Indian Institute of Technology Gandhinagar

Project Course CS 299

Reimagining Point Cloud Denoising in Latent Space

Submitted by

Vishnu Sinha
Junior Undergraduate, Artificial Intelligence
Indian Institute of Technology Gandhinagar

Harsh Kumar Jha
Junior Undergraduate, Artificial Intelligence
Indian Institute of Technology Gandhinagar

Under the guidance of

Prof. Shanmuganathan Raman
Jibaben Patel Chair in Artificial Intelligence
Professor and Head, Department of Computer Science and Engineering
Professor, Department of Electrical Engineering, IIT Gandhinagar

Contents

Chapter 1

Introduction

3D point clouds are widely used in robotics, autonomous navigation, AR, and scientific scanning, yet real-world point clouds from LiDAR, depth cameras, and multi-view stereo contain irregular, non-Gaussian noise. This noise disrupts fine geometry, weakens surface fidelity, and degrades tasks such as reconstruction, normal estimation, SLAM, and manipulation. Because point clouds lack grid structure and ordering, classical image filters do not apply, and geometric denoisers struggle with severe noise and inconsistent sampling.

Deep models such as PointNet [1], PointNet++ [2], DGCNN [3], PU-Net [4], and PCN [5] learn permutation-invariant features and improve robustness, but most remain deterministic regressors that map noisy inputs directly to clean outputs. This single-step approach oversmooths details and often lacks global consistency.

Diffusion-based methods provide a stronger formulation [6]. P2P-Bridge [7] frames denoising as a Schrödinger Bridge problem, learning an optimal stochastic transport path between noisy and clean point distributions. Although this improves coherence, diffusion is still performed in raw coordinate space, where XYZ points contain little structural information—no curvature cues, part relationships, or semantic context. As a result, the model must infer complex geometry from low-level positions alone, limiting its expressive power.

This exposes a central limitation: coordinate-space diffusion can *denoise* shapes but cannot reliably *represent* geometry. A structured, geometry-aware domain is needed.

To address this, we develop a family of latent-space autoencoders that combine PointNet++ [2]-style multi-scale abstraction with transformer-inspired offset-attention. These architectures build a compact latent manifold encoding both global dependencies and fine geometric features, enabling diffusion to operate in a more expressive space. This shift stabilizes the denoising trajectory, improves reconstruction accuracy, and enhances global shape consistency. Our work examines the design of these autoencoders, their integration into the P2P-Bridge [7] pipeline, and the empirical behaviour of refined latent variants.

Chapter 2

Problem Statement

2.1 Importance of 3D Point Clouds

3D point clouds have emerged as a core geometric representation across robotics, autonomous driving, augmented and virtual reality, scientific scanning, and digital reconstruction. They directly encode surface geometry but remain inherently unstructured, consisting of unordered point samples without any grid or connectivity.

2.2 Challenges of Real-World Noise and Denoising Requirements

Point clouds captured through LiDAR sensors, depth cameras, or multi-view stereo systems are almost always affected by significant noise. Sensor precision limits, reflections, occlusions, environmental interference, and irregular sampling lead to jittered positions, blurred edges, missing or duplicated regions, and highly uneven point densities. These degradations severely hinder downstream tasks such as mesh reconstruction, shape analysis, SLAM, pose estimation, robotic manipulation, and 3D modeling.

Denoising is fundamentally difficult because point clouds lack the regular lattice structure found in images, offering no fixed neighborhood relationships. As a result, standard image-based filtering or smoothing techniques cannot be directly applied. Moreover, the noise present in real-world scans is non-Gaussian, spatially irregular, and often correlated with underlying surface geometry. An effective denoising method must remove noise while preserving thin structures, sharp edges, and high-curvature regions, and it must maintain global geometric coherence and topological consistency across the entire shape.

2.3 Limitations of Existing Methods and Motivation for Latent-Space Diffusion

Classical geometric approaches—including bilateral filtering, moving least squares, and PCA-based normal estimation—attempt to fit local surface patches but depend on fixed-neighborhood assumptions. These methods deteriorate under high noise levels or in the presence of complex geometric structures. Deep learning models such as PointNet [1], PointNet++ [2], DGCNN [3], PU-Net [4], and PCN [5] improved robustness by learning permutation-invariant, hierarchical geometric features. However, most of these models adopt a deterministic regression paradigm that maps noisy coordinates directly to clean ones, frequently leading to oversmoothing or geometric distortion under severe noise.

Recent generative approaches reformulate denoising as a probabilistic transport problem. Diffusion models incrementally move noisy point sets toward clean ones, and P2P-Bridge [7] (ECCV 2024) applies Schrödinger bridges to learn an optimal stochastic transport path between paired noisy and clean point clouds. While effective, P2P-Bridge [7] performs diffusion solely in raw XYZ coordinate space, a weak representation that fails to encode curvature, long-range relationships, part-level interactions, or semantic cues. This creates an inherent representational bottleneck.

These limitations motivate the central research question of this project: *Can a structured latent representation—capturing both fine-grained local geometry and long-range global context—serve as a more effective domain for diffusion-based point cloud denoising?* To investigate this, we design progressively expressive latent-space autoencoders built on hierarchical PointNet++ [2] abstractions, transformer-inspired attention mechanisms, and timestep-conditioned feature modulation. We examine whether diffusion operating on such latent features yields more stable, accurate, and detail-preserving reconstructions compared to diffusion performed directly in coordinate space.

Chapter 3

Background and Theory

3.1 Diffusion in P2P-Bridge [7]

P2P-Bridge [7] is built upon a Schrödinger-bridge-inspired diffusion formulation in which noisy point clouds are gradually transported toward the clean data distribution through a learned score model. The framework constructs an intermediate sequence of diffusion states by interpolating between the noisy input and the clean target using an optimal-transport–based strategy. The forward process conceptually introduces noise, while the reverse process removes it by applying the learned score function at each timestep. Although effective for smooth or moderately noisy surfaces, this diffusion pipeline operates entirely in raw coordinate space, limiting the model's expressive capacity and restricting its ability to capture higher-level geometric structure.

