Feedforward 3D Shape Generation: Aligning Semantic Latents with Geometric Spaces

MSc Data Science and Artificial Intelligence Technology

Xingyu Han



Delft University of Technology



TU Delft

Feedforward 3D Shape Generation: Aligning Semantic Latents with Geometric Spaces

by

Xingyu Han

Supverisors:	Dr. Petr Kellnhofer
Thesis Committee:	Dr. Petr Kellnhofer & Dr. Xucong Zhang
Project Duration:	December 2025 - August 2026
Faculty:	Electrical Engineering, Mathematics and Computer Science (EEMCS)

Cover: AI-generated image created with ChatGPT (OpenAI), based on the author's design concept, 2026.

TU Delft

Abstract

Current text-to-3D generation frameworks are bottlenecked by extreme computational demands, relying heavily on expensive test-time optimization or massive datasets to learn multi-modal alignments. This research investigates exploiting the highly structured semantic latent space of pre-trained Large Language Models (LLMs) to directly predict continuous 3D shapes, bypassing the need to build intricate alignment networks from scratch. To bridge the topological mismatch between language and 3D space, we propose a decoupled, two-stage architecture utilizing continuous Signed Distance Functions (SDFs). First, an auto-decoder optimization framework explicitly aligns the semantic text space with the geometric shape space to ensure high structural fidelity. Second, a mapping network learns the non-linear projection between the frozen LLM text latents and the optimized geometric latents. Extensive evaluations demonstrate that this decoupled paradigm prevents the geometric collapse observed in fully joint, end-to-end training, and confirms that pre-trained LLM embeddings serve as superior geometric priors compared to random initializations. Ultimately, this methodology establishes a stable, optimization-free inference pipeline, enabling feedforward synthesis of 3D geometries directly from raw textual prompts.

Preface

This thesis marks the completion of my MSc program in Data Science and Artificial Intelligence Technology at Delft University of Technology.

I would like to begin by expressing my deepest gratitude to my parents. Their support made it possible for me to pursue both my bachelor's and master's studies abroad, far from home. Throughout these years, they have always been there for me—supporting me through every difficulty I encountered and sharing in the joy of every achievement I made. Their constant encouragement and unconditional support have given me the confidence to keep moving forward, and I am truly grateful for everything they have done for me.

I would also like to sincerely thank my supervisor, Dr. Petr Kellnhofer, for his continued guidance, insightful feedback, and support throughout my thesis project. His advice has played an essential role in shaping this work and helping me learn how to conduct research throughout the process. I am also grateful to Dr. Xucong Zhang for serving as a member of my thesis committee and for taking the time to evaluate my work.

Finally, I would like to thank all the friends I met during my five years in Delft. The experiences we shared, both within and beyond university life, have made these years truly special. Regardless of where the future takes me, my five years in Delft will always remain a treasured memory and an invaluable chapter of my life.

Xingyu Han
Delft, August 2026

Contents

Abstract i

Preface ii

1 Introduction 1

2 Background and Related Work 3
2.1 3D Shape Representations . . . 3
2.1.1 Explicit Representations . . . 3
2.1.2 Implicit Representations . . . 4
2.2 3D Synthesis Methodologies . . . 4
2.3 Text-to-3D Generation . . . 6

3 Methodology 8
3.1 Problem Definition . . . 8
3.2 Model Architecture . . . 10
3.2.1 Geometric SDF Decoder . . . 10
3.2.2 Latent Mapper . . . 10
3.3 Training Scheme . . . 10
3.3.1 Stage 1: Auto-Decoder Training (Geometric Alignment) . . . 10
3.3.2 Stage 2: Latent Mapper Training (Feedforward Projection) . . . 11
3.3.3 Inference Pipeline . . . 11

4 Experiments 12
4.1 Datasets and Data Processing . . . 12
4.2 Experiment Setups . . . 12
4.3 Implementation Details . . . 14
4.4 Evaluation Metrics . . . 15

5 Results and Discussion 17
5.1 Quantitative Evaluation . . . 17
5.1.1 Analysis of Semantic Priors . . . 17
5.1.2 Analysis of the Auto-Decoder Architecture . . . 19
5.1.3 Efficacy of the Latent Mapper . . . 19
5.1.4 Decoupled vs. End-to-End Training . . . 20
5.1.5 Assessing the Optimization Upper Bound . . . 20

6 Conclusion 22
6.1 Limitations and Future Work . . . 22
6.2 Relation to the Broader CS and DSAIT Field . . . 23
6.3 Societal Impact and Temporal Context . . . 23
6.4 Overall Conclusion . . . 23

References 25

A Supplementary Material 28
A.1 Use of AI Tools . . . 28
A.2 Additional Results . . . 28

1

Introduction

The generation of three-dimensional content stands as a cornerstone of modern digital technology, playing an increasingly vital role across a wide spectrum of applications ranging from industrial automation and autonomous driving to game development, virtual reality, and the creative workflows of digital artists. Unlike 2D images, which project a flat perspective of our surroundings, the 3D format naturally aligns with human perception of physical reality. It preserves spatial topology, depth, and structural continuity, allowing users to interact with objects and environments from arbitrary viewpoints. However, the manual creation of high-fidelity 3D assets remains a notorious bottleneck in the production pipeline, demanding extensive expertise, expensive software suites, and hours of meticulous labor by specialized artists. Consequently, automating this creation process via generative models has emerged as an essential endeavor in computer vision and computer graphics.

Over the past decade, 3D shape generation has transitioned from traditional procedural modeling to data-driven machine learning methodologies. Early deep learning approaches leaned heavily on architectures adopted from the 2D domain. Generative Adversarial Networks (GANs) [1] and Variational Autoencoders (VAEs) [2] initially pioneered the synthesis of explicit 3D structures like voxel grids and point clouds, though they frequently suffered from scalability constraints or a lack of surface topological connectivity. The introduction of implicit neural representations, most notably continuous signed distance fields, offered a breakthrough in capturing watertight geometries with arbitrary resolution.

Recently, the field has experienced a profound shift with the emergence of Large Language Models (LLMs) and Diffusion Models [3], whose immense generative capacities have pushed the boundaries of conditional 3D synthesis. Modern architectures have successfully leveraged these developments to generate complex assets from flexible inputs; for instance, autoregressive sequence models like ShapeFormer [4] encode geometries into discrete embeddings, while modern frameworks like ShapeGPT [5], Point-E [6], Shap-E [7], and DreamFusion [8] have demonstrated the ability to produce highly diverse shapes and texturized scenes directly driven by natural language text.

Despite these remarkable strides, contemporary state-of-the-art text-to-3D generation frameworks present significant limitations. A primary drawback is their extraordinary requirement for computational resources. Many optimization-based approaches rely on computing expensive score distillation gradients through thousands of rendering iterations at inference time, while native 3D diffusion or transformer-based networks require massive multi-modal training datasets to establish the cross-modal relationship between text and geometry from scratch.

This raises a fundamental question: given that pre-trained LLMs already possess a highly structured and expressive latent space capable of understanding rich language semantics, can we directly exploit these pre-existing text latents as conditioning inputs to predict continuous 3D shapes without building intricate multi-modal alignment networks from the ground up?

To investigate this hypothesis, we choose Signed Distance Functions (SDFs) [9] as our underlying 3D representation due to their continuous mathematical formulation and topological flexibility. We first

construct a baseline coordinate-based regression network that takes a raw, pre-trained LLM text embedding $\mathbf{z}_{\text{LLM}}$ along with a spatial query coordinate $\mathbf{x} \in \mathbb{R}^3$ as inputs to directly predict the corresponding scalar signed distance value $s = f(\mathbf{x}, \mathbf{z}_{\text{LLM}})$. This baseline serves to empirically evaluate how effectively raw textual latents map to continuous spatial structures compared to standard random initializations.

To bridge the structural discrepancy between semantic text embeddings and geometric spatial properties, we further adapt the auto-decoder optimization framework popularized by DeepSDF. By treating the conditional variables as optimization targets, we optimize the text latents during training to force an explicit alignment between the LLM text space and the 3D geometric shape space, improving the surface fidelity and generation quality of the synthesized shapes.

While the auto-decoder alignment strategy yields high-quality reconstructions, it introduces a test-time optimization bottleneck that requires access to the ground-truth 3D shape at inference time—a constraint that renders direct text-to-shape inference impossible. To circumvent this limitation, we train a feedforward Multi-Layer Perceptron (MLP) mapping network. This model learns the non-linear projection between the original, frozen LLM text latents and the optimized geometric latents obtained from the auto-decoder space. By introducing this mapping module, our framework bypasses test-time optimization entirely, enabling instantaneous feedforward generation of explicit 3D SDFs from raw textual prompts without requiring ground-truth shapes during inference.

Finally, we conduct a systematic architectural analysis to evaluate whether an end-to-end training scheme or a decoupled two-stage training paradigm fits better within this cross-modal latent alignment scenario.

In summary, the primary contributions of this work are as follows:

- Evaluation of Pre-trained LLM Latents for Direct Geometric Prediction: We establish a rigorous benchmark evaluating the direct performance of leveraging raw, frozen LLM text embeddings as input conditions to coordinate-based networks for predicting continuous SDF fields, validating its efficacy over random latent benchmarks.
- Auto-decoder Mediated Space Alignment: We adapt an auto-decoder optimization framework to structurally align the semantic text embedding space of LLMs with the continuous 3D geometric space, substantially enhancing the resolution and structural fidelity of the generated objects.
- Optimization-Free Feedforward Inference Pipeline: We introduce and train an MLP-based mapping network that learns to bridge raw textual embeddings and optimized geometric codes, effectively bypassing test-time optimization and removing the requirement of ground-truth geometric inputs during generation.
- Empirical Validation of Multi-Stage Training: We provide a comprehensive comparative analysis of training schemes, demonstrating that a decoupled two-stage training pipeline yields superior stability and geometric reconstruction accuracy compared to a fully joint, end-to-end approach in this setting.

2

Background and Related Work

This chapter reviews the foundational concepts and literature surrounding 3D shape synthesis. It first covers fundamental 3D shape representations, highlighting the distinctions between explicit and implicit formats. Next, it outlines the progression of core 3D synthesis methodologies. Finally, it examines contemporary text-to-3D generation frameworks, exploring how natural language is integrated to drive shape generation and establishing the context for our proposed approach.

2.1. 3D Shape Representations

The choice of 3D representation is a foundational design decision in any generative framework, since it determines how geometry is stored, processed, reconstructed, and ultimately generated. Different representations offer different trade-offs in terms of structural regularity, memory efficiency, computational cost, geometric fidelity, and compatibility with neural architectures. In general, 3D representations can be divided into two major categories: explicit representations, which directly describe surfaces or volumes, and implicit representations, which define shapes through continuous functions over 3D space.

2.1.1. Explicit Representations

Explicit representations parameterize geometry directly, either as occupied volumetric cells, discrete surface samples, or connected surface elements. Their main advantage is that the represented geometry is directly accessible. However, this explicitness often comes at the cost of memory consumption, irregular structure, or limited geometric continuity.

Voxel Grids. Voxel grids can be understood as the 3D extension of 2D pixels, where a shape is represented on a regular volumetric grid. Each voxel typically stores occupancy, density, color, or other geometric attributes. The regular structure of voxel grids makes them highly compatible with standard 3D convolutional neural networks, which is beneficial for learning local spatial patterns [10, 11]. However, their main limitation is scalability: both memory usage and computational cost grow cubically with resolution. As a result, generating high-fidelity 3D shapes at high resolutions quickly becomes prohibitively expensive unless sparse structures are introduced.

Point Clouds. Point clouds represent a shape as an unordered set of 3D coordinates, often augmented with additional attributes such as color, normals, or semantic labels. Compared with voxel grids, point clouds are significantly more memory-efficient because they store only sampled points near or on the object surface rather than allocating a dense volumetric grid [12, 13]. This makes them well suited for representing sparse geometry and for processing raw 3D sensor data such as LiDAR or depth scans. However, unlike voxel grids, point clouds do not possess a regular neighborhood structure, which makes them less directly compatible with conventional convolutional operations. More importantly, point clouds lack explicit topological connectivity and do not directly define continuous surfaces. Consequently, reconstructing smooth, watertight, and editable geometry from point clouds often requires additional surface reconstruction procedures. Therefore, while point clouds are more compact and flexible than voxel grids, they generally provide weaker guarantees of surface continuity and topological

consistency.

