3D Visual Computing Course Project

Zixuan Wang

wang-zx23@mails.tsinghua.edu.cn

June 16, 2025

Contents

Chapter 1 Literature Review

1.1 Introduction

Generative frameworks such as Generative Adversarial Networks (GANs) Goodfellow et al. (2014) and Variational Autoencoders (VAEs) Kingma and Welling (2013) were foundational in data synthesis. More recently, diffusion models have become the state-of-the-art for 2D image generation, delivering superior photorealism, diversity, and training stability rooted in a solid probabilistic framework Ho et al. (2020a); Croitoru et al. (2023). However, extending these 2D successes to 3D content generation is not straightforward. The inherent complexity of 3D data introduces distinct challenges that prevent the direct application of 2D methods Wang et al. (2025). To successfully apply diffusion models in the 3D domain, researchers must make critical design choices regarding two key aspects: the 3D data representation and the diffusion methodology. Consequently, 3D diffusion modeling has evolved into a distinct research area with its own unique problems and solutions. The demand for high-quality 3D content is surging across industries like gaming, film, architecture, virtual reality (VR), and scientific visualization. Traditional 3D modeling workflows are resource-intensive, demanding significant manual labor, specialized skills, and computational power. This production bottleneck underscores the need for automated, efficient, and accessible methods for 3D content creation, a gap that generative models are poised to fill.

1.2 Fundamentals of Diffusion Methods

Denoising Diffusion Probabilistic Models (DDPMs)

Denoising Diffusion Probabilistic Models (DDPMs) are latent variable models that consist of two primary processes Ho et al. (2020b). The **forward process** systematically introduces Gaussian noise to data x_0 over T discrete timesteps. This is governed by a fixed Markov chain, gradually transforming the data into an isotropic Gaussian distribution x_T. The transition at each step t is defined by a fixed variance schedule β_t:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{1.2.1}$$

The **reverse process** learns to restore the original data from noise. It begins with a standard Gaussian distribution $p(x_T) = \mathcal{N}(x_T; \mathbf{0}, I)$ and iteratively denoises the data using a neural network, p_θ, which is trained to approximate the true posterior transitions. Each reverse step is a learned Gaussian transition:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{1.2.2}$$

In practice, the variance $\Sigma_\theta(x_t, t)$ is often fixed to a time-dependent constant, and the network is trained to predict the noise component ϵ added at the corresponding forward step. This is accomplished by minimizing a simplified objective function derived from the variational lower bound:

$$\mathbb{E}_{t,x_0,\epsilon}[||\epsilon - \epsilon_\phi(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2] \tag{1.2.3}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$. This objective effectively trains the model ϵ_ϕ to predict and remove the noise from a given noisy sample x_t.

Stochastic Differential Equation (SDE) Formulation

Diffusion models can also be generalized to a continuous-time framework using stochastic differential equations (SDEs), a perspective also known as score-based generative modeling Po et al. (2023). In this view, the forward process that corrupts data with noise is described by the SDE:

$$d\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t dt + \sqrt{\beta(t)}d\mathbf{w}_t \tag{1.2.4}$$

where $\mathbf{w}_t$ is a standard Wiener process and $\beta(t)$ is a continuous noise schedule. The data generation process is defined by a corresponding reverse-time SDE, which transforms noise back into data:

$$d\mathbf{x}_t = \left(-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)\right)dt + \sqrt{\beta(t)}d\bar{\mathbf{w}}_t \tag{1.2.5}$$

Solving this reverse SDE requires estimating the **score function**, $\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)$, which is the gradient of the log-density of the noisy data at time t. A time-dependent neural network is trained to approximate this score function.

Score Distillation Sampling (SDS) for 3D Generation

Score Distillation Sampling (SDS) is a pivotal technique that leverages pre-trained 2D text-to-image diffusion models to optimize 3D representations, such as Neural Radiance Fields (NeRFs) Poole et al. (2023). The core idea is to treat the 2D diffusion model as a powerful prior, providing gradients to update the parameters θ of a differentiable 3D generator g. The process starts by rendering an image $x = g(\theta)$ from a random viewpoint. A noisy version x_t is created by adding noise ϵ to the image. The pre-trained diffusion model's noise-prediction network, ϵ_ϕ, then estimates the added noise, conditioned on a text prompt y. The SDS loss gradient updates the 3D model's parameters θ to make its renderings more plausible under the 2D model's learned distribution:

$$\nabla_\theta \mathcal{L}_{SDS}(\phi, g(\theta)) = \mathbb{E}_{t,\epsilon}\left[w(t)\left(\epsilon_\phi(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon; y, t) - \epsilon\right)\frac{\partial x}{\partial \theta}\right] \tag{1.2.6}$$

Here, $w(t)$ is a weighting function dependent on timestep t. The key innovation of SDS is its ability to distill knowledge from powerful 2D models for 3D generation without requiring large-scale, paired text-3D datasets.