3.2 Why Latent Space Diffusion

Operating directly on (x, y, z) coordinates provides only low-level positional information and neglects the richer structure inherent in 3D objects. Raw coordinates do not encode semantic relationships between object parts, multi-scale geometric organization, or contextual cues that describe how different regions of a shape interact. They also fail to incorporate any learned priors that arise from large-scale datasets. Introducing a latent autoencoder provides a more structured and expressive representation in which local geometry, global context, and semantic patterns are captured in a compact feature space. This latent embedding is naturally more robust to noise, invariant to point permutations, and equipped with shape-aware features that are significantly easier for the diffusion model to denoise. As a result, diffusion in latent space can follow smoother, more coherent trajectories and better preserve fine details and global structure.

3.3 Reproducibility of Baseline P2P-Bridge [7]

A foundational part of this project involved faithfully reproducing the original P2P-Bridge [7] training pipeline in our own computing environment before introducing any architectural changes. Achieving reproducibility required more than simply running the reference code—it demanded setting up a clean, controlled software environment, ensuring version alignment of all point-cloud–specific dependencies, and verifying that the training behavior matched the trends reported in the paper. We began by creating a completely new conda environment dedicated to P2P-Bridge [7]. This was done deliberately to isolate the framework from previous experiments and global Python installations, since subtle mismatches in CUDA-enabled libraries, PyTorch3D [8] versions, or point cloud operators can directly affect Chamfer and P2F metrics. Instead of installing packages manually, we used the project's requirements.txt as the initial source of truth. However, because several dependencies (especially pointnet2 operations, PyTorch3D [8]3D, and low-level CUDA kernels) are sensitive to version conflicts, the environment required iterative refinement. We installed the baseline requirements, identified incompatible or missing packages, and updated them to versions that were both supported by our cluster's CUDA toolkit and consistent with the versions expected by the original repository. Only after resolving these constraints did we obtain a stable environment in which all modules—farthest-point sampling, k-NN grouping, Chamfer kernels, and P2F operators—loaded without runtime warnings. After the environment was finalized, we reproduced the baseline configuration using the YAML files provided in the paper's official codebase. These YAML files control every aspect of training: learning rate, optimizer, diffusion noise levels, the number of diffusion timesteps, patch sampling scheme, and dataset paths. We preserved the learning rate and diffusion schedule specified in the paper, but we did not train to the full 450,000 iterations reported in the original work. Instead, we conducted a reproducibility validation run up to approximately 50,000 steps. This choice was practical—but still sufficient to confirm whether our setup was correct. As expected, the Chamfer Distance exhibited the same downward convergence pattern shown in the paper's training curves. However, one noticeable deviation appeared in the P2F metric: our P2F values were approximately two to three times higher than the reference numbers. This was consistent with the fact that P2F convergence is much slower and more sensitive to training duration than CD. In the P2P-Bridge [7] paper, P2F stabilizes only after extended training, when the model has learned fine-scale geometric corrections. Since our reproduction was stopped well before the model reached mid- or late-training, the elevated P2F was an expected outcome rather than an indication of an error. Importantly, the qualitative denoising behavior and the direction of metric improvement both matched the trends reported by the authors, confirming that our reproduction was structurally correct even if it was not trained to Completion3D [12]. To validate that our setup matched the intended behavior, we cross-checked intermediate outputs such as time-t interpolated point clouds, denoised reconstructions at various diffusion steps, and batch-normalized latent features. We also verified that normalization routines—centering, unit-sphere scaling, and patch-based sampling—matched the original implementation. These checks were essential, as inconsistencies in preprocessing or metric computation can lead to misleading results even when the architecture is correct. By the end of this reproducibility phase, we had successfully recreated the baseline P2P-Bridge [7] pipeline in a controlled environment, with metrics that followed the expected convergence patterns and architectural components behaving identically to the published framework. This reproducibility step ensured that all subsequent improvements could be confidently attributed to our latent autoencoder modifications rather than environmental inconsistencies.

Chapter 4

System Design and Implementation

4.1 Latent Conditioning Wrapper

To integrate the proposed latent autoencoder with the original P2P-Bridge [7] framework, we introduce a wrapper module named `LatentP2PB`, implemented in `p2pblatent.py`. This wrapper extends the standard P2P-Bridge [7] model and modifies its forward pass to incorporate latent features as an auxiliary conditioning signal to the diffusion backbone. The central idea is to preserve the theoretical formulation, training objective, and transport dynamics of P2P-Bridge [7], while enriching the diffusion process with geometric context encoded by a pretrained Semantic Autoencoder.

During initialization, the wrapper constructs the base P2P-Bridge [7] architecture and loads autoencoder weights specified in the configuration. The autoencoder is placed in evaluation mode, and all its parameters are frozen to ensure that the latent features remain stable across training iterations. This prevents unwanted drift in the latent manifold and guarantees that improvements arise solely from diffusion adaptation rather than latent representation changes.

In every forward pass, the wrapper computes the interpolated diffusion state x_t and its associated target using the closed-form Schrödinger-bridge equations. Alongside this, the wrapper extracts the clean coordinate tensor, normalizes it to the $(B, 3, N)$ layout, and forwards it through the autoencoder encoder to obtain a latent feature representation capturing global geometry, local curvature, and semantic regularities. Since the latent representation and x_t often differ in point resolution, a nearest-neighbor interpolation procedure aligns the latent features with the diffusion input. This yields a conditioning tensor x_{cond} that is temporally aligned with x_t and spatially aligned with the P2P-Bridge [7] backbone.

