Text-Conditioned 3D Object Generation with Latent Diffusion Prior in NeRF

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent advancements in 3D generative models have focused on synthesizing high-quality 3D objects through diverse approaches, including voxel-based representations, multi-view consistency frameworks, and radiance field modeling. While these methods demonstrate significant progress, they often face challenges such as limited resolution, computational inefficiency, or lack of flexibility. To address these limitations, we propose a novel text-conditioned NeRF model that integrates a latent diffusion prior to generating high-quality 3D objects. Our approach leverages textual guidance to improve adaptability, efficiency, and control in 3D generation tasks. Experiments demonstrate the superiority of our method in both fidelity and generalization across diverse scenes.

1 Introduction

The generation of high-quality 3D objects has become a critical focus in computer vision and graphics, driven by applications ranging from virtual reality and gaming to digital content creation. Traditional 3D modeling methods, such as voxel-based and mesh-based approaches, often suffer from scalability and flexibility limitations, hindering their applicability to large-scale or diverse datasets. Recent advancements in generative modeling, such as neural implicit representations and diffusion models, have introduced promising solutions for synthesizing detailed 3D scenes.

Despite these advancements, challenges remain. Voxel-based approaches are constrained by resolution limits, leading to suboptimal visual quality [Wu et al., 2016]. Multi-view consistency frameworks, while addressing some of these issues, often incur significant computational costs due to iterative generation processes [Watson et al., 2022]. Meanwhile, radiance field modeling methods achieve impressive quality, but require substantial resources, making them less efficient for scalable applications [Müller et al., 2023].

To tackle these challenges, we introduce a text-conditioned NeRF model that integrates a latent diffusion prior for flexible and efficient 3D object generation. By leveraging textual descriptions as a guiding signal, our approach achieves superior adaptability and control, enabling high-quality 3D synthesis of diverse scenarios. In this paper, we outline our model design, training process, and experimental validation, demonstrating its effectiveness in overcoming the limitations of prior methods. Our main contribution are as follows:

- We propose a text-conditioned 3D object generation method that leverages textual guidance to generate high-quality and diverse 3D objects.
- Through experiments, we demonstrate that our method improves generation quality, enhances generalization ability, and boosts training stability and efficiency.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

2 Related Work

Recent advancements in 3D generative models have explored diverse methodologies to efficiently and effectively synthesize high-quality 3D objects. These approaches can be broadly categorized into voxel-based representations, multi-view consistency frameworks, and radiance field modeling.

Voxel-based methods were among the first attempts to generate 3D objects by mapping a low-dimensional latent space (e.g., Gaussian distribution) to a 3D voxel grid. For instance, early work such as 3D-GAN [Wu et al., 2016] employed a generator-discriminator framework where the generator synthesized voxelized 3D shapes, and the discriminator evaluated their realism. While effective in learning the structural properties of 3D objects, the voxel-based representation inherently struggles to capture fine-grained details due to its resolution constraints, leading to suboptimal visual quality.

Multi-view consistency frameworks aim to address the challenges of limited views and ensuring coherence across different perspectives. For example, diffusion-based methods like 3DiM [Watson et al., 2022] employ stochastic conditioning during denoising steps, where randomly selected views guide the generation of consistent multi-view objects. Utilizing shared UNet weights for clean conditioning views and target views, and incorporating cross-attention layers to mix information between input and output views, these frameworks significantly enhance 3D consistency. However, the iterative nature of diffusion-based processes results in substantial computational overhead, particularly for generating complete 3D objects with high fidelity.

Radiance field modeling has emerged as a powerful paradigm for synthesizing realistic 3D geometries and detailed images. Methods like DiffRF [Müller et al., 2023] build upon neural radiance field (NeRF) techniques by integrating them with diffusion models, enabling direct operation on 3D radiance fields. This approach achieves remarkable quality but requires significant computational resources, highlighting an ongoing trade-off between visual fidelity and efficiency.

These foundational approaches illustrate the inherent trade-offs in 3D object generation—balancing quality, efficiency, and consistency. Motivated by these insights, our research introduces a text-conditioned NeRF model that leverages the strengths of radiance field modeling while addressing the limitations of prior methods. Our approach seeks to achieve high-quality 3D object synthesis with improved computational efficiency and adaptability.