Variational Score Distillation (VSD)

Variational Score Distillation (VSD) was introduced as a more principled successor to SDS Wang et al. (2023). While SDS optimizes a single 3D asset, VSD frames the problem as aligning a distribution of 3D scenes with the 2D diffusion prior. This variational inference approach helps mitigate common SDS artifacts, such as over-saturation, over-smoothing, and a lack of fine detail. The VSD gradient calculation differs from SDS. Instead of comparing the model's noise prediction to the ground-truth noise ¡span class=”math-inline”¿ epsilon¡/span¿, it computes the difference between two noise predictions under different conditions, leading to a more stable gradient. The final loss function is:

$$\nabla_\theta \mathcal{L}_{VSD}(\phi, g(\theta)) = \mathbb{E}_{t,x,\epsilon}\left[w(t)\left(\epsilon_\phi(x_t; y, t) - \epsilon_\phi(x_t; y_c, t)\right)\frac{\partial x}{\partial \theta}\right] \tag{1.2.7}$$

where the gradient is derived from the difference between the noise predicted with a specific text prompt ¡span class=”math-inline”¿y¡/span¿ and a more general prompt ¡span class=”math-inline”¿y_c¡/span¿. This method has been shown to enhance the diversity and fidelity of generated 3D content.

1.3 3D Diffusion Approaches

Diffusion models for 3D generation can be classified based on the domain where the diffusion occurs and the use of pre-trained models. A prominent survey by Wang et al. Wang et al. (2025) categorizes methods into three groups: 2D space diffusion with pre-trained models, 2D space diffusion without pre-trained models, and 3D space diffusion. This framework helps organize the diverse strategies in the field. The primary division in 3D diffusion methods is between **”2D-lifting”** (2D-space diffusion) and **”3D-native”** (3D-space diffusion) approaches. This dichotomy reflects a fundamental trade-off. 2D-lifting methods capitalize on powerful, web-scale pre-trained 2D diffusion models, often achieving superior texture, detail, and generalization. However, they inherently struggle to maintain 3D geometric consistency, leading to artifacts like the Janus problem (an object with multiple fronts). In contrast, 3D-native approaches directly model 3D data distributions, ensuring better geometric integrity. Yet, they are hampered by the scarcity of high-quality 3D data, which can limit the detail and diversity of the generated assets.

2D-Space Diffusion (2D-Lifting): Generating 3D from 2D Priors

These methods optimize a 3D representation by aligning its 2D renderings with distributions learned by 2D diffusion models. Initially, ”2D-lifting” referred mainly to techniques like Score Distillation Sampling (SDS), which uses a fixed, pre-trained 2D model to optimize a single 3D scene Poole et al. (2023). More recent approaches have refined this paradigm by fine-tuning or training 2D diffusion models to be explicitly ”3D-aware” Shi et al. (2023); Lin et al. (2025). These models can generate consistent multi-view images or image-plus-normal map bundles, directly addressing the key weakness of 3D inconsistency. Some work uses existing text-to-image diffusion models as powerful priors, most commonly through SDS and its variants.

Another approch is training Novel 2D Diffusion Models for 3D-Aware. This involves training or fine-tuning 2D diffusion models to generate 2D representations that are inherently 3D-aware, such as consistent multi-view images.

- **Advantages:** Can explicitly learn 3D consistency from multi-view datasets. Inference is often much faster than per-scene optimization.
- **Challenges:** Requires large datasets of multi-view images derived from 3D assets. A separate reconstruction step is often needed to produce a final 3D model.
- **Examples:** MVDream generates consistent multi-view images from text prompts Shi et al. (2023). Kiss3DGen outputs a ”3D Bundle Image” containing multi-view RGB images and normal maps for reconstruction Lin et al. (2025).

3D-Space Diffusion (3D-Native): Direct Diffusion on 3D Data

In contrast to 2D-lifting, these methods apply the diffusion process directly to 3D representations like voxels or point clouds. They fundamentally require 3D datasets for training.

- **Advantages:** Inherently strong multi-view consistency and direct control over geometry. Generation is typically fast once the model is trained.
- **Challenges:** Severely limited by the scarcity and scale of high-quality 3D training data. The high dimensionality of 3D data also imposes significant computational costs.
- **Examples:** Early works include Point-E Nichol et al. (2022) and Shap-E Jun and Nichol (2023). More recent methods apply diffusion to latent codes from 3D autoencoders Zeng et al. (2022); Nam et al. (2022).

1.4 Conditional 3D Generation Methods

Conditional 3D generation provides explicit user control over the synthesis process by guiding it with specific inputs, such as text, images, or sketches. This paradigm moves beyond random sampling towards targeted content creation that aligns with a user’s intent. The core technical challenge across all modalities is to effectively translate the conditioning signal into a coherent and detailed three-dimensional structure. This section provides a detailed algorithmic description of the primary methods developed for this purpose, categorized by the input condition.

Text-to-3D Synthesis

Translating natural language into 3D assets is a primary goal of conditional generation. The approaches can be broadly divided by whether they operate directly in 3D space or leverage powerful, pre-existing 2D models.