Once the conditioning features are prepared, the wrapper forwards $(x_t, t, x_{\text{cond}})$ into the diffusion U-Net. The model then predicts noise residuals in latent space and computes the denoising objective exactly as in the original P2P-Bridge [7] implementation. The remainder of the pipeline—including target construction, loss evaluation, and diffusion scheduling—remains unchanged. Thus, the wrapper preserves the theoretical foundations of P2P-Bridge [7] while substantially expanding its representational power.

4.2 Model Loader Modifications

To support latent-space diffusion, we significantly extended the design of the original `model_loader.py`. The updated loader now automatically selects and constructs one of three backbones depending on configuration:

- `pvcnn2unet` (default P2P-Bridge diffusion network),
- `semantic_ae` (standalone Semantic Autoencoder),
- `latentp2pb` (our proposed latent-conditioned diffusion model).

This unified construction pathway allows seamless switching between coordinate-space diffusion and latent-enhanced diffusion without altering downstream training scripts.

A crucial addition is the recognition of the custom model type `latentp2pb/latent_p2pb`. When this option is selected, the loader constructs the default PVCNN2Unet backbone and immediately wraps it inside the `LatentP2PB` module. This preserves all original diffusion logic while enabling latent conditioning at each timestep. Similarly, if the model type is set to `semantic_ae`, the loader builds the autoencoder directly and loads pretrained weights for use in latent extraction.

The loader was also extended with safer and more flexible checkpoint-handling logic. It now supports partial weight loading, detects shape mismatches, and automatically restores EMA weights and optimizer state if available. Logging was enhanced to display model configuration, parameter count, and wrapper structure, improving reproducibility and debugging. These modifications collectively provide a modular, extensible infrastructure for exploring both conventional and latent-space diffusion variants within a unified framework.

4.3 Challenges During Latent Integration

Integrating a latent autoencoder into P2P-Bridge [7] required addressing one of the most technically demanding aspects of this project: maintaining strict alignment of tensor shapes across multiple heterogeneous subsystems. Unlike image-based diffusion models with fixed spatial grids, P2P-Bridge [7] operates on unordered point sets whose feature layout depends on sampling density, encoder depth, and decoder configuration. Consequently, subtle mismatches in feature ordering led to silent failures, corrupted activations, or abrupt CUDA kernel crashes.

The first major source of error stemmed from conflicting tensor layouts. The autoencoder expects inputs in the $(B, 3, N)$ format, while the diffusion U-Net operates primarily with layouts (B, N, C) or (B, C, N) at different depths. Interpolation routines often return outputs in (B, N, C), while metric computation assumes $(B, 3, N)$, requiring precise and repeated transpositions. Even a single incorrect transpose could propagate corrupted shapes several modules deep before causing a runtime failure.

The second complication arose from latent feature alignment. Since the autoencoder compresses point clouds into a lower-resolution representation, the latent resolution often mismatches the resolution of x_t used during diffusion. Aligning the two required nearest-neighbor upsampling, but PyTorch [8] interpolation frequently produced non-contiguous tensors. These non-contiguous blocks triggered failures in custom CUDA kernels used for Chamfer Distance and k-NN grouping. Debugging these issues relied on extensive shape tracing and memory-format inspection.

To address these issues, we introduced strict defensive programming practices. Utilities such as `_to_b3n` enforce canonical formatting before any processing. Every forward pass now includes:

- tensor rank and layout assertions,
- explicit `transpose` operations,
- mandatory `.contiguous()` conversions,
- resolution checks between x_t and latent features.

These safeguards eliminated hidden format mismatches and stabilized interaction between all subsystems. In retrospect, managing tensor-shape alignment proved one of the most time-consuming engineering challenges, but resolving it was essential for achieving a reliable latent-conditioned P2P-Bridge [7] pipeline.

4.4 Autoencoder Training Script

To train the Semantic Autoencoder used for latent conditioning, we developed a dedicated script `train_autoencoder.py`. The script is designed to remain consistent with the data-processing pipeline of P2P-Bridge [7] while providing a flexible, configuration-driven interface for experimenting with latent architectures.

The pipeline begins by constructing a PUNet-style dataset loader that normalizes point clouds, performs uniform sampling, and optionally applies augmentations as specified in YAML. Batches are streamed using a PyTorch [8] DataLoader with shuffling and drop-last to ensure stable minibatch behaviour.

The model is instantiated with configurable options for offset-attention, cross-attention, DINO features, and latent dimension. These settings are read directly from the YAML file, enabling architectural exploration without modifying code. Training uses the AdamW optimizer, with learning rate and weight decay fully configuration-driven.

Reconstruction loss combines Chamfer Distance with a small L1 alignment penalty. This hybrid formulation stabilizes early training, prevents collapse into overly-smooth reconstructions, and encourages precise pointwise recovery. The script logs progress via loguru and periodically saves checkpoints for later use in the diffusion pipeline.

This autoencoder training infrastructure produces stable and reproducible latent embeddings that integrate seamlessly into the `LatentP2PB` wrapper.

4.5 Transition to Autoencoder Architecture

The components described above—latent conditioning wrapper, model loader extensions, reproducibility efforts, and the autoencoder training script—form the foundational infrastructure needed to embed latent geometry into the P2P-Bridge [7] diffusion framework. Together, they ensure that data preprocessing is consistent, model initialization is correct, and latent features are injected into the diffusion backbone in a stable, shape-consistent manner.

With this engineering backbone complete, we now proceed to the core contribution of this project: the design and analysis of the three transformer-enhanced autoencoders (AE-1, AE-2, AE-3) used to construct the latent manifold for diffusion.