3 Baseline Model: Single-Stage Diffusion NeRF

3.1 Multiscene NeRF

Traditional NeRF methods are designed to learn a unique representation for each individual scene, training a separate model for every scene from scratch. This scene-specific approach, while effective for detailed 3D reconstructions, becomes computationally prohibitive and inflexible when dealing with large datasets encompassing multiple scenes. To address these limitations, Chen et al. [2023a] introduces a more scalable and efficient architecture that leverages a shared NeRF network across multiple scenes.

At the core of this approach is the concept of latent scene codes—compact, learnable embeddings that encode the unique characteristics of each scene. Instead of feeding raw positional and camera data directly into the network, Multiscene NeRF uses these latent codes in conjunction with scene-specific inputs. The NeRF network then functions as an auto-decoder (Park et al. [2019]), reconstructing the 3D structure of a scene based on its corresponding latent representation. This design allows for a single network to handle multiple scenes by simply switching the latent code, making the model both memory-efficient and adaptable.

The introduction of latent codes not only reduces the storage and computational requirements but also enhances the model's ability to generalize across scenes. By sharing parameters across scenes, Multiscene NeRF can learn a more robust representation, enabling tasks like few-shot scene synthesis or interpolation between scenes.

3.2 Latent Diffusion Model

Latent Diffusion Models (LDMs) are a class of generative models that leverage the power of diffusion processes within a latent space to generate high-quality data samples efficiently. Unlike standard

diffusion models that operate directly in pixel space, LDMs compress input data into a lower-dimensional latent space using a pre-trained encoder, such as a variational autoencoder (VAE from Kingma and Welling [2022]) or a similar structure. The diffusion process then operates within this latent space, significantly reducing computational overhead while maintaining fidelity in the generated outputs. By iteratively denoising latent variables, LDMs progressively refine samples from noise to coherent outputs.

In the context of multiscene NeRF, LDMs act as a crucial approach in generating latent scene codes. These codes act as compact representations of scenes, which the NeRF model uses to reconstruct diverse 3D scenes. The integration of LDMs allows for efficient and flexible generation of latent scene representations, enabling the model to synthesize a wide variety of scenes while retaining high quality. Moreover, LDM also provides great generalization ability for multiscene NeRF, enabling for novel and diversified scene genreation capability, making it a powerful framework for tasks requiring efficient and scalable 3D reconstruction.

3.3 Single-Stage Diffusion NeRF (SSDNeRF)

In our project, we adopt the Single-Stage Diffusion NeRF (SSDNeRF) from Chen et al. [2023b] model as our baseline. This is a multiscene NeRF that leverage latent diffusion model for latent scene codes generation. However, different from previous similar approaches using two-stage training scheme like Bautista et al. [2022], Müller et al. [2023], Dupont et al. [2022], Shue et al. [2022], this model introduce a single-stage training approach, enabling the model to learn less noisy latent scene codes, as well as empowerinng its few-shot generation ability.

3.3.1 Single-Stage Training Paradigm

One of the key differentiators of SSDNeRF from prior multiscene NeRF approaches is its single-stage training process. Traditional methods typically adopt a two-stage training pipeline:

1. Train the NeRF auto-decoder to learn per-scene latent codes.
2. Use these latent codes as ground truth to train a latent diffusion model (LDM).

This two-stage approach suffers from several drawbacks. First, the learned latent codes are often noisy, leading to suboptimal scene representations. Additionally, learning from sparse views becomes challenging, further reducing the quality of generated scenes.

In contrast, SSDNeRF integrates the training of the diffusion prior (LDM) and the NeRF decoder into a single-stage framework. This joint optimization ensures that the latent scene codes are refined throughout the process, leading to clearer and more accurate scene representations, even under sparse view conditions. Experimental results have demonstrated the superiority of single-stage training, with notable improvements in the clarity and consistency of the reconstructed scenes compared to the two-stage approach.

