LN3Diff: Scalable Latent Neural Fields Diffusion for Speedy 3D Generation

Yushi Lan[1], Fangzhou Hong[1], Shuai Yang[2], Shangchen Zhou[1], Xuyi Meng[1], Bo Dai[3], Xingang Pan[1], and Chen Change Loy[1]

[1] S-Lab, Nanyang Technological University, Singapore
[2] Wangxuan Institute of Computer Technology, Peking University
[3] Shanghai Artificial Intelligence Laboratory

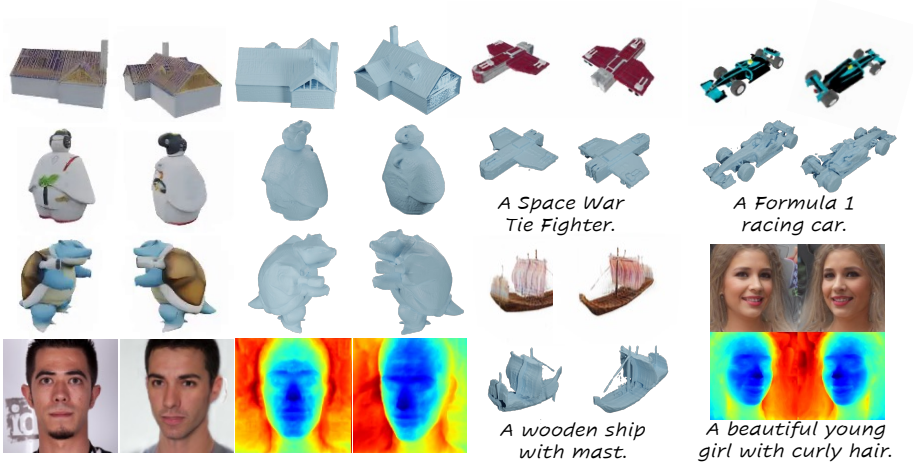

Fig. 1: We present LN3Diff, which performs efficient 3D diffusion learning over a compact latent space. The resulting model enables both high-quality monocular 3D reconstruction and text-to-3D synthesis.

Abstract. The field of neural rendering has witnessed significant progress with advancements in generative models and differentiable rendering techniques. Though 2D diffusion has achieved success, a unified 3D diffusion pipeline remains unsettled. This paper introduces a novel framework called **LN3Diff** to address this gap and enable fast, high-quality, and generic conditional 3D generation. Our approach harnesses a 3D-aware architecture and variational autoencoder (VAE) to encode the input image(s) into a structured, compact, and 3D latent space. The latent is decoded by a transformer-based decoder into a high-capacity 3D neural field. Through training a diffusion model on this 3D-aware latent space, our method achieves superior performance on Objaverse, ShapeNet

and FFHQ for conditional 3D generation. Moreover, it surpasses existing 3D diffusion methods in terms of inference speed, requiring no per-instance optimization. Video demos can be found on our project webpage: https://nirvanalan.github.io/projects/ln3diff.

Keywords: Generative Model · Reconstruction · Latent Diffusion Model

1 Introduction

The advancement of generative models [23,32] and differentiable rendering [90] has paved the way for a new research direction called neural rendering [90]. This field is continuously pushing the limits of view synthesis [56], editing [47], and particularly 3D object synthesis [6]. While 2D diffusion models [32, 84] have outperformed GANs in image synthesis [15] in terms of quality [73], controllability [107], and scalability [77], a unified 3D diffusion pipeline has yet to be established.

3D object generation methods using diffusion models can be categorized into 2D-lifting and feed-forward 3D diffusion models. In 2D-lifting methods, score distillation sampling (SDS) [68, 98] and Zero-123 [50, 79] achieve 3D generation by leveraging pre-trained 2D diffusion models. However, SDS-based methods require costly per-instance optimization and are prone to the multi-face Janus problem [68]. Meanwhile, Zero-123 fails to enforce strict view consistency. On the other hand, feed-forward 3D diffusion models [9, 38, 49, 57, 97, 105] enable fast 3D synthesis without per-instance optimization. However, these methods typically involve a two-stage pre-processing approach. First, during the data preparation stage, a shared decoder is learned over a large number of instances to ensure a shared latent space. This is followed by per-instance optimization to convert each 3D asset in the datasets into neural fields [101]. After this, the feed-forward diffusion model is trained on the prepared neural fields.

While the pipeline above is straightforward, it poses extra challenges to achieve high-quality 3D diffusion: **1) Scalability.** In the data preparation stage, existing methods face scalability issues due to using a shared, low-capacity MLP decoder for per-instance optimization. This approach is data inefficient, requiring over 50 views per instance [9, 57] during training. Consequently, computation cost scales linearly with the dataset size, hindering scalability for large, diverse 3D datasets. **2) Efficiency.** Employing 3D-specific architectures [12, 69, 93] is computationally intensive and necessitates representation-specific designs [109]. Consequently, existing methods compress each 3D asset into neural fields [101] before training. However, this compression introduces high-dimensional 3D latent, increasing computational demands and training challenges. Limiting the neural field size [9] might mitigate these issues but at the cost of reconstruction quality. In addition, the auto-decoding paradigm can result in an unclean latent space [24, 38, 81], unsuitable for 3D diffusion training [73]. **3) Generalizability.** Existing 3D diffusion models primarily focus on unconditional generation over single classes, neglecting high-quality conditional 3D generation (*e.g.*, text-to-3D) across generic, category-free 3D datasets. Furthermore, projecting monocular input images into

the diffusion latent space is crucial for conditional generation and editing [48,107], but this is challenging with the shared decoder designed for multi-view inputs.

In this study, we propose a novel framework called **L**atent **N**eural fields **3**D **Diff**usion (LN3Diff) to address these challenges and enable fast, high-quality and generic conditional 3D generation. Our method involves training a variational autoencoder [44] (VAE) to compress input images into a lower-dimensional 3D-aware latent space, which is more expressive and flexible compared to pixel-space diffusion [15,32,35,84]. From this space, a 3D-aware transformer-based decoder gradually decodes the latent into a high-capacity 3D neural field. This autoencoding stage is trained amortized with differentiable rendering [90], incorporating novel view supervision for multi-view datasets [8,14] and adversarial supervision for monocular dataset [39]. Thanks to the high-capacity model design, our method is more *view efficient*, requiring only two views per instance during training. After training, we leverage the learned 3D latent space for conditional 3D diffusion learning, ensuring effective utilization of the trained model for high-quality 3D generation. The pre-trained encoder can amortize the data encoding over incoming data, thus streamlining operations and facilitating efficient 3D diffusion learning while remaining compatible with advances in 3D representations.

To enhance efficient information flow in the 3D space and promote coherent geometry reconstruction, we introduce a novel 3D-aware architecture tailored for fast and high-quality 3D reconstruction while maintaining a structured latent space. Specifically, we employ a convolutional tokenizer to encode the input image(s) into a *KL*-regularized 3D latent space, leveraging its superior perceptual compression ability [20]. We employ transformers [16,65] to enable flexible 3D-aware attention across 3D tokens in the latent space. Finally, we up-sample the 3D latent and apply differentiable rendering for image-space supervision, making our method a self-supervised 3D learner [83].

In summary, we contribute a 3D-representation-agnostic pipeline for building generic, high-quality 3D generative models. This pipeline provides opportunities to resolve a series of downstream 3D vision and graphics tasks. Specifically, we propose a novel 3D-aware reconstruction model that achieves high-quality 3D data encoding in an amortized manner. Learning in the compact latent space, our model demonstrates state-of-the-art 3D generation performance on the ShapeNet benchmark [8], surpassing both Generative Adversarial Network (GAN)-based and 3D diffusion-based approaches. Our method shows superior performance in monocular 3D reconstruction and conditional 3D generation on ShapeNet, FFHQ, and Objaverse datasets, with a fast inference speed, *e.g.*, 3× faster against existing latent-free 3D diffusion methods [1].

2 Related Work

3D-aware GANs. GANs [23] have shown promising results in generating photorealistic images [3,40,41], inspiring researchers to explore 3D-aware generation [29,58,63]. Motivated by the recent success of neural rendering [54,56,64], researchers have introduced 3D inductive bias into the generation task [5,78], demon-

strating impressive 3D-awareness synthesis through hybrid designs [6,25,33,60,62]. This has made 3D-aware generation applicable to a series of downstream applications [47,85,86,106]. However, GAN-based methods suffer from mode collapse [92] and struggle to model datasets with larger scale and diversity [15]. Besides, 3D reconstruction and editing with GANs require elaborately designed inversion algorithms [48].

3D-aware Diffusion Models. The success of 2D diffusion models [32,84] has inspired their application to 3D generation. DreamFusion [36,68,98] adapted 2D models for 3D, but faces challenges like expensive optimization, mode collapse, and the Janus problem. Some methods propose learning the 3D prior in a 2D manner [7,50,51,91]. While these can produce photorealistic results, they lack view consistency and fail to fully capture the 3D structure. A canonical 3D diffusion pipeline involves a two-stage training process. First, an auto-decoder is pre-trained with multi-view images [17,38,49,57,81,97]. Then, 3D latent codes serve as the training corpus for diffusion. However, the auto-decoding stage leads to an unclean latent space and limited scalability. Moreover, the large latent codes, e.g., $256 \times 256 \times 96$ [97], hinder efficient diffusion learning [35].

Prior works, such as RenderDiffusion [1] and DMV3D [102], propose latent-free 3D diffusion by integrating rendering into diffusion sampling. However, this approach involves time-consuming volume rendering at each denoising step, significantly slowing down sampling. SSDNeRF [9] suggests a joint 3D reconstruction and diffusion approach but requires a complex training schedule and shows performance only in single-category unconditional generation. In contrast, our proposed LN3Diff trains 3D diffusion on a compressed latent space without rendering operations. As shown in Section 4, our method outperforms others in 3D generation and monocular 3D reconstruction, achieving three times faster speed. Additionally, we demonstrate conditional 3D generation over diverse datasets, whereas RenderDiffusion and SSDNeRF focus on simpler classes. Other approaches, like 3DGen [27] and VolumeDiffusion [88], perform diffusion in the 3D latent space but heavily rely on 3D data (e.g., point clouds and voxels) and do not support monocular datasets like FFHQ [39]. Moreover, their methods are designed for U-Net, whereas our DiT-based architecture offers greater scalability.