Optimization-Based Methods: Score Distillation Sampling

The most prominent and highest-fidelity method for text-to-3D generation is an optimization process reliant on Score Distillation Sampling (SDS) Poole et al. (2023). This algorithm does not train a new generative model but rather optimizes a single 3D scene representation on a per-prompt basis. The process begins by initializing a differentiable 3D representation, such as a Neural Radiance Field (NeRF) Mildenhall et al. (2020) or, more recently, a set of 3D Gaussians Tang et al. (2023). In each optimization step, a virtual camera is placed at a randomly sampled viewpoint, and a differentiable renderer produces the corresponding 2D image. This rendered image is then treated as a clean sample and is perturbed with a random amount of Gaussian noise, simulating a single step of a forward diffusion process. The key insight is to then use a large, pre-trained 2D text-to-image diffusion model Rombach et al. (2022) as a powerful, non-parametric prior. This 2D model, guided by the user’s text prompt via its cross-attention layers, predicts the noise component from the noisy rendered image. The SDS

Table 1.1: Comparative Analysis of 2D-Lifting vs. 3D-Native Diffusion Models

Feature	**2D-Lifting (Pre-trained 2D Priors)**	**2D-Lifting (Novel/Finetuned 3D-Aware 2D Models)**	**3D-Native Diffusion**
Primary Data Source	Vast 2D image-text datasets (indirectly)	Multi-view 2D renderings from 3D assets	3D datasets (point clouds, meshes, voxels, etc.)
3D Consistency	Prone to inconsistencies (e.g., Janus problem)	Improved consistency due to explicit multi-view training	Inherently better geometric consistency
Detail/Texture Quality	Often high due to powerful 2D priors	Can achieve good detail, dependent on 2D model capacity	Can struggle with high-frequency details due to 3D data limits
Generalization	Good generalization to diverse styles/objects	Generalization depends on diversity of MV training data	Limited by diversity of 3D training data
Training Cost	No training of core 2D diffusion model needed	Requires training/fine-tuning of 2D diffusion model on MV data	Expensive training due to high-dimensional 3D data
Inference Speed	Often slow per-scene optimization (SDS)	Can be fast (feed-forward 2D generation + reconstruction)	Can be fast once trained (feed-forward 3D generation)
Susceptibility to Janus Problem	High	Reduced, as model is trained for multi-view consistency	Low, as 3D structure is directly modeled
Key Examples	SDS (e.g., DreamFusion Poole et al. (2023))	Multi-view diffusion (e.g., MVDream Shi et al. (2023)), Bundle generation (e.g., Kiss3DGen **?**)	Direct diffusion on 3D reps (e.g., Point-E Nichol et al. (2022))

loss is calculated from the difference between the predicted noise and the actual noise added. Crucially, because the entire rendering pipeline is differentiable, the gradient from this loss can be backpropagated through the renderer to update the parameters of the 3D representation. By repeating this process for thousands of iterations from a wide distribution of camera poses, the 3D representation is gradually sculpted until its renderings from all angles are consistent with the text prompt. This mechanism was further refined by Variational Score Distillation (VSD), which re-frames the gradient computation to align distributions rather than single samples, leading to improved diversity and photorealism Wang et al. (2023).

Feed-Forward Methods: Decoupling Generation and Reconstruction

To overcome the significant time cost of per-scene optimization, feed-forward methods aim to generate 3D assets in a single pass. This is typically achieved by decoupling the problem into two stages: first generating a 2D or 2.5D representation, and then reconstructing the 3D model from it. One dominant approach involves training a specialized multi-view diffusion model, such as MVDream Shi et al. (2023), on large-scale datasets of rendered 3D objects Deitke et al. (2023). This model is architected to accept a text prompt and directly output a set of geometrically consistent multi-view images. These images can then be processed by a separate reconstruction module to produce the final 3D asset. An even faster pipeline has emerged that leverages pre-trained Large Reconstruction Models (LRMs) Hong et al. (2023). In this workflow, a standard text-to-image model generates just one or a few views of the target object. These views are then fed into the LRM, which is a highly optimized model trained specifically to infer a complete 3D shape from sparse image inputs. This design, exemplified by systems like One-2-3-45 Liu et al. (2023a), dramatically accelerates the text-to-3D process, enabling near-real-time creation.

3D-Native Generative Models

In contrast to 2D-lifting, 3D-native approaches apply a conditional diffusion process directly to a 3D data representation. The mechanism for incorporating the text condition is analogous to that in 2D models. The text prompt is encoded into a semantic embedding using a text encoder like CLIP. This embedding is then injected into the architecture of a 3D denoising network (e.g., a 3D U-Net) at each timestep, typically via cross-attention layers. This allows the text to directly guide the denoising process as it refines a noisy 3D representation—such as a voxel grid, a point cloud, or the latent code of a 3D autoencoder—into a clean, final shape. Models like Diffusion-SDF Li et al. (2023) apply this to generate Signed Distance Fields, while LION Zeng et al. (2022) operates within the compressed latent space of a point cloud VAE, showcasing how text can directly control the generation of diverse 3D geometric structures.