2.1.2. Implicit Representations

In contrast to explicit representations, implicit representations define shapes through continuous functions rather than through discrete geometric primitives. A surface is usually recovered as a level set of a learned or analytical function. This formulation allows implicit representations to model geometry at arbitrary spatial resolution and to avoid the fixed discretization constraints of voxel grids or point clouds.

Neural Radiance Fields (NeRF) [14]. Neural Radiance Fields represent a scene as a continuous volumetric radiance field. Typically, a multi-layer perceptron (MLP) maps a 3D spatial coordinate and a viewing direction to a volume density and an RGB color. This representation has achieved strong performance in novel view synthesis because it jointly models geometry and appearance through differentiable volumetric rendering. However, NeRFs are primarily optimized for photorealistic rendering rather than for explicit geometric reconstruction. As a result, although they can produce visually convincing views, the underlying geometry may be difficult to extract, edit, or manipulate precisely.

Signed Distance Functions (SDF) [9]. Signed Distance Functions provide a continuous implicit representation of 3D geometry by assigning each spatial point a scalar value that describes its signed distance to the object surface $\partial\Omega$. Formally, an SDF f maps any 3D coordinate $\mathbf{x} \in \mathbb{R}^3$ to a real-valued distance:

$$f(\mathbf{x}) = \begin{cases} d(\mathbf{x}, \partial\Omega), & \text{if } \mathbf{x} \text{ is outside } \Omega, \\ 0, & \text{if } \mathbf{x} \in \partial\Omega, \\ -d(\mathbf{x}, \partial\Omega), & \text{if } \mathbf{x} \text{ is inside } \Omega \end{cases} \tag{2.1}$$

where

$$d(\mathbf{x}, \partial\Omega) = \min_{\mathbf{y} \in \partial\Omega} |\mathbf{x} - \mathbf{y}|_2. \tag{2.2}$$

The surface of the object is then defined as the zero-level set:

$$S = \{\mathbf{x} \in \mathbb{R}^3 \mid f(\mathbf{x}) = 0\}. \tag{2.3}$$

The sign of the function indicates whether a point lies inside or outside the object volume, while the magnitude represents the shortest Euclidean distance to the surface.

Compared with voxel grids and point clouds, SDFs provide a more continuous and geometry-aware representation. Unlike voxel grids, they are not limited to a fixed spatial resolution, and unlike point clouds, they define a complete surface implicitly rather than relying on disconnected samples. SDFs are also topology-agnostic, naturally continuous, and compatible with constructive solid geometry operations. These properties make SDFs particularly suitable for high-fidelity shape generation and reconstruction.

Overall, each representation offers a different balance between efficiency and geometric expressiveness. Voxel grids are simple and convolution-friendly but memory-intensive; point clouds are compact and sensor-aligned but lack explicit surface structure; NeRFs are powerful for appearance and view synthesis but less convenient for direct geometry manipulation; and SDFs offer a continuous, compact, and surface-oriented formulation.

2.2. 3D Synthesis Methodologies

The development of 3D shape generation has been strongly influenced by the adaptation of generative models from the 2D domain to different 3D representations. Depending on model architectures, existing methods can be broadly categorized into GAN-based, VAE-based, autoregressive, and diffusion-based approaches. Each family of methods introduces a different strategy for learning the distribution of

3D shapes, while also exposing different trade-offs in terms of scalability, geometric fidelity, semantic controllability, and computational cost.

Generative Adversarial Networks (GANs) [1]. GANs were among the earliest deep generative frameworks applied to 3D shape synthesis. Early methods such as r-GAN [15] and 3D-GAN [16] directly generated explicit 3D representations, including point clouds and voxel grids. These approaches demonstrated that adversarial learning could be extended from image generation to 3D geometry generation. However, explicit volumetric representations suffer from severe scalability limitations, since memory usage and computational cost grow cubically with spatial resolution. As a result, high-resolution 3D generation becomes difficult when dense voxel grids are used.

To alleviate the limitations of explicit representations, later GAN-based methods explored implicit geometric representations. For example, SDF-StyleGAN [17] generates shapes through signed distance fields, enabling a more continuous and resolution-independent formulation of geometry. EG3D [18] further improves computational efficiency and 3D awareness by utilizing a hybrid explicit-implicit network architecture that maps learned features onto 3D tri-planes.

Overall, GAN-based methods provide an important foundation for 3D synthesis, but they often require careful training and may struggle with mode collapse, structural consistency, and high-resolution geometry.

Variational Autoencoders (VAEs) [2]. VAEs introduced a probabilistic formulation for 3D generation by learning a latent distribution from which new shapes can be sampled. Instead of directly mapping random noise to geometry, VAEs encode input data into a latent space and decode latent samples back into 3D structures. This makes them particularly suitable for learning compact and structured shape representations.

Several VAE-based methods have been developed for different types of 3D data. SDM-Net [19], for instance, models structured meshes composed of deformable parts, allowing the generated shapes to preserve meaningful part-level organization. TM-Net [20] further extends this idea by generating texture maps in a part-aware manner, thereby combining geometric structure with appearance modeling. With the development of neural scene representations, NeRF-VAE [21] extends variational generation from object-level geometry to scene-level representations by incorporating Neural Radiance Fields.

A particularly influential implicit representation method is DeepSDF [22]. Although it is not a conventional VAE, DeepSDF adopts an auto-decoder formulation in which each shape is associated with an optimizable latent code. Given a latent vector and a spatial coordinate, the network predicts the signed distance value at that coordinate. In this way, DeepSDF learns a continuous signed distance function for each shape, representing the entire 3D volume without requiring a predefined mesh template or a fixed voxel resolution. This decoder-based formulation has strongly influenced later work on implicit 3D representation learning and provides an important foundation for the latent-space optimization of geometry within our proposed method.

Autoregressive Models. Autoregressive models approach 3D generation by decomposing a shape into a sequence of elements and modeling their joint probability as a product of conditional probabilities. This formulation allows complex 3D structures to be generated step by step, making it compatible with sequence models such as Transformers.

PointGrow [23] autoregressively generates point clouds by predicting points sequentially, using self-attention mechanisms to capture dependencies among previously generated points. PolyGen [24] extends autoregressive generation to mesh data by modeling vertices and faces as sequences, allowing the generation of explicit polygonal meshes. ShapeFormer [4] further combines autoregressive modeling with implicit representations by encoding incomplete inputs into feature sequences and using a Transformer [25] to sample a complete sequence from which a deep implicit function can be reconstructed.

More recently, token-based mesh generation has further strengthened the connection between 3D generation and language modeling. MeshGPT [26], for example, quantizes mesh face features into discrete tokens and uses a decoder-only Transformer to generate shape token sequences. These approaches show that once 3D geometry is converted into discrete tokens, powerful sequence models can be directly applied to shape synthesis. However, the quality of the generated geometry depends

heavily on the design of the tokenizer, the expressiveness of the discrete codebook, and the amount of available 3D training data.

Diffusion Models [3]. Diffusion models have recently achieved remarkable success in 3D generation by learning to reverse a gradual noise corruption process. Their iterative denoising formulation enables them to capture complex data distributions and generate diverse, high-quality samples. In the 3D domain, diffusion models have been applied to different representations, including point clouds, voxel grids, meshes, NeRFs, and SDFs.

A major breakthrough in text-guided 3D optimization was the introduction of Score Distillation Sampling (SDS) in DreamFusion [8]. Instead of training a large-scale 3D generative model from scratch, SDS uses a pretrained text-to-image diffusion model as a strong 2D prior. A differentiable renderer is used to render the current 3D representation from multiple views, and the diffusion model provides gradients that guide the rendered images toward the input text prompt. In this way, 2D diffusion priors can supervise the optimization of 3D representations such as Neural Radiance Fields, even when large paired text-3D datasets are unavailable.

Several variants have been proposed to improve the stability and quality of this optimization process. Score Jacobian Chaining (SJC) [27], for example, also leverages pretrained 2D diffusion models to provide optimization signals for 3D generation. These methods demonstrate that strong image priors can be transferred to the 3D domain through differentiable rendering. However, optimization-based methods are often computationally expensive, since they require iterative per-prompt optimization and multi-view rendering.

In parallel, the emergence of Transformer-based diffusion architectures such as DiT [28] has enabled large-scale 3D generative models to be trained more directly. Models such as CLAY [29] use large-scale 3D datasets and diverse conditioning signals to generate high-quality shapes and textures. Compared with earlier optimization-based methods, these models can produce 3D assets more efficiently at inference time. Nevertheless, they typically require substantial training data, large model capacity, and extensive computational resources.

2.3. Text-to-3D Generation

While general 3D synthesis methods establish the geometric foundations for shape generation, text-to-3D generation introduces an additional semantic control mechanism. By conditioning generation on natural language, these methods allow users to create 3D assets from intuitive textual descriptions rather than from manually specified geometric parameters. This direction is particularly important for scalable 3D content creation, since text provides a flexible interface for specifying object categories, attributes, styles, and structural constraints.

Early work such as Text2Shape [30] learns a joint embedding space between textual descriptions and 3D shape geometries, enabling text-conditioned shape retrieval and generation. With the development of diffusion and autoregressive models, recent text-to-3D methods have achieved substantially higher diversity, fidelity, and semantic alignment.

Diffusion-based Text-to-3D. Diffusion-based text-to-3D methods commonly rely on vision-language alignment to connect textual semantics with visual or geometric representations. Contrastive Language-Image Pre-training (CLIP) [31] maps images and text into a shared embedding space, making it possible to compare visual outputs with text prompts in a semantically meaningful way. Early methods such as CLIP-Forge [32] use pretrained CLIP priors to guide 3D shape generation, enabling text-to-shape synthesis without requiring expensive test-time optimization.

Other methods combine 2D diffusion priors with 3D generative models. Point-E [6], for example, first generates text-conditioned synthetic images using a 2D diffusion model and then conditions a 3D diffusion model on these images to produce point clouds. This two-stage design transfers the semantic generation capability of text-to-image models into the 3D domain, while avoiding direct optimization of a 3D representation for each prompt.

Optimization-based approaches provide another important direction. DreamFusion [8] uses SDS to optimize a 3D representation by aligning its multi-view renderings with a text prompt through a pretrained

2D diffusion model. Although this approach avoids the need for large paired text-3D datasets, it can suffer from issues such as slow optimization, limited geometric precision, and view inconsistency. To improve resolution and geometry quality, Magic3D [33] adopts a coarse-to-fine strategy. It first optimizes a coarse neural representation, extracts the underlying geometry as a mesh, and then refines the high-resolution textured mesh using differentiable rendering and SDS.

With the emergence of large-scale annotated 3D datasets such as Objaverse [34], recent large-scale diffusion models such as CLAY [29] are able to generate high-quality 3D assets from diverse conditioning signals. Compared with optimization-based methods, these feed-forward approaches can be more efficient at inference time. However, they usually require large-scale 3D data, extensive training, and careful alignment between text, images, and geometry.

Autoregressive and LLM-based Text-to-3D. Autoregressive methods provide an alternative formulation by converting 3D shapes into discrete token sequences. This discretization makes it possible to apply language-model-style architectures to 3D generation. AutoSDF [35], for instance, learns a quantized codebook for local regions of truncated signed distance functions using a VQ-VAE [36]. A 3D shape can then be represented as a sequence of discrete codebook tokens and generated using a Transformer-based model.

Recent methods further extend this idea by aligning 3D tokens with the token space of large language models. In this unified multimodal paradigm, 3D generation is formulated as a language generation problem, where the model predicts discrete 3D tokens conditioned on textual input. Methods such as ShapeGPT [5], SAR3D [37], and ShapeLLM-Omni [38] demonstrate this direction by leveraging pretrained language models for text-to-shape generation. By mapping 3D geometry into discrete token vocabularies, these methods benefit from the semantic reasoning and sequence modeling capabilities of LLMs.