Generalized 3D Reconstruction and View Synthesis. To bypass the per-scene optimization of NeRF, researchers have proposed learning a prior model through image-based rendering [11,34,75,96,103]. However, these methods are primarily designed for view synthesis and lack explicit 3D representations. LoLNeRF [72] learns a prior through auto-decoding but is limited to simple, category-specific settings. Moreover, these methods are intended for view synthesis and cannot generate new 3D objects. VQ3D [76] adapts the generalized reconstruction pipeline to 3D generative models. However, it uses a 2D architecture with autoregressive modeling over a 1D latent space, ignoring much of the inherent 3D structure. NeRF-VAE [45] directly models 3D likelihood with a VAE posterior but is constrained to simple 3D scenes due to the limited capacity of VAE. Concurrently, LRM [11,34] has proposed a feedforward model for generalized monocular reconstruction. However, its latent space is not specifically

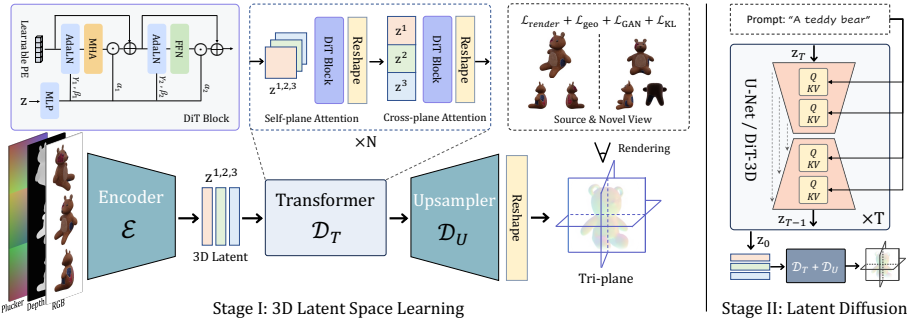

Fig. 2: Pipeline of LN3Diff. In the 3D latent space learning stage, a convolutional encoder $\mathcal{E}_{\phi}$ encodes a set of images $\mathcal{I}$ into the KL-regularized latent space. The encoded 3D latent is further decoded by a 3D-aware DiT transformer $\mathcal{D}_T$, in which we perform self-plane attention and cross-plane attention. The transformer-decoded latent is upsampled by a convolutional upsampler $\mathcal{D}_U$ towards a high-res tri-plane for rendering supervisions. In the next stage, we perform conditional diffusion learning over the compact latent space using either U-Net or DiT.

designed for learning a generative model, which limits its effectiveness for 3D diffusion learning.

3 Scalable Latent Neural Fields Diffusion

This section introduces our latent 3D diffusion model, which learns efficient diffusion prior over the compressed latent space by a dedicated variational autoencoder. Specifically, the goal of training is to learn a variational encoder $\mathcal{E}_{\phi}$ that maps a set of posed 2D image(s) $\mathcal{I} = \{I_i, ..., I_V\}$, of a 3D object to a latent code $\mathbf{z}$, a denoiser $\boldsymbol{\epsilon}_{\theta}(\mathbf{z}_t, t)$ to denoise the noisy latent code $\mathbf{z}_t$ given diffusion time step t, and a decoder $\mathcal{D}_{\psi}$ (including a Transformer $\mathcal{D}_T$ and an Upsampler $\mathcal{D}_U$) to map $\mathbf{z}_0$ to the 3D tri-plane $\widetilde{\mathcal{X}}$ corresponding to the input object.

Such design offers several advantages: (1) By explicitly separating the 3D data compression and diffusion stage, we avoid representation-specific 3D diffusion design [1, 38, 57, 81, 109] and achieve 3D representation/rendering-agnostic diffusion, which can be applied to any neural rendering techniques. (2) By leaving the high-dimensional 3D space, we reuse the well-studied Latent Diffusion Model (LDM) architecture [65, 73, 95] for computationally efficient learning and achieve better sampling performance with faster speed. (3) The trained 3D compression model in the first stage serves as an efficient and general-purpose 3D tokenizer, whose latent space can be easily re-used over downstream applications or extended to new datasets [13, 104].

In the following subsections, we first discuss the compressive stage with a detailed framework design in Sec. 3.1. Based on that, we introduce the 3D diffusion generation stage in Sec. 3.2 and present the condition injection in Sec. 3.3. The method overview is shown in Fig. 2.

3.1 Perceptual 3D Latent Compression

As analyzed in Sec. 1, directly leveraging neural fields for diffusion training hinders model scalability and performance. Inspired by previous work [20, 73], we propose to take multi-view image(s) as a proxy of the underlying 3D scene and compress the input image(s) into a compact 3D latent space. Though this paradigm is well-adopted in the image domain [20, 73] with similar trials in specific 3D tasks [4, 48, 55, 76], we, for the first time, demonstrate that a high-quality compression model is feasible, whose latent space serves as a compact proxy for efficient diffusion learning.

Encoder. Given a set of image(s) $\mathcal{I}$ of an 3D object where each image within the set $I \in \mathbb{R}^{H\times W\times 3}$ is an observation of an underlying 3D object from viewpoints $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_V\}$, LN3Diff adopts a convolutional encoder $\mathcal{E}_{\boldsymbol{\phi}}$ to encode the image set $\mathcal{I}$ into a latent representation $\mathbf{z} \sim \mathcal{E}_{\boldsymbol{\phi}}(\mathcal{I})$. To inject camera condition, we concatenate Plucker coordinates $\mathbf{r}_i = (\mathbf{d}_i, \mathbf{p}_i \times \mathbf{d}_i) \in \mathbb{R}^6$ channel-wise as part of the input [82], where $\mathbf{d}_i$ is the normalized ray direction and $\mathbf{p}_i$ is the camera origin corresponding to the camera $\mathbf{c}_i$, and $\times$ denotes the cross product. For challenging datasets like Objaverse [14], we also concatenate the rendered depth map, making our input a dense 3D colored point cloud [100].

Unlike existing works [20, 76] that operate on 1D order latent and ignore the internal structure, we choose to output 3D latent $\mathbf{z} \in \mathbb{R}^{h\times w\times d\times c}$ to facilitate 3D-aware operations, where $h = H/f, w = W/f$ are the spatial resolution with down-sample factor f, and d denotes the 3D dimension. Here we set $f = 8$ and $d = 3$ to make $\mathbf{z} \in \mathbb{R}^{h\times w\times 3\times c}$ a tri-latent, which is similar to tri-plane [6, 66] but in the compact 3D latent space. We further impose *KL-reg* [44] to encourage a well-structured latent space to facilitate diffusion training [73, 95].

Decoder Transformer. The decoder aims to decode the compact 3D codes $\mathbf{z}$ for high-quality 3D reconstruction. Existing image-to-3D methods [4, 6, 24, 48] adopt convolution as the building block, which lacks 3D-aware operations and impede information flow in the 3D space. Here, we adopt ViT [16, 65] as the decoder backbone due to its flexibility and effectiveness. Inspired by Rodin [97], we made the following reformulation to the raw ViT decoder to encourage 3D inductive bias and avoid the mixing of uncorrelated 3D features: (1) *Self-plane Attention Block*. Given the input $\mathbf{z} \in \mathbb{R}^{l\times 3\times c}$ where $l = h \times w$ is the sequence length, we treat each of the three latent planes as a data point and conduct self-attention within itself. This operation is efficient and encourages local feature aggregation. (2) *Cross-plane Attention Block*. To further encourage 3D inductive bias, we roll out $\mathbf{z}$ as a long sequence $l \times 3 \times c \rightarrow 3l \times c$ to conduct attention across planes, so that all tokens in the latent space could attend to each other. In this way, we encourage global information flow for more coherent 3D reconstruction and generation. Compared to Rodin, our design is fully attention-based and naturally supports parallel computing without the expensive axis pooling aggregation.

Empirically, we observe that using DiT [65] block and injecting the latent $\mathbf{z}$ as conditions yields better performance compared to the ViT [16, 61] block, which takes the latent $\mathbf{z}_0$ as the regular input. Specifically, the adaptive layer norm (adaLN) layer [65] fuses the input latent $\mathbf{z}$ with the learnable positional encoding

for attention operations. Moreover, we interleave the two types of attention layers to make sure the overall parameters count consistent with the pre-defined DiT length, ensuring efficient training and inference. As all operations are defined in the token space, the decoder achieves efficient computation against Rodin [97] while promoting 3D priors.

Decoder Upsampler. After all the attention operations, we obtain the tokens from the last transformer layer $\widetilde{\mathbf{z}}$ as the output. The context-rich tokens are reshaped back into spatial domain [28] and up-sampled by a convolutional decoder to the final tri-plane representation with shape $\hat{H} \times \hat{W} \times 3C$. Here, we adopt a lighter version of the convolutional decoder for efficient upsampling, where the three spatial latent of $\widetilde{\mathbf{z}}$ are processed in parallel.

Learning a Perceptually Rich and Intact 3D Latent Space. Adversarial learning [23] has been widely applied in learning a compact and perceptually rich latent space [20, 73]. In the 3D domain, the adversarial loss can also encourage correct 3D geometry when novel-view reconstruction supervisions are inapplicable [5, 42, 76], *e.g.*, the monocular dataset such as FFHQ. Inspired by previous research [42, 76], we leverage adversarial loss to bypass this issue. Specifically, we impose an input-view discriminator to maintain perceptually-reasonable input view reconstruction, and an auxiliary novel-view discriminator to distinguish the rendered images between the input and novel views. We observe that if asking the novel-view discriminator to differentiate novel-view renderings and real images instead, the reconstruction model will suffer from *posterior collapse* [52], which outputs input-irrelevant but high-fidelity results to fool the novel-view discriminator. This phenomenon has also been observed by Kato *et al.* [42].

