathbb{R}^{M \times D}$ with the accompanying coordinates, to 3D space$\mathcal{X}_t \in \mathbb{R}^{N \times 3}$. Similarly to Sec. 4.2.3, we voxelize the $\hat{\mathcal{X}}_{out}^{m}$ into volume $\tilde{\boldsymbol{V}_{out}} \in \mathbb{R}^{L \times L \times L \times D}$ and following

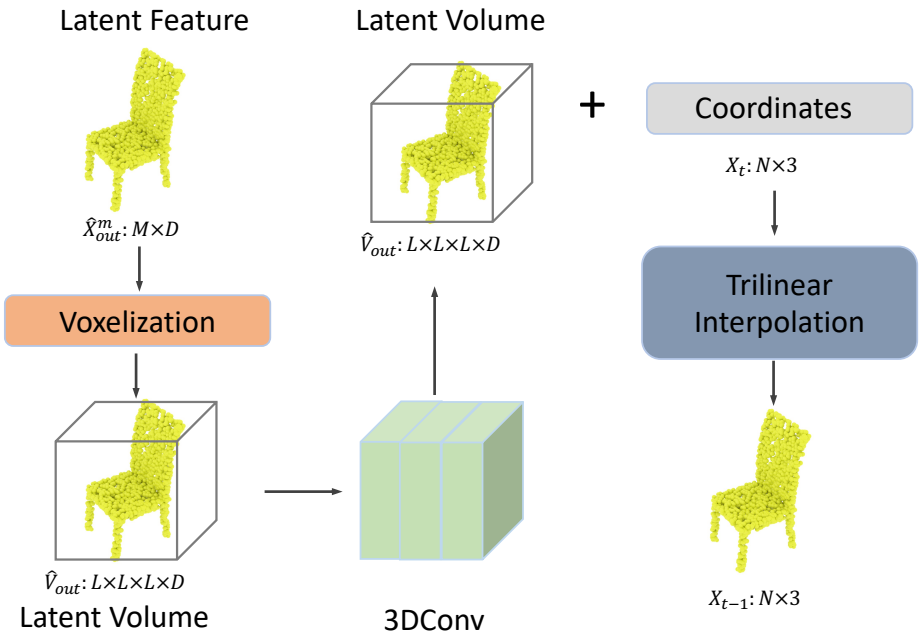

Figure 4.6: The overview of decoder. The final prediction X_{t-1} can be obtained by querying the latent volume V_{out} with the coordinates.

an additional 3D convolutional network while preserving the original shape, then query use $\mathcal{X}_t$, thereby obtaining the final prediction of the noise ϵ_θ. By adopting the TF-Encoder and DM-Block, we not only overcome the computational bottleneck of raw 3D data processing but also retain high-frequency details at the correct diffusion phase. This holistic design integrates time-variant frequency emphasis with state space modeling in a straightforward yet novel manner.

4.3 Experiments

We evaluate our proposed TFDM architecture against state-of-the-art 3D point cloud generation approaches on the established ShapeNet-v2 [104] benchmark dataset.

4.3.1 Experimental Setup

ShapeNet-v2 Benchmark Dataset: For a fair comparison on ShapeNet-v2, we follow the common practice that focuses on training and evaluating only select key categories, namely *chair*, *car*, and *airplane*. From each shape, we sample 2048 points

out of the 5000 available points in the training set and the test set, with normalization applied across the entire dataset. We adhere to the pre-processing steps and data split strategy as outlined in PointFlow [23].

Evaluation Metrics: Following the popular practice of prior work [25, 32], we use 1-NNA (and the derived 1-NNA-Abs50) and COV to evaluate generation quality and diversity, alongside CD and EMD, which measure point-wise and distributional differences:

- **1-NNA** (1-Nearest Neighbor) Accuracy: Measures the leave-one-out accuracy of a 1-NN classifier, reflecting both quality and diversity of generated samples.
- **1-NNA-Abs50** (Absolute 50-Shifted 1-NNA): Transforms the aforementioned 1-NNA x into $|x - 50|$, making it more sensitive to deviations from the ideal 50%; a lower score indicates an ideal generated distribution closer to real data.
- **COV** (Coverage): Evaluates how many reference point clouds are matched to at least one generated shape, where a higher value indicates greater diversity in generation.
- **CD** (Chamfer Distance): Measures point-wise similarity between generated and reference point clouds by computing the average nearest neighbor distance.
- **EMD** (Earth Mover's Distance): Captures the minimal cost of transforming one distribution into another, providing a global similarity measure between point clouds.

Implementation: For the frequency-based encoder, we set $k = 32$ for the k-NN graph, and the percentage ζ of high-pass points is set to 0.875. The diffusion model timestep is set to 1000, and the threshold τ for the sampling strategy is 50. In the Mamba layer, we apply 8 Mamba blocks for each stream with a latent size of 512. To enhance computational efficiency, the Mamba layer is applied to only 256 latent points. For training, we used an NVIDIA A100 GPU (80 GB) to train each category for 10,000 epochs with a batch size of 32. The learning rate was set to 0.0002, and we employed an Adam optimizer with a weight decay of 0.98.

Method	Chair				Airplane				Car			
	1-NNA-Abs50 (↓)		COV (↑)		1-NNA-Abs50 (↓)		COV (↑)		1-NNA-Abs50 (↓)		COV (↑)	
	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD
r-GAN [22]	33.69	49.70	24.27	15.13	48.40	46.79	30.12	14.32	44.46	49.01	19.03	6.539
PointFlow [23]	12.84	10.40	46.84	47.35	25.68	20.74	47.04	40.52	8.10	6.52	35.40	44.60
SoftFlow [98]	9.21	10.55	41.39	47.43	26.05	15.80	46.24	40.25	18.58	15.98	36.34	45.25
DPM [32]	10.05	24.77	44.86	35.50	26.42	36.91	48.64	33.83	18.89	29.97	44.03	34.94
PVD [25]	7.89	23.68	40.66	42.71	16.44	26.26	47.34	42.15	4.55	3.83	41.19	50.56
LION [52]	3.70	2.34	48.94	52.11	17.41	11.23	47.16	49.63	3.41	1.14	50.00	56.53
DiT-3D [53]	**0.89**	**0.73**	52.45	54.32	12.35	8.67	53.16	**54.39**	1.76	0.65	50.00	56.38
FrePolad [54]	3.53	3.23	50.28	50.93	15.25	12.10	45.16	47.80	1.89	0.26	50.14	55.23
TIGER [24]	4.61	2.71	-	-	21.85	5.82	-	-	4.31	2.24	-	-
NSO [112]	5.51	7.63	-	-	18.63	11.85	-	-	9.66	3.55	-	-
LPCG [113]	1.77	3.62	**56.39**	**58.38**	**11.43**	**3.47**	**55.94**	51.66	**1.69**	1.18	50.00	54.16
TFDM (ours)	3.25	1.68	49.84	54.98	18.31	8.88	51.38	52.25	4.21	**0.14**	**50.56**	**57.90**

Table 4.1: Comparison results (%) on ShapeNet-v2 with shape metrics Absolute 50-Shifted 1-Nearest Neighbor Accuracy (1-NNA-Abs50) and Convergence (COV), Chamfer Distance (CD) and Earth Mover's Distance (EMD), where CD is multiplied by 10^3 and EMD is multiplied by 10^2; – denotes unavailable result from original authors; **Best**/2nd best highlighted.

	Serialization Strategy	Freq Decom	Latent Block	CD ↓ (1-NNA-Abs50)	EMD ↓ (1-NNA-Abs50)	CD ↑ (COV)	EMD ↑ (COV)
(a)	None		Conv	9.24	5.95	45.26	50.43
(b)	None		Transformer	6.21	1.43	49.65	54.15
(c)	Hilbert		Mamba	5.98	1.10	49.10	54.21
(d)	Hilbert	✓	Mamba	4.76	0.85	49.99	56.10
(e)	Hilbert + Hilber-Trans		Mamba	4.64	0.64	50.11	55.73
(f)	Hilbert + Hilber-Trans	✓	Mamba	4.53	0.35	50.25	56.65
(g)	Z + Z-Trans	✓	Mamba	**4.21**	**0.14**	**50.56**	**57.90**

Table 4.2: Component-wise ablation of TFDM on ShapeNet-v2 (car category): latent block,serialization strategy, frequency-based component, and latent block.

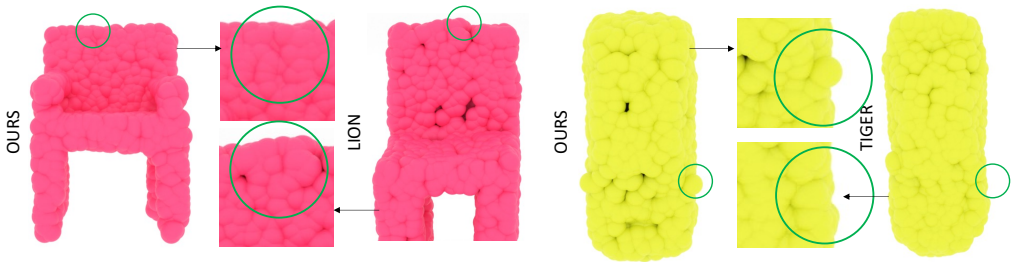

Figure 4.7: Details of Qualitative Result. Back of chair (Pink) - smooth(ours)/deformed(other) , Car side-view mirror (Yellow) - retained(ours)/missing(other)

4.3.2 Comparison with State-of-the-Art

Performance: In table 4.1, we compare TFDM with multiple point cloud generation approaches. Notably, TIGER (CVPR24) [24], FrePolad (ECCV24) [54], and DiT-3D (NeurIPS23) [53] are very recent methods. Among these, TIGER and FrePolad are relatively lightweight, yet we surpass both on the *chair* and *car* categories: for example, TFDM achieves a 0.25% improvement in 1-NNA CD and 0.98% improvement in 1-NNA-abs EMD on *chairs* compared to the better of the two, and a 0.12% gain on *cars*. DiT-3D, while offering strong performance, incurs extremely high computational overhead, requiring 1700 GPU hours and 711 million parameters. Even so, TFDM outperforms it on three out of four metrics for the *car* category, including a 0.51% gain in 1-NN EMD and 1.52% in COV EMD. These results highlight the efficiency and effectiveness of our approach.

Efficiency: As shown in table 4.3, our full model achieves the best results, requiring only slightly more training time than TIGER. Furthermore, compared to other top-performing methods, our approach significantly reduces training hours and parameter size while still achieving the highest overall performance. For further efficiency improvement, our single-stream variant achieves the lowest computational cost and fastest inference time, albeit with a slight performance trade-off compared to DiT-3D and our full model.