This formulation provides a promising path toward semantically rich and controllable 3D generation. However, it also introduces several challenges. First, the quality of the generated shape depends heavily on the expressiveness of the 3D tokenizer and codebook. Second, aligning 3D tokens with language tokens often requires large-scale multimodal training. Third, the discretization process may lose fine geometric details, especially when continuous surfaces are compressed into a limited token vocabulary.

Motivation for Our Approach. While diffusion and autoregressive frameworks have achieved remarkable performance in 3D generation, they remain highly resource-intensive, often requiring massive datasets and extensive multi-modal fine-tuning to establish cross-domain alignment from scratch. To circumvent these computational bottlenecks, we reframe the text-to-3D generation problem as a direct feature regression task. Inspired by recent findings demonstrating that pre-trained LLMs inherently function as robust, high-level feature extractors [39], we propose utilizing frozen LLM latents directly as conditional priors to distinguish shapes and guide the geometric synthesis process. To ensure high-fidelity outputs and bypass the resolution constraints typical of explicit structures, we adopt continuous Signed Distance Functions (SDFs) as our regression target. However, because LLMs are optimized primarily for linguistic corpora rather than spatial topologies, an inherent domain gap exists between these semantic embeddings and the 3D geometric space. To successfully bridge this discrepancy, we propose introducing an auto-decoder optimization strategy, adapted from DeepSDF, to explicitly align the semantic and geometric manifolds. The detailed formulation, architectural design, and training scheme of this proposed alignment framework are presented in the following chapter.

3

Methodology

This chapter covers the methodology for our proposed method. We first define the formal representation of the method and describe the model architecure in details. Then we explain the difference of two different training schemes applied to our proposed methods.

3.1. Problem Definition

Our primary objective is to synthesize continuous 3D geometries directly from natural language prompts by leveraging the highly structured semantic space of pre-trained Large Language Models (LLMs). We represent the target 3D geometry as a continuous Signed Distance Function (SDF). Let $\mathbf{x} \in \mathbb{R}^3$ denote a 3D spatial query coordinate, and let $\mathbf{z}_{\text{text}} \in \mathbb{R}^{768}$ represent the frozen semantic latent embedding extracted from an LLM given an input text prompt.

A naive baseline approach would attempt to directly map the text embedding to the geometric space by learning a coordinate-based neural network f_θ, parameterized by θ, such that $s = f_\theta(\mathbf{z}_{\text{text}}, \mathbf{x})$, where s is the predicted signed distance. However, we hypothesize that the semantic language space and 3D geometric space possess inherently different topologies, where forcing a direct mapping leads to suboptimal geometric fidelity.

To bridge this cross-modal domain gap, we adopt an auto-decoder framework inspired by DeepSDF. Instead of directly feeding the frozen LLM latent $\mathbf{z}_{\text{text}}$ into the network, we introduce an optimizable geometric latent vector, $\mathbf{z}_{\text{opt}} \in \mathbb{R}^{768}$. During training, we treat $\mathbf{z}_{\text{opt}}$ as an explicit optimization target alongside the network weights θ. To ensure that the optimized geometric latent remains grounded in the original textual semantics, we apply a regularization constraint that anchors $\mathbf{z}_{\text{opt}}$ to $\mathbf{z}_{\text{text}}$.

The learning problem is therefore formulated as the following joint optimization task:

$$\begin{aligned} \underset{\theta,\, \mathbf{z}_{\text{opt}}}{\arg\min} \quad & \mathcal{L}_{\text{total}} = \mathcal{L}_{\text{SDF}}\left(f_\theta(\mathbf{z}_{\text{opt}}, \mathbf{x}),\, s\right) + \lambda \mathcal{L}_{\text{anchor}}\left(\mathbf{z}_{\text{opt}},\, \mathbf{z}_{\text{text}}\right), \\ & \mathcal{L}_{\text{SDF}}\left(f_\theta(\mathbf{z}_{\text{opt}}, \mathbf{x}),\, s\right) = \left\| f_\theta(\mathbf{z}_{\text{opt}}, \mathbf{x}) - s \right\|_1, \\ & \mathcal{L}_{\text{anchor}}\left(\mathbf{z}_{\text{opt}},\, \mathbf{z}_{\text{text}}\right) = \left\| \mathbf{z}_{\text{opt}} - \mathbf{z}_{\text{text}} \right\|_2^2. \end{aligned} \tag{3.1}$$

Where the components of the formulation are defined as follows:

- f_θ: The learned coordinate-based neural network (the SDF decoder).
- $\mathbf{z}_{\text{opt}}$: The optimizable 3D geometric latent vector corresponding to the specific shape.
- $\mathbf{z}_{\text{text}}$: The frozen LLM latent vector acting as the semantic anchor.
- $\mathbf{x}$: The continuous 3D query point.
- s: The ground-truth signed distance value at point $\mathbf{x}$.

- $\mathcal{L}_{\text{SDF}}$: The primary reconstruction loss measuring the error between the predicted signed distance and the ground-truth value s.
- $\mathcal{L}_{\text{anchor}}$: The regularization term penalizing the deviation of the geometric latent $\mathbf{z}_{\text{opt}}$ from the semantic text embedding $\mathbf{z}_{\text{text}}$.
- λ: The hyperparameter controlling the weight of the anchor regularization.

While this joint optimization yields high-fidelity shape reconstructions, finding $\mathbf{z}_{\text{opt}}$ requires test-time optimization (TTO) using ground-truth SDF values, which prevents direct zero-shot generation from text at inference. Consequently, our secondary objective is to learn a latent transformation mapping—specifically, a Multi-Layer Perceptron (MLP)—that projects $\mathbf{z}_{\text{text}} \rightarrow \mathbf{z}_{\text{opt}}$. By training this transformation model, we can completely bypass test-time optimization, enabling feedforward 3D shape generation purely from text.

The overall framework is illustrated in Figure 3.1, summarizing the two-stage training strategy and the resulting optimization-free text-to-3D inference pipeline.

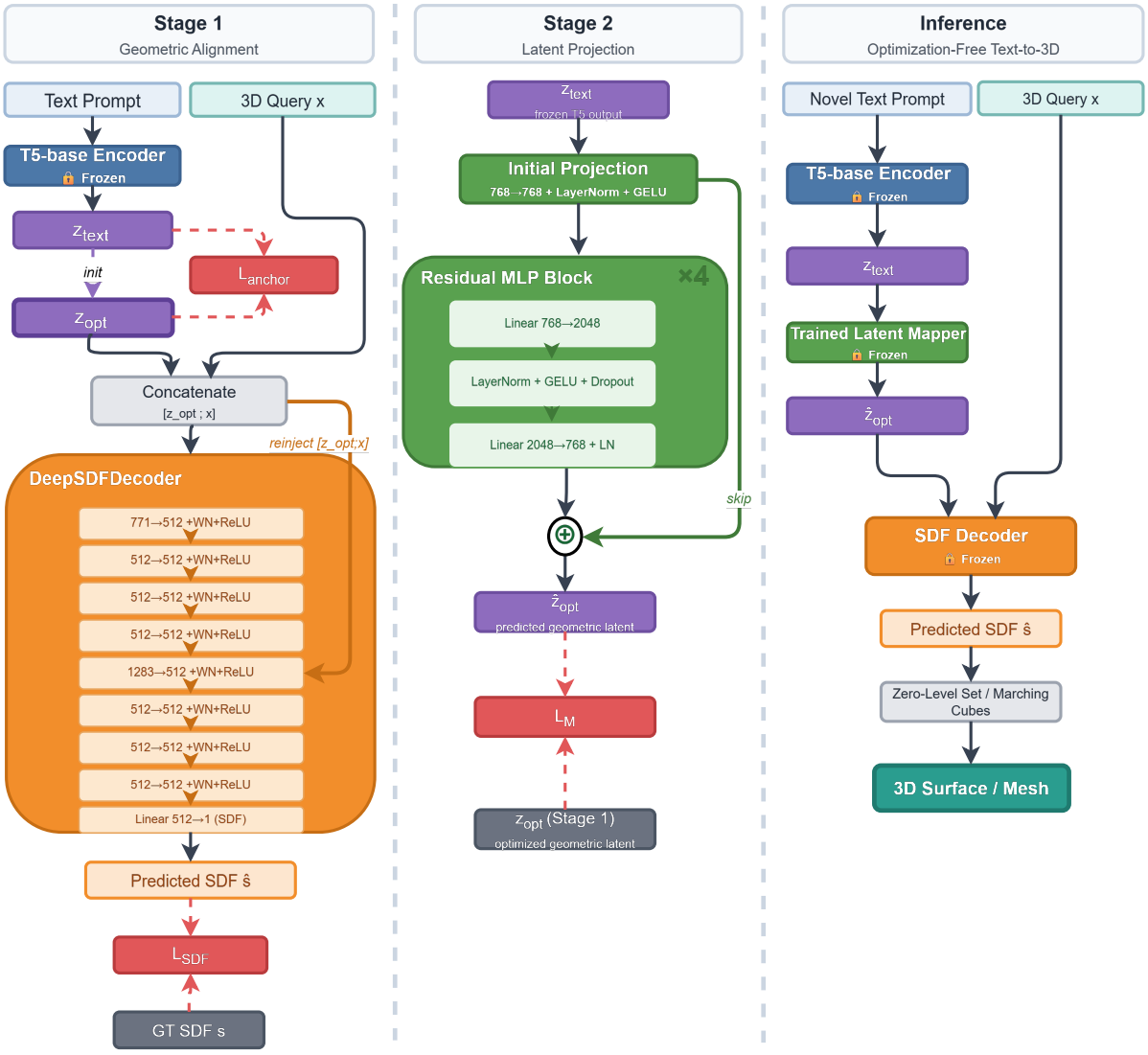


Figure 3.1: Overview of the proposed two-stage text-to-3D generation framework. Stage 1 optimizes the geometric latent space and SDF decoder under semantic anchoring, Stage 2 learns the projection from frozen text embeddings to the optimized geometric latents, and inference combines the trained mapper and decoder to enable optimization-free text-to-3D generation.

3.2. Model Architecture

Our framework consists of two core components as shown in Figure 3.1: an auto-decoder-based geometric SDF decoder and a learnable latent mapping network. The architecture is designed to map high-level semantic embeddings from an LLM into a refined geometric latent space that can be directly decoded into a Signed Distance Function.

3.2.1. Geometric SDF Decoder

We adapt the DeepSDF [22] architecture, which functions as a continuous coordinate-based MLP. Given a geometric latent vector $\mathbf{z}_{\text{opt}} \in \mathbb{R}^{768}$ and a 3D spatial query point $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$, the decoder learns the mapping $f_\theta : (\mathbf{z}_{\text{opt}}, \mathbf{x}) \to \mathbb{R}$.

Our implementation utilizes an 8-layer MLP with weight normalization to stabilize the training of the implicit field. The network adopts a ResNet-style skip connection at the fifth layer, where the concatenated input vector (the latent $\mathbf{z}_{\text{opt}}$ and the spatial coordinate $\mathbf{x}$) is re-injected into the hidden layers. This structural design ensures that the model preserves high-frequency spatial details while conditioning the global geometry on the latent vector, effectively allowing the network to distinguish between different shape topologies based on the latent input.

3.2.2. Latent Mapper

To eliminate the requirement for test-time optimization, we introduce a LatentMapper network, defined as $\mathbf{z}_{\text{opt}} = \mathcal{M}(\mathbf{z}_{\text{text}})$. This mapper acts as a bridge between the frozen, pre-trained LLM semantic latent $\mathbf{z}_{\text{text}}$ and the optimized geometric latent $\mathbf{z}_{\text{opt}}$ required by the decoder. The LatentMapper $\mathcal{M}$ is constructed as a deep residual network comprising:

Initial Projection: A linear layer followed by Layer Normalization and GELU activation to project the raw text embedding into the model's internal feature dimension.

Residual Blocks: A sequence of N ResBlock modules. Each block consists of a two-layer MLP with a bottleneck hidden dimension, Layer Normalization, and Dropout. The residual connection $\mathbf{x} + \text{net}(\mathbf{x})$ ensures stable gradient flow, allowing the network to learn the non-linear transformation from semantic space to geometric space without losing the original textual information.