Training. After the decoder $\mathcal{D}_{\psi}$ decodes a high-resolution neural fields $\hat{\mathbf{z}_0}$ from the latent, we have $\hat{I} = \mathtt{R}(\widetilde{\mathcal{X}}) = \mathtt{R}(\mathcal{D}_{\psi}(\mathbf{z})) = \mathtt{R}(\mathcal{D}_{\psi}(\mathcal{E}_{\phi}(\mathcal{I})))$, where $\mathtt{R}$ stands for differentiable rendering [90] and we take $\mathcal{D}_{\psi}(\mathbf{z}) = \mathtt{R}(\mathcal{D}_{\psi}(\mathbf{z}))$ for brevity. Here, we choose $\widetilde{\mathcal{X}}$ as tri-plane [6, 66] and $\mathtt{R}$ as volume rendering [56] for experiments. Note that our compression model is 3D representations/rendering agnostic and new neural rendering techniques [43] can be easily integrated by alternating the decoder architecture [87]. The final training objective reads as

$$\mathcal{L}(\phi, \psi) = \mathcal{L}_{\text{render}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}} + \lambda_{\text{kl}}\mathcal{L}_{\text{KL}} + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}}, \tag{1}$$

where $\mathcal{L}_{\text{render}}$ is a mixture of the L_1 and perceptual loss [108], $\mathcal{L}_{\text{reg}}$ encourages smooth geometry [99], $\mathcal{L}_{\text{KL}}$ is the *KL-reg* loss to regularize a structured latent space [73], and $\mathcal{L}_{\text{GAN}}$ improves perceptual quality and enforces correct geometry for monocular datasets. Note that $\mathcal{L}_{\text{render}}$ is applied to both input-view and randomly sampled novel-view images.

For category-specific datasets such as ShapeNet [8], we only supervise *one* novel view, which already yields good enough performance. For category-free datasets with diverse shape variations, *e.g.*, Objaverse [14], we supervise *four* novel views. Our method is more data-efficient against the existing state-of-the-art 3D diffusion method [9, 57], which requires 50 views to converge. The implementation details are included in the supplementary material.

3.2 Latent Diffusion and Denoising

Latent Diffusion Models. LDM [73, 95] is designed to acquire a prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}_0)$ within the perceptual latent space, whose training data is the latent obtained online from the trained $\mathcal{E}_{\boldsymbol{\phi}}$. Here, we use the score-based latent diffusion model [95], which is the continuous derivation of DDPM variational objective [32]. Specifically, the denoiser $\boldsymbol{\epsilon}_\theta$ parameterizes the score function score [84] as $\nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t) := -\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)/\sigma_t$, with continuous time sequence t. By training to predict a denoised variant of the noisy input $\mathbf{z}_t$, $\boldsymbol{\epsilon}_\theta$ gradually learns to denoise from a standard Normal prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ by solving a reverse SDE [32].

Following LSGM [95], we formulate the learned prior at time t geometric mixing $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t) := \sigma_t(1 - \boldsymbol{\alpha}) \odot \mathbf{z}_t + \boldsymbol{\alpha} \odot \boldsymbol{\epsilon}'_\theta(\mathbf{z}_t, t)$, where $\boldsymbol{\epsilon}'_\theta(\mathbf{z}_t, t)$ is the denoiser output and $\alpha \in [0, 1]$ is a learnable scalar coefficient. Intuitively, this formulation can bring the denoiser input closer to a standard Normal distribution, which the reverse SDE can be solved faster. Similarly, Stable Diffusion [67, 73] also scales the input latent by a factor to maintain a unit variance, which is pre-calculated on the billion-level dataset [77]. The training objective reads as

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathcal{E}_{\boldsymbol{\phi}}(I), \boldsymbol{\epsilon} \sim \mathcal{N}(0,1), t} \left[\frac{w_t}{2} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, c)\|_2^2 \right], \tag{2}$$

where $t \sim \mathcal{U}[0, 1]$ and w_t is an empirical time-dependent weighting function, c is the corresponding condition.

The denoiser $\boldsymbol{\epsilon}_\theta$ is realized by a time-dependent U-Net [74] or DiT [65]. During training, we obtain $\mathbf{z}_0$ online from the fixed $\mathcal{E}_{\boldsymbol{\phi}}$, roll-out the tri-latent $h \times w \times 3 \times c \rightarrow h \times (3w) \times c$, and add time-dependent noise to get $\mathbf{z}_t$. Here, we choose the importance sampling schedule [95] with *velocity* [53] parameterization, which yields more stable behavior against ϵ parameterization for diffusion learning. After training, the denoised samples can be decoded to the 3D neural field (*i.e.*, tri-plane here) with a single forward pass through $\mathcal{D}_{\boldsymbol{\psi}}$, on which neural rendering can be applied.

3.3 Conditioning Mechanisms

Compared with the existing approach that focuses on category-specific unconditional 3D diffusion model [1], we propose to inject CLIP embeddings [71] into the latent 3D diffusion model to support image/text-conditioned 3D generation. Given input condition $\mathbf{y}$, the diffusion model formulates the conditional distribution $p(\mathbf{z}|\mathbf{y})$ with $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{y})$. The inputs $\mathbf{y}$ can be text captions for datasets like Objaverse, or images for general datasets like ShapeNet and FFHQ.

Text Conditioning. For datasets with text caption, we follow Stable Diffusion [73] to directly leverage the CLIP text encoder CLIP_T to encode the text caption as the conditions. All the output tokens 77×768 are used and injected to diffusion denoiser with cross attention blocks.

Image Conditioning. For datasets with images only, we encode the input I corresponding to the latent code $\mathbf{z}_0$ using CLIP image encoder CLIP_I and adopt the output embedding as the condition. To support both image and text

conditions, we re-scale the image latent code with a pre-calculated factor to match the scale of the text latent.

Classifier-free Guidance. We adopt *classifier-free guidance* [31] for latent conditioning to support conditional and unconditional generation. During diffusion model training, we randomly zero out the corresponding conditioning latent embeddings with 15% probability to jointly train the unconditional and conditional settings. During sampling, we perform a linear combination of the conditional and unconditional score estimates:

$$\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t, \boldsymbol{\tau}_\theta(y)) = s\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \boldsymbol{\tau}_\theta(y)) + (1-s)\boldsymbol{\epsilon}_\theta(\mathbf{z}_t), \tag{3}$$

where s is the guidance scale to control the mixing strength to balance sampling diversity and quality.

4 Experiments

Datasets. Following most previous work, we use ShapeNet [8] to benchmark the 3D generation performance. We use the Car, Chair, and Plane categories with 3514, 6700, and 4045 instances correspondingly. Each instance is randomly rendered from 50 views following a spherical uniform distribution. Moreover, to evaluate the performance over diverse high-quality 3D datasets, we also include the experiments over Objaverse [14], which is the largest 3D dataset with challenging categories and complicated geometry. We use the renderings provided by G-Objaverse [70] and choose a high-quality subset with around $176K$ 3D instances, where each consists of 40 random views.

Training Details. For ShapeNet and FFHQ training, we adopt a monocular input setting with $V = 1$ and target rendering size 128×128. For Objaverse, we adopt multi-view inputs with $V = 6$ and target rendering size 192×192. All training images are resized to $H = W = 256$ as input. The encoder $\mathcal{E}_\phi$ has down-sample factor $f = 8$ and the decoder upsampler $\mathcal{D}_U$ outputs tri-plane with size $\hat{H} = \hat{W} = 128$ and $C = 32$. To trade off rendering resolution and training batch size, we impose supervision over 80×80 randomly cropped patches. For adversarial loss, we use DINO [61] in vision-aided GAN [46] with non-saturating GAN loss [26] for discriminator training. For conditional diffusion training, we use the CLIP image embedding for ShapeNet and FFHQ, and CLIP text embedding from the official text caption for Objaverse. Both the autoencoding model and diffusion model are trained for $800K$ iterations, which take around 7 days with 8 A100 GPUs in total.

Metrics. Following prior work [9, 22, 57, 81], we adopt both 2D and 3D metrics to benchmark the generation performance: Fréchet Inception Distance (FID@50K) [30] and Kernel Inception Distance (KID@50K) [2] to evaluate 2D renderings, as well as Coverage Score (COV) and Minimum Matching Distance (MMD) to benchmark 3D geometry. We calculate all metrics under 128×128 for fair comparisons across all baselines.

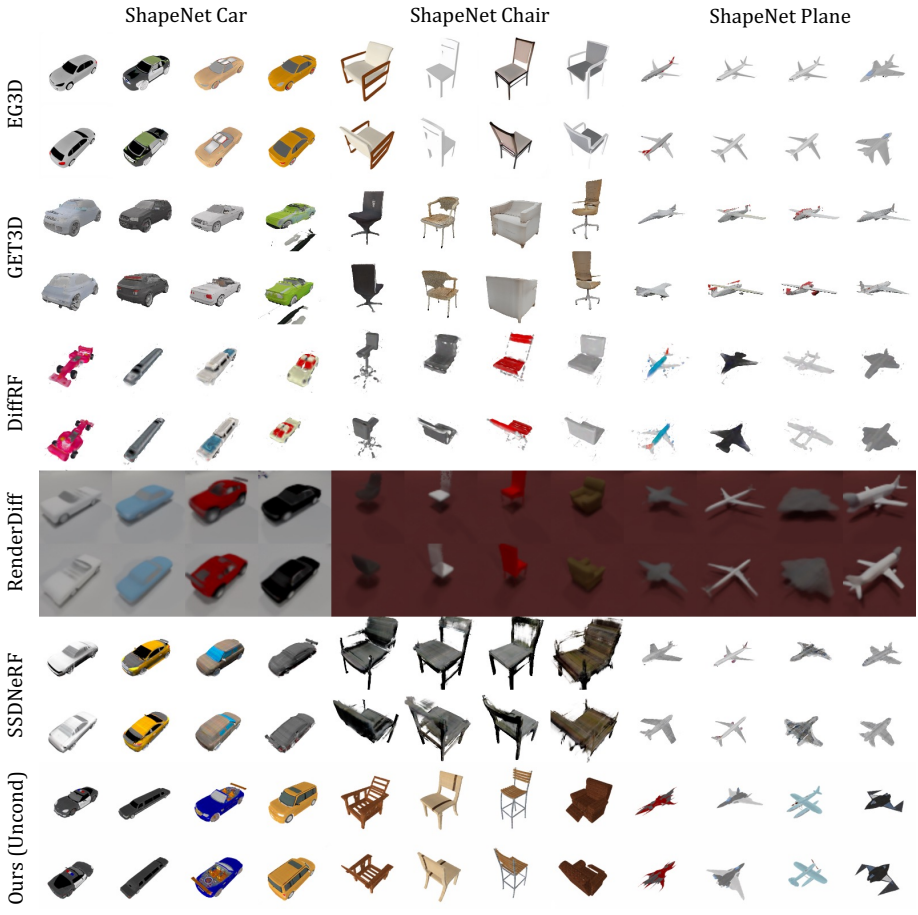

Fig. 3: ShapeNet Unconditional Generation. We show four samples for each method. Zoom in for the best view.