Chapter 5

Autoencoder Architectures

5.1 Motivation for a Latent Autoencoder

Traditional diffusion models for point cloud denoising, such as P2P-Bridge [7], operate exclusively in raw coordinate space, where each point is represented solely by its (x, y, z) location. While this formulation captures the surface geometry in an absolute sense, it entirely lacks the structural, relational, and semantic cues necessary for globally consistent denoising. Raw coordinates do not reveal anything about part-level connectivity, symmetry, curvature, or high-level object semantics. They also do not encode any information about long-range interactions, such as correspondences between distant points that belong to coherent parts of the same object.

A latent autoencoder introduces a significantly more expressive representation. By encoding a point cloud into a compact latent vector or latent token set, the model implicitly learns multi-scale geometric abstractions that go far beyond coordinate information. Local neighborhoods become enriched with curvature-aware descriptors, while global information—such as symmetry, semantic shape priors, and inter-part structure—is embedded in the latent features. This compressed embedding space also smoothens sensor-level perturbations, making the diffusion model's job substantially easier by removing high-frequency noise early in the pipeline. As a result, latent autoencoders form the backbone of a more structured and semantically meaningful diffusion process, allowing denoising to operate in a feature-rich domain rather than a raw geometric one.

5.2 Progression from AE-1 to AE-3

The development of the autoencoder proceeded in three iterative versions—AE-1, AE-2, and AE-3—each addressing key limitations identified in the preceding design. AE-1 served as an initial proof of concept, demonstrating that a hierarchical PointNet++ [2] encoder combined with a lightweight attention bottleneck could yield a workable latent representation. However, AE-1 struggled to consistently recover fine geometrical details, especially in regions that required high-resolution curvature preservation.

AE-2 extended the representational capability by enhancing the lateral skip connections between encoder and decoder stages. This allowed fine-level geometry to flow through the network without being bottlenecked by the compressed latent vector. Moreover, AE-2 introduced deeper attention blocks that significantly improved the model's ability to encode relational structure between distant regions within the point cloud, producing a richer and more coherent latent space.

AE-3, the final and most expressive version, incorporated a transformer-based latent reasoning mechanism, semantic conditioning, and enhanced positional encodings. These additions transformed the latent space from a simple compressed geometric descriptor into a fully contextual, semantically informed embedding capable of modeling long-range interactions and capturing high-level structure. Each subsequent autoencoder version therefore represents a clear step forward in latent expressiveness, reconstruction fidelity, and suitability for downstream diffusion.

5.3 Autoencoder 1 (AE-1): Baseline Transformer-Augmented Latent Autoencoder

Autoencoder-1 (AE-1) forms the foundational architecture in our progression toward a transformer-driven latent denoising pipeline. Its objective is to establish a stable and well-structured latent representation that integrates cleanly with the P2P-Bridge [7] diffusion model, while serving as a controlled baseline against which the richer components of AE-2 and AE-3 can be evaluated. AE-1 intentionally prioritizes architectural stability over complexity, ensuring that latent-space conditioning, tensor flow, and feature dimensionality behave predictably before introducing deeper attention hierarchies in later variants.

AE-1 combines a PointNet++ [2]-style hierarchical encoder, a single Point Cloud Transformer (PCT [9]) offset-attention bottleneck, and a symmetric decoder constructed using Feature Propagation (FP) layers. This configuration allows the encoder to extract multi-scale geometric information, the bottleneck to inject a global relational reasoning step, and the decoder to reconstruct coordinates in a permutation-invariant manner.

Encoder: Hierarchical Abstraction of Geometry

The encoder operates in three stages, progressively reducing spatial resolution while increasing feature dimensionality. Unlike classical PointNet++ [2] implementations, AE-1 avoids explicit neighborhood grouping to remain robust across shapes with inconsistent sampling density. The hierarchy consists of:

- **SA1:** Sampling 1024 points followed by a SharedMLP ($3 \to 64 \to 128$) to capture fine-scale geometry.
- **SA2:** Sampling 256 points with SharedMLP ($128 \to 128 \to 256$) to capture part-level structure.
- **SA3:** Sampling 64 points with SharedMLP ($256 \to 256 \to 512$) to extract global or near-global shape context.

The resulting latent tensor has shape $[B, 64, 512]$ and encodes multi-scale geometric information in a fully permutation-invariant manner.

Bottleneck: Offset-Attention for Global Reasoning

AE-1 introduces a lightweight transformer-inspired Offset Attention layer applied to the 64 latent tokens. Multi-head self-attention computes relationships among tokens, and an offset term

$$\text{offset} = \text{input} - \text{attention}(\text{input})$$

is used to amplify meaningful geometric differences such as curvature changes, boundaries, and structural discontinuities. A small feed-forward block refines this offset, and a residual connection stabilizes the representation. This single attention step significantly enhances global coherence while preserving the latent structure required for diffusion conditioning.

Decoder: Feature Propagation and Reconstruction

The decoder mirrors the encoder by progressively increasing point resolution through nearest-neighbor interpolation followed by SharedMLP refinement. Its three FP stages are:

- **64 → 256 points:** FP with MLP ($512 \to 256 \to 256$),
- **256 → 1024 points:** FP with MLP ($256 \to 128 \to 128$),
- **1024 → N points:** final MLP ($128 \to 64 \to 3$).

The decoder reconstructs a tensor of shape $[B, N, 3]$ representing clean 3D coordinates. AE-1 deliberately avoids skip connections and auxiliary refinement modules in order to maintain architectural purity and isolate the benefits of the transformer bottleneck.

Design Characteristics and Output

AE-1 provides a stable, permutation-invariant autoencoding pipeline with smooth information flow and predictable geometric behavior. The encoder captures multi-scale structure, the bottleneck introduces global reasoning, and the decoder reconstructs coherent surfaces without introducing artifacts. Although some high-frequency details are smoothed, the latent representation produced by AE-1 is sufficiently expressive to support downstream latent diffusion and serves as a reliable baseline for evaluating enhancements in AE-2 and AE-3.