Multi-Class Generation: We train the TFDM model jointly without category conditioning on 10 object classes from ShapeNet-v2 (*cap, keyboard, earphone, pillow, bag, rocket, basket, bed, mug, bowl*). Training on such a diverse set of shapes poses

significant challenges due to the complexity and multimodal nature of the data. We present both qualitative (fig. 4.8) and quantitative (table 5.4) results. For comparison, we also train several baseline models under the same conditions, and the results demonstrate that our approach achieves the best overall performance across these methods.

Method	Para (M)	Training Time (h)	Inference Time (s)	EMD (1-NNA-Abs50) ↓
TIGER	<u>70.11</u>	<u>164</u>	<u>9.73</u>	2.24
DIT-3D	711.88	1688	100.13	<u>0.65</u>
LION	144.25	550	27.12	1.14
Ours (SS)	**48.84**	**138**	**8.12**	0.85
Ours (Full)	70.25	192	11.41	**0.14**

Table 4.3: Training time, inference time, model size and the corresponding evaluation results. For a fair comparison, we report these metrics on Nvidia V100 GPU with a batch size of 32. Training time and inference time, measured in GPU hours and second respectively, is averaged over three categories: chair, airplane and car. Where 'SS' indicates single-stream model.

4.3.3 Ablation Studies

In this section, we analyze the impact of various components and strategies within our proposed TFDM framework.

Different combinations of serialization methods: We further evaluate multiple combinations of serialization methods within the two-stream architecture to determine the most effective strategy for information flow between streams. Specifically, for the car category, the combination of z and z-transform serialization yields better performance than the combination of Hilbert and Hilbert transform. We compare row (f) with (e) in table 5.2, the combination of z and z-trans order performs better than another, which gets improvements of 0.32% in 1-NNA-Abs50 CD, 2.08% in 1-NNA-Abs50 EMD, 0.31% in COV CD, and 1.35% in COV EMD.

Effectiveness of frequency based model: We assess the impact of the frequency-based time-variant strategy by comparing models that incorporate this mechanism against those that do not. Our results reveal that incorporating frequency decomposition leads to further improvements across all metrics. Specifically, comparing

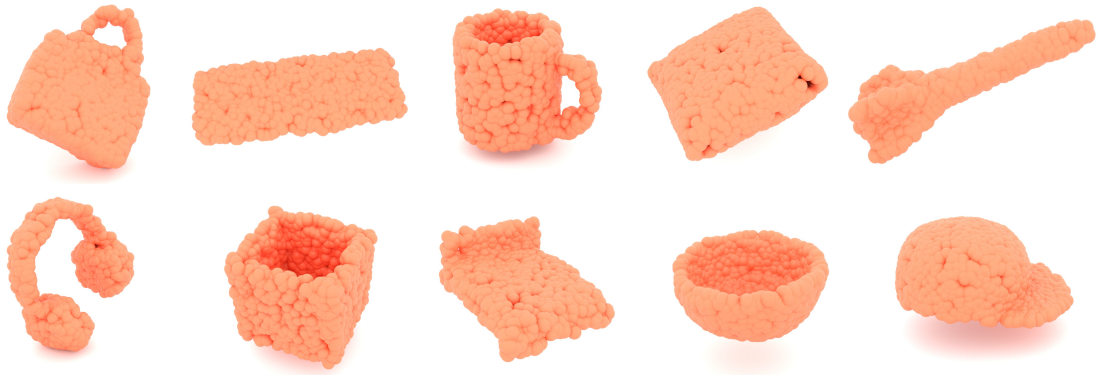

Figure 4.8: Qualitative results of our model jointly trained on ten categories, presented in the following order: *bag, keyboard, mug, pillow, rocket, earphone, basket, bed, bowl, and cap.*

Method	CD ↓ (1-NNA-Abs50)	EMD ↓ (1-NNA-Abs50)	CD ↑ (COV)	EMD ↑ (COV)
DPM	9.71	21.54	43.65	38.94
PVD	7.52	17.43	44.12	44.32
Tiger	0.88	0.98	56.25	57.64
Ours	**0.85**	**0.43**	**56.41**	**60.68**

Table 4.4: Comparison results (%) jointly trained on ten categories.

rows (d) and (e) in table 5.2, the frequency-based method achieves improvements of 0.09% in 1-NNA-Abs50 CD, 0.71% in 1-NNA-Abs50 EMD, 0.13% in COV CD, and 1.08% in COV EMD. Additionally, as shown in fig. 4.9, the diffusion model recovers finer details in the final timesteps, making it well-suited for frequency analysis.

Effectiveness of Mamba block: In table 5.2, comparing rows (a) and (b), we substitute the Mamba latent block (row, c) with standard 3D convolutional blocks (row, a) and Transformer block (row,b). The results demonstrate that substituting simple convolutional blocks with Mamba blocks significantly enhances both the quality and diversity of the generated outputs. Specifically, the Mamba blocks outperform the convolutional model by 3.26% and 4.85% in 1-NNA-Abs50 CD and EMD, respectively. Additionally, the Mamba block surpasses the Transformer block with 0.33% in 1-NNA-Abs50 EMD and 0.55% in COV CD while only containing half parameters of the Transformer block under the same conditions.

Effectiveness of the two-stream Mamba layer design: In table 5.2, we evaluate

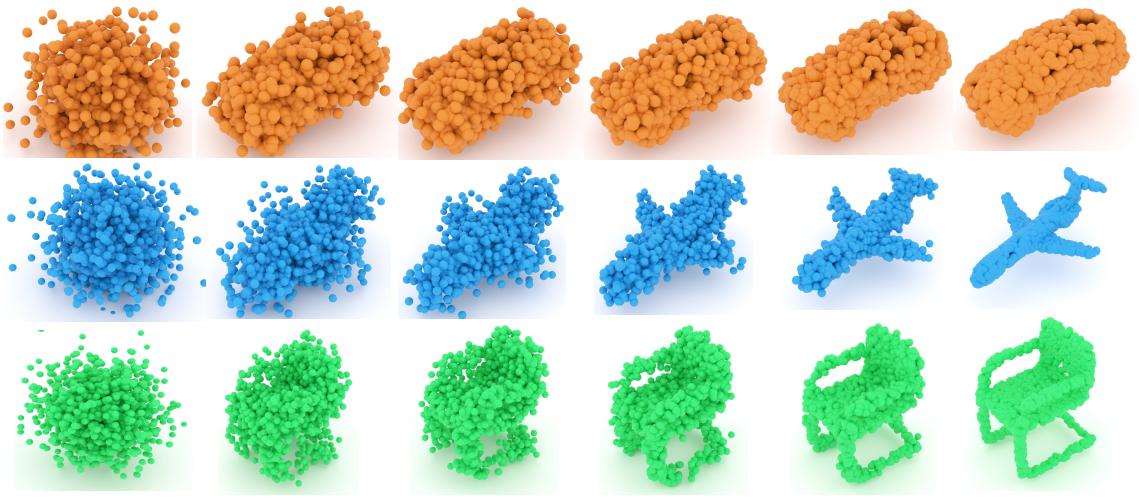

Figure 4.9: Illustrative examples of the reverse diffusion process demonstrating detailed information recovery at the final timesteps (left to right, timesteps progressing from T to 0).

	τ	ζ	CD ↓ (1-NNA-Abs50)	EMD ↓ (1-NNA-Abs50)	CD ↑ (COV)	EMD ↑ (COV)
(a)	25	0.875	3.54	1.99	49.01	53.99
(b)	50	0.875	3.25	1.68	49.84	54.98
(c)	50	0.75	4.15	2.37	48.93	53.46

Table 4.5: Ablations on hyperparameters. τ and ζ v.s. 1-NNA/COV.

the two-stream (row, d) versus the single-stream (row, b) architecture. table 5.2 demonstrates our two-stream architecture consistently achieves superior results compared to the single-stream, regardless of other components. Our two-stream design achieves improvements of 1.34% in 1-NNA-Abs50 CD, 1.09% in 1-NNA-Abs50 EMD, 1.01% in COV CD, and 1.52% in COV EMD.

Effectiveness of Hyperparameters: We also evaluate the impacts of hyperparamters τ and ζ, As shown in table 4.5 for chair category, the results indicates that $\tau = 50$ and $\zeta = 0.875$ yield the best performance. The complete results are provided in the Supplementary Material.

4.4 Summary

In this chapter, we propose a novel architecture that jointly leverages state-space models and frequency analysis within a point cloud diffusion framework for generative tasks. Our proposed DM-Block integrates latent space representations in the Mamba block to effectively address the challenge of efficiently applying diffusion via Mamba. Furthermore, we recognize that the diffusion process should recover fine-grained details during the final time steps. To this end, we introduce TF-Encoder, which includes a time-variant frequency-based point extraction method that achieves this without incurring high computational costs. Experimental results demonstrate that our method achieves state-of-the-art performance in certain categories while maintaining computational efficiency (with up to 10× less parameters and 9× shorter inference time than competitive approaches), and yields promising results across all categories. **Future Direction**: Our method effectively achieves high-quality and diverse point cloud generation through the integration of frequency analysis and Mamba in latent space. However, our current approach applies frequency analysis without a dynamic adaptation mechanism. A promising direction would be to integrate frequency analysis directly with the neural network for joint training.

CHAPTER 5

FLDCG: Frequency-Aware Latent Diffusion for 3D Point Cloud Generation

5.1 Introduction

Point clouds have emerged as a powerful representation in 3D data processing due to their high fidelity, ease of capture, and straightforward manipulation. 3D point cloud generation has gained increasing attention for its superior performance across various applications, including virtual reality, robotics [14], mesh generation, scene completion, and reconstruction [16, 17, 96]. Despite advancements in 2D image generation [60, 97], point clouds remain inherently discrete, unordered, and complex, presenting unique challenges that require further exploration to facilitate effective use with contemporary generative models.

