Learning Controllable 3D Diffusion Models from Single-view Images

Jiatao Gu[†], Qingzhe Gao[§], Shuangfei Zhai[†], Baoquan Chen[¶], Lingjie Liu[‡], Josh Susskind[†]
[†]Apple [‡]University of Pennsylvania [§]Shandong University [¶]Peking University
[†]{jgu32,szhai,jsusskind}@apple.com [‡] lingjie.liu@seas.upenn.edu
[§] gaoqingzhe97@gmail.com [¶] baoquan@pku.edu.cn

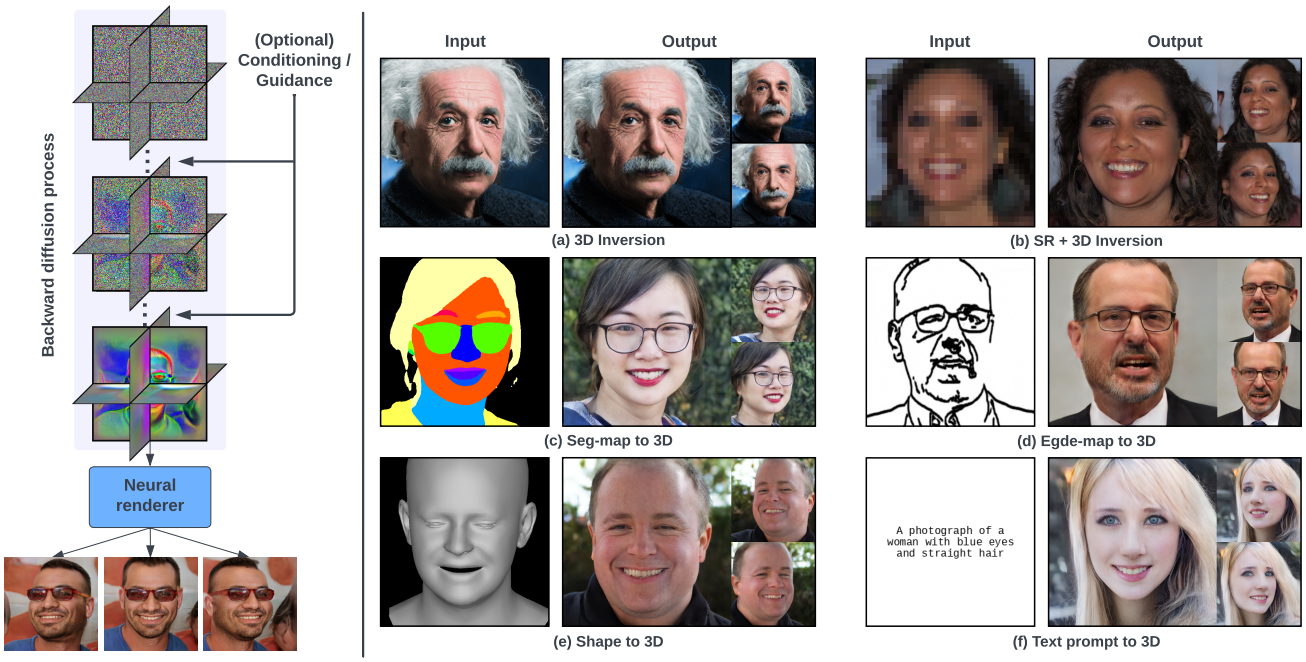


Figure 1: Left is the generation process, where a diffusion model samples a triplane which can be used for image rendering. Right are the examples of controllable generation given various conditioning inputs, showing generated frontal and side views from Control3Diff. The faces shown are all generated by models without real identities **due to concerns about individual consent** except for the input in (a).

Abstract

Diffusion models have recently become the de-facto approach for generative modeling in the 2D domain. However, extending diffusion models to 3D is challenging, due to the difficulties in acquiring 3D ground truth data for training. On the other hand, 3D GANs that integrate implicit 3D representations into GANs have shown remarkable 3D-aware generation when trained only on single-view image datasets. However, 3D GANs do not provide straightforward ways to precisely control image synthesis. To address these challenges, We present Control3Diff, a 3D diffusion model that combines the strengths of diffusion models and 3D GANs for versatile controllable 3D-aware image synthesis for single-view datasets. Control3Diff explicitly models the underlying latent distribution (optionally conditioned on external inputs), thus enabling direct control during the diffusion process. Moreover, our approach is general and applicable to any types of controlling inputs, allowing us to train it with the same diffusion objective without any auxiliary supervision. We validate the efficacy of Control3Diff on standard image generation benchmarks including FFHQ, AFHQ, and ShapeNet, using various conditioning inputs such as images, sketches, and text prompts. Please see the project website (`https://jiataogu.me/control3diff`) for video comparisons.

1. Introduction

The synthesis of photo-realistic 3D-aware images of real-world scenes from sparse controlling inputs is a long-

standing problem in both computer vision and computer graphics, with various applications including robotics simulation, gaming, and virtual reality. Depending on the task, sparse inputs can be single-view images, guiding poses, or text instructions, and the objective is to recover 3D representations and synthesize consistent images from novel viewpoints. This is a challenging problem, as the sparse inputs typically contain insufficient information to predict complete 3D details. Consequently, the selection of an appropriate *prior* during controllable generation is crucial for resolving uncertainties. Recently, significant progress has been made in the field of 2D image generation through the use of diffusion-based generative models [87, 31, 89, 20], which learn the *prior* and have achieved remarkable success in various conditional applications such as super-resolution [77, 49, 27], in-painting [54], image translation [75, 100] and text-guided synthesis [70, 73, 76, 30]. It is natural to consider applying similar approaches in 3D generation. However, learning diffusion models typically relies heavily on the availability of ground-truth data, which is not commonly available for 3D content, especially for single-view images.

To address this limitation, we propose a framework called Control3Diff, which links diffusion models to generative adversarial networks (GANs) [23] and takes advantage of the success of GANs in 3D-aware image synthesis [81, 10, 62, 25, 11, 64, 86]. The core idea behind 3D GANs is to learn a generator based on neural fields that fuse 3D inductive bias in modeling with volume rendering. By training 3D GANs on single-view data with random noises and viewpoints as inputs, we can avoid the need for 3D ground truth. Our proposed framework Control3Diff predicts the internal states of 3D GANs given any conditioning inputs by modeling the prior distribution of the underlying manifolds of real data using diffusion models. Furthermore, the proposed framework can be trained on synthetic generation from a 3D GAN, allowing for infinite examples to be used for training without worrying about over-fitting. Finally, by applying the guidance techniques [20, 32] in 2D diffusion models, we are able to learn controllable 3D generation with a single loss function for all conditional tasks. This eliminates the use of ad-hoc supervisions and constraints which were commonly needed in existing conditional 3D generation [8, 19].

To validate the proposed framework, we use a variant of the recently proposed EG3D [11] that learns an efficient tri-plane representation as the basis for Control3Diff. We extensively conduct experiments on six types of inputs and demonstrate the effectiveness of Control3Diff on standard benchmarks including FFHQ, AFHQ-cat, and Shapenet.

2. Preliminaries: Controllable Image Synthesis

In this section, we first define the problem of *controllable image synthesis in 2D and 3D-aware manners* and review the 2D solutions with diffusion models. Then, we pose the difficulties of applying similar methods to 3D scenario.