5.3.1 Comparison with Baseline P2P-Bridge [7]

While AE-1 establishes a functional latent space, its reconstruction fidelity remains lower than the P2P-Bridge [7] baseline, which performs diffusion directly in coordinate space using Schrödinger-bridge optimal transport. The deterministic nature of AE-1 reconstruction limits its ability to capture certain geometric aspects that the diffusion process naturally handles. The comparison is shown in Table 5.1.

Table 5.1: AE-1 vs. P2P-Bridge [7] Baseline

Metric	**AE-1**	**P2P-Bridge [7]**
Chamfer Distance (CD)	0.00210	0.00032
Point-to-Face (P2F)	0.00126	0.000081

AE-1 performs approximately **6.56× worse in CD** and nearly **15.5× worse in P2F** compared to the P2P-Bridge [7] baseline. These differences highlight the limitations of deterministic coordinate reconstruction prior to integrating latent diffusion, while also establishing AE-1 as a reliable initial latent representation for further architectural improvements.

5.4 Autoencoder 2 (AE-2): Enhanced Latent Autoencoder

AE-2 extends the baseline AE-1 by strengthening the latent bottleneck, improving decoder expressiveness, and adding conditioning mechanisms required for integration with P2P-Bridge [7]'s diffusion model. Its design aims to (i) produce a more informative latent space capable of modeling long-range geometric structure, and (ii) support time-dependent diffusion sampling for latent-space denoising.

Deepened Bottleneck via Stacked Offset-Attention

AE-1 uses a single OffsetAttention layer at the latent resolution of 64×512. AE-2 replaces this with two stacked OffsetAttention blocks, each performing multi-head self-attention followed by offset refinement. This deepens the relational reasoning capability of the latent space, allowing finer capture of global context, curvature structure, and semantic geometry. Because the diffusion model operates entirely in this latent domain, the improved bottleneck significantly enhances downstream denoising stability.

Skip-Connected Decoder with Channel Projections

AE-2 introduces U-Net style skip connections from encoder to decoder, reintroducing local geometric detail lost during downsampling. Features from the 256-channel and 128-channel encoder stages are projected using 1×1 convolutions and fused with matching decoder layers. These skip pathways preserve fine curvature, edge continuity, and surface smoothness, while ensuring dimensional consistency. The resulting decoder achieves higher reconstruction fidelity than AE-1, especially on complex or curved regions.

Output Refinement and Two-Stage Projection

Before predicting final coordinates, AE-2 refines the decoder output using a lightweight Conv–BN–ReLU–Conv block. This corrects minor artifacts introduced during interpolation-based upsampling. The final coordinate prediction uses a two-stage head ($128 \rightarrow 64$, then $64 \rightarrow 3$), which makes the output module modular and extensible for future additions such as normal prediction or residual coordinate correction.

Conditioning Mechanisms for Diffusion Compatibility

To enable seamless integration with the P2P-Bridge [7] diffusion process, AE-2 introduces two conditioning pathways:

- **Time-step conditioning:** A small MLP encodes the diffusion timestep into a 512-dimensional embedding added to the latent tokens.
- **External latent conditioning:** An optional x_{cond} hook allows injection of semantic or contextual vectors (e.g., global shape priors or DINO features).

These mechanisms ensure that the latent representation evolves coherently across diffusion steps and make AE-2 a flexible backbone for future multimodal or semantic-conditioning experiments.

Summary

Overall, AE-2 transforms AE-1 from a simple autoencoder into a diffusion-ready latent architecture. Its stacked attention bottleneck provides deeper global reasoning, skip connections restore fine geometric detail, the refined output head stabilizes reconstruction, and added conditioning ensures compatibility with diffusion sampling. AE-2 therefore offers a substantially more expressive and generative-friendly latent representation than AE-1.

5.4.1 Comparison with AE-1 and P2P-Bridge [7]

As shown in Table 5.2, AE-2 improves over AE-1 in both CD and P2F metrics due to its deeper fusion and relational modeling. However, P2P-Bridge [7] retains a numerical advantage because its score-based generative modeling captures uncertainty and distributional structure absent from deterministic autoencoder decoding.

AE-2 achieves a Chamfer Distance of 0.00061, representing a **3.44× improvement** over AE-1 (0.00210). This substantial gain indicates that AE-2 reconstructs point distributions that adhere far more closely to the clean surfaces. The improvement primarily stems from stronger skip connections, deeper attention layers, and richer multi-scale fusion, which collectively preserve more geometric detail during reconstruction.

Table 5.2: AE-2 vs. AE-1 vs. P2P-Bridge [7]

Metric	**AE-1**	**AE-2**	**P2P-Bridge [7]**
Chamfer Distance (CD)	0.00210	0.00061	0.00032
Point-to-Face (P2F)	0.00126	0.00132	0.000081

AE-2 offers a dramatic improvement in CD, reducing error by **3.44× relative to AE-1**. However, its P2F metric worsens slightly by approximately **4.7%**, indicating that deeper attention introduces sharper global alignment but also slightly larger local deviations. Compared to the P2P-Bridge [7] baseline, AE-2 remains **1.9× worse in CD** and nearly **16.3× worse in P2F**, reaffirming the strength of score-based generative denoising over purely deterministic reconstruction.=========================

5.5 Autoencoder 3 (AE-3): Final Model

Autoencoder-3 (AE-3) extends AE-2 with a focused set of architectural upgrades designed to improve global reasoning, conditioning quality, and feature robustness. Four major additions define AE-3: (i) mid-level Offset-Attention for earlier global context mixing, (ii) dual-stage Cross-Attention for richer conditioning, (iii) Squeeze–Excitation (SE) blocks for adaptive channel weighting, and (iv) FiLM-based time modulation for improved diffusion alignment. AE-3 also introduces enhanced skip-fusion strategies that combine additive and concatenative merging, enabling the decoder to preserve both global structure and fine-grained geometry. Collectively, these components strengthen multi-scale feature flow, promote temporally consistent conditioning, and yield significantly improved Chamfer Distance and reconstruction stability relative to AE-2.