3.3.2 Loss Functions

The loss function in SSDNeRF comprises two components:

$$\mathcal{L} = \lambda_{\text{rend}}\mathcal{L}_{\text{rend}}(\{x_i\}, \psi) + \lambda_{\text{diff}}\mathcal{L}_{\text{diff}}(\{x_i\}, \phi). \tag{1}$$

- $\mathcal{L}_{\text{render}}$: An ℓ_2 pixel-wise loss that measures the difference between the rendered image and the ground truth observation. This term ensures accurate 2D image rendering from the 3D scene representation.

$$\mathcal{L}_{\text{rend}}(\{x_i\}, \psi) = \mathbb{E}_i \left[\sum_j \frac{1}{2} \left\| y_{ij}^{\text{gt}} - y_\psi(x_i, r_{ij}^{\text{gt}}) \right\|^2 \right]. \tag{2}$$

- $\mathcal{L}_{\text{diffusion}}$: An ℓ_2 denoising loss that quantifies the difference between the denoised latent code produced by the diffusion prior and the actual scene code. This term drives the LDM to generate accurate latent scene codes.

$$\mathcal{L}_{\text{diff}}(\phi) = \mathbb{E}_{i,t,\epsilon} \left[\frac{1}{2} w^{(t)} \left\| \hat{x}_\phi(x_i^{(t)}, t) - x_i \right\|^2 \right]. \tag{3}$$

Both loss terms rely on the latent scene code, which is iteratively updated during training to balance the demands of image rendering and diffusion-based denoising.

In our project, we use the same loss functions for training.

3.3.3 Image-Conditioned Generation

SSDNeRF also supports image-conditioned generation. However, rather than directly generating a scene conditioned on a given image, this capability is implemented through a fine-tuning approach. Specifically, during fine-tuning, the denoising and NeRF decoder models are frozen, and gradients are backpropagated to adjust the noise and latent scene codes. This allows for the adaptation of the model to new scenes based on limited image inputs, enhancing its practical applicability in real-world scenarios.

In summary, SSDNeRF represents a state-of-the-art baseline for unified 3D generation and reconstruction, offering significant improvements in both training efficiency and output quality through its single-stage training paradigm.

4 Proposed Method

In this paper, we propose a novel architecture for text-guided 3D scene generation. An overview of the proposed framework is illustrated in Figure 1.

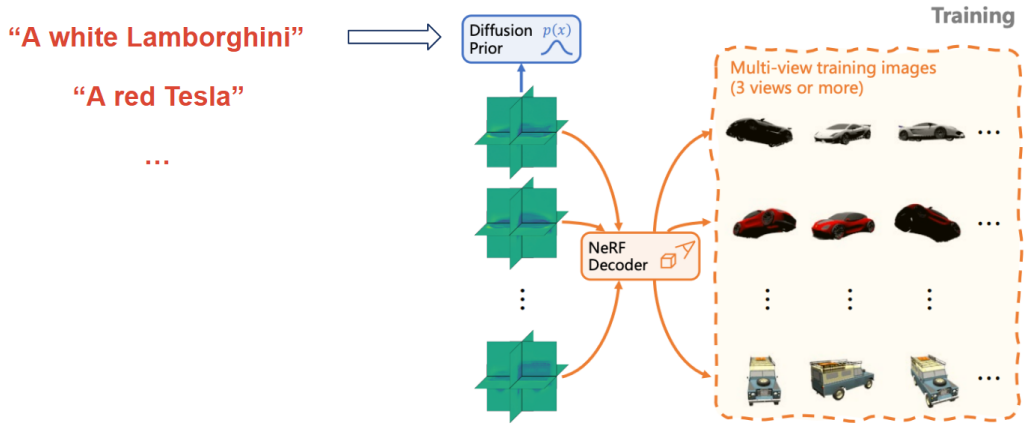


Figure 1: Illustration of text-conditioned diffusion prior for 3D scene generation, modified from Chen et al. [2023b]

The architecture leverages a diffusion model to generate a latent scene code that encapsulates the 3D object's structure, guided by the input text. Subsequently, a NeRF-based decoder reconstructs the 3D object from the generated scene code.

Our method integrates text information into the diffusion process through the following key steps:

Text embedding. We utilize the CLIP text encoder to transform each input sentence into a text embedding $x \in \mathbb{R}^d$.

Information extraction and reshape. We used a MLP to extract the information from the text embedding and reshape the embedding to the shape of the scene code. Specifically, we adopted the idea of low-rank approximation to decrease the computational complexity of our model.

$$x' = MLP(x), \text{where } x' \in \mathbb{R}^{d_s \times d_s}, \tag{4}$$

where d_s is the size of scene code with a default value 128.