2.1. Definition

2D. The goal of controllable synthesis is to learn a generative model that synthesizes diverse 2D images $\boldsymbol{x}$ conditioned on an input control signal $\boldsymbol{c}$. This can be mainly done by sampling images in the following two ways:

$$\boldsymbol{x} \sim \exp\left[-\ell(\boldsymbol{c}, \boldsymbol{x})\right] \cdot p_\theta(\boldsymbol{x}) \quad \text{Or} \quad p_\theta(\boldsymbol{x}|\boldsymbol{c}), \tag{1}$$

where θ is the parameters of the generative model. The former one is called *guidance*. At test-time, an energy function $\ell(\boldsymbol{c}, \boldsymbol{x})$ is to measure the alignment between the synthesized image $\boldsymbol{x}$ and the input $\boldsymbol{c}$ to guide the prior generation $p_\theta(\boldsymbol{x})$. Note that, only for the controllable tasks where the energy function $\ell(\boldsymbol{c}, \boldsymbol{x})$ can be defined, the *guidance* techniques can be applied. The latter one directly formulates it as a *conditional* generation problem $p_\theta(\boldsymbol{x}|\boldsymbol{c})$ if the paired data $(\boldsymbol{x}, \boldsymbol{c})$ is available. As $\boldsymbol{c}$ typically contains less information than $\boldsymbol{x}$, it is crucial to handle uncertainties with generative models.

3D. The above formulation can be simply extended to 3D. In this work, we assume a 3D scene represented by latent representation $\boldsymbol{z}$, and we synthesize 3D-consistent images by rendering $\boldsymbol{x} = \mathcal{R}(\boldsymbol{z}, \pi)$ given different camera poses π. Here, we do not restrict the space of $\boldsymbol{z}$, meaning that it can be any high-dimensional structure that describes the 3D scene. Similarly, we can define 3D-aware controllable image synthesis by replacing $\boldsymbol{x}$ with $\boldsymbol{z}$ in Eq. (1).

2.2. Diffusion Models

Standard diffusion models [87, 89, 31] are explicit generative models defined by a Markovian process. Given an image $\boldsymbol{x}$, a diffusion model defines continuous time latent variables $\{\boldsymbol{z}_t | t \in [0, 1], \boldsymbol{z}_0 = \boldsymbol{x}\}$ based on a fixed schedule $\{\alpha_t, \sigma_t\}$: $q(\boldsymbol{z}_t|\boldsymbol{z}_s) = \mathcal{N}(\boldsymbol{z}_t; \alpha_{t|s}\boldsymbol{z}_s, \sigma^2_{t|s} I),\ 0 \le s < t \le 1$, where $\alpha_{t|s} = \alpha_t/\alpha_s, \sigma^2_{t|s} = \sigma^2_t - \alpha^2_{t|s}\sigma^2_s$. Following this definition, we can easily derive the latent $\boldsymbol{z}_t$ at any time by $q(\boldsymbol{z}_t|\boldsymbol{z}_0) = \mathcal{N}(\boldsymbol{z}_t; \alpha_t \boldsymbol{z}_0, \sigma^2_t I)$. The model θ then learns the reverse process by denoising $\boldsymbol{z}_t$ to the clean target $\boldsymbol{x}$ with a weighted reconstruction loss $\mathcal{L}_\theta$:

$$\mathcal{L}_{\text{Diff}} = \mathbb{E}_{\boldsymbol{z}_t \sim q(\boldsymbol{z}_t|\boldsymbol{z}_0), t\sim[0,1]} \left[\omega_t \cdot \|\boldsymbol{z}_\theta(\boldsymbol{z}_t) - \boldsymbol{z}_0\|_2^2\right]. \tag{2}$$

Typically, θ is parameterized as a U-Net [74, 31] or ViT [66]. Sampling from a learned model p_θ can be performed using ancestral sampling rules [31] – staring with pure Gaussian noise $\boldsymbol{z}_1 \sim \mathcal{N}(0, I)$, we sample s, t following a uniformly spaced sequence from 1 to 0:

$$\boldsymbol{z}_s = \alpha_s \boldsymbol{z}_\theta(\boldsymbol{z}_t) + \sqrt{\sigma_s^2 - \bar{\sigma}^2}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t) + \bar{\sigma}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \tag{3}$$

where $\bar{\sigma} = \sigma_s \sigma_{t|s}/\sigma_t$ and $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t) = (\boldsymbol{z}_t - \alpha_t \boldsymbol{z}_\theta(\boldsymbol{z}_t))/\sigma_t$. By decomposing the sophisticated generative process into

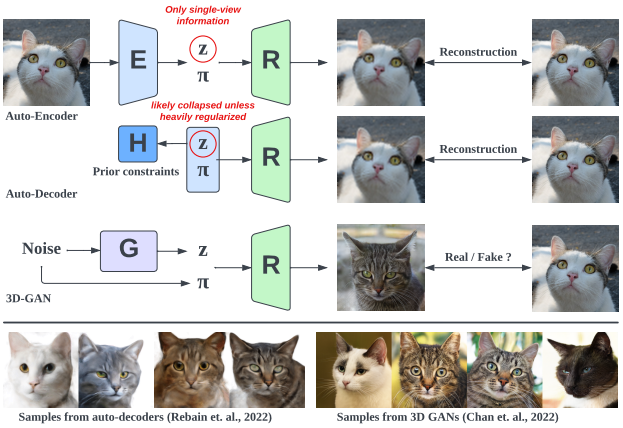


Figure 2: ↑ Comparison between reconstruction-based and GAN-based approaches for obtaining latents $\boldsymbol{z}$ for learning the LDM; ↓ learned samples from auto-decoders generally have lower quality than 3D GANs due to regularization and optimization difficulties.

hundreds of denoising steps, diffusion models effectively expand the modeling capacity, and have been shown superior performance than other types of generative models [20]. For better efficiency, Latent Diffusion Models (LDM [73]) have extended the process in latent space by learning an additional encoder $\boldsymbol{z}_0 = \mathcal{E}(\boldsymbol{x})$ to map input images to the latent space.

Due to the autoregressive nature, diffusion models are suitable for controllable generation (§ 2.1). Prior research studied *guidance* with a variety of classifiers, constraints or auxiliary loss functions [20, 44, 17, 24, 33, 26, 5]. Other works explored learning *conditional diffusion* models with parallel data (e.g., class labels [32], text prompts [73], aligned image maps [100]). Importantly, *classifier-free guidance* [32], which enables generation with a balance between sampling controlled quality and diversity, has become a basic building block for large-scale diffusion models [76, 70].

2.3. 3D-aware Image Synthesis

When extending image synthesis to 3D, one can model each 3D scene, which corresponds to a latent representation $\boldsymbol{z}$ (§ 2.1), as a neural radiance field (NeRF [57]) $f_{\boldsymbol{z}} : \mathbb{R}^5 \to \mathbb{R}^4_+$ which maps every spatial point and the viewing direction to its radiance and density. $f_{\boldsymbol{z}}$ is parameterized as MLPs [57] or tri-planes [11, 14] with upsamplers [62, 25]. Next, we can synthesize 3D-consistent images via volume rendering [56].

Despite the success in the 2D scenario, diffusion models have rarely been applied directly in controllable 3D-aware image synthesis with NeRF. There are three **key challenges**:

1. Learning diffusion models requires the 3D ground-truth $\boldsymbol{z}$ (shown in Eq. (2)) that is often unavailable.
2. While there exist approaches to acquire high-quality 3D labels from dense multi-view image collections, for most of the cases, only single-view images are available.
3. As discussed in § 2.1, we need either an energy function $\ell(.,.)$ for *guidance* or paired data for *conditional generation* in controllable synthesis. However, both of them are not straightforward to define in the latent space of implicit 3D representations (e.g., NeRF).