Single Image-to-3D Synthesis

Novel View Synthesis with 2D Diffusion Priors

The dominant methodology for image-to-3D generation involves a two-stage process of view synthesis followed by reconstruction. The process begins by encoding the input image into a compact feature vector using a vision encoder. This feature vector acts as the primary condition for a specialized novel view synthesis diffusion model, such as that in Zero-1-to-3 Liu et al. (2023b). This model is uniquely designed to also accept a target camera pose (as a relative transformation from the input view) as a secondary condition. By repeatedly invoking the model with the same image condition but a different target pose, a full set of multi-view images capturing the object from all sides can be generated. This collection of synthesized views is then passed to a reconstruction algorithm. While early methods used NeRF-fitting for high-quality results, the process was slow. More recent approaches, like DreamGaussian Tang et al. (2023), have adopted highly efficient representations like 3D Gaussian Splatting Kerbl et al. (2023) for the reconstruction stage, enabling rapid 3D model creation. This modular framework effectively leverages the generative power of 2D models for the geometrically complex task of view hallucination.

Direct Reconstruction with Large Reconstruction Models

A more recent and streamlined approach bypasses the intermediate step of generating multiple views. Instead, it utilizes a Large Reconstruction Model (LRM) 'Hong et al. (2023)'. An LRM is a transformer-based model specifically trained to take one or more images as input and directly output the parameters for a 3D representation, such as a triplane feature field. When applied to the image-to-3D task, the workflow is maximally efficient: the single input image is passed through the LRM in one forward pass to produce the 3D model. This eliminates the need for iterative optimization or generating dozens of intermediate images, representing the fastest path from a single image to a 3D asset.

Generation from Other Modalities

Sketch-Based Modeling

Sketches offer an intuitive medium for expressing 3D concepts. In a typical sketch-to-3D pipeline, the input sketch is first processed as a sparse image and fed into an encoder to extract its structural features. This feature representation then conditions a generative model that synthesizes the final 3D shape Chen et al. (2023). This process treats the sketch as a strong geometric guide. In an editing context, a sketch can be used to define a target deformation. In a system like SKED Mikaeili et al. (2023), the user's strokes are interpreted as geometric constraints that guide the non-rigid deformation of an existing 3D mesh, offering a direct and interactive modeling experience.

Layout-Conditioned Scene Generation

To generate complex, multi-object scenes, controlling the spatial layout is paramount. Layout-conditioned systems like CC3D Bahmani et al. (2023) achieve this through a hierarchical generation process. The user first defines an abstract scene layout, typically composed of labeled 3D bounding boxes. The model then performs a global diffusion process on a feature grid representing the entire scene volume, conditioned on this layout. This step establishes a shared context and ensures that inter-object relationships are coherent. Subsequently, the algorithm processes each bounding box individually. It crops the local features from the global grid corresponding to a specific box and uses them to condition a second, object-level generative model. This local model is responsible for synthesizing the detailed geometry of the individual object within its bounding box. This coarse-to-fine strategy ensures that generated scenes adhere to the specified global structure while populating it with contextually appropriate objects.

Chapter 2 Baseline Experiments

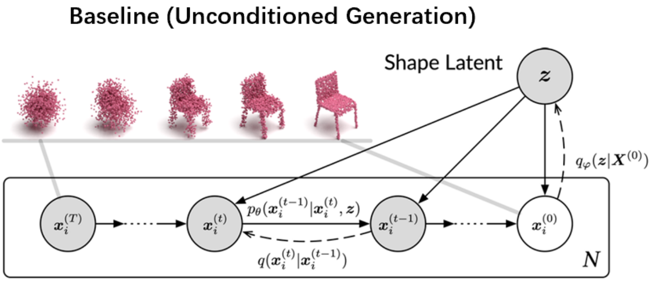
For baseline experiments, the early work of Luo and Hu (2021) was selected due to computational considerations.

2.1 Problem Formulation

The core idea of this work is to frame 3D point cloud generation as a probabilistic modeling task inspired by non-equilibrium thermodynamics. A point cloud, $X^{(0)} = \{x_i^{(0)}\}_{i=1}^{N}$, is treated as a collection of particles that diffuse over time from an original, structured distribution to a simple noise distribution Luo and Hu (2021). The generation task is therefore to learn the **reverse diffusion process**.

2.2 Pipeline

To generate a new point cloud, a latent code z is first sampled from the prior distribution $p(z)$. Then, points are sampled from a standard normal distribution, $X^{(T)} \sim \mathcal{N}(0, I)$, and are iteratively passed through the learned reverse Markov chain for $t = T, T-1, ..., 1$ to produce the final point cloud $X^{(0)}$.


Baseline (Unconditioned Generation)
Shape Latent
$\boldsymbol{z}$
$q_\varphi(\boldsymbol{z}|\boldsymbol{X}^{(0)})$
$\boldsymbol{x}_i^{(T)}$
$\boldsymbol{x}_i^{(t)}$
$p_\theta(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{z})$
$\boldsymbol{x}_i^{(t-1)}$
$\boldsymbol{x}_i^{(0)}$
$q(\boldsymbol{x}_i^{(t)}|\boldsymbol{x}_i^{(t-1)})$
N

Diffusion Probabilistic Models for 3D Point Cloud Generation

Shitong Luo, Wei Hu *
Wangxuan Institute of Computer Technology
Peking University
{luost, forhuwei}@pku.edu.cn

Figure 2.1

2.3 Experiments

The following table summarizes the quantitative results across 4 categories from the ShapeNet dataset. The metrics used for evaluation are 1-Nearest Neighbor Accuracy (1-NNA), Coverage (COV), Jensen-Shannon Divergence (JSD), and Minimum Matching Distance (MMD). For 1-NNA and COV, Chamfer Distance (CD) is used as the distance metric.