4.1 Evaluation

In this section, we compare our method with both state-of-the-art GAN-based methods: EG3D [6], GET3D [22] as well as recent diffusion-based methods: DiffRF [57], RenderDiffusion [1] and SSDNeRF [9]. Since LN3Diff only leverages $v = 2$ for ShapeNet experiments, for SSDNeRF, we include both the official 50-views trained $\text{SSDNeRF}_{\text{V=50}}$ version as well as the reproduced $\text{SSDNeRF}_{\text{V=3}}$ for fair comparison. We find SSDNeRF fails to converge with $V = 2$. We set the guidance scale in Eq. (3) to $s = 0$ for unconditional generation, and $s = 6.5$ for all conditional generation sampling.

Unconditional Generation on ShapeNet. To evaluate our methods against existing 3D generation methods, we include the quantitative and qualitative results for unconditional single-category 3D generation on ShapeNet in Tab. 1

Table 1: Quantitative Comparison of Unconditional Generation on ShapeNet. The proposed LN3Diff shows satisfactory performance on single-category generation.

Category	Method	FID@50K↓	KID@50K(%)↓	COV(%)↑	MMD(‰)↓
Car	EG3D [6]	33.33	1.4	35.32	3.95
	GET3D [22]	41.41	1.8	37.78	3.87
	DiffRF [57]	75.09	5.1	29.01	4.52
	RenderDiffusion [1]	46.5	4.1	-	-
	$\text{SSDNeRF}_{\text{V=3}}$ [9]	47.72	2.8	37.84	3.46
	$\text{SSDNeRF}^{*}_{\text{V=50}}$ [9]	45.37	2.1	**67.82**	2.50
	LN3Diff**(Ours)**	**17.6**	**0.49**	43.12	**2.32**
Plane	EG3D [6]	14.47	0.54	18.12	4.50
	GET3D [22]	26.80	1.7	21.30	4.06
	DiffRF [57]	101.79	6.5	37.57	3.99
	RenderDiffusion [1]	43.5	5.9	-	-
	$\text{SSDNeRF}_{\text{V=3}}$ [9]	21.01	1.0	42.50	2.94
	LN3Diff**(Ours)**	**8.84**	**0.36**	**43.40**	**2.71**
Chair	EG3D [6]	26.09	1.1	19.17	10.31
	GET3D [22]	35.33	1.5	28.07	9.10
	DiffRF [57]	99.37	4.9	17.05	14.97
	RenderDiffusion [1]	53.3	6.4	-	-
	$\text{SSDNeRF}_{\text{V=3}}$ [9]	65.04	3.0	**47.54**	6.71
	LN3Diff**(Ours)**	**16.9**	**0.47**	47.1	**5.28**

and Fig. 3. We evaluate all baseline 3D diffusion methods with 250 DDIM steps and GAN-based baselines with $psi = 0.8$ to guarantee each sample is intact for COV/MMD evaluation. For FID/KID evaluation, we re-train the baselines and calculate the metrics using a fixed upper-sphere ellipsoid camera trajectory [83] across all datasets. For COV/MMD evaluation, we randomly sample 4096 points around the extracted sampled mesh and ground truth mesh surface and adopt Chamfer Distance for evaluation.

As shown in Tab. 1, LN3Diff achieves quantitatively better performance against all GAN-based baselines regarding rendering quality and 3D coverage. Fig. 3 further demonstrates that GAN-based methods suffer greatly from mode collapse: in the ShapeNet Plane category, both EG3D and GET3D are limited to the white civil airplanes, which is fairly common in the dataset. Our methods can sample more diverse results with high-fidelity texture.

Compared against diffusion-based baselines, LN3Diff also shows better visual quality with better quantitative metrics. $\text{SSDNeRF}_{\text{V=50}}$ shows a better coverage score, which benefits from leveraging more views during training. However, our method with $V = 2$ shows comparative performance against $\text{SSDNeRF}_{\text{V=3}}$ on the ShapeNet Chair and even better performance on the remaining datasets.

Conditional 3D Generation. Conditional 3D generation has the potential to streamline the 3D modeling process in both the gaming and film industries. As visualized in Fig. 4, we present our conditional generation performance on the

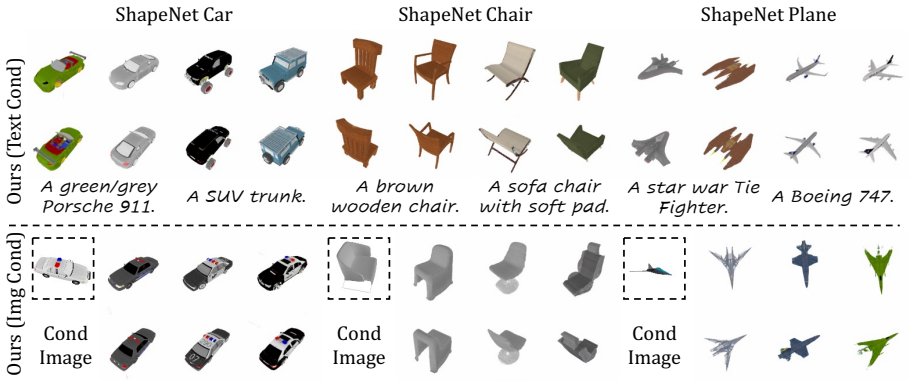

Fig. 4: ShapeNet Conditional Generation. We show conditional generation with both texts and image as inputs. Zoom in for the best view.

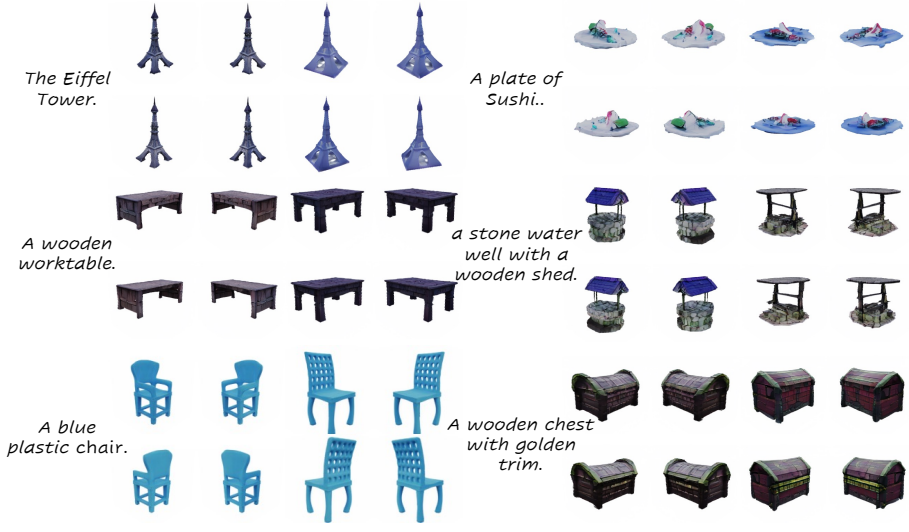

Fig. 5: Objaverse Conditional Generation Given Text Prompt. We show two samples for each prompt. Zoom in for the best view.

ShapeNet dataset, where either text or image serves as the input prompt. Visually inspected, our method demonstrates promising performance in conditional generation, closely aligning the generated outputs with the input conditions. For the image-conditioned generation, our method yields semantically similar samples while maintaining diversity.

We also demonstrate the text-conditioned generation of Objaverse in Fig. 5 and Tab. 2. As shown, the diffusion model trained over LN3Diff's latent space enables high-quality 3D generation over generic 3D datasets. This ability is unprecedented among existing 3D diffusion baselines and and marks a significant

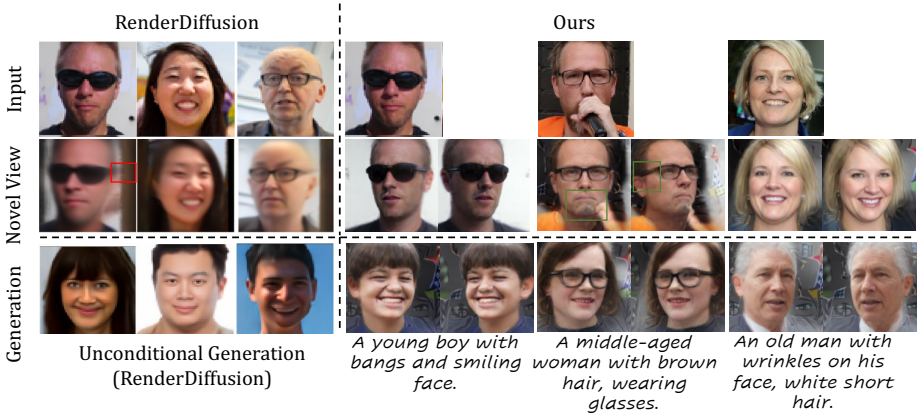

Fig. 6: FFHQ Monocular Reconstruction (upper half) and 3D Generation (lower half). For monocular reconstruction, we test our method with hold-out test set and visualize the input-view and novel-view. Compared to baseline, our method shows consistently better performance on both reconstruction and generation.

step toward highly controllable 3D generation. Qualitative comparisons against Shape-E and Point-E are provided in the supplementary materials.

We compare our method against RenderDiffusion, the only 3D diffusion method that supports 3D generation over FFHQ. As shown in the lower part in Fig. 6, beyond view-consistent 3D generation, our method further supports conditional 3D generation at 128×128 resolution, while RenderDiffusion is limited to 64×64 resolution due to the expensive volume rendering integrated into diffusion training. Quantitatively, our method achieves an FID score of 36.6, compared to 59.3 by RenderDiffusion.