More precisely, targeting on Challenge # 1, prior arts [6, 83, 60, 94] first reconstruct the latent $\boldsymbol{z}$ from dense multi-view images of each scene. In the rest of paper, we refer to them as *reconstruction-based* methods. That is, given a set of posed images $\{\boldsymbol{x}_i\}_{i=1}^N$, one can minimize:

$$\mathcal{L}_{\text{RC}} = \mathbb{E}_{\{\boldsymbol{x}_i\}\sim\text{data}}\left[\sum_i \|\mathcal{R}(\boldsymbol{z}, \pi_i) - \boldsymbol{x}_i\|_2^2 + \mathcal{H}(\boldsymbol{z})\right], \quad (4)$$

where π_i is the camera of $\boldsymbol{x}_i$, $\mathcal{R}$ is the differentiable volume renderer of $f_{\boldsymbol{z}}$, and $\mathcal{H}$ is the prior regularization over $\boldsymbol{z}$. Here, $z = \mathcal{E}(\{\boldsymbol{x}_i\}_{i=1}^N)$ represents either the backward process of $\nabla_{\boldsymbol{z}}\mathcal{L}_{\text{recon}}$ (also known as "auto-decoder" [85]) that updates $\boldsymbol{z}$ via SGD, or an amortized multi-view encoder [48, 78].

In spite of the good results with dense multi-view data for training, these methods perform poorly when only one view is available for each scene (Challenge #2). Single-view auto-decoders usually fail to learn geometry, even with strong regularization [71], the reconstructed quality is still limited. On the other side, using an encoder $\mathcal{E}(\boldsymbol{x})$ may ease the aforementioned issues after adopting various auxiliary losses with novel-view rendering [8]. Yet, due to limited view coverage and object occlusion, an image encoder is unable to predict fully determined 3D details, resulting in additional uncertainties. We illustrate a comparison in Fig. 2. Besides Challenge #1 and #2, #3 has rarely been studied in prior research. In the next section, we will elaborate on how we address these challenges to achieve controllable 3D-aware image synthesis with only single-view images for training.

3. Method: Control3Diff

We present Control3Diff, a controllable 3D-aware generation framework based on a 3D GAN (§ 3.1). We study two ways of controlling image synthesis with Control3Diff (§§ 3.2 and 3.3). The pipeline is illustrated in Fig. 3.

3.1. Latent Diffusion with 3D GANs

Instead of acquiring $\boldsymbol{z}$ from dense multi-view images $\{\boldsymbol{x}_i\}$ as done in reconstruction-based methods, we directly sample from the learned distribution of $\boldsymbol{z}$ of a 3D GAN model, which is trained on single-view images. In this paper, considering its state-of-the-art performance, we build Control3Diff based on EG3D [11]. EG3D first learns a tri-plane generator $\mathcal{G} : \boldsymbol{u} \in \mathbb{R}^{512} \to \boldsymbol{z} \in \mathbb{R}^{3\times256\times256\times32}$, mapping low-dimensional noises to an expressive tri-plane. The feature of a 3D point is obtained by projecting the point to three orthogonal planes and gathering local features from the three

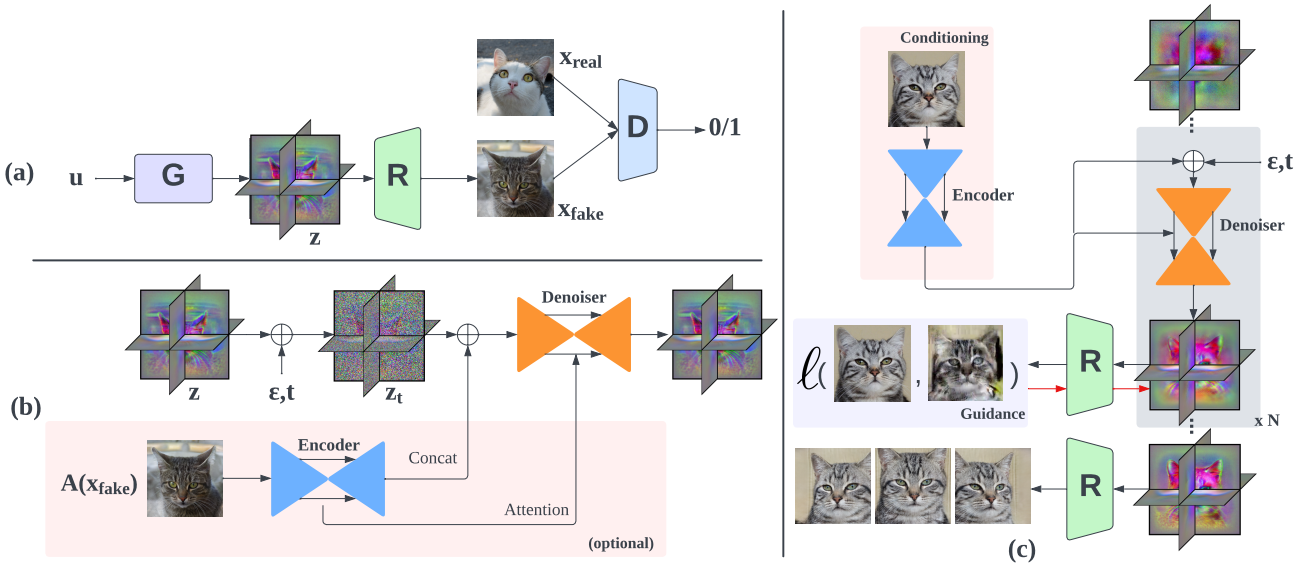


Figure 3: Pipeline of Control3Diff. (a) 3D GAN training; (b) Diffusion model trained on the extracted tri-planes can be trained with or without the input conditioning; (c) controllable 3D generation with the learned diffusion model, optionally with guidance. The tri-plane features are presented in three color planes, and the camera poses are omitted for better visual convenience.

planes, which is the input to the radiance function $f_{\boldsymbol{z}}$ for radiance and density prediction.

Training an EG3D model requires a joint optimization of a camera-conditioned discriminator $\mathcal{D}$, and we adopt the non-saturating logistic objective with R1 regularization:

$$\begin{aligned}\mathcal{L}_{\text{GAN}} = &\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(0,I),\pi\sim\Pi}\left[h\left(\mathcal{D}(\mathcal{R}\left(\mathcal{G}(\boldsymbol{u}),\pi\right),\pi\right)\right] \\ &+\mathbb{E}_{\boldsymbol{x},\pi\sim\text{data}}\left[h\left(-\mathcal{D}(\boldsymbol{x},\pi)\right)+\gamma\|\nabla_{\boldsymbol{x}}\mathcal{D}(\boldsymbol{x},\pi)\|_2^2\right],\end{aligned} \tag{5}$$

where $h = -\log(1+\exp(-u))$ and Π is the prior camera distribution. The adversarial learning enables the training on single-view images, as it only forces the model output to match the training data distribution rather than learns a one-to-one mapping as an auto-encoder. Note that, in order to train diffusion models more stable, besides Eq. (5), we also bound $\mathcal{G}(\boldsymbol{u})$ with $\tanh(.)$ and apply an additional L2 loss similar to [83] when training EG3D. However, we observed in our experiments that these additional constraints would not affect the performance of EG3D.

After EG3D is trained, as the second stage, we apply the denoising on the tri-plane to train a diffusion model with the renderer $\mathcal{R}$ frozen. Training follows the same denoising objective Eq. (2) and $\boldsymbol{z}_0 = \mathcal{G}(\boldsymbol{u})$. As $\boldsymbol{u}$ is randomly sampled, we can essentially learn from unlimited data. Different from [60, 94], we do not need any auxiliary loss or additional architectural change. Optionally, we can add the control signal as the conditioning to the diffusion network to formulate a *conditional* generation framework for controlling (§ 3.2).