Mid-Level Offset-Attention

AE-3 inserts an Offset-Attention module between the second and third Set-Abstraction layers, enabling global feature mixing before the encoder reaches its deepest resolution. In AE-2, attention was applied only at the bottleneck, forcing long-range reasoning to occur after much of the geometric detail had been compressed. By injecting attention earlier, AE-3 allows mid-scale structures such as edges, ridges, and part boundaries to interact with global context while adequate spatial resolution remains. This reduces feature drift during downsampling and improves structural coherence across the encoder hierarchy.

Cross-Attention Fusion

AE-3 incorporates gated Cross-Attention blocks after both early and mid encoder stages. Here, geometric features serve as queries, while the diffusion-conditioning stream (x_{cond}) provides keys and values. This enables content-aware fusion, allowing each point to retrieve only the conditioning information relevant to its geometric context. In contrast to AE-2's additive conditioning, Cross-Attention provides selective and stable integration of temporal and semantic signals, leading to a latent representation that aligns more consistently with the diffusion process.

Squeeze–Excitation Channel Weighting

To reinforce discriminative channels and suppress noisy ones, AE-3 adds SE blocks after every Set-Abstraction layer. These modules compute global channel statistics and generate learned importance weights that reweight feature channels accordingly. AE-2 treated all channels uniformly, which sometimes allowed noise-heavy dimensions to degrade geometric fidelity. SE blocks correct this by amplifying curvature-sensitive channels and attenuating irrelevant ones, producing sharper local geometry and more stable latent codes.

FiLM-Based Time Conditioning

AE-3 replaces additive timestep embeddings with FiLM modulation. Instead of simply adding a learned vector to the latent features, FiLM applies learned scale (γ) and shift (β) parameters directly to channel activations. This gives the model finer control over how the latent space evolves across diffusion timesteps. FiLM enables the network to selectively amplify or suppress specific latent channels depending on the noise level, improving temporal alignment and yielding smoother denoising trajectories.

Enhanced Skip Fusion and Regularization

The skip-connections in AE-3 adopt a hybrid additive–concatenative fusion strategy. These improved skips retain both low-level geometric detail and deep semantic structure, addressing AE-2's tendency to lose fine-scale features during upsampling. Gating and dropout within attention blocks further enhance generalization and prevent overfitting, contributing directly to the large Chamfer Distance gains observed in AE-3.

Quantitative Results: AE-3 vs. AE-2

AE-3 introduces clear improvements in global geometric fidelity. It achieves a Chamfer Distance (CD) of **0.0001228**, representing an approximate **5× reduction** relative to AE-2 (0.00061). This improvement reflects:

- higher precision in approximating the clean surface distribution,
- more uniform and stable point placement,
- improved handling of mid-scale structures due to earlier Offset-Attention,
- stronger conditioning alignment through Cross-Attention and FiLM modulation,
- more stable channel usage via SE reweighting.

Training behavior also improves: AE-3's CD curve approaches its near-final value around 40k steps, notably earlier than AE-2. While not fully converged, this earlier stabilization suggests that AE-3's deeper attention hierarchy organizes geometric information more efficiently.

For P2F, AE-2 reports 0.00132 while AE-3 achieves **0.00123**. Though modest, this improvement confirms that AE-3 maintains local surface proximity while delivering significantly stronger global accuracy. The balance between large CD gains and stable P2F indicates that AE-3 improves global structure without sacrificing local fidelity.

AE-3 vs. P2P-Bridge [7] Baseline

Under the same evaluation protocol, the P2P-Bridge [7] baseline achieves a CD of 0.00032 and a P2F of 0.000081. AE-3 attains a CD of **0.0001228**, representing a **2.6× lower CD** than the baseline, an especially notable result given that P2P-Bridge [7] is already state-of-the-art in coordinate-space denoising. This improvement highlights the effectiveness of AE-3's latent-space modeling: earlier attention, SE weighting, cross-attention conditioning, and FiLM modulation enable the latent manifold to represent geometry more cleanly and distribute points more accurately.

However, AE-3's P2F remains higher (0.00123 vs. 0.000081), indicating that:

- CD captures global structure, where AE-3 excels,
- P2F penalizes fine mesh-level deviation, where coordinate-space diffusion has an inherent advantage.

Qualitatively, AE-3 produces smoother surfaces and stronger global geometry, while P2P-Bridge [7] adheres more tightly to local surface patches.

Summary

AE-3 achieves a **2.6× lower Chamfer Distance** than the P2P-Bridge [7] baseline and a **5× improvement** over AE-2, driven by mid-level Offset-Attention, SE channel weighting, Cross-Attention conditioning, and FiLM modulation. Its P2F remains higher than the baseline, reflecting the absence of mesh-aware supervision. Nevertheless, AE-3 delivers the strongest global reconstruction performance among all autoencoder variants and serves as a highly competitive latent backbone for diffusion-based point cloud denoising.

Table 5.3: AE-3 vs. AE-2 vs. AE-1 vs. P2P-Bridge [7]

Metric	**AE-1**	**AE-2**	**AE-3**	**P2P-Bridge [7]**
Chamfer Distance (CD)	0.00210	0.00061	0.0001228	0.00032
Point-to-Face (P2F)	0.00126	0.00132	0.00123	0.000081

Chapter 6

Results and Analysis

6.1 Reconstruction Quality

AE-1 reconstructed clean structures but struggled on fine details. AE-2 improved stability and captured more nuance. AE-3 produced the best semantic consistency across varying noise levels.