Monocular 3D Reconstruction. Beyond the samples shown in Fig. 1, we also include monocular reconstruction results over FFHQ datasets in the upper half of Fig. 6 and compare against RenderDiffusion. As can be observed, our method demonstrates high fidelity and preserves semantic details even in self-occluded images. The novel view generated by RenderDiffusion appears blurry and misses semantic components that are not visible in the input view, such as the leg of the eyeglass.

Table 2: Quantitative Metrics on Text-to-3D. The proposed method outperforms Point-E and Shape-E on CLIP scores over two different backbones.

Method	ViT-B/32	ViT-L/14
Point-E [59]	26.35	21.40
Shape-E [38]	27.84	25.84
Ours	**29.12**	**27.80**

4.2 Ablation Study and Analysis

Reconstruction Arch Design. In Tab. 3, we benchmark each component of our auto-encoding architecture over a subset of Objaverse with $7K$ instances

Table 3: Ablation of Reconstruction Arch Design. We ablate the design of our auto-encoding architecture. Each component contributes to a consistent gain in the reconstruction performance, indicating an improvement in the modeling capacity.

Design	PSNR@100K
2D Conv Baseline	17.46
+ ViT Block	18.92
ViT Block → DiT Block	20.61
+ Plucker Embedding	21.29
+ Cross-Plane Attention	21.70
+ Self-Plane Attention	**21.95**

Table 4: Diffusion Sampling Speed and Latent Size. We provide the sampling time per instance evaluated on 1 V100, along with the latent size. Our method achieves faster sampling speed while maintaining superior generation performance.

Method	V100-sec	Latent Size
Get3D/EG3D	<0.5	256
SSDNeRF	8.1	$128^2 \times 18$
RenderDiffusion	15.8	-
DiffRF	18.7	$32^3 \times 4$
LN3DIFF$_{\text{uncond}}$	**5.7**	$32^2 \times 12$
LN3DIFF$_{\text{cfg}}$	**7.5**	$32^2 \times 12$

and record the PSNR at $100K$ iterations. Each component introduces consistent improvements with negligible parameter increases.

Novel View Discriminator for Monocular Dataset. Novel view discriminator is crucial for monocular datasets like FFHQ. As shown in Fig. 7, without it, the VAE model fails to yield a plausible novel view.

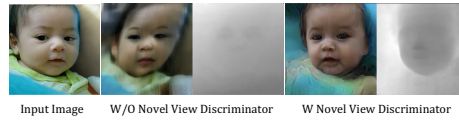

Fig. 7: Ablation of novel view discriminator.

Diffusion Sampling Speed and Latent Size. We report the sampling speed and latent space size comparison in Tab. 4. By performing on the compact latent space, our method achieves the fastest sampling while keeping the best generation performance. Though RenderDiffusion follows a latent-free design, its intermediate 3D neural field has a shape of $256^2 \times 96$ and hinders efficient diffusion training.

5 Conclusion and Discussions

In this work, we present a new paradigm of 3D generative model by learning the diffusion model over a compact 3D-aware latent space. A dedicated variational autoencoder encodes (multi-view) image(s) into a low-dim structured latent space, where conditional diffusion learning can be efficiently performed. We achieve state-of-the-art performance over ShapeNet and demonstrate our method over generic category-free Objaverse 3D datasets. Our work potentially facilitates numerous downstream applications in 3D vision and graphics tasks.

Limitations and Future Work. Our method comes with some limitations unresolved. VAE side, we observe that volume rendering is memory-consuming. Extending our decoder to more efficient 3D representations such as 3DGS [43] shall alleviate this issue. Besides, adding more real-world data such as MVImageNet [104] and more control conditions [107] is also worth exploring. Overall, our method is a step towards a general 3D diffusion model and can inspire future research in this direction.

Potential Negative Societal Impacts. The entity relational composition capabilities of LN3Diff could be applied maliciously to real human figures. Additional potential impacts are discussed in the Supplementary File in depth.

Acknowledgement. This study is supported under the RIE2020 Industry Alignment Fund Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contributions from the industry partner(s). It is also supported by Singapore MOE AcRF Tier 2 (MOE-T2EP20221-0011).

References

1. Anciukevičius, T., Xu, Z., Fisher, M., Henderson, P., Bilen, H., Mitra, N.J., Guerrero, P.: RenderDiffusion: Image diffusion for 3D reconstruction, inpainting and generation. In: CVPR (2023) 3, 4, 5, 8, 10, 11
2. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying MMD GANs. In: ICLR (2018) 9
3. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: ICLR (2019) 3
4. Cai, S., Obukhov, A., Dai, D., Van Gool, L.: Pix2NeRF: Unsupervised Conditional p-GAN for Single Image to Neural Radiance Fields Translation. In: CVPR (2022) 6, 24, 27
5. Chan, E., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: Pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In: CVPR (2021) 3, 7
6. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3D generative adversarial networks. In: CVPR (2022) 2, 4, 6, 7, 10, 11
7. Chan, E.R., Nagano, K., Chan, M.A., Bergman, A.W., Park, J.J., Levy, A., Aittala, M., Mello, S.D., Karras, T., Wetzstein, G.: GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In: arXiv (2023) 4
8. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. arXiv preprint arXiv:1512.03012 (2015) 3, 7, 9, 27
9. Chen, H., Gu, J., Chen, A., Tian, W., Tu, Z., Liu, L., Su, H.: Single-stage diffusion nerf: A unified approach to 3D generation and reconstruction. In: ICCV (2023) 2, 4, 7, 9, 10, 11
10. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., Li, Z.: Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis (2023) 22
11. Chen, Y., Wang, T., Wu, T., Pan, X., Jia, K., Liu, Z.: Comboverse: Compositional 3d assets creation using spatially-aware diffusion guidance. arXiv preprint arXiv:2403.12409 (2024) 4
12. Contributors, S.: SpConv: Spatially sparse convolution library. https://github.com/traveller59/spconv (2022) 2
13. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., VanderBilt, E., Kembhavi, A., Vondrick, C., Gkioxari, G., Ehsani, K., Schmidt, L., Farhadi, A.: Objaverse-xl: A universe of 10m+ 3d objects. arXiv preprint arXiv:2307.05663 (2023) 5
14. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3D objects. arXiv preprint arXiv:2212.08051 (2022) 3, 6, 7, 9
15. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. NeurIPS (2021) 2, 3, 4, 22, 24
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 3, 6

17. Dupont, E., Kim, H., Eslami, S.M.A., Rezende, D.J., Rosenbaum, D.: From data to functa: Your data point is a function and you can treat it like one. In: ICML (2022) 4
18. Dupont, E., Martin, M.B., Colburn, A., Sankar, A., Susskind, J., Shan, Q.: Equivariant neural rendering. In: International Conference on Machine Learning. pp. 2761–2770. PMLR (2020) 27
19. Eslami, S.M.A., Rezende, D.J., Besse, F., Viola, F., Morcos, A.S., Garnelo, M., Ruderman, A., Rusu, A.A., Danihelka, I., Gregor, K., Reichert, D.P., Buesing, L., Weber, T., Vinyals, O., Rosenbaum, D., Rabinowitz, N.C., King, H., Hillier, C., Botvinick, M.M., Wierstra, D., Kavukcuoglu, K., Hassabis, D.: Neural scene representation and rendering. Science **360**, 1204 – 1210 (2018) 27
20. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR (2021) 3, 6, 7
21. Fridovich-Keil and Yu, Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022) 24
22. Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., Fidler, S.: Get3D: A generative model of high quality 3D textured shapes learned from images. In: NeurIPS (2022) 9, 10, 11, 23
23. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014) 2, 3, 7
24. Gu, J., Gao, Q., Zhai, S., Chen, B., Liu, L., Susskind, J.: Learning controllable 3D diffusion models from single-view images. arXiv preprint arXiv:2304.06700 (2023) 2, 6
25. Gu, J., Liu, L., Wang, P., Theobalt, C.: StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis. In: ICLR (2021) 4
26. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein GANs. In: NeurIPS (2017) 9
27. Gupta, A., Xiong, W., Nie, Y., Jones, I., Oğuz, B.: 3dgen: Triplane latent diffusion for textured mesh generation (2023), https://arxiv.org/abs/2303.05371 4
28. He, K., Chen, X., Xie, S., Li, Y., Doll'ar, P., Girshick, R.B.: Masked autoencoders are scalable vision learners. In: CVPR (2022) 7
29. Henzler, P., Mitra, N.J., Ritschel, T.: Escaping plato's cave: 3D shape from adversarial rendering. In: ICCV (2019) 3
30. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017) 9
31. Ho, J.: Classifier-free diffusion guidance. In: NeurIPS (2021) 9
32. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020) 2, 3, 4, 8
33. Hong, F., Chen, Z., Lan, Y., Pan, L., Liu, Z.: EVA3D: Compositional 3D human generation from 2d image collections. In: ICLR (2022) 4
34. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. In: ICLR (2024) 4
35. Hoogeboom, E., Heek, J., Salimans, T.: simple diffusion: End-to-end diffusion for high resolution images. In: ICML (2023) 3, 4
36. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields. In: CVPR (2022) 4, 24
37. Jang, W., Agapito, L.: Codenerf: Disentangled neural radiance fields for object categories. In: ICCV. pp. 12949–12958 (2021) 27