We note that, training a diffusion model over $\mathcal{G}(\boldsymbol{u})$ samples differs from distilling a pre-trained GAN into another GAN generator, which is unsuitable for the controlling tasks. Although it is efficient to sample high-quality tri-planes $\boldsymbol{z}$, GANs are implicit generative models [58] and do not model the likelihood in the learned latent space. That is to say, we do not have a proper prior $p(\boldsymbol{z})$ given the latent representations of a 3D GAN. Especially in the high-dimensional space like tri-planes, any control without knowing the underlying density will easily fall off the learned manifold and output degenerated results. As a result, almost all existing works [8, 19] utilize 3D GANs for controlling the focus on low-dimensional spaces, which can be approximately assumed Gaussian. However, this has to scarify the controllability. In contrast, diffusion models explicitly learn the score functions of the latent distribution even with high-dimensionality [89], which fills in the missing pieces for 3D GANs. Also see experimental comparison in Table 1.

3.2. Conditional 3D Diffusion

We can synthesize controllable images by extending latent diffusion into a conditional generation framework. Conventionally, learning such conditional models requires labeling parallel corpus, e.g., large-scale text-image pairs [80] for the Text-to-Image task. Compared to acquiring 2D paired data, creating the paired data of the control signal and 3D representation is much more difficult. In our method, however, we can easily synthesize an infinite number of pairs of the control signal and triplanes by using the rendered images of the triplane from 3D GAN to predict the control signal with an off-the-shelf method. Now, the learning objective

can be written as follows:

$$\mathcal{L}_{\text{Cond}} = \mathbb{E}_{\boldsymbol{z}_0,\boldsymbol{z}_t,t,\pi}\left[\omega_t \cdot \|\boldsymbol{z}_\theta(\boldsymbol{z}_t, \mathcal{A}\left(\mathcal{R}(\boldsymbol{z}_0,\pi)\right)) - \boldsymbol{z}_0\|_2^2\right]. \tag{6}$$

where $\boldsymbol{z}_0 = \mathcal{G}(\boldsymbol{u})$ is the sampled tri-plane, $\mathcal{A}$ is the off-the-shelf prediction module that converts rendered images into $\boldsymbol{c}$ (e.g., "edge-detector" for edge-map to 3D generation), and $\pi \sim \Pi$ is a pre-defined camera distribution based on the testing preference. Here $\boldsymbol{z}_\theta$ represents a conditional denoiser that learns to predict denoised tri-planes given the condition. In early exploration, we noticed that the prior camera distribution Π significantly impacts the generalizability of the learned model, where for some datasets (e.g., FFHQ, AFHQ), the biased camera distribution in training set would cause degenerated results for rare camera views. Therefore, we specifically re-sample the cameras for these datasets.

Joint Diffusion with Camera Pose π_c Conditional models can be learned without camera input, which implicitly maps the input view to the global triplane space. It implies that conditional models can predict camera information through diffusion. In light of this observation, we propose to jointly predict the input camera pose π_c with $\boldsymbol{z}$ in one diffusion framework. Similar to 3D-aware generation, predicting cameras from a single view is also a challenging problem, requiring resolving ambiguities in natural images. At the same time, previous works either rely on external deterministic camera predictors [52] or optimize the cameras at inference time [46]. In this work, for simplicity, we flatten π_c into a vector, broadcast it, and concatenate it to the channels of $\boldsymbol{z}$ as the new diffusion target.

3.3. Guided 3D Diffusion

An alternative way to control image synthesis is to follow a similar recipe in 2D (as defined in § 2.1) to perform test-time guidance based on a task-specific energy function $\ell(\boldsymbol{c}, \boldsymbol{z})$. Nevertheless, directly defining such an energy function between $\boldsymbol{c}$ and 3D representation (i.e., a tri-plane NeRF) is challenging. We circumvent this by defining $\ell(.,.)$ to measure the closeness between $\boldsymbol{c}$ and the differentiablly rendered image $\mathcal{R}(\boldsymbol{z}, \pi_c)$. In this way, we can learn the 3D representation using 2D rendering guidance (e.g., CLIP score [68] for text-to-3D, and MSE or perceptual loss [103] for image inversion). Using 2D guidance for learning 3D representation is reasonable since the final targets of most controlling tasks we care about are images synthesized from certain viewpoints. The 2D rendering guidance can be implemented efficiently via replacing $\boldsymbol{z}_\theta(\boldsymbol{z}_t)$ in Eq. (3) with $\hat{\boldsymbol{z}}_\theta(\boldsymbol{z}_t)$ as:

$$\hat{\boldsymbol{z}}_\theta(\boldsymbol{z}_t) = \boldsymbol{z}_\theta(\boldsymbol{z}_t) - w_t \nabla_{\boldsymbol{z}_t} \ell\left[\boldsymbol{c}, \mathcal{R}(\boldsymbol{z}_\theta(\boldsymbol{z}_t), \pi_c)\right], \tag{7}$$

where $\boldsymbol{z}_\theta$ is the denoised tri-plane derived from the unconditional prior, w_t is the time-dependent weight.

Langevin correction steps While the 2D rendering guidance can provide a gradient to learning 3D representations, the optimization is not often stable due to the nonlinearity of mapping from 2D to 3D. Our initial experiments showed that early guidance steps get stuck in a local minimum with incorrect geometry prediction, which is hard to correct in the later denoising stage when the noise level decreases. Therefore, we adopt similar ideas from the predictor-corrector [90, 33] to include additional Langevin correction steps before the diffusion step (Eq. (3)):

$$\boldsymbol{z}_t = \boldsymbol{z}_t - \frac{1}{2}\delta\sigma_t\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{z}_t) + \sqrt{\delta}\sigma_t\boldsymbol{\epsilon}', \boldsymbol{\epsilon}' \sim \mathcal{N}(0, I), \tag{8}$$

where δ is the step size, and $\hat{\boldsymbol{\epsilon}}_\theta$ is derived from $\hat{\boldsymbol{z}}_\theta$ in Eq. (7). According to Langevin MCMC [55], the additional steps help $\boldsymbol{z}_t$ match the marginal distribution given certain σ_t.

Discussion: Conditioning v.s. Guidance Compared to guidance methods in § 3.3, training a conditional 3D diffusion model has several benefits. First, in guided diffusion, a proper-designed differentiable $\ell(.,.)$ is necessary to back-propagate the gradient guidance to the diffusion model, which, however, is not available for all kinds of conditional tasks. In contrast, conditional models do not have such requirements and can adapt any conditional distribution. Also, conditioning is computationally more efficient because the guidance requires rendering and back-propagating through the volume renderer $\mathcal{R}$ at each step. However, conditioning methods have a possible issue. As we train our models based on the images generated by a pretrained 3D GAN (Eq. (6)), the learned $p(\boldsymbol{z}|\boldsymbol{c})$ probably has domain gaps between real images and synthesized images. In such a case, guidance-based methods become more reliable as ℓ is directly computed upon real controls.

Optionally, we can combine the best of both worlds when ℓ is available. For instance, we learn a conditional diffusion model and generate samples jointly with guidance (see Fig. 3 (c)). This paradigm can also be used when the test camera is not given: the guidance is used to update the camera π_c predicted by the aforementioned conditional model.