The MLP output will then be reshaped as $[1, d_s, d_s]$ to match the shape of the scene code.

$$text\_cond = x'.reshape(1, d_s, d_s). \tag{5}$$

Condition concatenation and convolution. The original noisy scene code $code \in \mathbb{R}^{c \times d_s \times d_s}$ is augmented by concatenating the text condition $text\_cond$ along the channel dimension,

$$code\_cond = \text{concat}(code, text\_cond, axis = 0), \tag{6}$$

where $code\_cond$ has the shape of $[c + 1, d_s, d_s]$. Please note that we omit the dimension of batch size in the above equation.

A convolutional layer is then applied to restore the channel dimension to its original size. This layer not only adjusts the dimensions, but also facilitates information sharing across the channels.

With the text-guided diffusion input, the whole model can be trained using methods mentioned in 3.3.1. The illustration of the overall framework is shown in Figure 2.

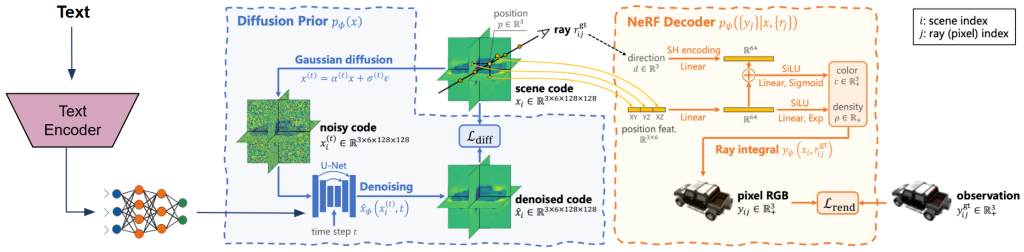


Figure 2: Illustration of the Text-Guided SSDNeRF framework: Text description will be passed to a text encoder (CLIP) to obtain embeddings, which is passed to a trainable MLP to generate text-condition embeddings and appended to the scene code in the diffusion model to create text-conditioned scene code. The scene code is passed to the NeRF decoder, which uses ray integral to reconstruct the 3D scene. Modified from Chen et al. [2023b]

5 Experiment

The objective of our training pipeline was to enable controllable and flexible 3D scene generation guided by textual descriptions. Traditional NeRF frameworks often rely on multiple images and extensive training times to generate 3D representations of specific scenes, which limits their adaptability and efficiency. While unconditional and image-conditioned generation approaches offer improvements, they lack the flexibility and control that textual guidance can provide. To address these limitations, we introduced text-conditioning (Figure 1) into the SSDNeRF framework.

Our training process incorporated scene descriptions into the diffusion model to guide the generation of scene codes. The pipeline (Figure 2) began with a CLIP text encoder, which converted input text descriptions into embeddings. These embeddings were expanded using a trainable MLP to match the dimensionality of the noisy scene code. After expansion, the embeddings were concatenated with the noisy scene code and processed through a convolutional layer to ensure effective integration of textual information. The conditioned noisy scene code was then passed through the diffusion model to generate the final scene code, which was subsequently decoded by the NeRF model to render the 3D scene.

For training and testing, we used the Amazon Berkeley Objects (ABO) dataset, focusing on the Table catalog. The metadata, such as product descriptions and color attributes, provided text guidance, while RGB images and camera poses were used for 3D object representation. We trained both our text-guided SSDNeRF and the baseline SSDNeRF for 120,000 iterations on a single NVIDIA H100 GPU, using a batch size of 8 and learning rates of 6×10^{-5} for the diffusion model and 6×10^{-4} for the decoder.

The trained models were evaluated on their ability to reconstruct 3D tables using text guidance. To assess overall 3D generation quality, we used a combined train + test partition of the ABO Tables

dataset. For generalization quality, we evaluated the models on unseen text descriptions from the test set. Metrics such as Fréchet Inception Distance (FID) and Kernel Inception Distance (KID) were used to quantify fidelity and diversity. Additionally, we visualized the reconstructed 3D tables to qualitatively compare the models.

6 Results

6.1 Quantitative Results

Our experiments demonstrate the significant advantages of introducing text-conditioning to the SSDNeRF framework. For quantitative evaluation, we used FID and KID, which are standard metrics for measuring fidelity and diversity in generative models. In terms of overall 3D generation quality, evaluated on the train + test partition of the ABO Tables dataset, our text-guided SSDNeRF outperformed the baseline SSDNeRF with a 19% lower FID (25.178 vs. 31.141) and a 25% lower KID (8.953 vs. 11.885). These results indicate that integrating textual guidance into the diffusion process improves the quality and diversity of the generated 3D scenes.