Category	Step[$\times 10^4$]	1-NNA-CD(↓)	COV-CD (↑)	JSD (↓) [$\times 10^{-2}$]	MMD-CD (↓) [$\times 10^{-3}$]
airplane	10	79.3%	44.4%	5.10	3.52
	20	78.2%	49.6%	4.16	3.47
	30	74.1%	44.4%	4.81	3.58
bag	10	10.0%	80.0%	23.95	21.4
	20	20.0%	80.0%	27.75	25.2
	30	30.0%	80.0%	23.92	24.9
car	10	85.0%	30.4%	3.73	4.78
	20	83.3%	31.1%	3.57	4.62
	30	81.5%	31.5%	3.50	4.54
table	10	66.6%	47.5%	1.97	13.7
	20	64.3%	43.7%	2.31	12.1
	30	63.4%	48.3%	2.06	12.9

Table 2.1: Quantitative results of the baseline model on four ShapeNet categories.

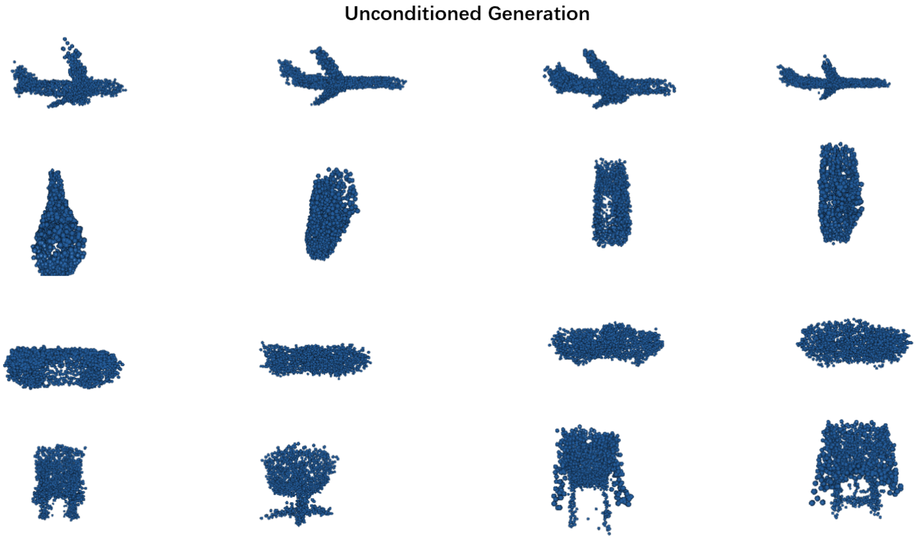


Figure 2.2

Chapter 3 Image-conditioned Point Cloud Generation

3.1 Problem Formulation

Building upon the foundation of unconditional generation, I further explore the task of generating a corresponding 3D point cloud conditioned on a single 2D image. The objective is to recover the 3D shape, represented as a point cloud $X^{(0)}$, from a given image I. We formalize this task as a conditional probabilistic generative model. My approach extends from the baseline unconditional diffusion model. In the original model, the reverse diffusion process (i.e., the generation process) is guided by a latent variable z, with its probabilistic form being $p_\theta(X^{(0)}|z)$. For the current conditional generation task, this latent variable z is replaced by a deterministic feature vector c_I, which is extracted from the input image I.

Consequently, the conditional reverse diffusion process can be expressed as:

$$p_\theta(X^{(0)}|c_I) = p(X^{(T)}) \prod_{t=1}^{T} p_\theta(X^{(t-1)}|X^{(t)}, c_I)$$

Here, $X^{(T)}$ is a noise sample drawn from a standard normal distribution, and $c_I = E_\phi(I)$ is the condition vector extracted from image I by an image encoder E_ϕ. The transition kernel of the reverse process is correspondingly conditioned at each step. Its mean is predicted by a neural network μ_θ that takes the noisy point cloud, the timestep, and the image condition as input:

$$p_\theta(x^{(t-1)}|x^{(t)}, c_I) = \mathcal{N}(x^{(t-1)}|\mu_\theta(x^{(t)}, t, c_I), \beta_t I)$$

The training objective is to optimize the network parameters θ and the image encoder parameters ϕ, enabling the model to accurately guide the progressive denoising of the point cloud based on the image features c_I, ultimately recovering a 3D shape that matches the image content.

3.2 Pipeline

To achieve generation from a 2D image to a 3D point cloud, I designed a two-stage training pipeline. The core idea is to first learn an effective image representation and then leverage this representation to guide the training of the conditional diffusion model. **Stage 1: Image Encoder Pre-training** The goal of this stage is to train an image encoder E_ϕ capable of compressing an input 2D image I into an information-rich, low-dimensional feature vector c_I. We employ an auto-encoder architecture, where an encoder extracts features, and a decoder reconstructs the original image from this feature vector. By minimizing the reconstruction loss, I compel the encoder to learn the key geometric and semantic information of the image. This process is unsupervised and uses only the rendered images from the ShapeNet dataset. **Stage 2: Conditional Diffusion Model Training** In the second stage, we train the conditional diffusion model on pairs of (Image I, 3D Point Cloud $X^{(0)}$). The weights of the pre-trained image encoder E_ϕ are loaded and serve as the conditioning module. During training, for each sample, we first extract the image condition via $c_I = E_\phi(I)$. This condition vector c_I is then fed into the diffusion model's denoising network μ_θ, along with the noisy point cloud $x^{(t)}$ and the timestep t.