4. Experiments

4.1. Experimental Settings

Dataset & Tasks We evaluate Control3Diff on three standard image generation benchmarks – FFHQ (512^2) [41], AFHQ-cat (512^2) [16], and ShapeNet (128^2) [85] including two categories *Cars* and *Chairs*. Following EG3D [11], each image is associated with its camera pose. We consider *six* controllable 3D-aware generation tasks. For all datasets, we test the standard *image-to-3D inversion* (original resolution and low-resolution inputs) and *edge-map to 3D* generations.

Figure 4: Comparison for 3D-inversion of *in-the-wild* images. We compare the proposed approach to direct prediction of the GAN's latent $\mathcal{W}$ and Tri-plane with a learned encoder, as well as an optimization based approach to infer the latent and expanded latent $\mathcal{W}$, $\mathcal{W}+$, as well as the Tri-plane, following [2]. Our method achieves better view consistency with higher output image quality compared to baselines.

Table 1: Quantitative comparison on inversion. Although optimizing the Tri-plane model can fit input views well, it falls short in generating realistic novel view images. Overall, our method achieves the best performance.

		FFHQ							AFHQ-Cat				
		PSNR ↑	SSIM ↑	LPIPS↓	ID↑	nvFID ↓	nvKID ↑	nvID↑	PSNR↑	SSIM ↑	LPIPS↓	nvFID ↓	nvKID ↑
Opt.	$\mathcal{W}$	15.93	0.68	0.42	0.60	39.26	0.023	0.57	16.08	0.57	0.42	9.15	0.004
	$\mathcal{W}+$	17.91	0.73	0.34	0.74	38.23	0.022	0.68	18.32	0.62	0.35	10.54	0.006
	Tri.	18.32	0.78	**0.11**	**0.92**	138.0	0.154	0.54	17.53	**0.71**	**0.14**	98.79	0.085
Pred.	$\mathcal{W}$	14.82	0.64	0.54	0.37	45.06	0.018	0.35	14.56	0.52	0.55	20.87	0.006
Ours		**22.30**	**0.79**	0.23	0.89	**13.48**	**0.005**	**0.81**	**20.11**	0.66	0.24	**7.03**	**0.003**

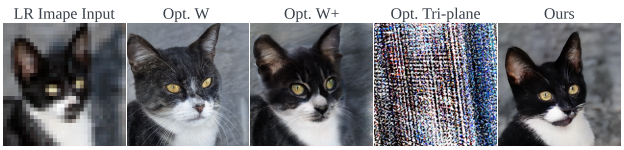


Figure 5: Comparison on the *SR+inversion* task. By learning the proper prior with diffusion models, Control3Diff is able to synthesize realistic and faithful cat faces from low-resolution inputs, while optimization-based approaches fail completely due to the lack of proper 3D prior.

For faces, we further explored *segmentation to 3D*, *head-shape to 3D* and *text-description to 3D* tasks to validate the controllability at various levels. To compare with previous work [19] for *Seg-to-3D*, we additionally train one model on CelebA-HQ [38]. Besides, we also report performance on unconditional generation with guidance in the ablation.

Baselines We choose the standard optimization-based and encoder-based [92, 47] approaches for image-to-3D inversion, and the recent Pix2Pix3D [19] as the major baseline to compare on the *Seg-to-3D* task. Note that we do not focus on achieving the state-of-the-art on a single task like inversion, but rather to highlight the potential of our generic framework in 3D-aware generation. Thus, our comparison limits to methods without fine-tuning the model weights [72].

Evaluation Metrics For image synthesis quality, we report five standard metrics: PSNR, SSIM, SG diversity [13], LPIPS [103], KID, and FID [29]. For face, we compute the cosine similarity of the facial embeddings generated by the

Table 2: Quantitative comparison on Seg2Face and Seg2Cat.

task	Seg2Face				Seg2Cat			
metric	FID↓	SG↑	mIoU↑	MPA↑	FID↓	SG↑	mIoU↑	MPA↑
p2p3D	21.28	**0.46**	0.52	0.63	15.46	**0.50**	0.64	0.76
ours	**12.85**	0.43	**0.61**	**0.72**	**11.66**	0.47	**0.67**	**0.79**

facial recognition network for a given pair of faces, utilizing it as ID metric. In the context of conditional generation tasks, following Pix2Pix3D [19], we evaluate methods using mean Intersection-over-Union (mIoU) and mean pixel accuracy (MPA) for segmentation maps.

Implementation Details We implemented all our models based on the standard U-Net architectures [20] where for conditional diffusion models, an U-Net-based encoder is adopted to encode the input image similar to [26], see Fig. 3 (b). We include the hyper-parameter details in Appendix.

4.2. Image-to-3D Inversion

In this section, we evaluate Control3Diff on 3D inversion tasks, comparing our methods in two cases: (1) standard inversion and (2) a more challenging 3D super-resolution task. To establish a baseline, we directly optimize the low-dimensional latent vectors ($\mathcal{W}$, $\mathcal{W}+$)*, following [2], as well as triplanes. As conventional GANs do not have learned priors in these spaces, optimization is performed with noise injection regularization. We also employ an encoder-based approach [47] that directly predicts $\mathcal{W}$ or triplanes. To predict triplanes, we train a separate encoder.

The results are shown in Table 1 and Fig. 4 where our methods significantly outperform the other methods in terms of both image quality and identity consistency. While direct optimization of triplanes may yield higher accuracy in the input view, it always results in collapsed novel view results due to a lack of prior. We also show the visual comparisons for 3D super-resolution in Fig. 5 where our diffusion-based approaches show more gains.

4.3. Seg-to-3D & Edge-to-3D Synthesis

We evaluate our methods on more general conditional 3D generation tasks where the input control is not necessarily the target view, e.g., *Seg-to-3D* and *Edge-to-3D* tasks. For the *Seg-to-3D* task, we train two additional parsing networks [99] with labels provided by Pix2Pix3D [19], where the segmentation ground-truth of cats is obtained via clustering the DINO feature as proposed by [3]. It has been observed that this clustering scheme adversely affects the performance of cat-parsing networks, resulting in lower accuracy than that achieved by face-parsing networks.

The results of our evaluation are presented in Table 2, which indicates that our method generates images with comparable alignment and quality. Furthermore, as illustrated in Fig. 6, our method is capable of producing more realistic faces in novel views. Additionally, our model successfully generates consistent 3D objects by taking as input edge maps, as illustrated in Fig. 6.

*$\mathcal{W}$, $\mathcal{W}+$ refers to the compact and expanded latent space of the GAN, respectively. See [2] for details.

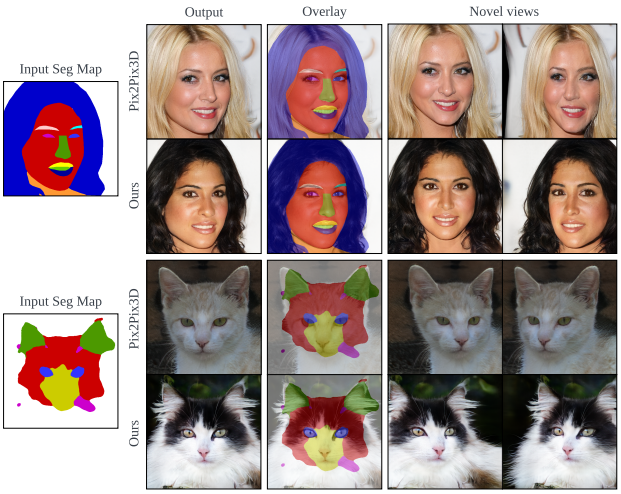


Figure 6: Comparison on *Seg-to-3D* generation. **All faces are model generated, and are not real identities.** Our proposed method generates images that achieve improved alignment with the segmentation map and greater 3D consistency.