For generalization performance on unseen text descriptions, both models achieved comparable FID scores (41.082 vs. 41.605). However, our text-guided SSDNeRF exhibited superior diversity, achieving a 13% lower KID (11.579 vs. 13.3215). These findings suggest that the text-conditioning mechanism not only enhances overall scene generation quality but also improves the model's ability to generalize to novel textual inputs.

6.2 Qualitative Results

Qualitative comparisons further validate the benefits of text-conditioning. Figures 3, 4, and 5 illustrate the training dynamics of both models. Across all loss metrics—DDPM loss, decoder loss, and reconstruction pixel loss—our text-guided SSDNeRF achieved faster and steadier convergence compared to the baseline SSDNeRF. Notably, the DDPM loss for the text-guided model displayed steady convergence, whereas the baseline exhibited higher variability, reflecting the stabilizing influence of textual guidance on training.

Visualization results provide additional insights. Figure 6 compares the 3D reconstructions generated by both models. Our text-guided SSDNeRF produced finer details, such as well-defined tabletop edges, while significantly reducing noise in the reconstructed geometry. Figure 7 demonstrates the model's ability to generate 3D scenes from unseen textual descriptions. The outputs align well with the provided descriptions, accurately capturing attributes such as table type and texture. However, ambiguities in text descriptions, such as conflicting colour specifications ("Grey; Green"), occasionally led to inconsistencies in generated attributes. Additionally, the visualized scene codes exhibited clear and structured representations, highlighting the diffusion model's ability to produce high-quality priors for NeRF decoding.

Table 1: Overall 3D generation comparison between SSDNeRF and Text-Guided SSDNeRF on the ABO Tables (train + test) dataset

Method (Train + Test)	FID ↓	KID/10^{-3} ↓
Original SSDNeRF (120K)	31.141	11.885
Text-guided SSDNeRF (120K)	**25.178**	**8.953**

Table 2: 3D generation generalization comparison between SSDNeRF and Text-Guided SSDNeRF on the ABO Tables (test) dataset

Method (Test)	FID ↓	KID/10^{-3} ↓
Original SSDNeRF (120K)	**41.0820**	13.3215
Text-guided SSDNeRF (120K)	41.605	**11.579**

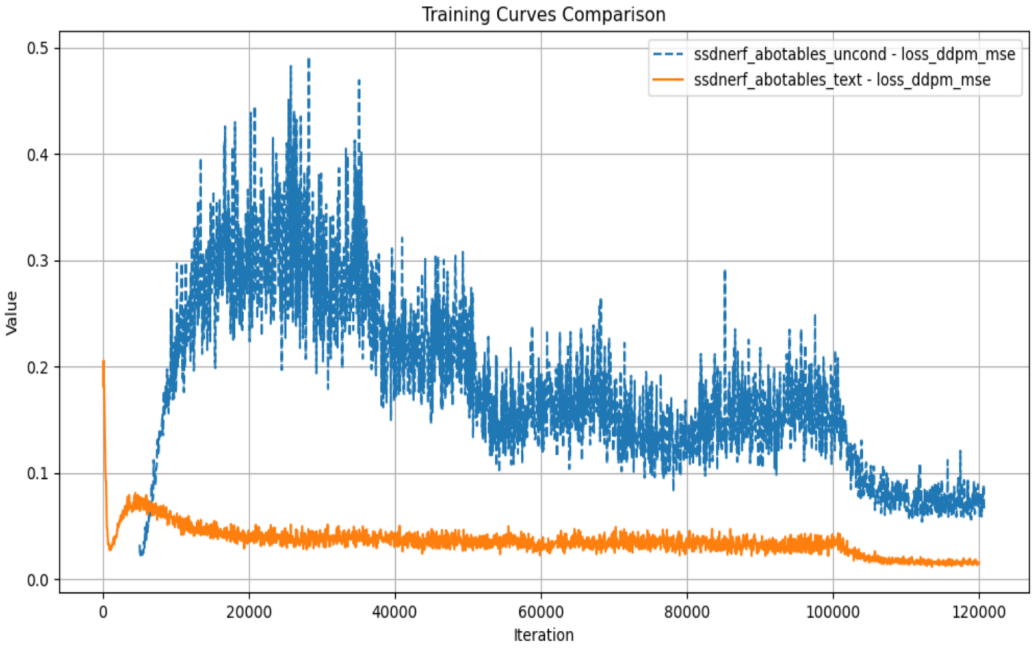


Figure 3: Comparison of the DDPM (Denoising Diffusion Probabilistic Model) training loss between SSDNeRF and Text-Guided SSDNeRF