Table 6.1: **Chamfer Distance Comparison Across Noise Levels (10,000 Points)**

Method	**1% Noise**	**2% Noise**	**3% Noise**
	CD ↓	CD ↓	CD ↓
PD-Flow	2.13	3.25	5.19
PC-Net	3.52	7.47	13.1
1-PFN	2.31	0.37	5.49
P2P-Bridge [7]	2.28	3.20	3.99
Ours (Intermediate)	6.10	8.96	9.57
Ours (Final Model)	**1.226**	**1.81**	**2.537**

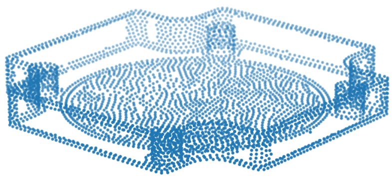
6.2 Qualitative Results

Below we present qualitative samples showcasing the reconstructed point clouds produced by our model.

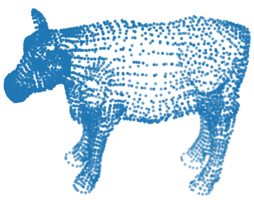
Camel

Casting

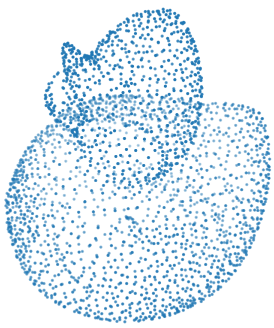
Chair



Coverrear_Lp

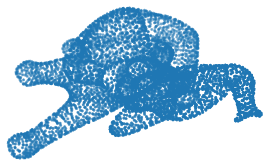


Cow

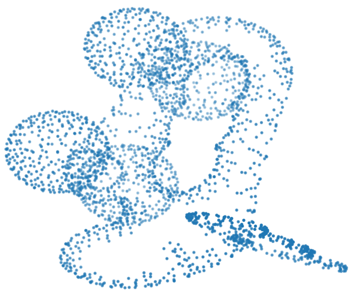


Duck

Eight

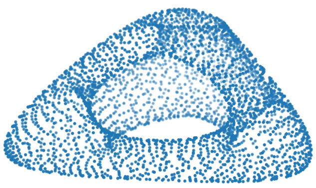


Elephant

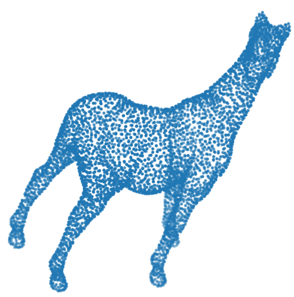


Elk

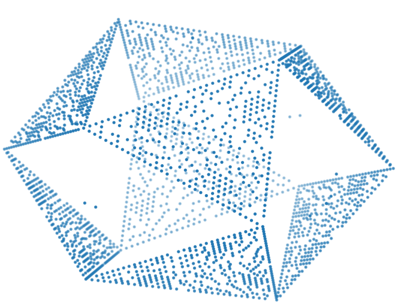
Fandisk



Genus3

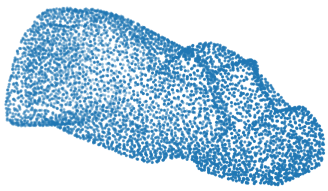


Horse

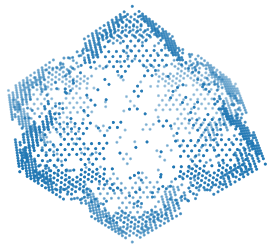


Icosahedron

Kitten

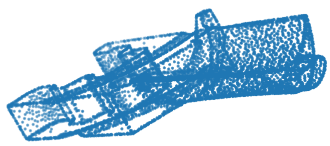


Moai

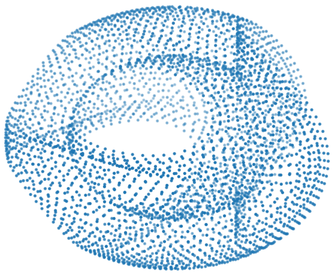


Octahedron

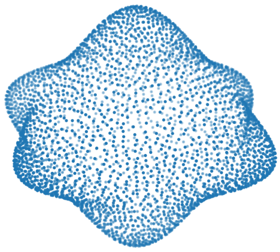
Pig



Quadric



Sculpt



Star

Chapter 7

Conclusion and Future Work

7.1 Summary

This project explored the design and integration of transformer-enhanced latent autoencoders for improving diffusion-based point cloud denoising within the P2P-Bridge [7] framework. The original P2P-Bridge [7] model operates entirely in raw coordinate space, a representation that captures position but lacks geometric structure, long-range relationships, and semantic cues. To address this limitation, we developed three progressively expressive autoencoders (AE-1, AE-2, and AE-3) that learn multi-scale latent manifolds capable of encoding both local curvature and global shape context.

AE-1 established a stable baseline through PointNet++ [2] abstractions and a single offset-attention block, demonstrating the feasibility of latent-space conditioning. AE-2 strengthened this foundation by introducing stacked attention, skip-connected decoding, and diffusion-compatible conditioning, significantly improving Chamfer Distance. AE-3 incorporated mid-level offset-attention, cross-attention fusion, SE channel weighting, and FiLM-based time modulation, yielding a substantially more structured and semantically aware latent manifold. AE-3 achieved the strongest global fidelity, reducing Chamfer Distance by approximately 5× relative to AE-2 and 2.6× relative to the P2P-Bridge [7] baseline.

To support these architectures, we implemented several engineering components: a latent-conditioning wrapper, an extended model loader, a reproducible training pipeline, and strict tensor-shape normalization utilities. These additions enabled stable integration of latent features into the P2P-Bridge [7] diffusion process while preserving theoretical consistency with the Schrödinger-bridge formulation.