By combining these modules, our architecture forms an end-to-end inference pipeline. During inference, the LatentMapper $\mathcal{M}$ processes the input text embedding to generate the necessary geometric latent, which is then fed into the DeepSDFDecoder f_θ to query any coordinate $\mathbf{x}$ in the 3D volume. This allows the model to synthesize the final surface as a zero-level set $f_\theta(\mathcal{M}(\mathbf{z}_{\text{text}}), \mathbf{x}) = 0$ in a single feedforward pass.

3.3. Training Scheme

To align the semantic and geometric spaces, we investigate two alternative training strategies: a fully joint end-to-end scheme and a decoupled two-stage scheme. In the joint scheme, the LatentMapper $\mathcal{M}$ and DeepSDFDecoder f_θ are optimized simultaneously, whereas the decoupled scheme separates geometric representation learning from semantic-to-geometric latent projection. We adopt the decoupled strategy for our proposed method, with the hypothesis that separating these two objectives may provide a more stable optimization process for the coordinate-based implicit field. The two training strategies are subsequently compared empirically to evaluate the effectiveness of this design choice.

3.3.1. Stage 1: Auto-Decoder Training (Geometric Alignment)

In the first stage, we exclusively train the DeepSDFDecoder f_θ alongside the geometric latents. For each text-shape pair in the training set, we extract the frozen LLM text embedding $\mathbf{z}_{\text{text}}$. We initialize a corresponding optimizable geometric latent vector $\mathbf{z}_{\text{opt}}$ (typically initialized to match $\mathbf{z}_{\text{text}}$ or drawn from a standard Gaussian distribution).

During the forward pass, spatial points $\mathbf{x}$ are sampled from the unit bounding box and concatenated with $\mathbf{z}_{\text{opt}}$ to predict the SDF value. Through backpropagation, we simultaneously update both the decoder's network parameters θ and the individual geometric latents $\mathbf{z}_{\text{opt}}$ to minimize the joint objective $\mathcal{L}_{\text{total}}$. The primary reconstruction loss $\mathcal{L}_{\text{SDF}}$ enforces geometric accuracy, while the regularization term

$\mathcal{L}_{\text{anchor}}(\mathbf{z}_{\text{opt}}, \mathbf{z}_{\text{text}})$ acts as an elastic tether, ensuring the optimized shape codes do not drift too far from their originating semantic text embeddings.

3.3.2. Stage 2: Latent Mapper Training (Feedforward Projection)

Once Stage 1 converges, the geometric space is fully structured and aligned. We freeze the decoder weights θ and extract the final, optimized geometric latents $\mathbf{z}_{\text{opt}}$ for all training samples. This creates a supervised, paired dataset consisting of the frozen semantic embeddings and their corresponding optimized geometric codes: $\{\mathbf{z}_{\text{text}}^{(i)}, \mathbf{z}_{\text{opt}}^{(i)}\}_{i=1}^{N}$.

We then train the LatentMapper network $\mathcal{M}$ in isolation. The mapper is optimized as a regression task to minimize the distance (e.g., Mean Squared Error) between its prediction and the target geometric latent:

$$\mathcal{L}_{\mathcal{M}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{M}(\mathbf{z}_{\text{text}}^{(i)}) - \mathbf{z}_{\text{opt}}^{(i)} \right\|_2^2 \tag{3.2}$$

By isolating the mapper's training to this second stage, the network strictly focuses on learning the non-linear manifold projection from the LLM space to the auto-decoder space without contending with a moving target.

3.3.3. Inference Pipeline

At test time, the trained LatentMapper $\mathcal{M}$ and DeepSDFDecoder f_θ are connected to form a seamless, end-to-end generative pipeline. Given a novel text prompt, the LLM extracts the semantic embedding $\mathbf{z}_{\text{text}}$. This vector is passed through the mapper to instantaneously yield the approximated geometric latent $\hat{\mathbf{z}}_{\text{opt}}$. The decoder then evaluates this latent across a grid of spatial queries $\mathbf{x}$ to extract the final 3D zero-level set. By relying on the mapper's learned projection, our framework entirely circumvents the computationally expensive test-time optimization process, enabling rapid, zero-shot 3D shape generation purely from text.

4

Experiments

This chapter details the experimental framework designed to evaluate our proposed text-to-3D generation methodology. We first outline the dataset preparation and data processing pipelines required to bridge the textual and spatial domains. Subsequently, we detail the various model configurations and baseline setups used to isolate and assess the contributions of our two-stage architecture. Finally, we provide a comprehensive rationale for our chosen evaluation metrics, demonstrating how a multi-faceted evaluation strategy is necessary to accurately capture both geometric fidelity and generative diversity.

4.1. Datasets and Data Processing

To validate our proposed text-to-3D generation framework, we utilize distinct data processing pipelines for the semantic text latents and the 3D geometric shapes.

SDF Data Preparation: For our geometric data, we select three structurally diverse subcategories from the ShapeNet [40] dataset: airplanes, cars, and chairs. We maintain the official train, validation, and test splits to ensure fair evaluation. Before extracting the implicit fields, all original 3D meshes are normalized to fit entirely within a unit bounding box.

Text Latent Extraction: To capture semantic language priors, we process textual descriptions for ShapeNet from Cap3D [41, 42] using a frozen pre-trained T5-base [43] encoder. Representative captions for the three evaluated object categories are shown in Table 4.1. For each input prompt, we extract the embeddings across all text tokens and compute their mean, resulting in a single 768-dimensional global semantic vector, denoted as z_{text}.

Coordinate-SDF Pair Sampling: To train the continuous implicit decoder, we generate paired spatial coordinates and signed distance values (x, s) using Point Cloud Utils [44]. Adopting a sampling strategy consistent with standard implicit neural representations, we sample a total of 50,000 points per shape. To ensure the network learns both high-frequency surface details and the overall global topology, the sampling is heavily biased toward the surface: 47,000 points are sampled aggressively near the mesh surface, while the remaining 3,000 points are sampled uniformly across the unit volume.

4.2. Experiment Setups

To rigorously evaluate the efficacy of our proposed two-stage architecture and its individual components, we establish several baselines and training configurations:

Direct Text-to-SDF (Baseline): A naive baseline where the frozen T5 [43] LLM latent z_{text} is fed directly into the continuous coordinate-based decoder. Only the decoder's network parameters are updated during training, effectively forcing a direct mapping between raw text semantics and 3D geometry. This configuration tests whether the pre-trained semantic space is already sufficiently compatible with the geometric decoding task. If the semantic and geometric latent spaces are poorly aligned, the

Table 4.1: Examples of textual descriptions from Cap3D used to obtain the semantic embeddings for the three ShapeNet categories considered in our experiments.

Category	Example Cap3D Caption
Airplane	"A small, lightweight aircraft featuring a classic bi-plane design with a predominantly white exterior, accented by a blue stripe and tail fin. It has a round propeller at the front, two wheels for landing gear, and displays the registration number N476BZ. The wings are broad and flat, providing stability during flight, while the cockpit is open, suggesting a vintage style. Overall, the object embodies a nostalgic aesthetic typical of early aviation."
Car	"A sleek, aerodynamic sports car featuring a white body with black accents, large rear wing for enhanced downforce, and intricately designed multi-spoke wheels. The car showcases distinctive front and rear styling with prominent air intakes and a low ground clearance, emphasizing its performance-oriented design. Additionally, it features tinted windows for a modern look."
Chair	"A modern office chair featuring a plush green fabric upholstery, providing comfort and support. It has a high backrest and cushioned seat, complemented by sleek black armrests. The chair is mounted on a five-point metal base with a light grey finish, equipped with rolling casters for mobility. The design seamlessly blends durability with contemporary style."

performance of this direct conditioning strategy is expected to be limited, motivating the explicit latent-alignment strategies introduced below.

Random Latent Prior (Baseline): This configuration follows the same decoder architecture and training procedure as Direct Text-to-SDF, but replaces z_{text} with a random latent vector $z_{\text{rand}} \sim \mathcal{N}(0, I)$ of the same dimensionality. Each shape is assigned a fixed random latent vector. Comparing this configuration with Direct Text-to-SDF allows us to assess whether the structure encoded by the pre-trained T5 embeddings provides a more informative conditioning signal than an unstructured latent representation.

End-to-End Joint Training: The LatentMapper M and SDF decoder f_θ are connected directly and optimized simultaneously, giving

$$\hat{s} = f_\theta(M(z_{\text{text}}), \mathbf{x}). \tag{4.1}$$

The T5 encoder remains frozen, while the mapper and decoder parameters are jointly optimized using the SDF reconstruction objective. This configuration assesses whether the semantic-to-geometric projection and the implicit geometric field can be learned simultaneously without an explicit intermediate latent-alignment stage.

End-to-End Joint Training (w/ Reg.): This configuration uses the same joint architecture but the optimization objective $\mathcal{L}_{\text{joint}}$ additionally constrains the mapped latent to remain close to the original semantic embedding:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}\text{SDF} + \lambda \|M(z_{\text{text}}) - z_{\text{text}}\|_2^2. \tag{4.2}$$

Comparing the two joint-training configurations allows us to assess whether explicitly preserving proximity to the pre-trained semantic space improves joint optimization.

Zero-Shot Semantic Decoding: After Stage 1 auto-decoder training, the optimized latent z_{opt} is replaced by the original frozen text embedding z_{text} and directly evaluated using the trained decoder. Neither latent optimization nor the LatentMapper is used at inference. This configuration aims to observe the zero-shot generalization capabilities of the geometric space.

Decoupled Latent Projection (Proposed): In Stage 1, the SDF decoder parameters θ and the per-shape geometric latents z_{opt} are jointly optimized according to Eq. (3.1). After convergence, the decoder is frozen and, in Stage 2, the LatentMapper M is trained to regress from z_{text} to the corresponding optimized z_{opt} according to Eq. (3.2). At inference, the predicted latent $\hat{z}_{\text{opt}} = M(z_{\text{text}})$ is directly decoded without access to ground-truth geometry or additional latent optimization. This configuration represents the proposed feedforward inference pipeline.

TTO Geometric Upper Bound (Requires Ground Truth): With the Stage 1 decoder frozen, the latent vector is optimized separately for each test shape using its ground-truth SDF observations. Because

this procedure requires access to the target geometry at test time, it is not considered a practical text-to-3D inference configuration and is not directly comparable to the feedforward methods in terms of deployment. Instead, it serves as an oracle reconstruction reference for estimating the capacity of the trained decoder and the remaining approximation gap introduced when z_{opt} is predicted by the feedforward mapper.

By comparing these different setups, we can empirically assess whether LLM latents provide superior efficacy over random latents, whether the auto-decoder alignment enhances geometric fidelity, and whether the mapper successfully bridges the gap toward the TTO upper bound. Furthermore, contrasting the proposed method with the End-to-End configurations allows us to validate the necessity of the decoupled two-stage training scheme and latent regularization in preserving implicit field stability.

4.3. Implementation Details

All models are implemented in PyTorch and trained using CUDA acceleration. Unless otherwise specified, all SDF-based configurations share the same architecture and optimization settings. The conditioning latent dimension is 768, and the DeepSDF decoder contains eight weight-normalized hidden layers with a hidden dimension of 512 and ReLU activations. The concatenated latent-coordinate input is re-injected before the fifth hidden layer through a skip connection. Training uses mini-batches of $B = 32$ shapes, with 16,384 spatial points sampled per shape at each optimization iteration, consisting of 8,192 volumetric and 8,192 surface samples. Predicted and ground-truth SDF values are clamped to $[-0.1, 0.1]$ before computing the L_1 reconstruction loss. The decoder-based configurations are optimized using Adam with a fixed learning rate of 1×10^{-4}, without weight decay or learning-rate scheduling.