Notably, the weights of the pre-trained image encoder also participate in gradient backpropagation and optimization during this stage. This is done to achieve an alignment between the 2D image feature space and the 3D geometry feature space, ensuring that the features extracted by the encoder are not only effective for image reconstruction but also for guiding 3D shape generation.

The entire pipeline is illustrated in the figure below.

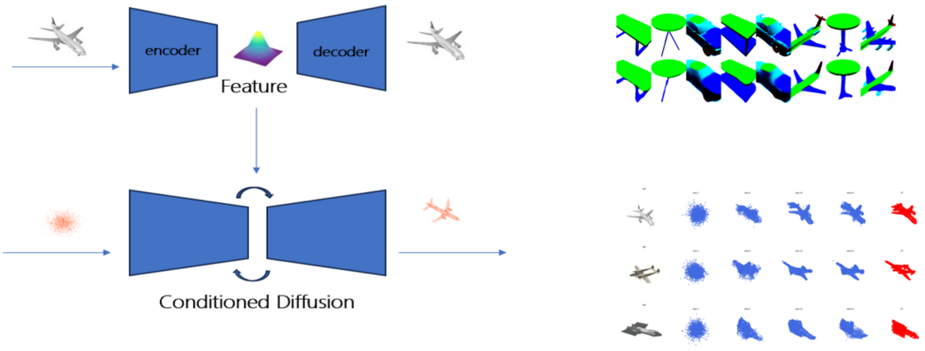


Figure 3.1: A schematic of the two-stage training pipeline for the conditional generation model.

3.3 Training

I conducted model training on the ShapeNet with rendering dataset, which provides a large collection of 3D models and their corresponding multi-view renderings, serving as the source for our (image, point cloud) data pairs. The training process strictly followed the two-stage procedure described previously. For training the conditional diffusion model, we used the Adam optimizer with an appropriate learning rate and batch size. The training loss curve is shown in the figure below, where the loss steadily decreases and eventually converges, indicating that our model was trained effectively.