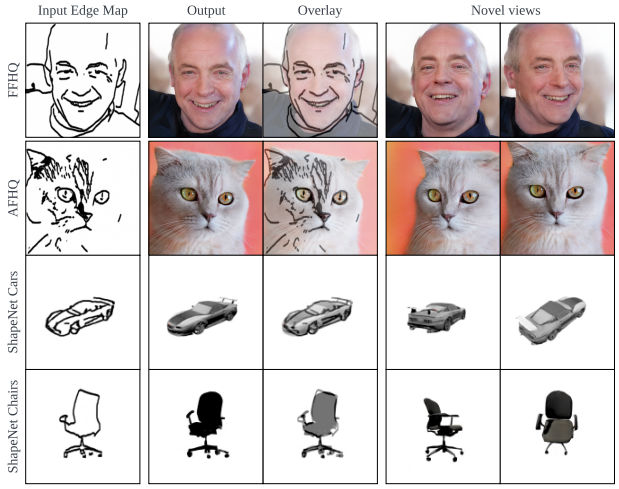
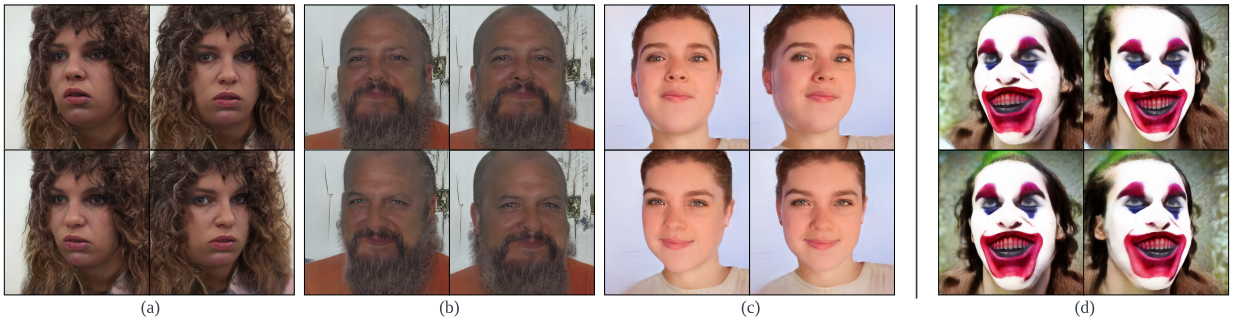


Figure 7: Qualitative results on *Edge-to-3D* generation on all three datasets. **All faces are model generated.**

4.4. Text-to-3D Synthesis

We demonstrate the versatility of our framework by applying it to text-to-3D generation. The qualitative results are shown in Fig. 8. For (a)-(c), we train Control3Diff as a



(a) (b) (c) (d)

Figure 8: Qualitative results on *Text-to-3D* synthesis based on given prompts: *(a) A middle-aged woman with curly brown hair and pink lips; (b) A middle-aged man with a receding hairline, a thick beard, and hazel eyes; (c) A young woman with freckles on her cheeks and brown hair in a pixie cut; (d) a photography of Joker's face.* **All faces are model generated, and are not real identities.**

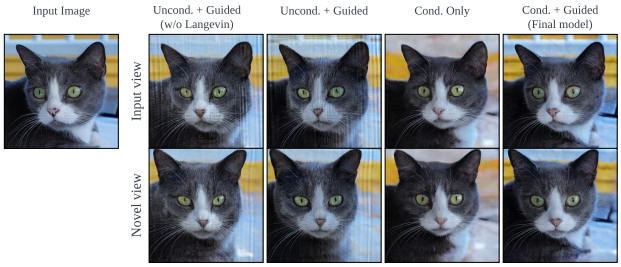


Figure 9: Comparison between conditional and guided diffusion.

conditional diffusion model where we adopt the normalized CLIP embedding of the model's rendering as conditioning. In test time, such a model can be seamlessly switched to text-control thanks to the multi-modal space of CLIP. We also conduct experiments with text feature guidance in (d), where we directly apply a pre-trained 2D diffusion model as a score function similar to DreamFusion [67], and guide the generation of an unconditional 3D diffusion model.

4.5. Ablation Study

We conducted an ablation study on the task of Image-to-3D inversion, evaluating the effects of conditioning and guidance on the visual performance of our method. As illustrated in Fig. 9, our results demonstrate that the inclusion of conditioning and guidance leads to superior visual performance, while their absence results in artifacts and an inability to fit target images. More specifically, if we only apply guidance on unconditional models, the generated outputs seem to have artifacts. On the other hand, when using the conditional model only, the model is unable to recover all details from the input image especially for background.

5. Related Work

Diffusion for 3D-aware Generation There have been recent attempts [95, 6, 61, 59, 4, 84, 15] to extend diffusion models to 3D. The key challenge here is to obtain 3D ground truth for training. Most works tackle this challenge by reconstructing 3D ground truth from dense multi-view data. Instead, our method can be trained only on single-view data by using a 3D GAN to synthesize infinite ground-truth 3D data. Another line of work [67, 93, 104, 18, 26] applies 2D diffusion priors to the sparse-view reconstruction or text-to-3D generation tasks. For example, NerfDiff [26] applies a test-time optimization by distilling 2D diffusion priors into NeRF for single-view reconstruction. Different from NerfDiff [26], our focus is 3D-aware image synthesis controlled by various control signals, and we apply denoising directly in 3D. Furthermore, our method can be trained on single-view datasets without the need of multi-view data.

Controllable Image Synthesis with GANs Conventional GANs [22, 41, 42] can generate photo-realistic images from low-dimensional randomly sampled latent vectors, but have limited controllability. Follow-up works enable controllability by either adding conditioning input along with the sampled vectors as input (named "Conditional GAN") [35, 65] or manipulating the sampled vectors [82, 28, 105]. These works only focus on 2D image synthesis with control, which cannot explicitly control 3D properties (e.g., cameras) and synthesize multi-view consistent images. Recently, 3D-GANs [81, 11, 12, 63, 25, 97] have been developed by integrating 3D representation and rendering into GANs. While these models can control 3D properties by manipulating the latent vectors, their controllability is limited to global camera poses or geometry. Many works [37, 91, 7, 36] support fine-grained geometry editing, but most of them have only demonstrated results on human face or body. Other conditional 3D GANs for general objects [8, 19] need additional constraints or architecture changes, however, their synthesis quality is still limited. In contrast, our method allows a variety of control signals (e.g., segmentation map) for fine-level 3D-aware image synthesis on various kinds of objects.

6. Conclusion

In summary, we propose Control3Diff, a versatile approach for 3D-aware image synthesis that combines the

strengths of 3D GANs and diffusion models. Our method enables precise control over image synthesis by explicitly modeling the underlying latent distribution. We validate our approach on standard benchmarks, demonstrating its efficacy with various types of conditioning inputs. Control3Diff represents a significant advancement in generative modeling in 3D, opening up new research possibilities in this area.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8296–8305, 2020.

[3] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

[4] Titas Anciukevicius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J. Mitra, and Paul Guerrero. RenderDiffusion: Image diffusion for 3D reconstruction, inpainting and generation. *arXiv*, 2022.

[5] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. *arXiv preprint arXiv:2302.07121*, 2023.

[6] 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. *arXiv*, 2022.

[7] Alexander W. Bergman, Petr Kellnhofer, Wang Yifan, Eric R. Chan, David B. Lindell, and Gordon Wetzstein. Generative neural articulated radiance fields. In *NeurIPS*, 2022.