Existing generative models targeting point clouds span a wide range of methods, including variational autoencoders (VAE) [21], generative adversarial networks (GAN) [22, 31], and normalizing flows [23, 51]. However, these methods often face challenges in achieving stable and high-fidelity generation, limiting their effectiveness in complex 3D point cloud tasks. Recently, denoising diffusion models [32] have demonstrated superior 3D point cloud generation performance, by defining a forward

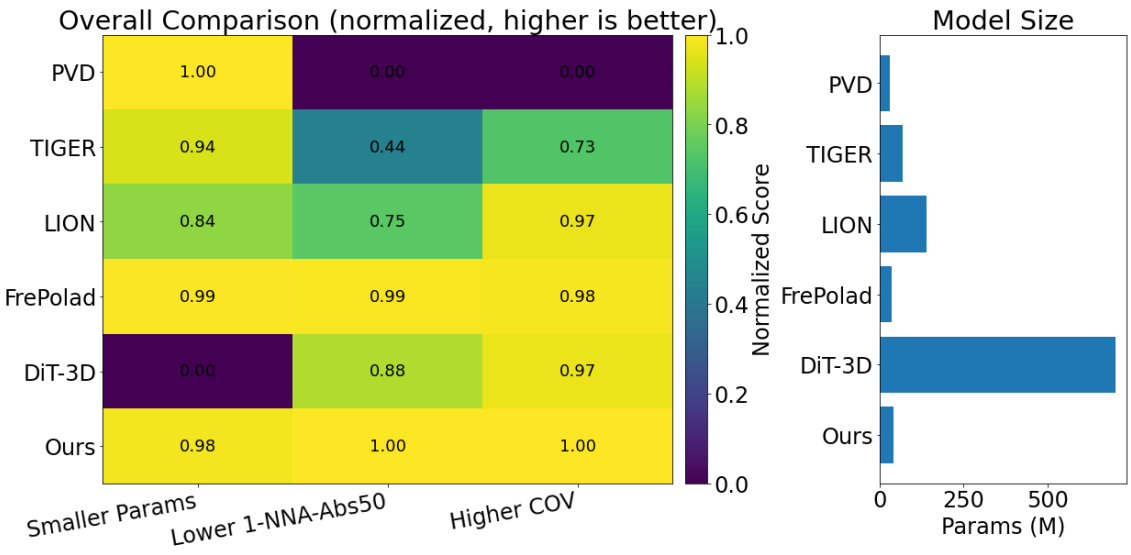

Figure 5.1: Normalized comparison of model efficiency and generative performance on ShapeNet-v2 (left). The heatmap summarizes three complementary aspects: model size (smaller parameters), generation quality and fidelity measured by 1-NNA-Abs50 EMD (lower is better), and diversity measured by COV EMD (higher is better). All metrics are normalized to a common scale where higher values indicate better performance. Darker cells therefore represent more favorable trade-offs. The accompanying bar chart reports the raw parameter counts (in millions), facilitating direct comparison of model complexity alongside normalized performance. Modelsize comparision among different methods (right).

process that gradually perturbs the point cloud into standard Gaussian noise, and then learns to recover that original cloud through a reverse denoising process. Once trained, new point clouds can be generated by directly sampling from the Gaussian distribution and using the same progressive denoising process, offering a more robust and accurate approach to 3D point cloud generation. Despite these advantages, the complexity of diffusion models imposes high computational demands, making scalability challenging for efficient point cloud generation.

Recent advancements in latent diffusion models (LDMs) [57] have demonstrated impressive performance in image generation tasks, primarily due to their scalability, improved training stability, and reduced computational cost compared to pixel-space diffusion models. By leveraging a pre-trained variational auto-encoder (VAE) to compress the input into a compact latent space, these models enable efficient and high-resolution generation while maintaining competitive visual fidelity. Inspired

by their success in the 2D domain, recent efforts have extended LDMs to the 3D domain, particularly for point cloud generation [52, 54]. In this context, the 3D data is similarly encoded into a latent space before applying the diffusion process, allowing for more tractable training and inference. However, this paradigm inherently relies on the quality of the VAE's latent representation. The compressed latent vectors often lack sufficient expressiveness to capture high-frequency geometric details and complex spatial structures in 3D data [114, 115]. As a result, existing 3D LDMs may suffer from degraded reconstruction quality or lose fine-grained surface information. This limitation motivates us to revisit the latent representation design for point cloud diffusion models.

Several recent works have investigated frequency-based representations [63, 65, 66, 101] in both 2D and 3D settings, demonstrating that frequency decomposition can enhance generative performance by isolating and emphasizing structural details across different spectral bands. In particular, some studies have explored the integration of frequency analysis with latent diffusion models [63, 64] and variational autoencoders (VAE) [116, 117], showing promising results in the 2D domain. However, extending such frequency-based approaches to 3D point clouds presents unique challenges. Unlike images, where spectral decomposition can be readily performed via Fourier or wavelet transforms, 3D point clouds are inherently sparse, irregular, and unordered. These characteristics complicate the application of classical frequency transforms and make it non-trivial to define or align spectral components consistently across different shapes. As a result, only a few works [54] have attempted to incorporate frequency-aware processing into point cloud diffusion models, and this remains an underexplored and technically demanding direction.

To address the aforementioned challenges, we propose FLDCG, a novel frequency-aware enhancement module for latent diffusion models. Specifically, we introduce a multi-band transformer architecture integrated within the VAE encoder to enrich the latent representation of 3D point clouds. Leveraging the Laplacian matrix derived from the graph process of point cloud, we compute a spectral decomposition to define frequency bands. Our key contribution lies in how these bands are utilized within the VAE: we design dedicated transformer branches for different frequency

components, allowing the model to capture both global structural information and high-frequency geometric details more effectively. The enhanced latent vector is then passed to a latent diffusion model (LDM) for point cloud generation, achieving strong performance with significantly reduced computational cost compared to previous approaches.

Overall, our contributions can be summarized as follows:

- We introduce a frequency-based multi-band transformer module within the VAE encoder, enabling frequency-aware latent representation learning for 3D point clouds.
- The resulting structured latent representation benefits the downstream latent diffusion model, improving generation quality while maintaining low computational overhead.
- We demonstrate that our method achieves state-of-the-art performance on the ShapeNet-v2 dataset in terms of both generation fidelity and sample efficiency.

5.2 Methodology

Our framework is motivated by the observation that point cloud generative modeling can benefit from a frequency-aware representation. We first compute a frequency score for each point to capture its structural importance. Based on these scores, points are sorted and partitioned into multiple frequency bands, each representing a different level of geometric detail. This multi-band representation allows the network to model coarse-to-fine structures explicitly. For each frequency band, we employ an independent Transformer encoder to capture intra-band dependencies, followed by attention pooling to aggregate its features. To adaptively control the contribution of each band, we introduce learnable frequency weights that are applied before concatenating the band features into a unified latent representation. This fused latent vector is then used as the input to the downstream latent diffusion model, enabling high-fidelity and detail-preserving point cloud generation. Our design combines frequency decomposition, Transformer-based feature extraction, and

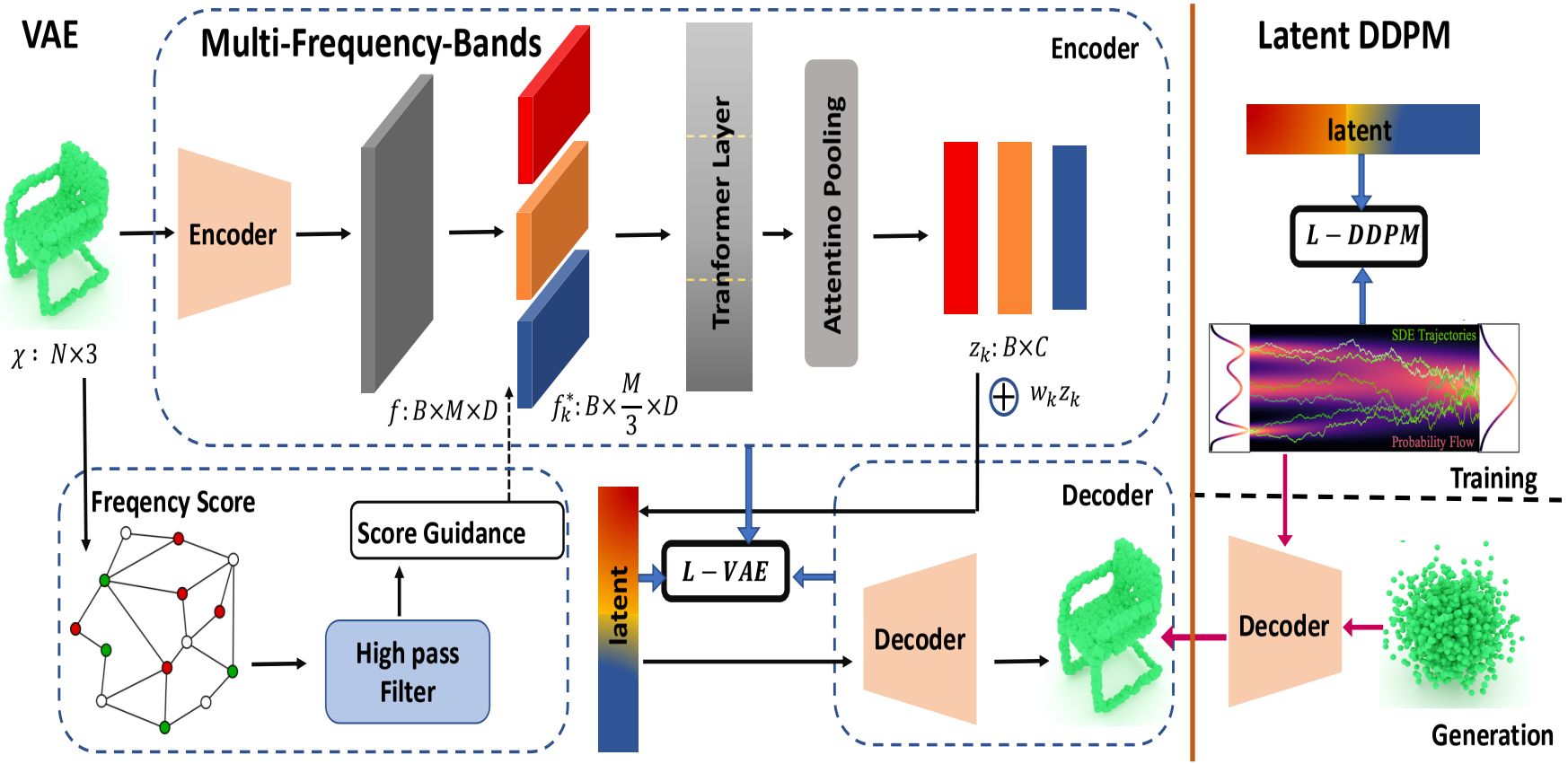

Figure 5.2: Illustration of the overall framework. The left part depicts the VAE training scheme, while the right side shows the latent DDPM training process (top) and generation process (bottom). During training, the input point cloud is encoded via our frequency-aware encoder, which simultaneously computes a frequency score to guide hierarchical segmentation and feature learning. For generation, the latent vector is sampled from the trained DDPM, and together with Gaussian noise, is passed through the CNF decoder to synthesize the final point cloud.

adaptive band weighting to effectively encode both global shape and fine details in the latent space.