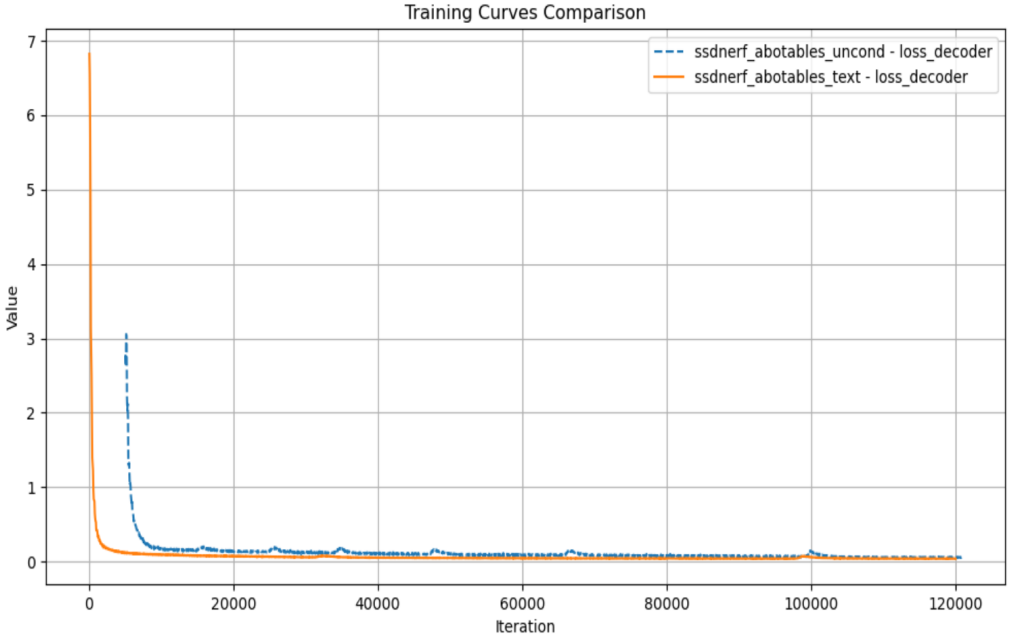


Figure 4: Comparison of the NeRF decoder training loss between SSDNeRF and Text-Guided SSDNeRF

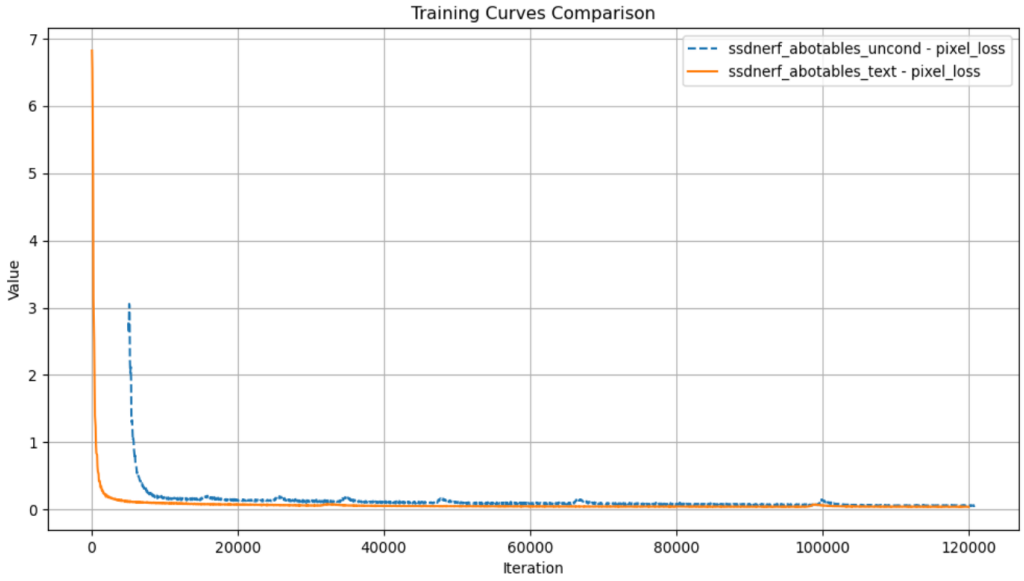


Figure 5: Comparison of the reconstruction pixel loss between SSDNeRF and Text-Guided SSDNeRF