The Direct Text-to-SDF, Random Latent Prior, and the two End-to-End Joint Training configurations follow this common setup and are each trained for 20,703 iterations. Direct Text-to-SDF directly conditions the decoder on the frozen T5 embedding. For the Random Latent Prior, each shape instead receives a fixed 768-dimensional random vector. Rather than sampling from a standard normal distribution, the random vectors are drawn independently per dimension according to the empirical mean and standard deviation of the T5 embedding distribution. A fixed random seed of 42 is used to ensure deterministic latent assignment. For the joint configurations, the LatentMapper and SDF decoder are optimized simultaneously using the same Adam optimizer and learning rate. The mapper consists of an initial 768-dimensional projection followed by four residual MLP blocks with a hidden dimension of 2048 and dropout 0.1. The unregularized model is optimized only with the SDF reconstruction loss, whereas the regularized variant additionally constrains the mapped latent toward the original T5 embedding using

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{SDF}} + \lambda \frac{1}{B} \sum_{i=1}^{B} \left\| M(z_{\text{text}}^{(i)}) - z_{\text{text}}^{(i)} \right\|_2, \tag{4.3}$$

with $\lambda = 10^{-3}$.

For Stage 1 of the proposed method, the same SDF decoder, sampling strategy, batch size, and reconstruction objective are retained. Each training shape is additionally associated with an optimizable 768-dimensional geometric latent z_{opt}, initialized from its corresponding frozen T5 embedding z_{text}. The decoder parameters and geometric latents are optimized using separate Adam optimizers, both with a fixed learning rate of 1×10^{-4}. The training objective is

$$\mathcal{L}_{\text{Stage1}} = \mathcal{L}_{\text{SDF}} + \lambda \frac{1}{B} \sum_{i=1}^{B} \left\| z_{\text{opt}}^{(i)} - z_{\text{text}}^{(i)} \right\|_2, \tag{4.4}$$

where the anchor regularization weight is set to $\lambda = 10^{-3}$. Stage 1 is trained for 38,934 iterations. During training, validation is performed using the original frozen semantic latents. In addition, every ten complete passes through the training set, validation latents are independently optimized for 200 steps with the decoder frozen, using Adam with a learning rate of 1×10^{-3}, to monitor the reconstruction capability of the learned geometric space.

For Stage 2, the Stage-1 decoder is frozen and the LatentMapper is trained independently using paired semantic and optimized geometric latents $(z_{\text{text}}, z_{\text{opt}})$. The mapper uses the same four-block resid-

ual architecture as in the joint configurations, with latent dimension 768, hidden dimension 2048, and dropout 0.1. Unlike the SDF-based configurations, mapper training uses the Smooth L_1 regression loss and AdamW with an initial learning rate of 1×10^{-4}, weight decay of 1×10^{-4}, and a batch size of 256. The learning rate is adjusted using cosine annealing over the complete training procedure, and gradients are clipped to a maximum L_2 norm of 1.0. With 9,901 latent pairs and incomplete final mini-batches discarded, the mapper is optimized for 7,600 iterations. The checkpoint achieving the lowest validation Smooth L_1 loss is retained for inference.

The Zero-Shot Semantic Decoding configuration requires no additional optimization and directly evaluates the trained Stage-1 decoder using the original z_{text}. The TTO Reconstruction Reference similarly introduces no additional network training; instead, the decoder parameters remain fixed while the latent representation of each test shape is optimized using its ground-truth SDF observations.

4.4. Evaluation Metrics

Because no single metric can fully encapsulate the quality of a generated 3D shape, we employ a diverse set of evaluation metrics that structurally compensate for one another, assessing surface accuracy, volumetric coherence, and distributional diversity.

Since the models predict continuous SDFs rather than point clouds directly, we first extract an explicit surface for evaluation. The predicted SDF is evaluated on a 128^3 regular grid covering the spatial range $[-1.1, 1.1]^3$, after which Marching Cubes is applied to obtain a triangular mesh. For the surface-based fidelity metrics, 30,000 points are sampled from the generated mesh with probability proportional to triangle area and compared against sampled ground-truth surface points. For the distribution-level metrics COV and 1-NNA, 2,048 surface points per shape are used to reduce the cost of pairwise shape comparisons.

o evaluate the structural fidelity of individual reconstructed objects, we use three complementary metrics. Chamfer Distance (CD) [12] measures the bidirectional average discrepancy between the sampled generated surface P and ground-truth surface Q:

$$\mathrm{CD}(P, Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p - q\|_2^2 + \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \|q - p\|_2^2. \tag{4.5}$$

Lower CD indicates greater surface similarity. In our implementation, the resulting value is multiplied by 10^3 for reporting. We additionally report the F1 Score at 1% [45], which evaluates the fraction of generated and ground-truth surface points lying within a distance threshold $\tau = 0.01$. Defining precision as the fraction of generated points whose nearest ground-truth point lies within τ, and recall analogously for ground-truth points, the F1 score is $2PR/(P + R)$. Higher F1 therefore indicates stronger agreement within the specified surface tolerance. Finally, 3D Intersection over Union (IoU) [46] evaluates volumetric occupancy rather than only surface proximity. Instead of voxelizing the reconstructed mesh, we evaluate the predicted and ground-truth SDFs at the stored volumetric query points and classify a point as occupied when its SDF value is non-positive. IoU is then computed as

$$\mathrm{IoU} = \frac{|\mathcal{O}_{\text{pred}} \cap \mathcal{O}_{\text{gt}}|}{|\mathcal{O}_{\text{pred}} \cup \mathcal{O}_{\text{gt}}|}, \tag{4.6}$$

where $\mathcal{O}_{\text{pred}}$ and $\mathcal{O}_{\text{gt}}$ denote the predicted and ground-truth occupied query points, respectively. Thus, CD and F1 characterize surface fidelity, while IoU assesses consistency of the predicted interior volume.

Beyond per-object reconstruction fidelity, we use Coverage (COV) [47] and 1-Nearest Neighbor Accuracy (1-NNA) [48] to compare the generated and ground-truth shape distributions. Pairwise shape distances are computed using Chamfer Distance on 2,048 sampled surface points per shape. Let $\mathcal{G}$ and $\mathcal{R}$ denote the generated and reference shape sets, respectively. COV measures the fraction of reference shapes that are selected as the nearest neighbor of at least one generated shape:

$$\mathrm{COV}(\mathcal{G}, \mathcal{R}) \frac{|\arg\min_{r \in \mathcal{R}} d(g, r) \mid g \in \mathcal{G}|}{|\mathcal{R}|}. \tag{4.7}$$

A higher COV indicates that the generated set spans a larger portion of the reference distribution. In contrast, 1-NNA evaluates how distinguishable the two distributions are. Generated and reference

shapes are pooled together, each shape is classified according to whether its nearest neighbor belongs to the same set, and the resulting classification accuracy is reported. For equally sized generated and reference sets, an accuracy close to 50% indicates that the two distributions are difficult to distinguish, whereas a substantially higher value indicates a larger distributional discrepancy. Together, COV and 1-NNA provide complementary measures of distributional coverage and similarity rather than the fidelity of individual reconstructions. For readability, Chamfer Distance is reported after multiplication by 10^3, while F1, IoU, COV, and 1-NNA are reported in percentage points, i.e., multiplied by 100.

5

Results and Discussion

In this chapter, we present the quantitative evaluation of the proposed framework across different experimental configurations. Based on these results, we critically analyze the performance of individual architectural components, highlighting both their contributions to geometric fidelity and the inherent limitations observed during the generation process.

5.1. Quantitative Evaluation

The quantitative performance of our proposed text-to-3D generation framework against established baselines and ablation setups is summarized in Table 5.1. The generated shapes are visualized using Polyscope [49] in Figure 5.1, with additional figures provided in A.2.

Table 5.1: Quantitative comparison of the evaluated text-to-3D configurations. CD is multiplied by 10^3; F1, IoU, COV, and 1-NNA are reported as percentages. Bold indicates the best result among feedforward configurations with valid reconstructions. The TTO result is shown in italics because it requires ground-truth geometry at test time and serves only as an oracle reconstruction reference.

Model Setup	CD$\times 10^3$(↓)	F1@1% (%) (↑)	IoU (%) (↑)	COV (%) (↑)	1-NNA (%) (ideal: 50%)
Direct Text-to-SDF (Baseline)	52.11	1.33	**12.18**	2.87	99.88
Random Latent Prior (Baseline)	91.10	2.49	10.35	1.78	99.94
End-to-End Joint Training	–†	–†	–†	–†	–†
End-to-End Joint Training (w/ Reg.)	61.74	1.02	1.81	1.46	99.93
Zero-Shot Semantic Decoding	34.23	3.24	8.52	4.21	99.67
Decoupled Latent Projection (Proposed)	**19.62**	**6.34**	11.39	**7.13**	**98.91**
TTO Geometric Upper Bound	*9.88*	*1.72*	*18.84*	*25.28*	*92.37*

† No valid zero-level surface could be recovered for the unregularized End-to-End Joint Training configuration. Consequently, mesh-based CD and F1, as well as the surface-distribution metrics COV and 1-NNA, are undefined. The resulting IOU comparison also does not correspond to a valid reconstructed shape and is not considered meaningful for shape-level evaluation.

5.1.1. Analysis of Semantic Priors

Comparing **Direct Text-to-SDF** with the **Random Latent Prior** examines whether the structure of the pretrained semantic embedding provides a useful conditioning signal for geometric prediction. As shown in Figure 5.2, the pretrained T5 embeddings exhibit a strongly structured latent organization even before geometric training: samples belonging to airplanes, cars, and chairs form clearly separated clusters in the projected space. This indicates that the semantic latent already encodes coarse category-level information that can distinguish the three shape classes. The quantitative results provide further evidence that this semantic structure is useful for geometric conditioning. **Direct Text-to-SDF** reduces the reported $10^3 \times$CD from 91.10 for the **Random Latent Prior** to 52.11 and improves IoU from 10.35% to 12.18%. COV also increases from 1.78% to 2.87%, while 1-NNA moves slightly toward the ideal value of 50%, from 99.94% to 99.88%. However, the **Random Latent Prior** achieves a higher F1@1% score (2.49% versus 1.33%), indicating that the benefit of semantic conditioning is not uniform across all surface metrics. Taken together with the category structure observed in Figure 5.2, these re-

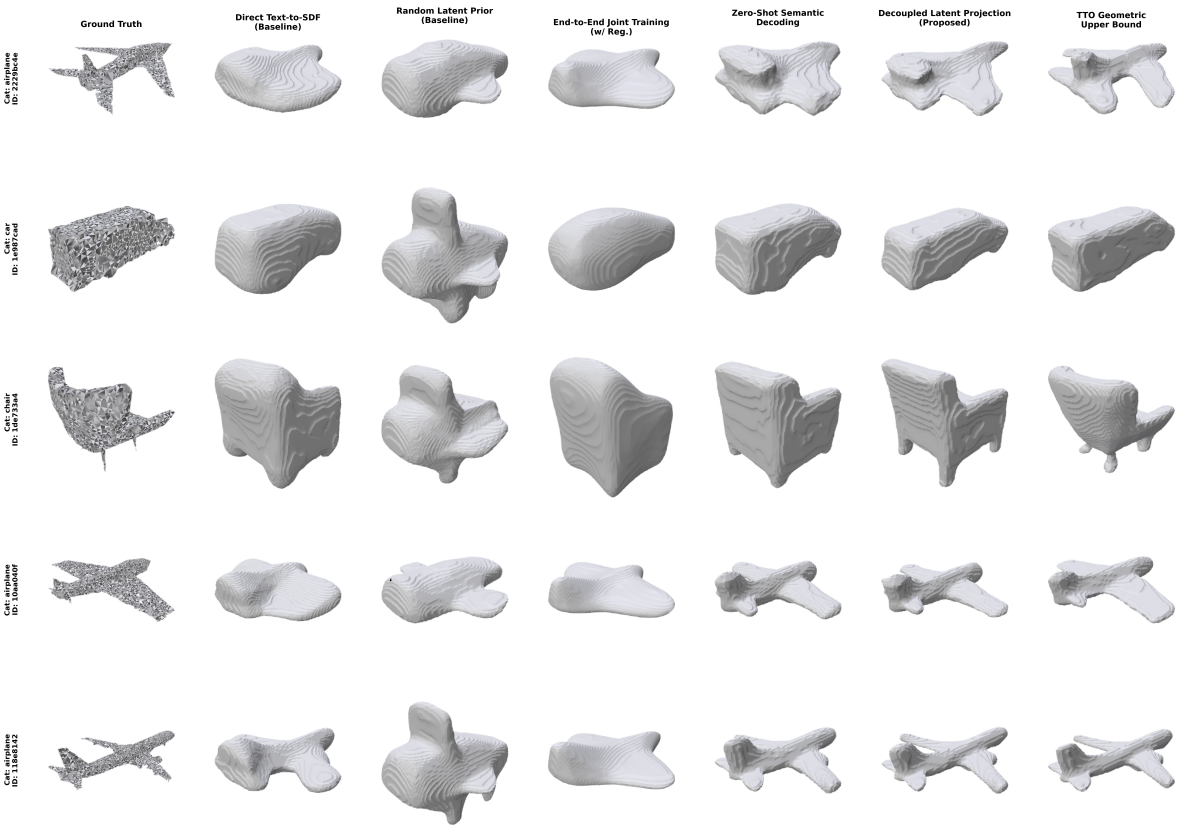


Figure 5.1: Qualitative comparison of representative test shapes across the evaluated configurations. From left to right: ground truth, Direct Text-to-SDF, Random Latent Prior, End-to-End Joint Training with latent regularization, Zero-Shot Semantic Decoding, Decoupled Latent Projection (proposed), and the TTO Geometric Upper Bound. The unregularized End-to-End Joint Training configuration is omitted because no valid zero-level surface could be extracted from its predicted SDF field.

sults suggest that pretrained T5 embeddings provide a more informative initialization and conditioning space than randomly assigned latent vectors, although their semantic organization alone is insufficient for accurate geometric reconstruction.