38. Jun, H., Nichol, A.: Shap-E: Generating conditional 3D implicit functions. arXiv preprint arXiv:2305.02463 (2023) 2, 4, 5, 13, 24
39. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: ICLR (2018) 3, 4
40. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019) 3
41. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: CVPR (2020) 3
42. Kato, H., Harada, T.: Learning view priors for single-view 3D reconstruction. In: CVPR (2019) 7
43. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4), 1–14 (2023) 7, 14
44. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv (2013) 3, 6
45. Kosiorek, A.R., Strathmann, H., Zoran, D., Moreno, P., Schneider, R., Mokr'a, S., Rezende, D.J.: NeRF-VAE: A geometry aware 3D scene generative model. ICML (2021) 4
46. Kumari, N., Zhang, R., Shechtman, E., Zhu, J.Y.: Ensembling off-the-shelf models for gan training. In: CVPR (2022) 9
47. Lan, Y., Loy, C.C., Dai, B.: DDF: Correspondence distillation from nerf-based gan. IJCV (2022) 2, 4, 28
48. Lan, Y., Meng, X., Yang, S., Loy, C.C., Dai, B.: E3dge: Self-supervised geometry-aware encoder for style-based 3D gan inversion. In: CVPR (2023) 3, 4, 6
49. Lan, Y., Tan, F., Qiu, D., Xu, Q., Genova, K., Huang, Z., Fanello, S., Pandey, R., Funkhouser, T., Loy, C.C., Zhang, Y.: Gaussian3Diff: 3D gaussian diffusion for 3D full head synthesis and editing. arXiv (2023) 2, 4
50. Liu, R., Wu, R., Hoorick, B.V., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3D object (2023) 2, 4
51. Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., et al.: Wonder3d: Single image to 3d using cross-domain diffusion. In: CVPR (2024) 4
52. Lucas, J., Tucker, G., Grosse, R.B., Norouzi, M.: Understanding posterior collapse in generative latent variable models. In: ICLR (2019) 7
53. Meng, C., Gao, R., Kingma, D.P., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: CVPR. pp. 14297–14306 (2022) 8
54. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy Networks: Learning 3D reconstruction in function space. In: CVPR (2019) 3
55. Mi, L., Kundu, A., Ross, D., Dellaert, F., Snavely, N., Fathi, A.: im2nerf: Image to neural radiance field in the wild. In: arXiv (2022) 6
56. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 2, 3, 7
57. Müller, N., Siddiqui, Y., Porzi, L., Bulo, S.R., Kontschieder, P., Nießner, M.: DiffRF: Rendering-guided 3D radiance field diffusion. In: CVPR (2023) 2, 4, 5, 7, 9, 10, 11
58. Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.: HoloGAN: Unsupervised Learning of 3D Representations From Natural Images. In: ICCV (2019) 3
59. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts (2022) 13, 24

60. Niemeyer, M., Geiger, A.: GIRAFFE: Representing scenes as compositional generative neural feature fields. In: CVPR (2021) 4
61. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision (2023) 6, 9
62. Or-El, R., Luo, X., Shan, M., Shechtman, E., Park, J.J., Kemelmacher-Shlizerman, I.: StyleSDF: High-resolution 3D-consistent image and geometry generation. In: CVPR (2021) 4
63. Pan, X., Dai, B., Liu, Z., Loy, C.C., Luo, P.: Do 2D GANs know 3D shape? Unsupervised 3D Shape Reconstruction from 2D Image GANs. In: ICLR (2021) 3
64. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: CVPR. pp. 165–174 (2019) 3
65. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023) 3, 5, 6, 8, 22, 23
66. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: ECCV (2020) 6, 7
67. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: Improving latent diffusion models for high-resolution image synthesis. In: arXiv (2023) 8
68. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: DreamFusion: Text-to-3D using 2D diffusion. ICLR (2022) 2, 4
69. Qi, C., Su, H., Mo, K., Guibas, L.: PointNet: Deep learning on point sets for 3D classification and segmentation. arXiv (2016) 2
70. Qiu, L., Chen, G., Gu, X., zuo, Q., Xu, M., Wu, Y., Yuan, W., Dong, Z., Bo, L., Han, X.: Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. arXiv preprint arXiv:2311.16918 (2023) 9, 23
71. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021) 8
72. Rebain, D., Matthews, M., Yi, K.M., Lagun, D., Tagliasacchi, A.: LOLNeRF: Learn from one look. In: CVPR (2022) 4
73. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 2, 5, 6, 7, 8, 22, 23, 28
74. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015) 8, 22
75. Sajjadi, M.S.M., Meyer, H., Pot, E., Bergmann, U., Greff, K., Radwan, N., Vora, S., Lucic, M., Duckworth, D., Dosovitskiy, A., Uszkoreit, J., Funkhouser, T., Tagliasacchi, A.: Scene Representation Transformer: Geometry-free novel view synthesis through set-latent scene representations. CVPR (2022) 4
76. Sargent, K., Koh, J.Y., Zhang, H., Chang, H., Herrmann, C., Srinivasan, P.P., Wu, J., Sun, D.: VQ3D: Learning a 3D-aware generative model on imagenet. ICCV (2023) 4, 6, 7
77. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: LAION-5B: An open large-scale dataset for training next generation image-text models. In: arXiv (2022) 2, 8

78. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: GRAF: Generative radiance fields for 3D-aware image synthesis. In: NeurIPS (2020) 3
79. Shi, R., Chen, H., Zhang, Z., Liu, M., Xu, C., Wei, X., Chen, L., Zeng, C., Su, H.: Zero123++: a single image to consistent multi-view diffusion base model. In: arXiv (2023) 2
80. Shi, Y., Wang, P., Ye, J., Mai, L., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3D generation. arXiv:2308.16512 (2023) 23
81. Shue, J., Chan, E., Po, R., Ankner, Z., Wu, J., Wetzstein, G.: 3d neural field generation using triplane diffusion. In: CVPR (2022) 2, 4, 5, 9
82. Sitzmann, V., Rezchikov, S., Freeman, W.T., Tenenbaum, J.B., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. In: NeurIPS (2021) 6
83. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene Representation Networks: Continuous 3D-structure-aware neural scene representations. In: NeurIPS (2019) 3, 11, 27
84. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2021) 2, 3, 4, 8
85. Sun, J., Wang, X., Shi, Y., Wang, L., Wang, J., Liu, Y.: Ide-3d: Interactive disentangled editing for high-resolution 3D-aware portrait synthesis. ACM Transactions on Graphics (TOG) **41**(6), 1–10 (2022) 4
86. Sun, J., Wang, X., Zhang, Y., Li, X., Zhang, Q., Liu, Y., Wang, J.: FENeRF: Face editing in neural radiance fields. In: arXiv (2021) 4
87. Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Splatter image: Ultra-fast single-view 3D reconstruction. In: arXiv (2023) 7
88. Tang, Z., Gu, S., Wang, C., Zhang, T., Bao, J., Chen, D., Guo, B.: Volumediffusion: Flexible text-to-3d generation with efficient volumetric encoder (2023) 4
89. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Multi-view 3d models from single images with a convolutional network (2016) 27
90. Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Xu, Z., Simon, T., Nießner, M., Tretschk, E., Liu, L., Mildenhall, B., Srinivasan, P., Pandey, R., Orts-Escolano, S., Fanello, S., Guo, M.G., Wetzstein, G., y Zhu, J., Theobalt, C., Agrawala, M., Goldman, D.B., Zollhöfer, M.: Advances in neural rendering. Computer Graphics Forum **41** (2021) 2, 3, 7
91. Tewari, A., Yin, T., Cazenavette, G., Rezchikov, S., Tenenbaum, J.B., Durand, F., Freeman, W.T., Sitzmann, V.: Diffusion with forward models: Solving stochastic inverse problems without direct supervision. In: NeurIPS (2023) 4
92. Thanh-Tung, H., Tran, T.: Catastrophic forgetting and mode collapse in gans. IJCNN pp. 1–10 (2020) 4
93. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: KPConv: Flexible and deformable convolution for point clouds. In: ICCV (2019) 2
94. Trevithick, A., Yang, B.: GRF: Learning a general radiance field for 3D scene representation and rendering. In: ICCV (2021) 27
95. Vahdat, A., Kreis, K., Kautz, J.: Score-based generative modeling in latent space. In: NeurIPS (2021) 5, 6, 8, 22
96. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.A.: IBRNet: Learning Multi-View Image-Based Rendering. In: CVPR (2021) 4
97. Wang, T., Zhang, B., Zhang, T., Gu, S., Bao, J., Baltrusaitis, T., Shen, J., Chen, D., Wen, F., Chen, Q., et al.: RODIN: A generative model for sculpting 3D digital avatars using diffusion. In: CVPR (2023) 2, 4, 6, 7

98. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. In: NeurIPS (2023) 2, 4
99. Weng, C.Y., Srinivasan, P.P., Curless, B., Kemelmacher-Shlizerman, I.: Person-NeRF: Personalized reconstruction from photo collections. In: CVPR. pp. 524–533 (June 2023) 7
100. Wu, C.Y., Johnson, J., Malik, J., Feichtenhofer, C., Gkioxari, G.: Multiview compressive coding for 3D reconstruction. arXiv preprint arXiv:2301.08247 (2023) 6
101. Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., Sridhar, S.: Neural fields in visual computing and beyond. Computer Graphics Forum **41** (2021) 2
102. Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzstein, G., Xu, Z., Zhang, K.: DMV3D: Denoising multi-view diffusion using 3D large reconstruction model. In: ICLR (2024) 4, 23
103. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: PixelNeRF: Neural radiance fields from one or few images. In: CVPR (2021) 4, 24, 27
104. Yu, X., Xu, M., Zhang, Y., Liu, H., Ye, C., Wu, Y., Yan, Z., Liang, T., Chen, G., Cui, S., Han, X.: MVImgNet: A large-scale dataset of multi-view images. In: CVPR (2023) 5, 14
105. Zhang, B., Tang, J., Nießner, M., Wonka, P.: 3DShape2VecSet: A 3d shape representation for neural fields and generative diffusion models. ACM Trans. Graph. **42**(4) (jul 2023). https://doi.org/10.1145/3592442, https://doi.org/10.1145/3592442 2
106. Zhang, J., Lan, Y., Yang, S., Hong, F., Wang, Q., Yeo, C.K., Liu, Z., Loy, C.C.: Deformtoon3d: Deformable 3D toonification from neural radiance fields. In: ICCV (2023) 4
107. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023) 2, 3, 14, 28
108. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 7
109. Zhou, L., Du, Y., Wu, J.: 3D shape generation and completion through point-voxel diffusion. In: ICCV (2021) 2, 5

In this supplementary material, we provide additional details regarding the implementations and additional results. We also discuss the limitations of our model.

Broader Social Impact. In this paper, we introduce a new latent 3D diffusion model designed to produce high-quality textures and geometry using a single model. As a result, our approach has the potential to be applied to generating DeepFakes or deceptive 3D assets, facilitating the creation of falsified images or videos. This raises concerns as individuals could exploit such technology with malicious intent, aiming to spread misinformation or tarnish reputations.