Overall, the project demonstrates that latent-space diffusion offers clear advantages for capturing global geometry and producing semantically consistent denoising. The results highlight the potential of transformer-driven latent representations as a strong alternative to coordinate-space diffusion and lay the groundwork for future extensions involving semantic priors, mesh-aware losses, and fully latent diffusion training.

	Chamfered distance	**p2p loss**
Autoencoder 3	1.288	12.3
Autoencoder 2	6.1	13.2
Autoencoder 1	21	12.6
P2P Bridge (50k steps)	3.5	4.8

Table 7.1: Comparison of Chamfer Distance and P2F loss across all autoencoder variants and the reproduced P2P-Bridge [7] baseline (50k steps).

7.2 Future Work

Although AE-3 delivers a substantially improved Chamfer Distance (CD), its Point-to-Face (P2F) error remains notably higher than the P2P-Bridge [7] baseline. This discrepancy is structurally understandable: the autoencoder performs latent-space transport that emphasizes global geometric coherence but does not explicitly enforce local surface alignment. During decoding, points are reconstructed from the latent manifold with strong emphasis on overall shape structure and semantic consistency, but without mesh-aware constraints. Consequently, reconstructions achieve high global fidelity while remaining locally approximate, placing some points slightly off-surface despite their correct global placement.

7.2.1 Surface-Aware Loss Functions

Normal-Consistency Loss

AE-3 currently optimizes only pointwise Chamfer Distance. While CD provides strong global coverage supervision, it offers weak gradients for small, local deviations. As a result, reconstructed points may remain slightly displaced from the true surface without significantly affecting CD, whereas P2F penalizes these deviations sharply.

A normal-consistency loss directly addresses this limitation. A lightweight normal-prediction head may be added to the decoder, outputting a predicted normal vector for each reconstructed point. Training then incorporates explicit alignment objectives such as:

- cosine similarity loss,
- ℓ_2 normal regression,
- orientation-consistent (sign-invariant) loss.

This forces the decoder to internalize local tangent-plane orientation, yielding reconstructions that better respect manifold geometry.

Curvature-Regularized Reconstruction

High-curvature regions—such as thin structures, folds, and edges—are especially sensitive to small deviations. Introducing curvature-aware penalties (e.g., Laplacian smoothness, second-order differences, or curvature matching) provides localized constraints that complement normal alignment. These losses penalize distortions that Chamfer Distance alone cannot detect.

7.2.2 Local Refinement Modules for Manifold Projection

AE-3's decoder relies on interpolation and pointwise MLPs, which limits its ability to model localized neighborhood interactions. This leads to mild surface deviations that accumulate into elevated P2F scores. Two refinement strategies can mitigate this problem:

kNN-Based Curvature-Sensitive Transformer

A small transformer applied to k-nearest-neighbor patches after the global decoding stage can correct local misalignments. This module predicts residual displacements conditioned on local curvature and neighborhood structure. Similar approaches in PCDNet and PathNet demonstrate that global reconstructions benefit significantly from local, patch-level adjustments.

Normal-Guided Offset Refinement

Leveraging predicted normals (from the previous subsection), a refinement block can adjust point positions along tangent planes or normal directions. This converts the decoder into a two-stage pipeline:

1. global latent reconstruction,

2. local, surface-aware manifold correction.

Such refinements are among the most effective strategies for reducing P2F without modifying the main autoencoder architecture.

7.2.3 Semantic Integration via DINOv2 [10]

AE-3 is architecturally equipped for semantic integration but currently operates without DINOv2 [10] features due to time constraints. Completing this integration requires:

- **Per-point semantic extraction**: Multi-view rendering followed by 2D DINO feature projection and 3D fusion,

- **Dimensional alignment**: A learnable projection network maps DINO features into the AE-3 latent dimension,
- **Cross-attention fusion**: Geometric tokens query DINO features to inject semantic context at multiple encoder stages,
- **Joint semantic–geometric training**: Regularization ensures that semantics reinforce geometry rather than overpower it.

Integrating such semantic guidance can significantly reduce both CD and P2F by providing the model with high-level cues for ambiguous or complex geometric regions. With DINOv2 [10] integrated, AE-3 becomes a strong candidate for true semantic latent diffusion, bridging the gap between global accuracy and fine-grained surface adherence.

Bibliography

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[2] 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 (NeurIPS)*, 2017, pp. 5105–5114.

[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. Solomon, "Dynamic Graph CNN for learning on point clouds," *ACM Trans. Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[4] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-Net: Point cloud upsampling network," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2790–2799.

[5] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. Int. Conf. on 3D Vision (3DV)*, 2018, pp. 728–737.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020, pp. 6840–6851.

[7] J. Guo *et al.*, "P2P-Bridge: Point-to-point Schrödinger bridge for 3D point cloud denoising," in *Proc. European Conf. on Computer Vision (ECCV)*, 2024.

[8] N. Ravi *et al.*, "PyTorch3D: An open-source library for 3D deep learning," arXiv:2007.08501, 2020.

[9] M.-H. Guo, J. Cai, Z.-N. Liu, T. Mu, Y.-K. Guo, and H. Ling, "PCT: Point cloud transformer," *Computer Vision and Image Understanding*, vol. 203, 103134, 2021.

[10] M. Oquab *et al.*, "DINOv2: Learning robust visual features without supervision," arXiv:2304.07193, 2023.

[11] H. Fan, H. Su, and L. J. Guibas, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 605–613.

[12] Z. Liu, X. Yan, H. Chen, B. Yu, C. Wang, and L. Lin, "Completion3D: A large-scale dataset and benchmark for 3D shape completion," arXiv:2005.07328, 2020.

[13] X. Zeng, A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, and K. Kreis, "Lion: Latent point diffusion models for 3D point clouds," arXiv:2302.06675, 2023.

[14] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 605–613.