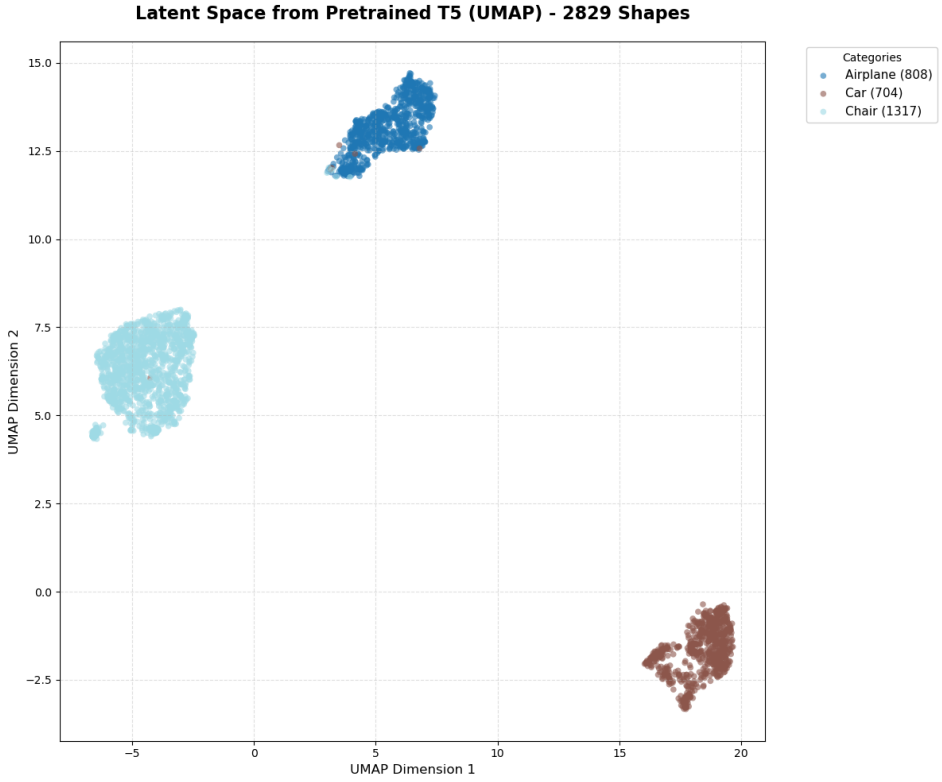


Figure 5.2: UMAP projection of pretrained T5 embeddings for 2,829 ShapeNet text descriptions. The clear separation between airplane, car, and chair samples illustrates that the frozen semantic embeddings contain substantial category-level structure prior to geometric training.

5.1.2. Analysis of the Auto-Decoder Architecture

The effect of the auto-decoder training can first be examined through **Zero-Shot Semantic Decoding**, which uses the Stage-1 decoder but directly supplies the original semantic latent at inference without latent optimization or the learned mapper. Compared with **Direct Text-to-SDF**, this configuration improves $10^3 \times$CD from 52.11 to 34.23 and F1 from 1.33% to 3.24%. COV also increases from 2.87% to 4.21%, while 1-NNA moves from 99.88% to 99.67%. In contrast, IoU decreases from 12.18% to 8.52%, indicating that improved surface proximity does not necessarily translate into improved volumetric occupancy. Figure 5.1 similarly shows more recognizable object structures for several zero-shot reconstructions, although substantial geometric errors remain. These results indicate that the latent space learned through auto-decoder training is more suitable for surface reconstruction, even before the explicit semantic-to-geometric mapper is introduced.

5.1.3. Efficacy of the Latent Mapper

Introducing the LatentMapper produces a consistent improvement over **Zero-Shot Semantic Decoding** across all five reported metrics. The proposed **Decoupled Latent Projection** reduces $10^3 \times$CD from 34.23 to 19.62, increases F1 from 3.24% to 6.34%, and improves IoU from 8.52% to 11.39%. At the distribution level, COV increases from 4.21% to 7.13%, while 1-NNA decreases from 99.67% to 98.91%, moving toward the ideal value of 50%. The qualitative examples in Figure 5.1 are consistent with these quantitative improvements, with the mapped reconstructions more closely following the ground-truth part arrangement and overall shape in several examples. Together, these results indicate that explicitly learning the projection from the frozen semantic latent to the optimized geometric latent recovers substantially more of the geometric information encoded by the auto-decoder space

than directly supplying the original text embedding.

5.1.4. Decoupled vs. End-to-End Training

The comparison with joint optimization highlights a major difference between the two training strategies. For the **unregularized End-to-End Joint Training** configuration, no valid zero-level surface could be recovered from the predicted field. It is therefore omitted from Figure 5.1 because there is no valid mesh to visualize. Adding latent regularization allows recoverable surfaces to be produced, but the resulting performance remains substantially below the proposed decoupled approach: $10^3 \times$CD is 61.74 versus 19.62, F1 is 1.02% versus 6.34%, and IoU is 1.81% versus 11.39%. COV likewise decreases from 7.13% to 1.46%, while 1-NNA moves from 98.91% to 99.93%. The visual difference is further illustrated in Figure 5.3. These observations support the use of separate geometric-space learning and semantic-to-geometric projection in the present experimental setting.

5.1.5. Assessing the Optimization Upper Bound

The TTO Geometric Upper Bound evaluates the decoder when the latent vector is optimized directly against ground-truth SDF observations at test time, and therefore represents an oracle reconstruction reference rather than a practical text-to-3D method. Relative to the proposed feedforward pipeline, TTO reduces $10^3 \times$CD from 19.62 to 9.88 and increases IoU from 11.39% to 18.84%. It also substantially improves distribution-level performance, increasing COV from 7.13% to 25.28% and moving 1-NNA from 98.91% to 92.37%.

Interestingly, however, its F1@1% score is lower than that of the proposed method (1.72% versus 6.34%). One possible explanation is the difference between the TTO optimization objective and the F1 evaluation criterion. TTO directly minimizes reconstruction error over sampled SDF observations, encouraging the predicted field to better approximate the target geometry globally, whereas F1@1% only counts surface samples whose nearest correspondence falls within the strict distance threshold of (0.01). We therefore hypothesize that TTO may produce a surface with lower overall geometric error but with a small systematic surface displacement that leaves many sampled points outside this narrow tolerance. In contrast, the proposed feedforward model may reconstruct some local surface regions more accurately within the threshold while exhibiting larger errors elsewhere, resulting in a higher F1 but worse average Chamfer Distance. This interpretation is consistent with the lower CD and higher IoU achieved by TTO, although additional analysis of the point-wise surface-error distribution would be required to verify this hypothesis.

Overall, the improvements in CD, IoU, COV, and 1-NNA obtained by TTO indicate that the optimized latent space retains considerably greater geometric reconstruction capacity than is currently recovered by the feedforward mapper, leaving substantial room for improving the approximation from semantic to optimized geometric latents.

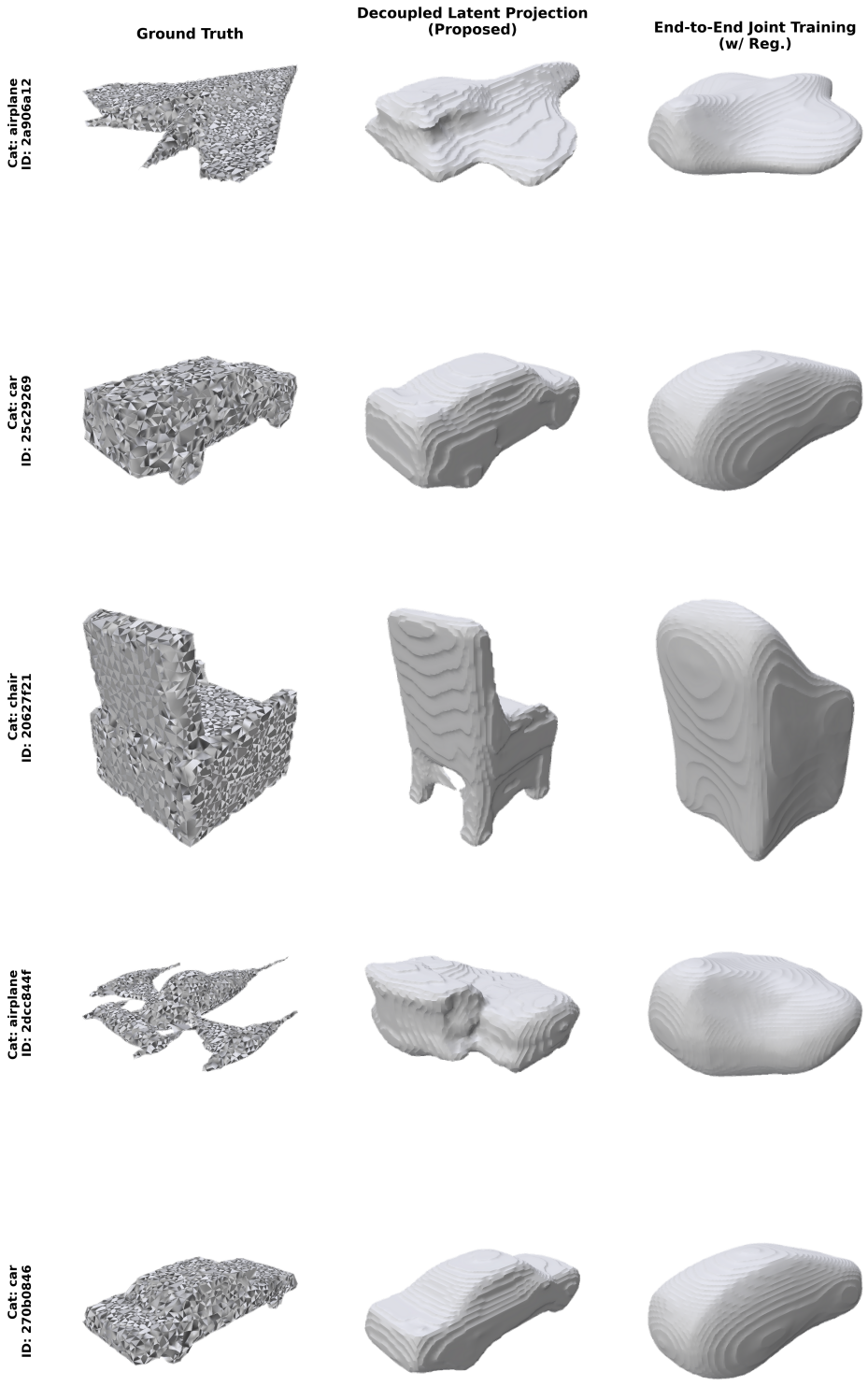


Figure 5.3: Focused qualitative comparison between the proposed decoupled training strategy and end-to-end joint training. From left to right: ground-truth geometry, Decoupled Latent Projection (proposed), and End-to-End Joint Training with latent regularization. The decoupled strategy more closely preserves the global structure and geometric characteristics of the target shapes.