5.2.1 Variational Autoencoder

Variational autoencoder (VAE) [115] is widely use in latent diffusion to model the latent distribution as it can access to a low-dimensional latent space and other works also prove this [21, 69]. VAE is probabilistic generative model which is able to model probability distribution with given datasets. For point cloud $\mathcal{X} \in \mathbb{R}^{N\times 3}$, VAE consists an encoder $q_\psi(z|X)$ and a decoder $p_\xi(X|z)$ parameterized by ψ and ξ. The encoder and decoder are jointly trained to maximize the evidence lower bound (ELBO):

$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{\psi}, \boldsymbol{\xi}; \boldsymbol{X}) := & \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_{\boldsymbol{\xi}}(\boldsymbol{X}|\boldsymbol{z})\right] \\ & - D_{\mathrm{KL}}\left(q_{\boldsymbol{\psi}}(\boldsymbol{z}|\boldsymbol{X}) \,\|\, p(\boldsymbol{z})\right),\end{aligned} \tag{5.1}$$

where $D_{\mathrm{KL}}(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence between the two distributions.

5.2.2 Diffusion Probabilistic Model

The VAE latent space is modelled by a denoising diffusion probabilistic model [60, 97], similar as Sec. 4.2.1, only substitute the x with latent z, thus the optimization objective becomes:

$$\mathcal{L} = \mathbb{E}_{t\sim[1,T]} w\left(t\right) \parallel \epsilon_\theta\left(\mathbf{z}_t, t\right) - \epsilon_0 \parallel_2^2 \tag{5.2}$$

After training, latent generation can be gained by sampling from a standard Gaussian distribution.

5.2.3 Point Cloud Graph Filter

Unlike 2D domain, where spectral analysis methods such as Fourier and wavelet transforms are straightforwardly applicable [65, 66, 69], the irregular, non-Euclidean nature of point clouds [105, 106] demands the development of alternative approaches for defining frequency components. The absence of structured grid in point cloud

data [29, 107] complicates the direct adoption of traditional spectral techniques, thus motivating a tailored method to effectively capture and process the inherent frequency characteristics of point cloud geometry.

To obtain a frequency-aware ordering of points, we compute a per-point frequency score from a sparse k-NN graph. We follow the graph construction and spectral interpretation in Sec. 4.2.1 and Sec. 4.2.3, then only describe the score used in this chapter.

Given a point cloud $\mathcal{X} = \{x_i\}_{i=1}^{N}$, we build a sparse weighted k-NN graph and apply a Laplacian-style high-pass operator $h(\tilde{\mathcal{A}}_w) = \mathbf{I} - \tilde{\mathcal{A}}_w$. The filtered response at each point is:

$$(h(\tilde{\mathcal{A}}_w)\mathcal{X})_i = x_i - \sum_{j=1}^{N} (\tilde{\mathcal{A}}_w)_{i,j} x_j. \tag{5.3}$$

We define the frequency score as the point-wise magnitude of the response:

$$S_i = \left\| (h(\tilde{\mathcal{A}}_w)\mathcal{X})_i \right\|_2, \tag{5.4}$$

where a larger S_i indicates stronger local variation relative to its neighborhood.

5.2.4 Multi-Frequency Bands Transformer VAE

As shown in Fig. 5.2, we propose multi-frequency bands Transformer VAE, a novel latent representation framework that explicitly incorporates frequency decomposition into point cloud encoding for the generation task. The proposed architecture operates in three main stages: (*i*) multi-frequency segmentation, where points are ordered and partitioned into distinct frequency bands based on a pre-computed frequency score; (*ii*) transformer-based encoding, where each frequency band is processed by an independent transformer module to capture both local and long-range dependencies within that band; and (*iii*) learnable attention-weighted aggregation, where the encoded features from different bands are adaptively fused via learnable per-band weights. This design enables the latent space to retain structured, frequency-aware information, thereby improving the generative capacity of downstream diffusion models while maintaining efficiency.

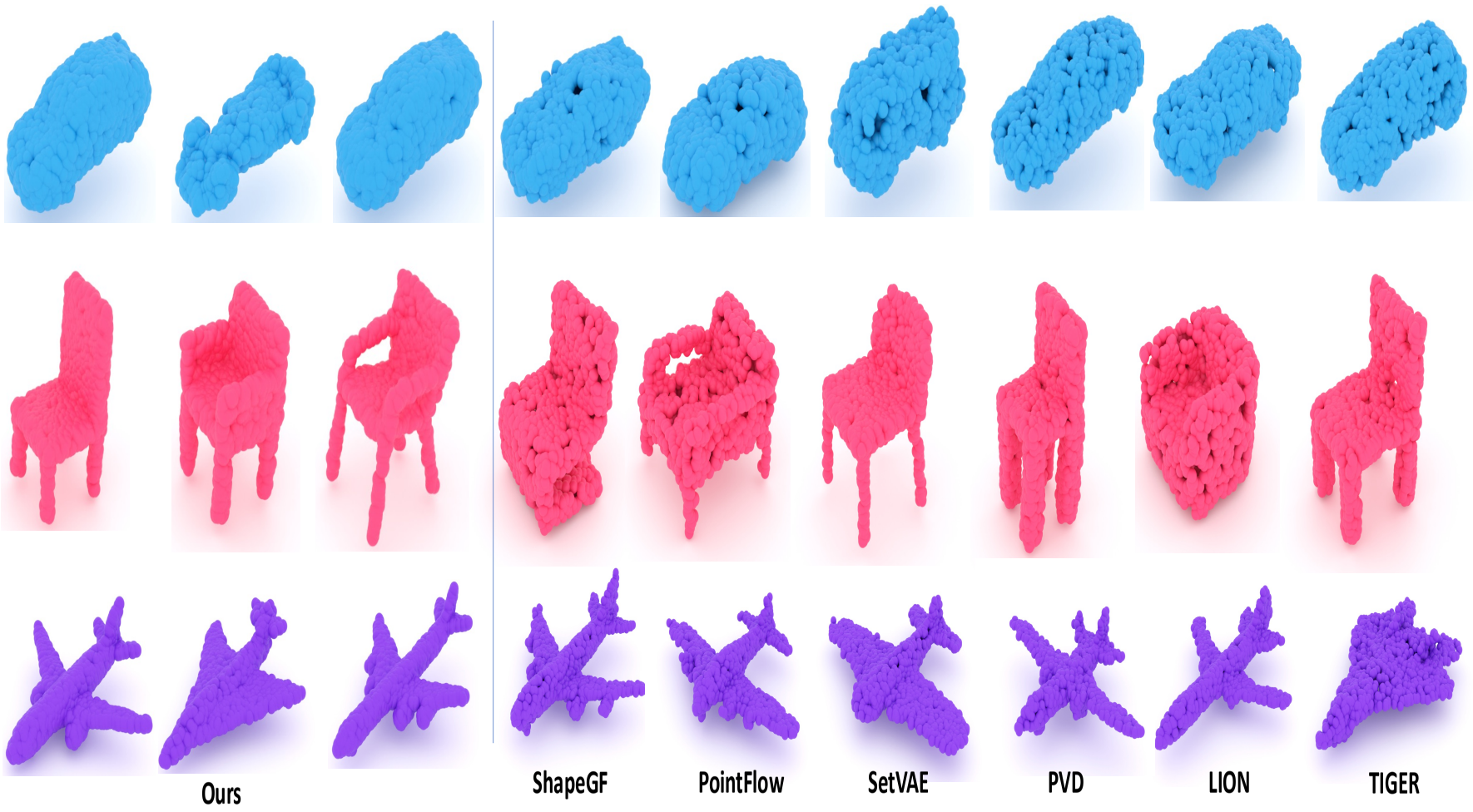

Figure 5.3: Qualitative comparisons for three illustrative object categories $\{cars, chairs, airplanes\}$: our approach (left) and other leading contemporary approaches (middle/right). TFDM generates high-quality and diverse point clouds.

Multi-Frequency Bands Given an input point cloud $\mathcal{X} \in \mathbb{R}^{B\times N\times 3}$, where B is the batch size, N is the number of points, and each point is represented by its 3D coordinates, we additionally associate each point with a pre-computed frequency score $\mathcal{S} \in \mathbb{R}^{B\times N}$ (as described in Sec. 5.2.3). We employ a U-Net backbone within the VAE encoder, producing intermediate point-wise features $f \in \mathbb{R}^{B\times M\times D}$, where M denotes the number of downsampled points and D is the feature dimension. To incorporate frequency-aware structure, we sort f along the point dimension according to the corresponding scores in $\mathcal{S}$, in descending order. The sorted feature is denoted by $f^* \in \mathbb{R}^{B\times M\times D} = \mathrm{Sort}(\mathbf{f}, \mathbf{S})$. We then divide f^* evenly into K non-overlapping frequency bands $\{f_k^*\}_{k=1}^K$, each corresponding to a distinct frequency range:

$$f_k^* \in \mathbb{R}^{B\times (M/K)\times D}, \quad k = 1, \ldots, K, \tag{5.5}$$

where lower k indicates higher-frequency components. *Note:* the frequency score is used solely for ordering and segmentation, and does not participate in gradient backpropagation.

Segment-wise Transformer Encoding. For each segmented frequency band $\mathbf{f}_k^* \in \mathbb{R}^{B\times P\times D}$, where $P = M/K$ is the number of points in the k-th band, we apply a dedicated Transformer encoder layer to capture *intra-band* dependencies and contextual relationships among points within the same frequency range. This design enables the model to learn specialized feature interactions for different frequency components, improving the expressiveness of the latent representation. Prior to Transformer processing, the features are permuted to match the required input format (P, B, D):

$$\tilde{f}_k = \mathrm{Transformer}_k\big(f_k^*\big), \quad \tilde{f}_k \in \mathbb{R}^{P\times B\times D}. \tag{5.6}$$

Following Transformer encoding, we employ an attention pooling module to aggregate point-level features into a compact band-level representation:

$$\mathbf{z}_k = \mathrm{AttnPool}\left(\tilde{f}_k\right), \quad \mathbf{z}_k \in \mathbb{R}^{B\times C}. \tag{5.7}$$

Here, C denotes the output channel dimension of the pooled features.