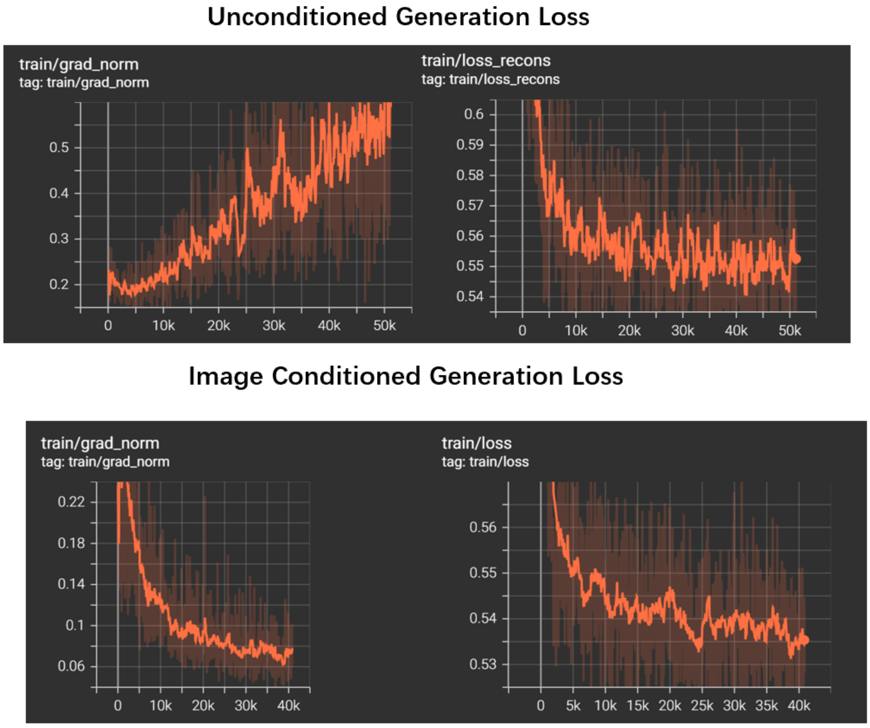
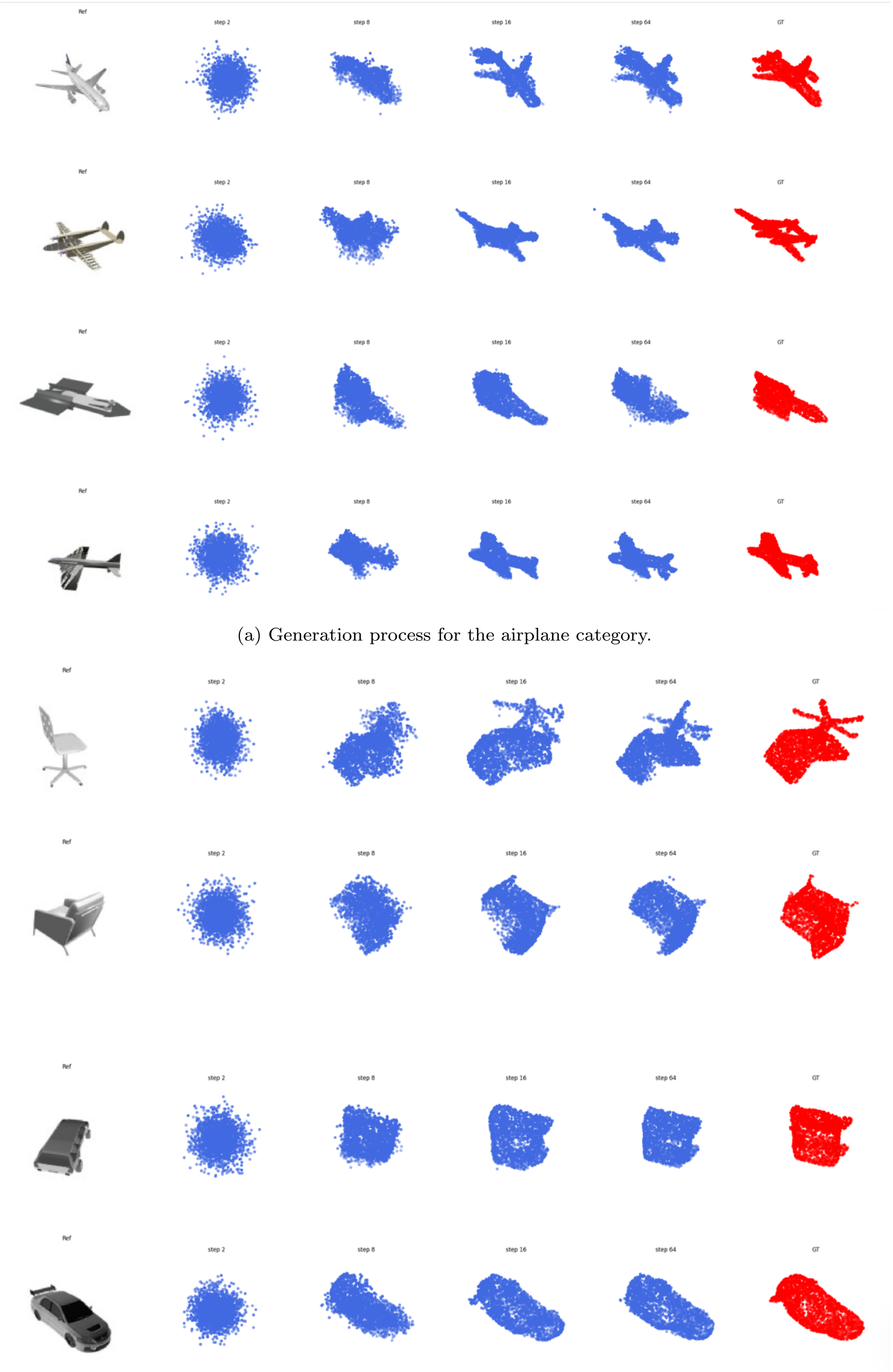


Figure 3.2: The loss curve during the training of the conditional diffusion model.

The figure below provides a visualization of the reverse diffusion process during training, starting from random noise (left) and progressively generating a complete shape (right).