6

Conclusion

This final chapter concludes the research presented in this thesis. We first critically examine the inherent limitations of our proposed text-to-3D generation framework, immediately pairing them with promising avenues for future research to address these specific challenges. We then reflect on the relationship of this work to the broader CS and DSAIT field and discuss its potential societal impact within the current development of generative AI. Finally, we provide an overall conclusion, synthesizing our core contributions and highlighting the primary strengths of the established methodology.

6.1. Limitations and Future Work

Although the proposed framework improves most reported metrics over other configurations, it also exhibits inherent weaknesses that limit its current performance. Building upon these observations, several logical directions for future research emerge:

Resolution Constraints and Spatial Scaling: A primary limitation of our experimental setup is the reliance on a relatively sparse 50,000-point sampling density during training. This provides an insufficient spatial signal for the network to mathematically reconstruct highly complex, watertight meshes with absolute precision, effectively capping the maximum achievable Intersection over Union (IoU) and leading to localized smoothing artifacts on thin topological structures. To overcome this bottleneck, future work should focus on scaling the spatial sampling resolution significantly beyond 50,000 points. Utilizing more advanced hierarchical sampling strategies or increased computational infrastructure would allow the continuous implicit field to capture finer, high-frequency surface details.

Approximation Gap and Expressive Architectures: While the feedforward mapper successfully bridges the semantic and geometric domains, it acts only as an approximation of a highly complex, non-linear optimization trajectory. Consequently, a considerable performance gap remains between the feedforward pipeline and the theoretical Test-Time Optimization (TTO) geometric upper bound. To narrow this gap, future iterations should explore more expressive neural architectures for the Latent Mapper. Replacing the simple MLP with more sophisticated attention-based networks or deeper residual architectures could yield a much more precise projection from the frozen semantic space to the optimized geometric space.

Evaluation Scope and Comparison with Existing Methods: The current evaluation is limited to three ShapeNet categories—airplanes, cars, and chairs—and therefore does not fully demonstrate generalization to broader object classes or open-vocabulary text prompts. In addition, we do not directly compare against existing text-to-3D methods, due to differences in datasets, representations, and evaluation protocols. The present experiments instead focus on controlled comparisons between variants of our framework to observe the effects of semantic priors, auto-decoder training, and latent projection. Future work should extend the evaluation to more diverse categories and establish a common benchmark for comparison with representative text-to-3D methods.

Lack of End-to-End Synergy and Hybrid Training: Our decoupled training paradigm, while strictly

necessary to maintain the stability of the coordinate-based implicit field, freezes the decoder's weights during the mapping stage. This prevents the geometric space from dynamically adjusting to accommodate the specific projection errors introduced by the MLP. Investigating hybrid training strategies could resolve this limitation. For instance, combining the stability of our decoupled pre-training with a carefully regularized, low-learning-rate end-to-end fine-tuning phase could restore dynamic synergy between the mapper and the decoder without collapsing the geometry.

6.2. Relation to the Broader CS and DSAIT Field

This research illustrates how knowledge developed across different areas of computer science and Data Science and Artificial Intelligence Technology (DSAIT) can be combined to address problems in 3D computer graphics. Rather than learning the entire text-to-3D relationship from scratch, our framework reuses the semantic representations of a pre-trained language model and connects them to a continuous geometric representation through latent-space optimization and projection. In this sense, the proposed method combines concepts from natural language processing, representation learning, transfer learning, and implicit neural representations within a single 3D generation framework.

More broadly, this work reflects an increasingly important direction within DSAIT that focuses on exploiting pre-trained representations and adapting them to specialized downstream domains. Our results suggest that useful information contained in one learned representation space can be transferred to a structurally different domain through an appropriate alignment mechanism. Although this thesis focuses specifically on semantic-to-geometric alignment, the same principle may be relevant to other cross-domain problems in computer science, where pre-trained semantic or perceptual representations need to interact with domain-specific structured representations. Future research could therefore investigate similar latent-alignment strategies across other modalities and application areas.

6.3. Societal Impact and Temporal Context

Automated 3D content generation has the potential to influence how digital assets are created. Traditional 3D modeling requires substantial technical expertise, specialized software, and considerable manual effort. By enabling geometry to be generated from natural-language descriptions, text-to-3D systems can lower the technical barrier to early-stage asset creation and provide artists and designers with additional ways to rapidly prototype and explore ideas. Such tools should be viewed primarily as complementary creative instruments because they can provide initial geometries and enable faster iteration while leaving detailed artistic decisions, refinement, and validation to human creators.

This work is also situated within a period in which generative AI systems are becoming increasingly large and computationally demanding. Instead of training a large multimodal generative model from scratch or performing iterative optimization for every input prompt, our framework reuses a frozen pre-trained semantic representation and performs generation through a comparatively compact latent mapper and SDF decoder. Removing test-time optimization may therefore reduce the marginal computational cost of generating individual assets and, consequently, has the potential to reduce associated energy consumption when deployed at scale. However, energy usage was not directly evaluated in this thesis, and the computational cost of pre-training the underlying language model must also be acknowledged. Nevertheless, the results motivate further research into smaller, reusable, and feedforward generative architectures as a complementary direction to the continuing trend toward increasingly large foundation models.

6.4. Overall Conclusion

The manual creation of high-fidelity 3D assets remains a notorious bottleneck in digital production pipelines, traditionally demanding extensive expertise and meticulous labor. While recent state-of-the-art text-to-3D generation frameworks have made remarkable strides toward automation, they often require extraordinary computational resources for iterative optimization or rely on massive multi-modal datasets to build alignment networks from the ground up.

To address these fundamental challenges, this research investigated the hypothesis that pre-trained Large Language Models (LLMs)—which possess a highly structured and expressive semantic space—can be directly exploited as conditioning inputs to predict continuous 3D shapes. We successfully

developed a decoupled two-stage architecture that leverages Signed Distance Functions (SDFs) to bridge the structural discrepancy between semantic text embeddings and explicit geometric spatial properties.

Our findings demonstrate that by adapting an auto-decoder optimization framework, we can effectively align the semantic text space with the 3D geometric space, significantly enhancing the resolution and structural fidelity of the generated objects compared to naive direct-regression baselines. Furthermore, the introduction of a dedicated feedforward mapping network proved highly effective in bypassing computationally expensive test-time optimization entirely.

Ultimately, this thesis establishes a robust and optimization-free inference pipeline. By utilizing the structured priors of LLM latents, the proposed framework enables the feedforward generation of diverse 3D SDFs directly from raw textual prompts, representing an alternative to traditional optimization-based synthesis.

References

[1] Ian J Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).

[2] P Kingma Diederik and Welling Max. “An introduction to variational autoencoders”. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[4] Xingguang Yan et al. “Shapeformer: Transformer-based shape completion via sparse representation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 6239–6249.

[5] Fukun Yin et al. “Shapegpt: 3d shape generation with a unified multi-modal language model”. In: *IEEE Transactions on Multimedia* 27 (2025), pp. 4107–4120.

[6] Alex Nichol et al. “Point-e: A system for generating 3d point clouds from complex prompts”. In: *arXiv preprint arXiv:2212.08751* (2022).

[7] Heewoo Jun and Alex Nichol. “Shap-e: Generating conditional 3d implicit functions”. In: *arXiv preprint arXiv:2305.02463* (2023).

[8] Ben Poole et al. “DreamFusion: Text-to-3D using 2D Diffusion”. In: *The Eleventh International Conference on Learning Representations*. 2022.

[9] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*. Vol. 153. Springer Science & Business Media, 2003.

[10] Daniel Maturana and Sebastian Scherer. “VoxNet: A 3D Convolutional Neural Network for real-time object recognition”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 922–928. DOI: 10.1109/IROS.2015.7353481.

[11] Charles R Qi et al. “Volumetric and multi-view cnns for object classification on 3d data”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5648–5656.

[12] Haoqiang Fan, Hao Su, and Leonidas J Guibas. “A point set generation network for 3d object reconstruction from a single image”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 605–613.

[13] Charles Ruizhongtai Qi et al. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space”. In: *Advances in neural information processing systems* 30 (2017).

[14] Ben Mildenhall et al. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[15] Panos Achlioptas et al. “Learning Representations and Generative Models for 3D Point Clouds”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 40–49.

[16] Jiajun Wu et al. “Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling”. In: *Advances in neural information processing systems* 29 (2016).

[17] X. Zheng et al. “SDF-StyleGAN: Implicit SDF-Based StyleGAN for 3D Shape Generation”. In: *Computer Graphics Forum* 41.5 (2022), pp. 52–63.

[18] Eric R Chan et al. “Efficient geometry-aware 3d generative adversarial networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 16123–16133.

[19] Lin Gao et al. “SDM-NET: Deep generative network for structured deformable mesh”. In: *ACM Transactions on Graphics (ToG)* 38.6 (2019), pp. 1–15.

[20] Lin Gao et al. “Tm-net: Deep generative networks for textured meshes”. In: *ACM Transactions on Graphics (TOG)* 40.6 (2021), pp. 1–15.

[21] Adam R Kosiorek et al. “NeRF-VAE: A Geometry Aware 3D Scene Generative Model”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 5742–5752.

[22] Jeong Joon Park et al. “Deepsdf: Learning continuous signed distance functions for shape representation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 165–174.

[23] Yongbin Sun et al. “Pointgrow: Autoregressively learned point cloud generation with self-attention”. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2020, pp. 61–70.

[24] Charlie Nash et al. “Polygen: An autoregressive generative model of 3d meshes”. In: *International conference on machine learning*. PMLR. 2020, pp. 7220–7229.

[25] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[26] Yawar Siddiqui et al. “MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024.

[27] Haochen Wang et al. “Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 12619–12629.

[28] William Peebles and Saining Xie. “Scalable Diffusion Models with Transformers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2023, pp. 4195–4205.

[29] Longwen Zhang et al. “Clay: A controllable large-scale generative model for creating high-quality 3d assets”. In: *ACM Transactions on Graphics (TOG)* 43.4 (2024), pp. 1–20.

[30] Kevin Chen et al. “Text2Shape: Generating Shapes from Natural Language by Learning Joint Embeddings”. In: *Computer Vision – ACCV 2018*. Cham: Springer International Publishing, 2019, pp. 100–116.

[31] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. PmLR. 2021, pp. 8748–8763.

[32] Aditya Sanghi et al. “Clip-forge: Towards zero-shot text-to-shape generation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 18603–18613.

[33] Chen-Hsuan Lin et al. “Magic3d: High-resolution text-to-3d content creation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 300–309.

[34] Matt Deitke et al. “Objaverse-XL: A Universe of 10M+ 3D Objects”. In: *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2023.

[35] Paritosh Mittal et al. “Autosdf: Shape priors for 3d completion, reconstruction and generation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 306–315.

[36] Aaron Van Den Oord, Oriol Vinyals, et al. “Neural discrete representation learning”. In: *Advances in neural information processing systems* 30 (2017).

[37] Yongwei Chen et al. “SAR3D: Autoregressive 3D Object Generation and Understanding via Multi-scale 3D VQVAE”. In: *CVPR*. 2025.

[38] Junliang Ye et al. “ShapeLLM-Omni: A Native Multimodal LLM for 3D Generation and Understanding”. In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2026.

[39] Eric Tang, Bangding Yang, and Xingyou Song. “Understanding LLM Embeddings for Regression”. In: *Transactions on Machine Learning Research* (2025).

[40] Angel X Chang et al. “Shapenet: An information-rich 3d model repository”. In: *arXiv preprint arXiv:1512.03012* (2015).

[41] Tiange Luo et al. “Scalable 3d captioning with pretrained models”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 75307–75337.

[42] Tiange Luo, Justin Johnson, and Honglak Lee. “View selection for 3d captioning via diffusion ranking”. In: *European Conference on Computer Vision*. Springer. 2024, pp. 180–197.