Attention-Weighted Fusion To adaptively modulate the relative contribution of each frequency band, we introduce a learnable parameter vector $\boldsymbol{\omega} \in \mathbb{R}^K$, where K denotes the number of bands. The parameters are normalized through a softmax operation to obtain attention weights:

$$\boldsymbol{\alpha} = \text{Softmax}(\boldsymbol{\omega}) \in \mathbb{R}^K, \quad \mathbf{z} = \sum \oplus(\alpha_k \mathbf{z}_k), \tag{5.8}$$

where $\mathbf{z}_k \in \mathbb{R}^{B \times C}$ represents the pooled feature from the k-th frequency band, and α_k denotes its normalized importance weight.

The concatenate feature vector $\mathbf{z} \in \mathbb{R}^{B \times 3C}$ is subsequently processed by a multi-layer perceptron (MLP) to yield the VAE latent parameters:

$$\mathbf{h} = \text{MLP}(\mathbf{z}) \in \mathbb{R}^{B \times 2C'}, \quad \boldsymbol{\mu}, \boldsymbol{\sigma} = \mathbf{h}[:, : C'], ; \mathbf{h}[:, C' :]. \tag{5.9}$$

Where the $\boldsymbol{\mu}, \boldsymbol{\sigma}$ can be trained via Eq. 5.1. At the diffusion stage, we adopt a deterministic latent representation by directly taking $\boldsymbol{\mu}$ as the input to the generative process, which empirically facilitates faster convergence and improved stability. Consequently, $\boldsymbol{\sigma}$ is utilized solely for computing the KL regularization term during VAE training and is omitted from the diffusion model. The resulting deterministic latent code $\boldsymbol{\mu}$ serves as the initial condition for the downstream diffusion-based generative model.

Continuous Normalizing Flow Decoder To decode the latent representation into a full-resolution point cloud, we employ a Conditional Continuous Normalizing Flow (CNF) as the decoder, denoted by f_θ. This flow-based model defines a bijective mapping between a simple base distribution and the target point cloud distribution, conditioned on the shape latent $\hat{z}$ obtained from either the VAE encoder or the latent DDPM. Training the CNF is done jointly with the VAE using a maximum likelihood objective derived from Eq. 5.1, which incorporates both the base distribution and the learned transformation. This formulation ensures that the decoder learns an accurate and invertible mapping from latent codes to point clouds, making it suitable for high-fidelity generation under our framework.

Method	Chair				Airplane				Car			
	1-NNA-Abs50 (↓)		COV (↑)		1-NNA-Abs50 (↓)		COV (↑)		1-NNA-Abs50 (↓)		COV (↑)	
	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD
r-GAN [22]	33.69	49.70	24.27	15.13	48.40	46.79	30.12	14.32	44.46	49.01	19.03	6.539
r-GAN (CD) [22]	18.58	33.94	19.71	15.43	37.30	43.95	38.52	21.23	16.49	38.78	38.92	23.58
r-GAN (EMD) [22]	21.90	14.65	38.07	44.04	39.49	26.91	38.27	38.52	21.16	16.19	37.78	45.17
PointFlow [23]	12.84	10.40	46.84	47.35	25.68	20.74	47.04	40.52	8.10	6.52	35.40	44.60
SoftFlow [98]	9.21	10.55	41.39	47.43	26.05	15.80	46.24	40.25	18.58	15.98	36.34	45.25
SetVAE [21]	8.84	10.57	46.83	44.36	24.80	15.65	48.10	40.35	13.04	15.53	40.99	46.59
DPF-Net [51]	12.00	8.03	42.08	41.75	16.53	5.44	45.82	46.55	12.53	4.48	45.85	48.56
DPM [32]	10.05	24.77	44.86	35.50	26.42	36.91	48.64	33.83	18.89	29.97	44.03	34.94
PVD [25]	7.89	23.68	40.66	42.71	16.44	26.26	47.34	42.15	4.55	3.83	41.19	50.56
LION [52]	3.70	<u>2.34</u>	48.94	52.11	17.41	11.23	47.16	49.63	3.41	1.14	50.00	**56.53**
TIGER [24]	4.61	2.71	-	-	21.85	**5.82**	-	-	4.31	2.24	-	-
FrePolad [54]	3.53	3.23	50.28	50.93	<u>15.25</u>	12.10	45.16	47.80	1.89	<u>0.26</u>	<u>50.14</u>	55.23
DiT-3D [53]	**0.89**	**0.73**	**52.45**	<u>54.32</u>	**12.35**	<u>8.67</u>	**53.16**	**54.39**	**1.76**	0.65	50.00	<u>56.38</u>
NSO [112]	5.51	7.63	-	-	18.63	11.85	-	-	9.66	3.55	-	-
TFDM (ours)	<u>3.43</u>	2.88	<u>50.52</u>	**54.79**	15.31	10.75	<u>48.95</u>	<u>51.25</u>	<u>1.79</u>	**0.22**	**50.28**	56.25

Table 5.1: Comparison results (%) on ShapeNet-v2 with shape metrics Absolute 50-Shifted 1-Nearest Neighbor Accuracy (1-NNA-Abs50) and Convergence (COV), Chamfer Distance (CD) and Earth Mover's Distance (EMD), where CD is multiplied by 10^3 and EMD is multiplied by 10^2; – denotes unavailable result from original authors; **Best**/<u>2nd best</u> highlighted.

5.2.5 Latent Diffusion as a Learned Prior

While it is possible to sample shape latents from a simple Gaussian prior during generation, such a restricted prior often fails to capture the complexity of the encoder distribution $q_\psi(\mathbf{z}|\mathbf{X})$. This mismatch, commonly referred to as the *prior hole problem* [21, 117], leads to degraded reconstruction quality. To overcome this limitation, we employ a denoising diffusion probabilistic model (DDPM) to learn a more expressive prior directly in the latent space. Specifically, the DDPM is trained on latents $\mathbf{z}$ sampled from $q_\psi(\mathbf{z}|\mathbf{X})$, replacing the standard Gaussian prior with a learned distribution $p_\zeta(\mathbf{z})$ that more accurately matches the true latent distribution $p(\mathbf{z})$. Then we can train the diffusion model across the latent $\mathbf{z}$ via Eq. 5.2.

When generation, the CNF transforms a set of initial noise points $u \sim \mathcal{N}(0, \mathbf{I})$, sampled from a standard 3D Gaussian distribution, into the desired output points $\mathbf{x} \in \mathbb{R}^3$ through an ODE-based transformation with conditioning on the latent μ. This formulation allows flexible control over point cardinality, as any number of initial points can be sampled and deterministically transformed into structured outputs. Additionally, the CNF enables expressive modeling of complex shape geometries, capturing fine-grained variations that are preserved from the frequency-aware latent representation.

Integrating latent diffusion with our multi-frequency Transformer VAE offers two main benefits. First, the frequency-aware encoding ensures that high-frequency geometric details preserved in the latent space remain intact during prior learning. Second, modeling the distribution of shape latents rather than raw point clouds greatly reduces the dimensionality of the generative task, thereby lowering both training and sampling costs (see Tab. Tab. 5.3) while enabling scalable synthesis.

5.3 Experiments

We evaluate our proposed TFDM architecture against state-of-the-art 3D point cloud generation approaches on the established ShapeNet-v2 [104] benchmark dataset.

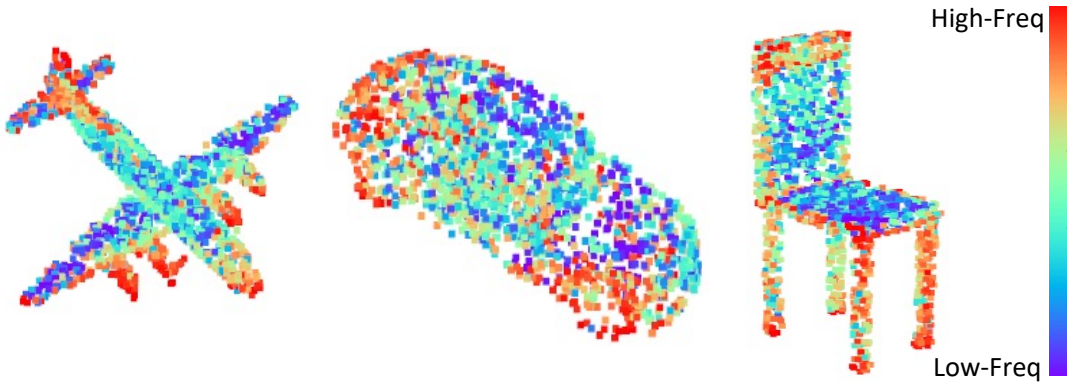

Figure 5.4: The illustration of the frequency distribution in different categories. Where red and blue represent the high and low frequency respectively.

5.3.1 Experimental Setup

ShapeNet-v2 Benchmark Dataset: For a fair comparison on ShapeNet-v2, we follow the common practice that focuses on training and evaluating only select key categories, namely *chair*, *car*, and *airplane*. From each shape, we sample 2048 points out of the 5000 available points in the training set and the test set, with normalization applied across the entire dataset. We adhere to the pre-processing steps and data split strategy as outlined in PointFlow [23].

Evaluation Metrics: Following established practices in prior work [25, 32], we use 1-NNA (and the derived 1-NNA-Abs50) and COV to evaluate the quality and diversity of the generation, together with CD and EMD, which measure the differences by point and distribution. Since the interpretation of 1-NNA can be ambiguous, we propose 1-NNA-Abs50 (Absolute 50-Shifted 1-NNA), a clearer alternative for evaluation. It transforms the aforementioned 1-NNA x into $|x - 50|$, making it more sensitive to deviations from the ideal 50%; a lower score indicates an ideal generated distribution closer to real data.

- **1-NNA** (1-Nearest Neighbor) Accuracy: Measures the leave-one-out accuracy of a 1-NN classifier, reflecting both quality and diversity of generated samples. A value close to 50% indicates a better result.

- **1-NNA-Abs50** (Absolute 50-Shifted 1-NNA): Since the interpretation of 1-NNA can be ambiguous, we propose a clearer alternative for evaluation. Transforms the

aforementioned 1-NNA x into $|x - 50|$, making it more sensitive to deviations from the ideal 50%; a lower score indicates an ideal generated distribution closer to real data.

- **COV** (Coverage): Evaluates how many reference point clouds are matched to at least one generated shape, where a higher value indicates greater diversity in generation.
- **CD** (Chamfer Distance): Measures point-wise similarity between generated and reference point clouds by computing the average nearest neighbor distance.
- **EMD** (Earth Mover's Distance): Captures the minimal cost of transforming one distribution into another, providing a global similarity measure between point clouds.