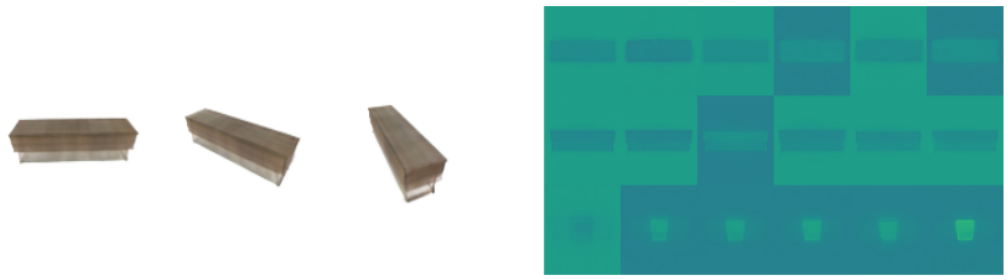
Figure 6: Comparison of the 3D reconstruction mesh between SSDNeRF and Text-Guided SSDNeRF (Ours)

7 Discussions

The results highlight the significant advantages of introducing text-conditioning to the SSDNeRF framework. By leveraging textual descriptions, our approach achieves superior fidelity, diversity, and generalization compared to the baseline SSDNeRF. This demonstrates the effectiveness of text as a flexible and controllable conditioning signal, particularly in applications requiring adaptable and customizable outputs.

Despite its advantages, there are areas for improvement. One challenge lies in handling ambiguities or inconsistencies in text descriptions, which occasionally result in inaccuracies in generated attributes. Future work could focus on refining the text-conditioning process to better handle such cases, potentially through advanced natural language understanding techniques or multi-modal input fusion. Another avenue for exploration involves a deeper analysis of scene codes, particularly how their structure influences NeRF decoding. Insights from such studies could optimize the diffusion process further and enhance the model's performance.

Expanding the dataset to include more complex scenes or diverse object categories could also push the boundaries of text-conditioned 3D generation. Integrating richer text descriptions or exploring



"Amazon Brand – Rivet Modern Six-Compartment TV Media Console, Walnut;Brown"

'Successful Home Dining Table Sturdy Wooden Table top, Comfortable and Roomy (Grey);Green"

"Amazon Brand – Rivet Molly Round Marble and Stainless Steel Side End Table;Grey"

Figure 7: Examples of text-conditioned 3D generation from unseen table descriptions. The left are generated 3D scenes, the right is the $3 \times 6 \times 128 \times 128$ scene code obtained from the diffusion model, the bottom is the text description used to conditionally generate the scene code.

multi-modal conditioning—such as combining text and image inputs—could provide more robust and flexible guidance signals. These improvements would not only enhance model performance but also broaden the applicability of text-conditioned 3D generative models in various real-world contexts.

8 Conclusion

In this paper, we introduced a novel text-guided 3D scene generation framework that integrates a diffusion model with NeRF-based decoding. Our approach effectively incorporates textual information into the diffusion process, improving both the quality and diversity of the generated 3D scenes. Our quantitative experimental results show that our method outperforms the baseline in both fidelity and diversity, and has superior generalization capabilities. Qualitative results highlight the benefits of text-conditioning in achieving finer details, reduced noise, and improved convergence during training. Future work could focus on enhancing our model's ability to follow text guidance with greater accuracy and extending its capabilities to generate 3D objects across different categories based on language instructions.

References

Miguel Angel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, Afshin Dehghan, and Josh Susskind. Gaudi: A neural architect for immersive 3d scene generation, 2022. URL https://arxiv.org/abs/2207.13751.

Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. Factor fields: A unified framework for neural fields and beyond, 2023a. URL https://arxiv.org/abs/2302.01226.

Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction, 2023b. URL https://arxiv.org/abs/2304.06714.

Emilien Dupont, Hyunjik Kim, S. M. Ali Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one, 2022. URL https://arxiv.org/abs/2201.12204.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4328–4338, 2023.

Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion, 2023. URL https://arxiv.org/abs/2212.01206.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation, 2019. URL https://arxiv.org/abs/1901.05103.

J. Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion, 2022. URL https://arxiv.org/abs/2211.16677.

Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.