[43] Colin Raffel et al. “Exploring the limits of transfer learning with a unified text-to-text transformer”. In: *Journal of machine learning research* 21.140 (2020), pp. 1–67.

[44] Francis Williams. *Point Cloud Utils*. https://www.github.com/fwilliams/point-cloud-utils. 2022.

[45] C.J. Van Rijsbergen. *Information Retrieval*. Butterworths, 1979. ISBN: 9780408709293. URL: `https://books.google.nl/books?id=t-pTAAAAMAAJ`.

[46] Christopher B Choy et al. “3d-r2n2: A unified approach for single and multi-view 3d object reconstruction”. In: *European conference on computer vision*. Springer. 2016, pp. 628–644.

[47] Panos Achlioptas et al. “Learning representations and generative models for 3d point clouds”. In: *International conference on machine learning*. PMLR. 2018, pp. 40–49.

[48] David Lopez-Paz and Maxime Oquab. “Revisiting Classifier Two-Sample Tests”. In: *International Conference on Learning Representations*. 2017.

[49] Nicholas Sharp et al. *Polyscope*. www.polyscope.run. 2019.

A

Supplementary Material

A.1. Use of AI Tools

AI tools, including ChatGPT and Gemini, were used to support code debugging, improve writing quality, and perform grammar checking. In addition, AI was used as a critical discussion partner when developing research ideas and implementation plans, helping to challenge assumptions, identify potential weaknesses, and uncover issues that might otherwise have been overlooked. ChatGPT was also used for generating the cover image.

A.2. Additional Results

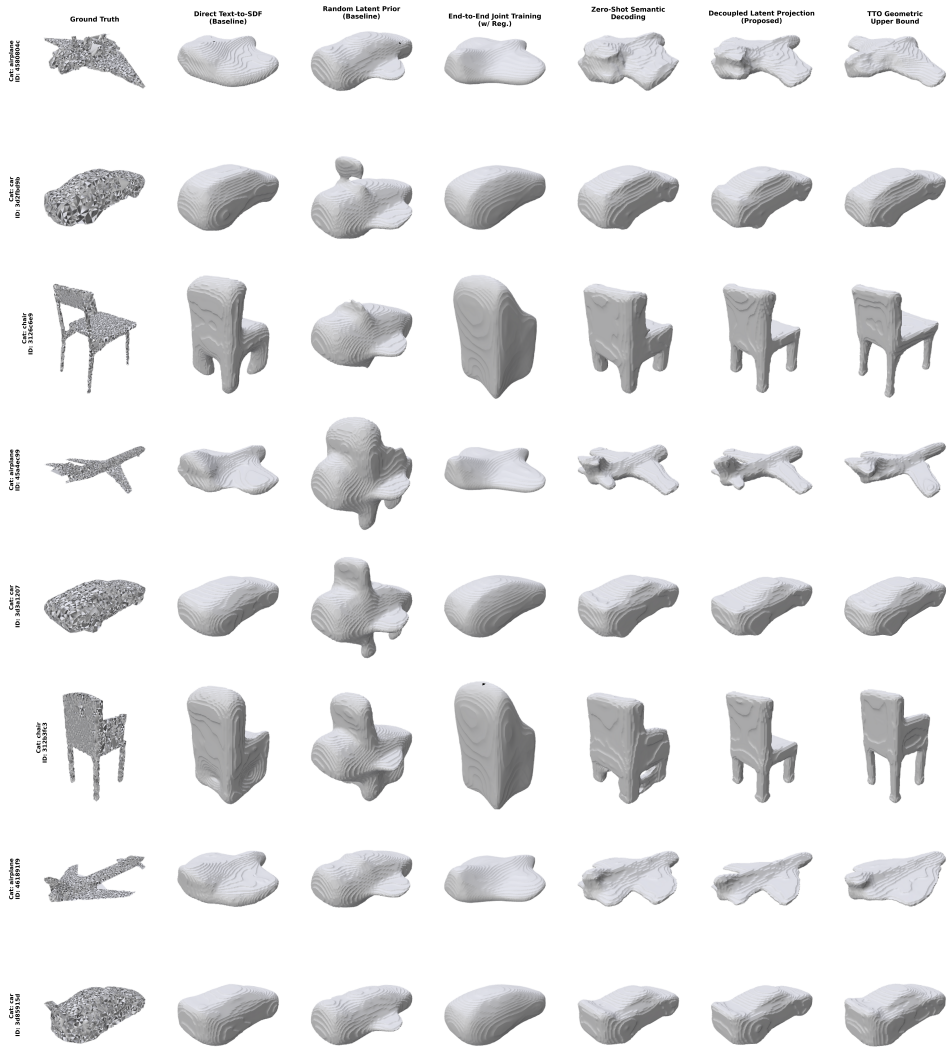


Figure A.1: Additional qualitative comparison between different configurations. From left to right: ground truth, Direct Text-to-SDF, Random Latent Prior, End-to-End Joint Training with latent regularization, Zero-Shot Semantic Decoding, Decoupled Latent Projection (proposed), and the TTO Geometric Upper Bound.

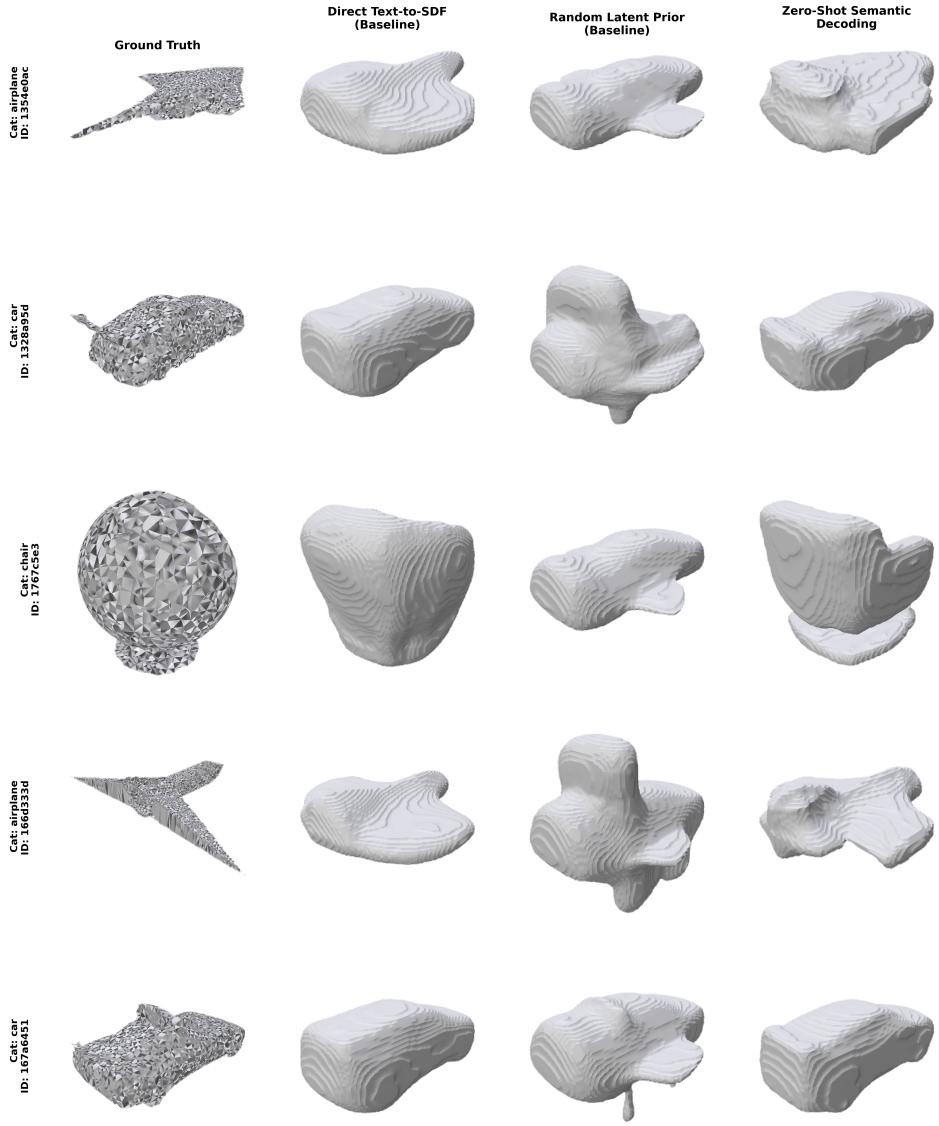


Figure A.2: Additional qualitative exapmples. From left to right: ground truth, Direct Text-to-SDF, Random Latent Prior, Zero-Shot Semantic Decoding.

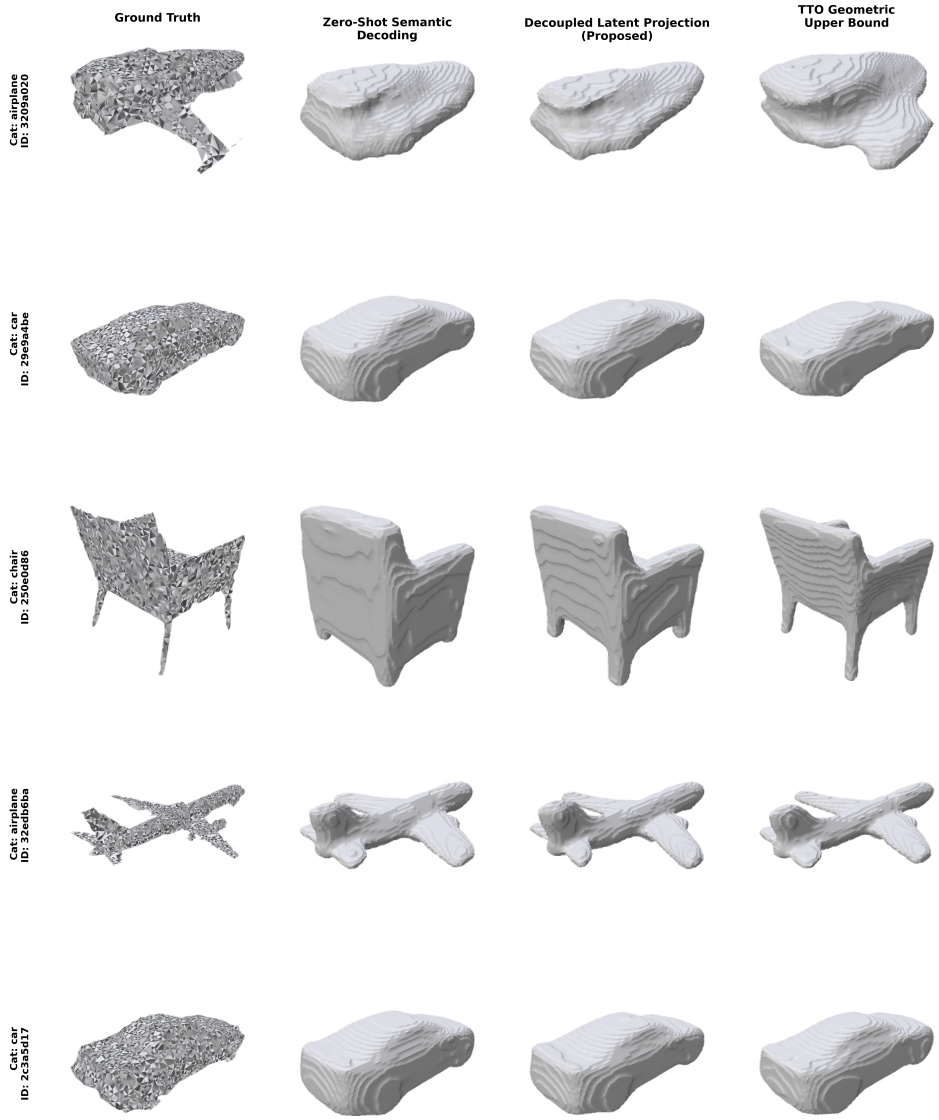


Figure A.3: Additional qualitative exapmples of another 3 configurations. From left to right: ground truth, Zero-Shot Semantic Decoding, Decoupled Latent Projection (proposed), and the TTO Geometric Upper Bound.