Implementation: For the point cloud high-pass filter, we set $k = 32$ for constructing the k-NN graph. The diffusion process is performed with 1000 timesteps. For the VAE model, we adopt a learning rate of 10^{-3} and a weight decay of 10^{-6}, with the latent dimension $C^{'} = 1024$. The number of frequency segments (bands) is optimally chosen as 3 for chair and car categories, and 4 for airplane, reflecting the higher geometric complexity of the latter. For latent diffusion, we use a learning rate of 10^{-5} and a weight decay of 10^{-8}, with a batch size of 64 and 1000 timesteps. All experiments are conducted on an NVIDIA A100 GPU (80GB) using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$.

5.3.2 Comparison with State-of-the-Art

Performance: In Tab. 5.1, we compare TFDM with multiple point cloud generation approaches. Notably,DIT-3D (NIPS23) TIGER (CVPR24), FrePolad (ECCV24) and NSO (ICLR25) are very recent methods. Among these, TIGER, FrePolad and NSO are relatively lightweight, yet we surpass both in most cases among the three categories: for example, TFDM achieves a 0.24% improvement in COV CD and 2.68% improvement in COV EMD on *chairs* compared to the best of the three, and a 1.62% gain on *airplane* with COV EMD. DiT-3D, while offering strong performance,

	Multi Band	MFB Block	Band Number	CD ↓ (NNA)	EMD ↓ (NNA)	CD ↑ (COV)	EMD ↑ (COV)
(a)		MLP		5.83	4.53	45.54	50.12
(b)		Transformer		4.86	3.43	47.65	52.15
(c)	✓	MLP	3	2.65	1.26	49.31	53.51
(d)	✓	Transformer	2	1.88	0.43	50.11	<u>55.88</u>
(e)	✓	Transformer	3	**1.79**	**0.22**	**50.28**	56.25
(f)	✓	Transformer	4	<u>1.86</u>	<u>0.35</u>	<u>50.13</u>	55.73

Table 5.2: Component-wise ablation of FLDCG on ShapeNet-v2 (car category). Evaluating the impact of multi-frequency band, transformer-based encoding, and the number of bands. NNA denotes the 1-NNA-Abs50 metric.

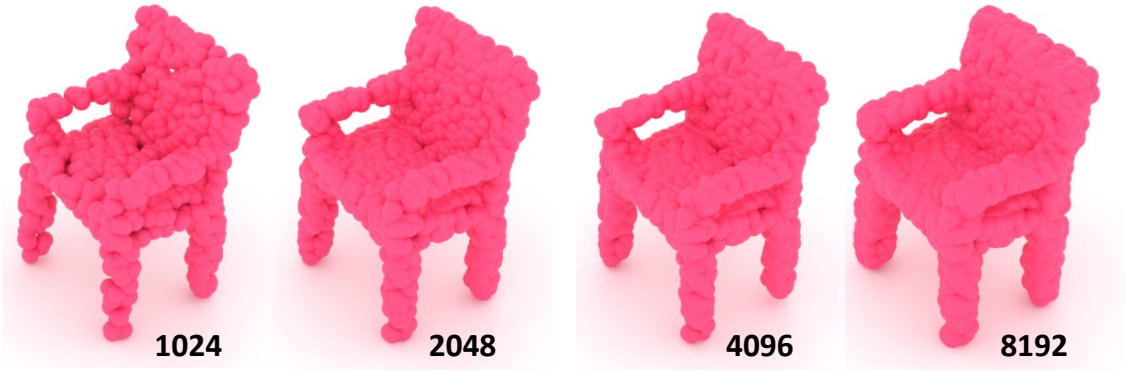

Figure 5.5: Varying point cardinality. Our model generates point clouds with arbitrary numbers of points.

incur extremely high computational overhead, requiring 1700 GPU hours and 711 million parameters. Even so, TFDM outperforms it on two out of four metrics for the *car* category, including a 0.43% gain in 1-NNA-Abs50 EMD and 0.28% in COV CD. These results highlight the efficiency and effectiveness of our approach.

Efficiency: As shown in Tab. 5.3, our proposed TFDM achieves a favorable balance between accuracy and efficiency. Although it introduces frequency-based segmentation and multiple transformer branches, the overall parameter count is substantially smaller (41.35M vs. 144.25M in LION and 711.88M in DiT-3D), leading to a dramatic reduction in both training time (24h vs. 550h/1688h) and inference cost (5.21s vs. 27.12s/100.13s). By leveraging latent diffusion in conjunction with a VAE backbone, our method achieves competitive generative performance with fewer computational resources.

Method	Para (M)	Training Time (h)	Inference Time (s)	EMD (1-NNA-Abs50) ↓
TIGER	70.11	164	9.73	2.24
DIT-3D	711.88	1688	100.13	0.65
LION	144.25	550	27.12	1.14
Ours	**41.35**	**24**	**5.21**	**0.22**

Table 5.3: Comparison on training and inference time, model size, and the corresponding evaluation results. Time is measured on the same device, and averaged over three categories: chair, airplane and car.

5.3.3 Ablation Studies

Effectiveness of Frequency-Band: To assess the effectiveness of decomposing features into multiple frequency bands, we compare our model against a baseline that employs transformer branches without band separation. As shown in Tab. 5.2, introducing frequency-aware band segmentation consistently improves generation quality. This validates that frequency bands provide a beneficial inductive bias for structuring latent representations. In particular, on the *car* category with three bands, our model achieves a relative improvement of 3.07% in 1-NNA-Abs50 CD and 3.21% in 1-NNA-Abs50 EMD over the band-free baseline.

Effectiveness of Transformer Architecture: We compare our multi-frequency transformer encoder with a shared MLP-based encoder to evaluate the impact of self-attention in modeling intra-band interactions. As reported in Tab. 5.2, the transformer-equipped model outperforms the MLP variant across all metrics. The attention mechanism facilitates better contextual encoding across spatially ordered points, leading to an improvement of 2.11% in COV CD and 1.97% in COV EMD over the band-free baseline.

Effectiveness of number of frequency band:We further analyze the sensitivity of our approach to the number of frequency bands. As shown in Tab. 5.2, a moderate number of bands provides the best trade-off between representation capacity and over-segmentation. Increasing the number of bands beyond this point brings only marginal gains, while reducing the bands compromises expressiveness. For instance, in the *car* category, the optimal setting of three bands yields improvements of 0.21% and 0.13% in 1-NNA-Abs50 CD compared with using one fewer or one additional

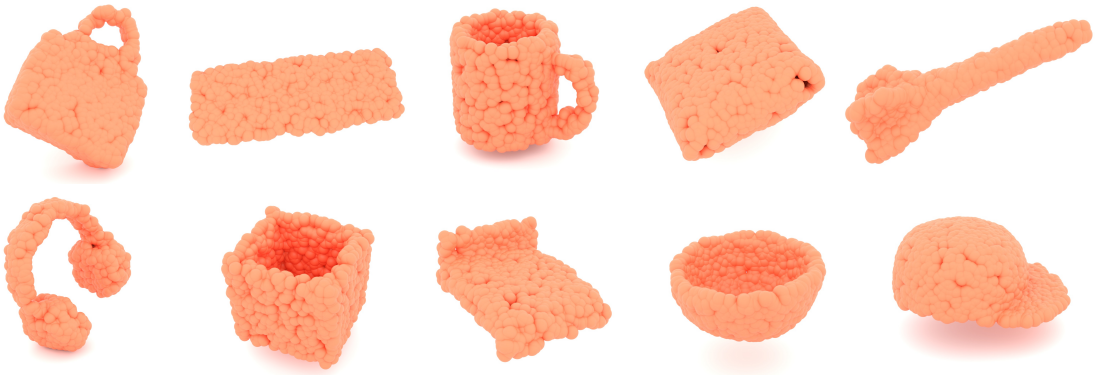

Figure 5.6: The illustration of multi-class generation.

band.

5.3.4 Multi-Class Generation

Multi-Class Generation: We train jointly without category conditioning on 10 categories from ShapeNet-v2 (*cap, keyboard, earphone, pillow, bag, rocket, basket, bed, mug, bowl*), and each category is operated as 3 frequency bands. Training on such a diverse set of shapes poses significant challenges due to the complexity and multimodal nature of the data. We present both qualitative and quantitative (Tab. 5.4) results. For comparison, we train baseline models under same conditions, and our approach shows the best.

Scalability: By modeling each point cloud as a distribution in the latent shape space, our method naturally supports sampling with arbitrary point cardinalities capability. As shown in Fig. 5.5 Our method can generate point clouds with varying numbers of points while preserving both geometric fidelity and structural consistency. This flexibility arises because the latent representation is set-based and independent of the number of input points.

5.4 Conclusion

This chapter presents a novel frequency-aware generative framework for 3D point cloud modeling, which integrates a Multi-Frequency Bands Transformer VAE with a latent diffusion model. By leveraging frequency scoring derived from point cloud

Method	CD ↓ (1-NNA-Abs50)	EMD ↓ (1-NNA-Abs50)	CD ↑ (COV)	EMD ↑ (COV)
DPM	9.71	21.54	43.65	38.94
PVD	7.52	17.43	44.12	44.32
Tiger	**0.88**	0.98	56.25	57.64
Ours	0.91	**0.86**	**56.31**	**58.12**

Table 5.4: Comparison results (%) jointly trained on 10 categories.

Laplacian structures, we introduce a frequency-based ordering and segmentation strategy that enables hierarchical feature decomposition. Each frequency band is processed by dedicated Transformer encoders and fused through attention-weighted aggregation, resulting in a more structured and informative latent representation. Extensive experiments demonstrate the effectiveness of our proposed method in generating diverse and detailed point clouds while maintaining computational efficiency. Our design provides a strong foundation for scalable, frequency-aware 3D generative modeling, and opens up new directions for structured latent representation learning in point cloud domains.

Future Direction A promising direction is to integrate learnable, frequency-aware modules directly into the architecture rather than relying on fixed priors. We also aim to extend the model to multimodal 2D rendering conditioning on or jointly training with images, where classical frequency tools (e.g., Fourier transforms) can be applied directly.

We could also consider the extension to noisy and non-uniform point clouds. In real-world scenarios, point clouds often exhibit measurement noise and non-uniform sampling density, which may affect the stability of the frequency score and subsequent band partition. To improve robustness, several practical extensions can be adopted. First, noise sensitivity can be reduced at the graph level by constructing the k-NN graph with additional constraints, such as mutual k-NN or radius-limited neighborhoods, and by using normalized graph operators. These strategies prevent noisy points or long-range connections from dominating the Laplacian response. Second, non-uniform sampling can be addressed by degree-normalized Laplacian formulations, which mitigate bias toward densely sampled regions. This allows the

frequency score to better reflect geometric variation rather than local point density.