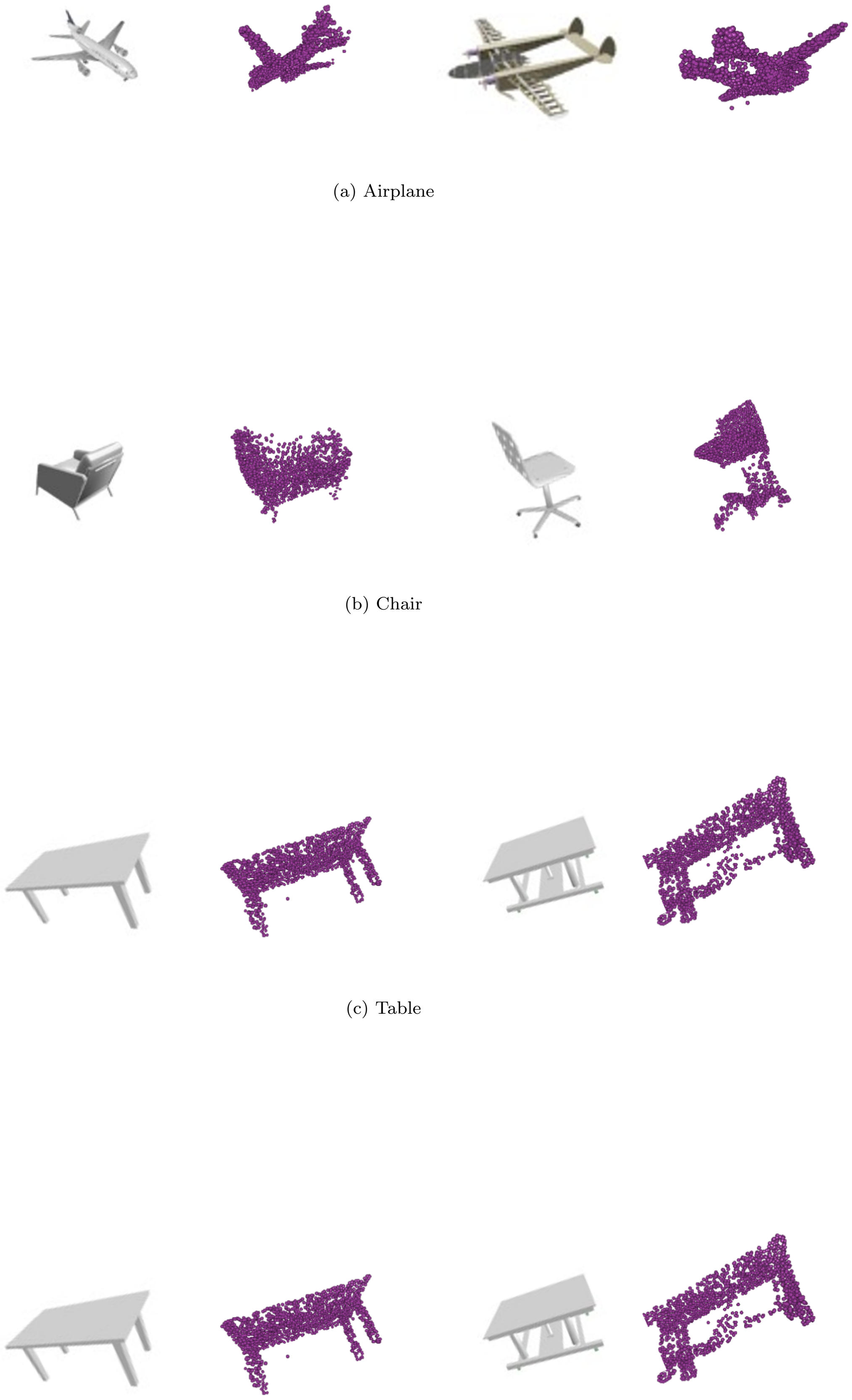
(a) Generation process for the airplane category.

(b) Generation process for the some other categories.

Figure 3.3: Visualization of the reverse denoising process from noise to the target shape.

3.4 Results

I performed a qualitative evaluation of the trained model's generation capabilities across several categories. Given a rendered image from the test set (unseen during training), the model can generate a point cloud of the corresponding category. The model successfully recovers plausible 3D shapes from the input 2D images. For objects with relatively rigid structures like airplanes and cars, the model effectively captures the overall silhouette and major components, such as wings, fuselage, and car bodies. For objects with more slender components, such as chairs and tables, the generated results are also largely correct, can distinguish some of structures like chair backs and legs. These results provide an initial validation of the effectiveness of my proposed image-conditioned point cloud generation method.



(a) Airplane

(b) Chair

(c) Table

(d) Car

Figure 3.4:

Bibliography

Sherwin Bahmani, Jeong Joon Park, Despoina Paschalidou, Xingguang Yan, Gordon Wetzstein, Leonidas Guibas, and Andrea Tagliasacchi. Cc3d: Layout-conditioned generation of compositional 3d scenes. *arXiv preprint arXiv:2303.12074*, 2023.

Tianrun Chen, Chenglong Fu, Ying Zang, Lanyun Zhu, Jia Zhang, Papa Mao, and Lingyun Sun. Deep3dsketch+: Rapid 3d modeling from single free-hand sketches. In *International Conference on Multimedia Modeling*, pages 16–28. Springer, 2023.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020a.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020b. URL `https://arxiv.org/abs/2006.11239`.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12642–12651, 2023.

Jiantao Lin, Xin Yang, Meixi Chen, Yingjie Xu, Dongyu Yan, Leyi Wu, Xinli Xu, Lie XU, Shunsi Zhang, and Ying-Cong Chen. Kiss3dgen: Repurposing image diffusion models for 3d asset generation, 2025. URL `https://arxiv.org/abs/2503.01370`.

Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023a.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023b.

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

Aryan Mikaeili, Or Perel, Mehdi Safaee, Daniel Cohen-Or, and Ali Mahdavi-Amiri. SKED: Sketch-guided text-based 3d editing. In *ICCV*, 2023.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

Gimin Nam, Mariem Khlifi, Andrew Rodriguez, Alberto Tono, Linqi Zhou, and Paul Guerrero. 3d-ldm: Neural implicit 3d shape generation with latent diffusion models. *arXiv preprint arXiv:2212.00842*, 2022.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T Barron, Amit H Bermano, Eric Ryan Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, et al. State of the art on diffusion models for visual computing. *arXiv preprint arXiv:2310.07204*, 2023.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3d using 2d diffusion. In *Int. Conf. Learn. Represent.*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.

Chen Wang, Hao-Yang Peng, Ying-Tian Liu, Jiatao Gu, and Shi-Min Hu. Diffusion models for 3d generation: A survey. *Computational Visual Media*, 11(1):1–28, 2025. doi: 10.26599/CVM.2025.9450452. URL `https://www.sciopen.com/article/10.26599/CVM.2025.9450452`.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023.

Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022.