[8] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional π-gan for single image to neural radiance fields translation. *arXiv preprint arXiv:2202.13162*, 2022.

[9] Caroline Chan, Frédo Durand, and Phillip Isola. Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7915–7925, 2022.

[10] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *arXiv*, 2020.

[11] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. *arXiv preprint arXiv:2112.07945*, 2021.

[12] Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. *CVPR*, 2021.

[13] Anpei Chen, Ruiyang Liu, Ling Xie, Zhang Chen, Hao Su, and Jingyi Yu. Sofgan: A portrait image generator with dynamic styling. *ACM Transactions on Graphics (TOG)*, 41(1):1–26, 2022.

[14] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022.

[15] 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, 2023.

[16] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[17] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[18] Congyue Deng, Chiyu "Max" Jiang, Charles R. Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, and Dragomir Anguelov. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors, 2022.

[19] Kangle Deng, Gengshan Yang, Deva Ramanan, and Jun-Yan Zhu. 3d-aware conditional image synthesis. *arXiv preprint arXiv:2302.08509*, 2023.

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[21] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021.

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[24] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *arXiv preprint arXiv:2206.09012*, 2022.

[25] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021.

[26] Jiatao Gu, Alex Trevithick, Kai-En Lin, Josh Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. *arXiv preprint arXiv:2302.10109*, 2023.

[27] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Miguel Angel Bautista, and Josh Susskind. f-dm: A multi-stage diffusion model via progressive signal transformation. *arXiv preprint arXiv:2210.04955*, 2022.

[28] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020.

[29] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[30] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[31] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[32] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[33] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

[34] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. *arXiv preprint arXiv:2301.11093*, 2023.

[35] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[36] Kaiwen Jiang, Shu-Yu Chen, Feng-Lin Liu, Hongbo Fu, and Lin Gao. Nerffaceediting: Disentangled face editing in neural radiance fields, 2022.

[37] Sun Jingxiang, Wang Xuan, Wang Lizhen, Li Xiaoyu, Zhang Yong, Zhang Hongwen, and Liu Yebin. Next3d: Generative neural texture rasterization for 3d-aware head avatars. *arXiv preprint arXiv:2205.15517*, 2022.

[38] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *International Conference on Learning Representations*, 2018.

[39] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020.

[40] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021.

[41] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

[42] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020.

[43] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[44] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793*, 2022.

[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[46] Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 3d gan inversion with pose optimization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2967–2976, 2023.

[47] Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 3d gan inversion with pose optimization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2967–2976, 2023.

[48] Adam R. Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Soňa Mokrá, and Danilo J. Rezende. NeRF-VAE: A Geometry Aware 3D Scene Generative Model. *ICML*, 2021.

[49] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022.

[50] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017.

[51] Yu-Jhe Li, Tao Xu, Bichen Wu, Ningyuan Zheng, Xiaoliang Dai, Albert Pumarola, Peizhao Zhang, Peter Vajda, and Kris Kitani. 3d-aware encoding for style-based neural radiance fields. *arXiv preprint arXiv:2211.06583*, 2022.

[52] Connor Z Lin, David B Lindell, Eric R Chan, and Gordon Wetzstein. 3d gan inversion for controllable portrait image animation. *arXiv preprint arXiv:2203.13441*, 2022.

[53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[54] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[55] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 28, 2015.

[56] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

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

[58] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

[59] Norman Müller, , Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner.

[60] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias

Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. *arXiv preprint arXiv:2212.01206*, 2022.

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

[62] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. *CVPR*, pages 11453–11464, 2021.

[63] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[64] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513, 2022.

[65] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

[66] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[67] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[68] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[69] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[70] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[71] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. Lolnerf: Learn from one look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1558–1567, 2022.

[72] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42(1):1–13, 2022.

[73] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[74] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net : Convolutional Networks for Biomedical Image Segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[75] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[76] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[77] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv:2104.07636*, 2021.

[78] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene Representation Transformer: Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations. *CVPR*, 2022.

[79] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[80] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[81] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020.

[82] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020.

[83] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. *arXiv preprint arXiv:2211.16677*, 2022.

[84] J. Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion, 2022.

[85] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. *Advances in Neural Information Processing Systems*, pages 1119–1130, 2019.

[86] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *arXiv preprint arXiv:2206.10535*, 2022.

[87] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[88] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[89] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[90] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[91] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *arXiv preprint arXiv:2205.15517*, 2022.

[92] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021.

[93] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation, 2022.

[94] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. *arXiv preprint arXiv:2212.06135*, 2022.

[95] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, and Baining Guo. Rodin: A generative model for sculpting 3d digital avatars using diffusion, 2022.

[96] 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.

[97] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. 2021.

[98] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural Radiance Fields from One or Few Images. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[99] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129:3051–3068, 2021.

[100] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[101] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An End-to-End Deep Learning Architecture for Graph Classification. *AAAI*, 2018.

[102] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[103] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[104] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction, 2022.

[105] Jiapeng Zhu, Ceyuan Yang, Yujun Shen, Zifan Shi, Deli Zhao, and Qifeng Chen. Linkgan: Linking gan latents to pixels for controllable image synthesis. *arXiv preprint arXiv:2301.04604*, 2023.

Appendix

A. Dataset Details

FFHQ contains 70k images of real human faces in resolution of 1024^2. We directly adopted the downsampled, realigned version provided by EG3D [11], which re-cropped the face and estimate the camera poses.

AFHQ-cat contains in total 5K images of cat faces in resolution of 512^2. The same as FFHQ, we directly download the data with estimated camera poses.

ShapeNet Cars & Chairs are standard benchmarks for single-image view synthesis [85]. We use the data modified by pixelNeRF [98] [†]. The chairs dataset consists of 6591 scenes, and the cars dataset has 3514 scenes, both with a predefined train/val/test split. Each training scene contains 50 posed images taken from random points on a sphere. Each testing scene contains 250 posed images taken on an Archimedean spiral along the sphere. All images are rendered at a resolution of 128^2.

CelebA-HQ Dataset [38] is comprised of 30,000 high-resolution images, each with dimensions of 1024^2 pixels. For the Seg-to-3D task, we utilize camera poses and labels supplied by Pix2Pix3D [19].

StyleGAN3-synthetic . Owing to concerns regarding individual consent, we utilize the StyleGAN3 [40] algorithm to synthesize 165 images that subsequently facilitate qualitative analysis and video production. This methodology adheres to ethical guidelines while effectively enabling the visualization and evaluation of our findings. We adhere to the same pre-processing procedure utilized by EG3D [11] for these synthetic images. This approach involves re-centering the faces and estimating the camera positions, thus ensuring a consistent methodology across datasets.

B. Implementation Details

B.1. 3D GAN Settings

Model Our method is largely based on EG3D [11] [‡] which adopts tri-plane representations to achieve efficient rendering process. We use the same hyper-parameters as stated in the original paper [11], where the triplane dimensions are set $3 \times 256 \times 256 \times 32$ for all datasets. To stabilize training of diffusion models, we constraints the value of triplanes by bounding its values to $(-1, 1)$ with $\tanh(.)$. We set the neural rendering resolution to be 128×128 for FFHQ and AFHQ-cat following a $\times 8$ 2D-upsampler, while 64×64 for ShapeNet Cars and Chairs following a $\times 2$ 2D-upsampler.