CHAPTER 6

Conclusions

This thesis presented three contributions in the field of point cloud learning, each targeting a critical aspect of segmentation or generation:

In Chapter. 3, we introduced $U3DS^3$, a holistic unsupervised segmentation framework that requires no human annotations or pretraining. The method begins with geometric superpoint construction, followed by clustering and iterative pseudo-label refinement. To further strengthen representation learning, $U3DS^3$ leverages invariance and equivariance constraints in voxelized space, ensuring robustness to geometric transformations. Unlike prior unsupervised methods that were often limited to object-level segmentation, $U3DS^3$ generalizes to scene-level segmentation in both indoor and outdoor datasets. Experiments on ScanNet, SemanticKITTI, and S3DIS demonstrated competitive or state-of-the-art performance, validating that semantic scene understanding can be achieved in a fully annotation-free setting.

Chapter. 4 proposed TFDM, an end-to-end diffusion model that integrates frequency-aware encoding with state space modeling. A time-variant frequency encoder aligns the generative trajectory with the coarse-to-fine nature of diffusion: low-frequency structures are emphasized in early steps, while high-frequency details are refined in later steps. To improve efficiency, we designed dual latent Mamba

blocks, reducing reliance on costly attention mechanisms with lightweight state space operations that still capture long-range dependencies. On ShapeNet-v2, TFDM achieved state-of-the-art fidelity while reducing parameters and inference time by up to an order of magnitude, showing that frequency-aware design and efficient architectures can complement each other in generative modeling.

In Chapter. 5, we developed FLDCG, a two-stage latent diffusion framework that enhances latent representations with explicit frequency decomposition. By applying spectral graph decomposition to point clouds, we separated representations into multiple frequency bands. A multi-band transformer architecture was then designed, where each branch specialized in a different frequency component, jointly encoding global low-frequency structure and local high-frequency detail. This enriched latent representation was subsequently used by a diffusion model, yielding higher fidelity generation with significantly reduced computational cost. Experiments on ShapeNet-v2 demonstrated improved coverage and minimum matching distance compared to baselines, with over 20× efficiency gains.

Overarching Themes

Taken together, these three contributions converge toward two overarching research themes:

- **Efficiency in scene understanding.** Both segmentation and generation suffer from computational bottlenecks annotation cost in segmentation and iterative denoising in generation. $U3DS^3$ eliminates annotation overhead, while TFDM and FLDCG substantially reduce the computational burden of diffusion-based generation. These works demonstrate that efficiency can be achieved without sacrificing performance, and in some cases even improves generalization.

- **Representation learning for geometric understanding.** High-quality point cloud understanding requires models that capture both global and fine-grained geometry. $U3DS^3$ demonstrates that invariance and equivariance can

guide unsupervised feature learning; TFDM leverages time-variant frequency encodings to align coarse-to-fine generation with geometry; and FLDCG explicitly models multi-band frequency components to enhance latent representations. These approaches collectively highlight frequency and geometric modeling as essential for advancing fidelity and robustness in 3D learning.

By uniting efficiency and representation quality, this thesis provides a comprehensive framework for scalable, annotation-free, and high-fidelity point cloud understanding.

Future Directions

While this thesis makes significant progress, several avenues remain open for exploration:

Unsupervised scene segmentation beyond static data. $U3DS^3$ is currently restricted to static indoor and outdoor scenes, limiting its ability to handle dynamic environments or multi-modal data. Extending the framework to temporal point clouds and incorporating cross-modal fusion (e.g., combining LiDAR with RGB or event-based sensors) would enhance its robustness and broaden its applicability in robotics and autonomous driving.

Toward faster and more flexible generative modeling. Although TFDM achieves significant efficiency gains, it still requires tens or hundreds of denoising steps due to the iterative nature of diffusion. Future research could explore alternative paradigms such as flow matching, rectified diffusion, or score distillation to enable single-step or few-step generation, and extend the framework toward conditional generation guided by text, images, or multimodal priors for greater controllability.

The proposed frameworks can be naturally extended from static 3D point clouds to 4D point cloud sequences by incorporating temporal structure into both representation learning and generative modeling. For segmentation, temporal consistency constraints can be introduced to encourage stable superpoint assignments across frames, enabling unsupervised learning of dynamic scenes. For generative models, frequency-aware encoding can be generalized to the spatio-temporal domain, where low-frequency

components capture global motion patterns and high-frequency components model fine-grained geometric and temporal variations. In this context, state space models such as Mamba are particularly well suited, as they can efficiently model long-range temporal dependencies with linear complexity, offering a scalable alternative to attention-based temporal transformers. These extensions would allow the proposed methods to handle dynamic environments, such as human motion, traffic scenes, or long-term LiDAR sequences.

Text-conditioned point cloud generation. Another promising direction is to extend the generative frameworks toward text-conditioned point cloud synthesis. The latent diffusion formulation in Chapter. 4 and Chapter. 5 provides a natural interface for multimodal conditioning, where text embeddings derived from pretrained language–vision models can be injected into the diffusion process via cross-attention or conditional normalization. Frequency-aware latent representations further offer an interpretable mechanism for semantic control, as textual descriptions may primarily influence low-frequency global structure (e.g., object category or shape attributes), while leaving high-frequency geometric details to be refined by the generative model. Such an extension would enable controllable 3D generation from natural language, bridging point cloud modeling with recent advances in multimodal foundation models.

Frequency-aware modeling across tasks. FLDCG depends on spectral graph decomposition, which may be sensitive to graph construction and less scalable to very large datasets. This motivates the development of lightweight or learnable frequency decomposition strategies and the extension of frequency-aware design beyond generation to tasks such as completion, registration, and part-level semantic reasoning.

Advanced hybrid architectures. While TFDM leverages Mamba modules and FLDCG adopts transformers, the integration of state space models and attention mechanisms in 3D deep learning remains underexplored. Designing hybrid architectures that combine the efficiency of state space models with the expressive power of transformers could enable scalable and more generalizable 3D representation learning for complex real-world scenes.

In summary, this thesis advances the state of point cloud learning by addressing efficiency, annotation efficiency, and representation quality across segmentation and generation. By integrating unsupervised learning, frequency-aware modeling, and advanced architectures, we provide new pathways toward scalable and high-fidelity 3D understanding.

Bibliography

[1] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2016.

[2] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Int. Conf. Comput. Vis. (ICCV)*, October 2019.

[4] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," No. 34, pp. 226–231, 1996.

[5] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[6] D. Griffiths and J. Boehm, "A review on deep learning techniques for 3d sensed data classification," *Remote Sensing*, vol. 11, no. 12, p. 1499, 2019.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 652–660, 2017.

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems (NIPS)* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[9] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4490–4499, 2018.

[10] H.-Y. Meng, L. Gao, Y.-K. Lai, and D. Manocha, "Vv-net: Voxel vae net with group convolutions for point cloud segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8500–8508, 2019.

[11] H. Zhou, Y. Feng, M. Fang, M. Wei, J. Qin, and T. Lu, "Adaptive graph convolution for point cloud analysis," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4965–4974, 2021.

[12] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[13] D. Garrido, R. Rodrigues, A. Augusto Sousa, J. Jacob, and D. Castro Silva, "Point cloud interaction and manipulation in virtual reality," in *2021 5th International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 15–20, 2021.

[14] I. Kapelyukh, V. Vosylius, and E. Johns, "Dall-e-bot: Introducing web-scale diffusion models to robotics," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3956–3963, 2023.

[15] Y. Shi, P. Wang, J. Ye, M. Long, K. Li, and X. Yang, "Mvdream: Multi-view diffusion for 3d generation," *arXiv preprint arXiv:2308.16512*, 2023.

[16] L. Nunes, R. Marcuzzi, B. Mersch, J. Behley, and C. Stachniss, "Scaling diffusion models to real-world 3d lidar scene completion," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 14770–14780, June 2024.

[17] T. Anciukevičius, Z. Xu, M. Fisher, P. Henderson, H. Bilen, N. J. Mitra, and P. Guerrero, "Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 12608–12618, 2023.

[18] L. Jiang, S. Shi, Z. Tian, X. Lai, S. Liu, C.-W. Fu, and J. Jia, "Guided point contrastive learning for semi-supervised point cloud semantic segmentation," in *Int. Conf. Comput. Vis. (ICCV)*, 2021.

[19] Y. Zhang, Y. Qu, Y. Xie, Z. Li, S. Zheng, and C. Li, "Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 15520–15528, October 2021.

[20] L. Li, H. P. H. Shum, and T. P. Breckon, "Less is more: Reducing task and model complexity for 3d point cloud semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 9361–9371, June 2023.

[21] J. Kim, J. Yoo, J. Lee, and S. Hong, "Setvae: Learning hierarchical composition for generative modeling of set-structured data," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 15059–15068, 2021.

[22] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," 2018.

[23] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, "Pointflow: 3d point cloud generation with continuous normalizing flows," in *Int. Conf. Comput. Vis. (ICCV)*, October 2019.

[24] Z. Ren, M. Kim, F. Liu, and X. Liu, "Tiger: Time-varying denoising model for 3d point cloud generation with diffusion process," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 9462–9471, 2024.

[25] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 5826–5835, October 2021.

[26] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58443–58469, 2020.

[27] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17853–17862, 2023.

[28] P. Kim, J. Chen, and Y. K. Cho, "Slam-driven robotic mapping and registration of 3d point clouds," *Automation in Construction*, vol. 89, pp. 38–48, 2018.

[29] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 16259–16268, 2021.

[30] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 35, pp. 33330–33342, 2022.

[31] F. Bordes, S. Honari, and P. Vincent, "Learning to generate samples from noise through infusion training," in *Int. Conf. Learn. Represent. (ICLR)*, 2022.

[32] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2021.

[33] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Int. Conf. Comput. Vis. (ICCV)*, October 2019.

[34] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2020.

[35] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 3075–3084, 2019.

[36] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[37] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2018.

[38] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, 2019.

[39] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel cnn for efficient 3d deep learning," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 32, 2019.

[40] B. Graham, M. Engelcke, and L. van der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[41] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point transformer v3: Simpler faster stronger," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 4840–4851, 2024.

[42] Y. Zhang, Z. Li, Y. Xie, Y. Qu, C. Li, and T. Mei, "Weakly supervised semantic segmentation for large-scale point cloud," in *AAAI*, no. 4, pp. 3421–3429, 2021.