A Implementation details

A.1 Training details

Diffusion. We mainly adopt the diffusion training pipeline implementation from ADM [15], continuous noise schedule from LSGM [95] with the spatial transformer attention implementation from LDM [73]. For ShapeNet and FFHQ dataset, we adopt U-Net [74] architecture and list the hyperparameters in Tab. 5. For Objaverse dataset, we adopt DiT-L [65] architecture with cross attention design, as proposed in PixArt [10]. The diffusion transformer is built with 24 layers with 16 heads and 1024 hidden dimension, which result in 458M parameters.

Table 5: Hyperparameters and architecture of diffusion model ϵ_θ.

Diffusion Model Details	
Learning Rate	$2e-5$
Batch Size	96
Optimizer	AdamW
Iterations	500K
U-Net base channels	320
U-Net channel multiplier	1, 1, 2, 2, 4, 4
U-Net res block	2
U-Net attention resolutions	4,2,1
U-Net Use Spatial Transformer	True
U-Net Learn Sigma	False
U-Net Spatial Context Dim	768
U-Net attention head channels	64
U-Net pred type	v
U-Net norm layer type	GroupNorm
Noise Schedules	Linear
CFG Dropout prob	15%
CLIP Latent Scaling Factor	18.4

VAE Architecture. For the convolutional encoder $\mathcal{E}_{\phi}$, we adopt a lighter version of LDM [73] encoder with channel 64 and 1 residual blocks for efficiency. When training on Objaverse with $V = 6$, we incorporate 3D-aware attention [80] in the middle layer of the convolutional encoder. For convolutional upsampler $\mathcal{D}_U$, we further half the channel to 32. All other hyper-parameters remain at their default settings. Regarding the transformer decoder $\mathcal{D}_T$, we employ the DiT-L/2 architecture, and overall saved VAE model takes around 1.5 GiB storage. The input dimension of z to the MLP in each DiT block is $h \times w \times c$ for self-plane attention, and $h \times w \times 3 \times c$ in cross-plane attention. When ablating the 3D-aware attention in Tab.3, we adopt channel-wise concatenated latent $h \times w \times (3c)$ for model input, as in SSDNeRF. Note that we trade off a smaller model with faster training speed due to the overall compute limit, and a heavier model would certainly empower better performance [65, 102]. We ignore the plucker camera condition for the ShapeNet and FFHQ dataset, over which we find raw RGB input already yields good enough performance.

A.2 Data and Baseline Comparison

Training data. For ShapeNet, following GET3D [22], we use the blender to render the multi-view images from 50 viewpoints for all ShapeNet datasets with foreground mask. Those camera points sample from the upper sphere of a ball with a 1.2 radius. For Objaverse, we use a high-quality subset from the pre-processed rendering from G-buffer Objaverse [70] for experiments. Since G-buffer Objaverse splits the subset into 10 general categories, we use all the 3D instances except from "Poor-quality": Human-Shape, Animals, Daily-Used, Furniture, Buildings&Outdoor, Transportations, Plants, Food and Electronics. The ground truth camera pose, rendered multi-view images and depth maps are used for stage-1 VAE training.

Evaluation. The 2D metrics are calculated between 50k generated images and all available real images. Furthermore, for comparison of the geometrical quality, we sample 4096 points from the surface of 5000 objects and apply the Coverage Score (COV) and Minimum Matching Distance (MMD) using Chamfer Distance (CD) as follows:

$$
\begin{aligned}
CD(X,Y) &= \sum_{x \in X} \underset{y \in Y}{min} ||x-y||_2^2 + \sum_{y \in Y} \underset{x \in X}{min} ||x-y||_2^2, \\
COV(S_g, S_r) &= \frac{|\{\arg\min_{Y \in S_r} CD(X,Y) | X \in S_g\}|}{|S_r|} \quad , \\
MMD(S_g, S_r) &= \frac{1}{|S_r|} \sum_{Y \in S_r} \underset{X \in S_g}{min} CD(X,Y)
\end{aligned}
\tag{4}
$$

where $X \in S_g$ and $Y \in S_r$ represent the generated shape and reference shape.

Note that we use 5k generated objects S_g and all training shapes S_r to calculate COV and MMD. For fairness, we normalize all point clouds by centering in the original and recalling the extent to [-1,1]. Coverage Score aims to evaluate

the diversity of the generated samples, and MMD is used for measuring the quality of the generated samples. 2D metrics are evaluated at a resolution of 128 $\times$ 128. Since the GT data contains intern structures, we only sample the points from the outer surface of the object for results of all methods and ground truth.

For FID/KID evaluation, since different methods have their unique evaluation settings, we standardize this process by re-rendering each baseline's samples using a fixed upper-sphere ellipsoid camera pose trajectory of size 20. With $2.5K$ sampled 3D instances for each method, we recalculate FID@50K/KID@50K, ensuring a fair comparison across all methods.

Details about Baselines. We reproduce EG3D, GET3D, and SSDNeRF on our ShapeNet rendering using their officially released codebases. In the case of RenderDiffusion, we use the code and pre-trained model shared by the author for ShapeNet experiments. Regarding FFHQ dataset, due to the unavailability of the corresponding inference configuration and checkpoint from the authors, we incorporate their unconditional generation and monocular reconstruction results as reported in their paper. For DiffRF, given the absence of the public code, we reproduce their method with Plenoxel [21] and ADM [15].

B More Results

B.1 More Qualitative 3D Generation Results

We include more uncurated samples generated by our method on ShapeNet in Fig. 8, and on FFHQ in Fig. 9. For Objaverse, we include its qualitative evaluation against state-of-the-art generic 3D generative models (Shape-E [38] and Point-E [59]in Fig. 10, along with the quantitative benchmark in Tab. 2 in the main paper. We use CLIP-precision score in DreamField [36] to evaluate the text-3D alignment. As can be seen, LN3Diff shows more geometry and appearance details with higher CLIP scores against Shape-E and Point-E.

B.2 More Monocular 3D Reconstruction Results

We further benchmark the generalization ability of our stage-1 monocular 3D reconstruction VAE. For ShapeNet, we include the quantitative evaluation in Tab. 6. Our method achieves a comparable performance with monocular 3D reconstruction baselines. Note that strictly saying, our stage-1 VAE shares a similar setting with Pix2NeRF [4], whose encoder also has a latent space for generative modeling. Other reconstruction-specific methods like PixelNeRF [103] do not have these requirements and can leverage some designs like pixel-aligned features and long-skip connections to further boost the reconstruction performance. We include their performance mainly for reference and leave training the stage-1 VAE model with performance comparable with those state-of-the-art 3D reconstruction models for future work.

Besides, we visualize LN3Diff's stage-1 monocular VAE reconstruction performance over our Objaverse split in Fig. 12. As can be seen, though only

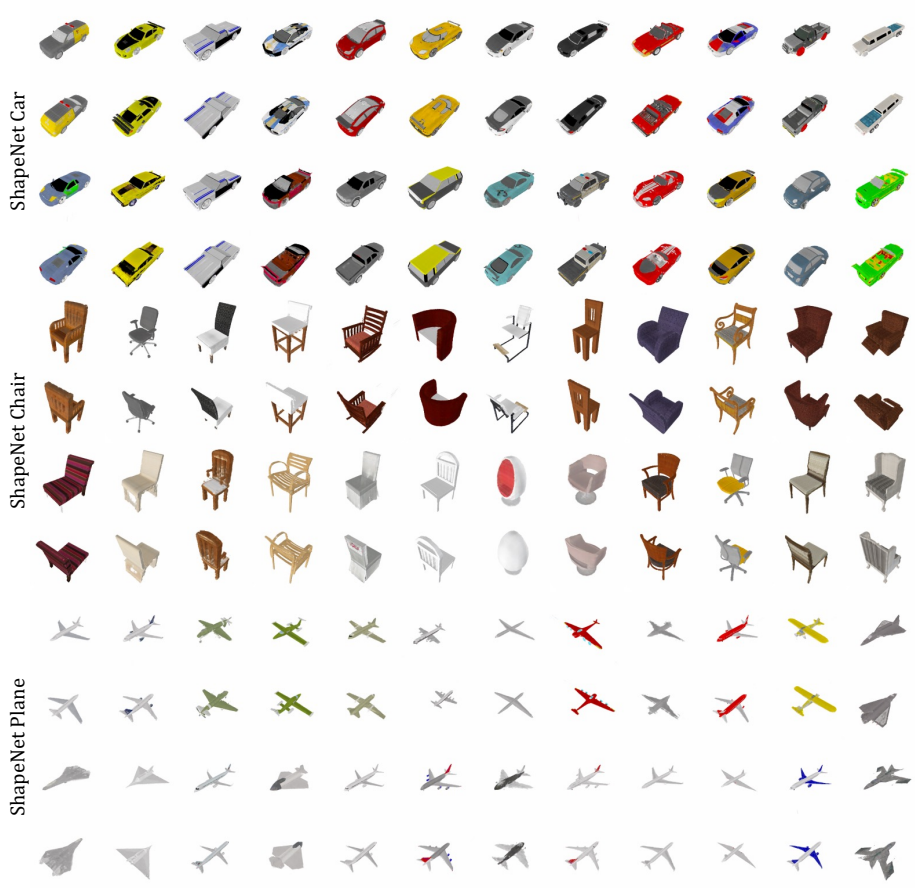

Fig. 8: Unconditional 3D Generation by LN3Diff (Uncurated). We showcase uncurated samples generated by LN3Diff on ShapeNet three categories. We visualize two views for each sample. Better zoom in.

one view is provided as the input, our monocular VAE reconstruction can yield high-quality and view-consistent 3D reconstruction with a detailed depth map. Quantitatively, the novel-view reconstruction performance over our whole Objaverse dataset achieves an average PSNR of 26.14. This demonstrates that our latent space can be treated as a compact proxy for efficient 3D diffusion training.

C Limitation and Failure Cases

We have included a brief discussion of limitations in the main submission. Here we include more details along with the visual failure cases for a more in-depth analysis of LN3Diff's limitations and future improvement directions.