[43] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Eur. Conf. Comput. Vis. (ECCV)*, September 2018.

[44] Y. Liu, Y. Zheng, D. Zhang, H. Chen, H. Peng, and S. Pan, "Towards unsupervised deep graph structure learning," in *Proceedings of the ACM Web Conference 2022*, pp. 1392–1403, 2022.

[45] Y. Mo, L. Peng, J. Xu, X. Shi, and X. Zhu, "Simple unsupervised graph representation learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, pp. 7797–7805, 2022.

[46] S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu, "Unsupervised part segmentation through disentangling appearance and shape," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 8355–8364, June 2021.

[47] Z. Song and B. Yang, "Ogc: Unsupervised 3d object segmentation from rigid dynamics of point clouds," in *Advances in Neural Information Processing Systems (NIPS)* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 30798–30812, Curran Associates, Inc., 2022.

[48] W. Li, W. Liu, J. Zhu, M. Cui, H. Xiansheng, and L. Zhang, "Box-supervised instance segmentation with level set evolution," in *European Conference on Computer Vision (ECCV)*, pp. 1–18, 2022.

[49] J. H. Cho, U. Mall, K. Bala, and B. Hariharan, "Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 16794–16804, June 2021.

[50] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[51] R. Klokov, E. Boyer, and J. Verbeek, "Discrete point flow networks for efficient point cloud generation," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 694–710, Springer, 2020.

[52] X. Zeng, A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, and K. Kreis, "Lion: Latent point diffusion models for 3d shape generation," in *Adv. Neural Inform. Process. Syst. (NeurIPS)* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 10021–10039, Curran Associates, Inc., 2022.

[53] S. Mo, E. Xie, R. Chu, L. Hong, M. Niessner, and Z. Li, "Dit-3d: Exploring plain diffusion transformers for 3d shape generation," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 36, 2024.

[54] C. Zhou, F. Zhong, P. Hanji, Z. Guo, K. Fogarty, A. Sztrajman, H. Gao, and C. Oztireli, "Frepolad: Frequency-rectified point latent diffusion for point cloud generation," in *Eur. Conf. Comput. Vis. (ECCV)*, 2024.

[55] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," in *International Conference on Machine Learning*, pp. 32211–32252, PMLR, 2023.

[56] F.-Y. Wang, Z. Huang, A. Bergman, D. Shen, P. Gao, M. Lingelbach, K. Sun, W. Bian, G. Song, Y. Liu, *et al.*, "Phased consistency models," *Advances in neural information processing systems*, vol. 37, pp. 83951–84009, 2024.

[57] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 10684–10695, 2022.

[58] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," *arXiv preprint arXiv:2212.08751*, 2022.

[59] S. Mo, E. Xie, Y. Wu, J. Chen, M. Nießner, and Z. Li, "Fast training of diffusion transformer with extreme masking for 3d point clouds generation," 2023.

[60] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Int. Conf. Mach. Learn. (ICML)*, pp. 2256–2265, PMLR, 2015.

[61] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[62] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[63] H. Phung, Q. Dao, and A. Tran, "Wavelet diffusion models are fast and scalable image generators," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 10199–10208, June 2023.

[64] X. Yang, D. Zhou, J. Feng, and X. Wang, "Diffusion probabilistic model made slim," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 22552–22562, June 2023.

[65] M. Xu, J. Zhang, Z. Peng, M. Xu, X. Qi, and Y. Qiao, "Learning geometry-disentangled representation for complementary understanding of 3d object point cloud," *AAAI*, vol. 35, pp. 3056–3064, 05 2021.

[66] H. Huang and Y. Fang, "Adaptive wavelet transformer network for 3d shape representation learning," in *Int. Conf. Learn. Represent.*, 2022.

[67] Y. Liu, D. Chen, S. Fu, P. T. Mathiopoulos, M. Sui, J. Na, and J. Peethambaran, "Segmentation of individual tree points by combining marker-controlled watershed segmentation and spectral clustering optimization," *Remote Sensing*, vol. 16, no. 4, p. 610, 2024.

[68] Z. Jing, H. Guan, P. Zhao, D. Li, Y. Yu, Y. Zang, H. Wang, and J. Li, "Multispectral lidar point cloud classification using se-pointnet++," *Remote Sensing*, vol. 13, no. 13, p. 2516, 2021.

[69] H. Phung, Q. Dao, and A. Tran, "Wavelet diffusion models are fast and scalable image generators," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 10199–10208, June 2023.

[70] K.-H. Hui, R. Li, J. Hu, and C.-W. Fu, "Neural wavelet-domain diffusion for 3d shape generation," in *SIGGRAPH Asia 2022 conference papers*, pp. 1–9, 2022.

[71] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[72] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, *et al.*, "Rethinking attention with performers," *arXiv preprint arXiv:2009.14794*, 2020.

[73] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.

[74] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[75] Z. Chang, G. A. Koulieris, and H. P. H. Shum, "On the design fundamentals of diffusion models: A survey," *arXiv*, 2023.

[76] J. Lei, C. Deng, K. Schmeckpeper, L. Guibas, and K. Daniilidis, "Efem: Equivariant neural field expectation maximization for 3d object segmentation without scene supervision," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023.

[77] J. Hou, B. Graham, M. Nießner, and S. Xie, "Exploring data-efficient 3d scene understanding with contrastive scene contexts," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 15587–15597, 2021.

[78] Z. Zhang, B. Yang, B. Wang, and B. Li, "Growsp: Unsupervised semantic segmentation of 3d point clouds," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 17619–17629, June 2023.

[79] C.-K. Yang, Y.-Y. Chuang, and Y.-Y. Lin, "Unsupervised point cloud object co-segmentation by co-contrastive learning and mutual attention sampling," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 7335–7344, October 2021.

[80] C. Ma, Y. Yang, J. Guo, F. Pan, C. Wang, and Y. Guo, "Unsupervised point cloud completion and segmentation by generative adversarial autoencoding network," in *Advances in Neural Information Processing Systems (NIPS)* (A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, eds.), 2022.

[81] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Int. Conf. Comput. Vis. (ICCV)*, October 2019.

[82] Y. Ouali, C. Hudelot, and M. Tami, "Autoregressive unsupervised image segmentation," in *Eur. Conf. Comput. Vis. (ECCV)* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 142–158, Springer International Publishing, 2020.

[83] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, and L. Van Gool, "Unsupervised semantic segmentation by contrasting object mask proposals," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 10052–10062, October 2021.

[84] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka, "Labels4free: Unsupervised segmentation using stylegan," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 13970–13979, October 2021.

[85] M. Jaritz, T.-H. Vu, R. d. Charette, E. Wirbel, and P. Perez, "xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2020.

[86] W. Liu and F. Su, "Unsupervised adversarial domain adaptation network for semantic segmentation," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 11, pp. 1978–1982, 2020.

[87] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation - supervoxels for point clouds," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2013.

[88] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, (New York, NY, USA), p. 1177–1178, Association for Computing Machinery, 2010.

[89] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[90] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2021.

[91] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2019.

[92] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *DAGM-Symposium*, 2003.

[93] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," in *Advances in Neural Information Processing Systems (NIPS)*, 2022.

[94] L. Landrieu and M. Boussaha, "Point cloud oversegmentation with graph-structured deep metric learning," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2019.

[95] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems (NIPS)* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[96] A. Gupta, W. Xiong, Y. Nie, I. Jones, and B. Oğuz, "3dgen: Triplane latent diffusion for textured mesh generation," *arXiv preprint arXiv:2303.05371*, 2023.

[97] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 33, pp. 6840–6851, 2020.

[98] H. Kim, H. Lee, W. H. Kang, J. Y. Lee, and N. S. Kim, "Softflow: Probabilistic framework for normalizing flow on manifolds," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 33, pp. 16388–16397, 2020.

[99] D. Liang, X. Zhou, W. Xu, X. Zhu, Z. Zou, X. Ye, X. Tan, and X. Bai, "Pointmamba: A simple state space model for point cloud analysis," in *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2024.

[100] T. Zhang, X. Li, H. Yuan, S. Ji, and S. Yan, "Point could mamba: Point cloud learning via state space model," *arXiv preprint arXiv:2403.00762*, 2024.

[101] X. Yang, D. Zhou, J. Feng, and X. Wang, "Diffusion probabilistic model made slim," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 22552–22562, June 2023.

[102] V. T. Hu, S. A. Baumann, M. Gui, O. Grebenkova, P. Ma, J. S. Fischer, and B. Ommer, "Zigma: A dit-style zigzag mamba diffusion model.," *arXiv preprint arXiv:2403.13802*, 2024.

[103] L. Fu, X. Li, X. Cai, Y. Wang, X. Wang, Y. Shen, and Y. Yao, "Md-dose: A diffusion model based on the mamba for radiotherapy dose prediction," *arXiv preprint arXiv:2403.08479*, 2024.

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

[105] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 652–660, 2017.

[106] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[107] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 7163–7172, 2019.

[108] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2018.

[109] Y. Wu and K. He, "Group normalization," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 3–19, 2018.

[110] J. L. Ba, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[111] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[112] K.-H. Hui, C. Liu, X. Zeng, C.-W. Fu, and A. Vahdat, "Not-so-optimal transport flows for 3d point cloud generation," in *The Thirteenth International Conference on Learning Representations*, 2025.

[113] D. Huang, X. Huang, C. Zhang, and Y. Shi, "Lpcg: A self-conditional architecture for labeled point cloud generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 3635–3643, 2025.

[114] Y. Li, Y. Dou, X. Chen, B. Ni, Y. Sun, Y. Liu, and F. Wang, "3dqd: Generalized deep 3d shape prior via part-discretized diffusion process," 2023.

[115] Z. Wang, C. Pei, M. Ma, X. Wang, Z. Li, D. Pei, S. Rajmohan, D. Zhang, Q. Lin, H. Zhang, *et al.*, "Revisiting vae for unsupervised time series anomaly detection: A frequency perspective," in *Proceedings of the ACM web conference 2024*, pp. 3096–3105, 2024.

[116] X. Lin, Y. Li, J. Hsiao, C. Ho, and Y. Kong, "Catch missing details: Image reconstruction with frequency augmented variational autoencoder," 2023.

[117] Z. Wang, C. Pei, M. Ma, X. Wang, Z. Li, D. Pei, S. Rajmohan, D. Zhang, Q. Lin, H. Zhang, J. Li, and G. Xie, "Revisiting vae for unsupervised time series anomaly detection: A frequency perspective," 2024.