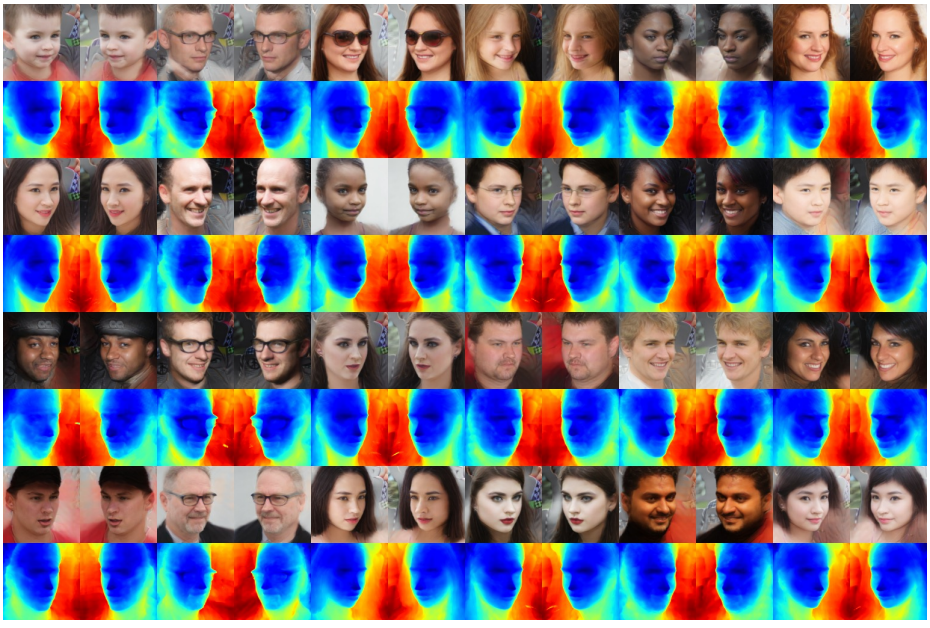

Fig. 9: Unconditional 3D Generation by LN3Diff (Uncurated). We showcase uncurated samples generated by LN3Diff on FFHQ. We visualize two views for each sample along with the extracted depth. Better zoom in.

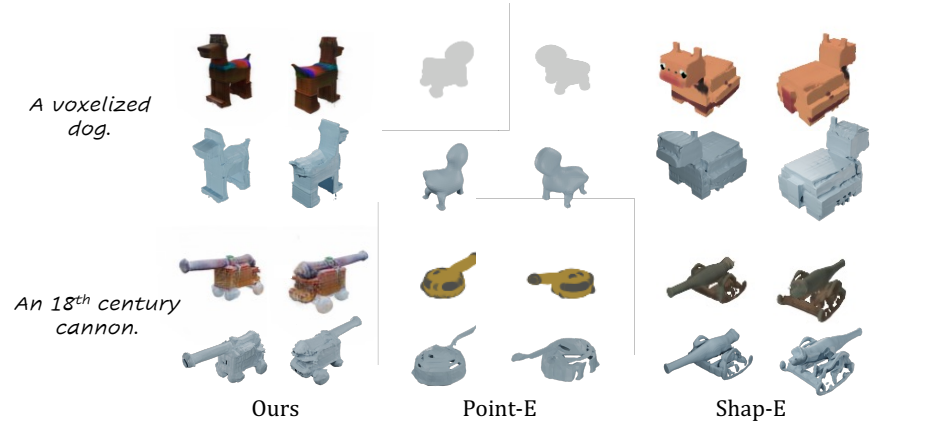

Fig. 10: Qualitative Comparison of Text-to-3D We showcase uncurated samples generated by LN3Diff on ShapeNet three categories. We visualize two views for each sample. Better zoom in.

Table 6: Quantitative results on ShapeNet-SRN [8, 83] chairs evaluate on 128×128. Legend: * – requires test time optimization. Note that our stage-1 VAE shares the same setting only with Pix2NeRF [103], which also has an explicit latent space for generative learning. Other baselines are included for reference.

Method	PSNR ↑	SSIM ↑
GRF [94]	21.25	0.86
TCO [89]	21.27	0.88
dGQN [19]	21.59	0.87
ENR [18]	22.83	-
SRN* [83]	22.89	0.89
CodeNeRF* [37]	22.39	0.87
PixelNeRF [103]	**23.72**	**0.91**
Pix2NeRF [4] conditional	18.14	0.84
Ours	20.91	0.89

Fig. 11: Limitation analysis. We showcase the deficiency to generate composed 3D scenes by LN3Diff. As shown here, the prompt **Two** chair yields similar results with **A** chair.

C.1 VAE Limitations

We have demonstrated that using a monocular image as encoder input can achieve high-quality 3D reconstruction. However, we noticed that for some challenging cases with diverse color and geometry details, the monocular encoder leads to blurry artifacts. As labeled in Fig. 12, our method with monocular input may yield floating artifacts over unseen viewpoints. We hypothesize that these artifacts are largely due to the ambiguity of monocular input and the use of regression loss (L2/LPIPS) during training. These observations demonstrate that switching to a multi-view encoder is necessary for better performance.

Besides, since our VAE requires plucker camera condition as input, the pre-trained VAE method cannot be directly applied to the unposed dataset. However, we believe this is not a research issue at the current time, considering the current methods still perform lower than expected on existing high-quality posed 3D datasets like Objaverse.

C.2 3D Diffusion Limitations

As one of the earliest 3D diffusion models that works on Objaverse, our method still suffers from several limitations that require investigation in the future. **(1)** The support of image-to-3D on Objaverse. Currently, we leverage $\text{CLIP}_{\text{text}}$

encoder with the 77 tokens as the conditional input. However, unlike 2D AIGC with T2I models [73], 3D content creation can be greatly simplified by providing easy-to-get 2D images. An intuitive implementation is by using our ShapeNet 3D diffusion setting, which provides the final normalized CLIP text embeddings as the diffusion condition. However, as shown in the lower half of Fig. 4 in the main submission, the CLIP encoder is better at extracting high-level semantics rather than low-level visual details. Therefore, incorporating more accurate image-conditioned 3D diffusion design like ControlNet [107] to enable monocular 3D reconstruction and control is worth exploring in the future. **(2)** Compositionality. Currently, our method is trained on object-centric dataset with simple captions, so the current model does not support composed 3D generation. For example, the prompt "Two yellow plastic chair with armchests" will still yield one chair, as visualized in Fig. 11. **(3)** UV map. To better integrate the learning-based method into the gaming and movie industry, a high-quality UV texture map is required. A potential solution is to disentangle the learned geometry and texture space and build the connection with UV space through dense correspondences [47].

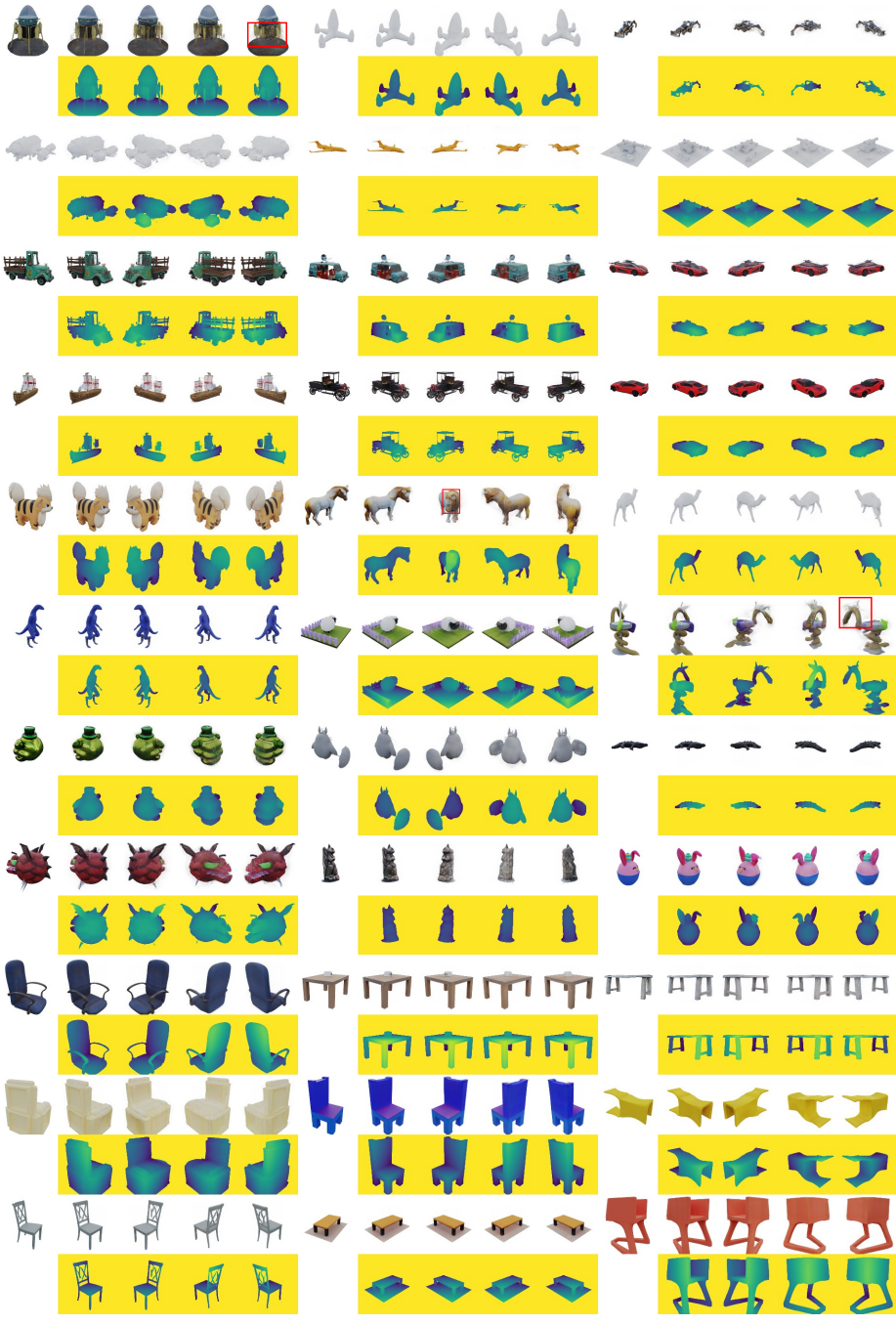

Fig. 12: Monocular 3D Reconstruction by LN3Diff stage-1 VAE on Objaverse (Uncurated). We showcase uncurated samples monocular-reconstructed by LN3Diff on Objaverse. From left to right, we visualize the input image, four reconstructed novel views with the corresponding depth maps. Artifacts are labeled in Red. Better zoom in.