[†] https://github.com/sxyu/pixel-nerf

[‡] https://github.com/NVlabs/eg3d.git

Training We follow similar recipes [11] for training EG3D models on four datasets. For FFHQ and ShapeNet, we train EG3D from scratch with $\gamma = 1$ and $\gamma = 0.3$, respectively. We first train FFHQ model at 64×64 resolution for 25M images, and another 2.5M images at 128×128. For ShapeNet, we train both datasets with 10M images. AFHQ-cat is a much smaller set, so we fine-tune the FFHQ checkpoint directly at 128×128 with $\gamma = 5$ and data augmentation [39] for 4.5M images. We additionally train an EG3D model on CelebAHQ for comparing on *Seg-to-3D* tasks. For this model, we fine-tune from the pre-trained FFHQ checkpoint with cameras provided by [19]. Both human and cat face models are trained with "generator pose conditioning (GPC)". Moreover, to encourage a smooth learned tri-plane space, we apply an additional regularization over the L2 norm over the tri-plane with weight $\lambda = 1$ for all experiments. We use a batch size of 32 on 8 NIVIDA A100 GPUs, and training approximately takes 3 days for 25M images.

Inference The trained EG3D models are used in both 3D diffusion training & inference. More precisely, we keep the neural renderer (NeRF MLPs + 2D upsamplers, see Fig. 1 for illustration) as the final stage of the tri-plane diffusion, which renders the denoised tri-plane into images given the camera input. To make sure the rendering solely depending on the tri-plane and viewing directions, we adopt the center-camera for GPC, and input the EMA style vector $\boldsymbol{w}_{\text{avg}}$ as well as constant noise to the upsamplers. We did not notice any quality difference by replacing with the average vectors.

B.2. 3D Diffusion Settings

Unconditional Model We use the improved UNet-based architecture [74, 20] for all of our main experiments of tri-plane space diffusion. In the exploration stage, we also tried different architectures such as Transformers [66], however, we did not notice significant difference on generation, and keep UNet as the basic backbone. Since the tri-plane size is fixed across various datasets, we apply exactly the same architecture and hyperparameters for all experiments. Our initial experiments showed that predicting the noise $\boldsymbol{\epsilon}$ (default setting as suggested by DDPM [31]) or the velocity $\boldsymbol{v}$ [79] tend to produce noticeable high-frequency artifacts on the generated tri-planes. We suspect it is due to the tri-plane space is naturally noisier than images, and all our models are trained with the signal $\boldsymbol{z}_0$ prediction as presented in Eq. (2) with $\omega_t = \text{Sigmoid}(\log(\alpha_t^2/\sigma_t^2))$.

Conditional Model The main settings of conditional diffusion models are identical to the unconditional models, except for the interaction module between the conditioning input. For tasks like *3D inversion*, *3D SR*, *Seg-to-3D* and *Edge-to-3D*, we transform the input into RGB images, resize the spatial resolution into 256×256. Then we jointly train a UNet-based encoder which has the same number of layers

and hidden dimensions as the denoiser. Note that, due to the use of self-attention layers [20], the UNet-based encoder is able to globally adjust the features even the input images are not spatially aligned with the canonical tri-plane space. Additionally, similar to [96, 26], we include a cross-attention layer between each self-attention outputs of the encoder and denoiser to strengthen the conditional modeling. On the other hand, for both the *Shape-to-3D* and *Text-to-3D* (with CLIP) tasks, we do not train another encoder, but treating the conditioning as vectors which are linearly transformed and combined with the time-embeddings.

Training We adopt the same training scheme for all our diffusion experiments including unconditional and conditional cases, which uses AdamW [53] optimizer with a learning rate of $2e-5$ and a EMA decaying rate of 0.9999. To encourage our high-resolution denosier to learn sufficiently on noisy tri-planes, we adopt a shifted cosine schedule ($256^2 \rightarrow 64^2$) inspired by [34]. We train all models with a batch size of 32 for 500K iterations on 8 NVIDIA A100 GPUs.

Conditioning camera As pointed out in § 3.2, it is critical to train conditional diffusion models with balanced camera poses, wheres the camera viewpoints from natural images (e.g., FFHQ, AFHQ) are typically biased toward the center view. Unlike training 3D GANs where matching the camera distribution is important for learning the 3D space, we found it crucial to have an unbiased input camera distribution when the 3D space is already learned. Otherwise, the performance of conditional generation degenerates heavily when the input image is not center-aligned. Therefore, for human and cat faces, we re-sample the input cameras which looks at the origin and distributes uniformly. To simulate errors in camera prediction, we augment the intrinsic matrix (focal length, f_x, f_y) with random Gaussian noises. We do not perform resampling and directly use the training set cameras for ShapeNet as it already covers all viewpoints uniformly.

Sampling Due to the requirements of proper score function $\ell(.,.)$, we only explored guided diffusion for 3D inversion and supper-resolution, while for the remaining tasks, we use the standard sampling strategy. No classifier-free guidance [32] is applied. By default, the standard ancestral sampling [31] takes 250 denoising steps for all of the experiments. For 3D inversion, we choose $\ell(.,.)$ to be VGG loss [101] with $w_t = 7e5 \cdot \sigma_t$ in Eq. (7). We notice that it is essential to use a large decreasing weight to take effective guidance. For supper-resolution tasks, we use exactly the same objective for guidance, while the loss is computed after down-sampling the rendered image into the input resolution. For cases using Langevin correction, we additionally apply 10 correction steps as described in Eq. (8) where $\delta = 0.25$. We only add Langevin steps for the first 50 denoising steps to save computational cost. The Langevin correction steps are particularly useful for unconditional models.

B.3. Application Details

Image-to-3D Inversion In this task, we independently and randomly select 1,000 images from both the FFHQ and AFHQ datasets, with the results presented in the main paper. To enhance the experimental rigor, we additionally choose 1,000 random images from the test set of CelebA-HQ dataset. We follow the EG3D methodology to re-crop the face and estimate the camera pose for enhanced processing. The results of CelebA-HQ are presented in Table 3. We select 5 camera poses with yaw angles of -35°, -17°, 0°, 17°, and 35°, and a roll angle of 0° to generate novel view images. The generated images are employed to compute the Fréchet Inception Distance (nvFID) to the original dataset and the ID metric (nvID) in relation to the input image.

Seg-to-3D Following a recent work (Pix2Pix3D [19]), in the Seg2Face process, we randomly select 500 images from the CelebA-HQ dataset, accompanied by their segmentation maps, and generate 10 images per input label using different random seeds. Subsequently, we predict the segmentation map for each generated image using a pretrained face-parsing network [99].In the Seg2Cat task, we employ a similar setting. The main distinction lies in the segmentation prediction process. We use the labels from Pix2Pix3D to train the parsing network and subsequently apply it to predict labels from the generated images. We evaluate the performance by calculating the mean Intersection over Union (MIOU) and average pixel accuracy (MPA) between the input labels and the predicted labels from the generated images. The Fréchet Inception Distance (FID) is computed between the generated images and all images in the CelebAHQ dataset. Single Generation Diversity (SG Diversity) is obtained by measuring the LPIPS metric between each pair of generated images, given a single conditional input.

Edge-to-3D We extract the edges for all datasets using informative drawing [9] [§].

Shape-to-3D We employ the FLAME template model [50] to represent facial shapes and utilize DECA [21] for extracting the corresponding FLAME parameters.

Text-to-3D For this task, we utilize CLIP [69] to extract image and text features. During the training phase, we employ the image features, while in the testing phase, we directly use the text features. While it is commonly known that the text and image spaces of CLIP are not fully aligned [70], we find the conditioning is effective as long as both features are normalized before diffusion.

[§] https://github.com/carolineec/informative-drawings.git

Table 3: Quantitative comparison on inversion.

		CelebA-HQ					
		PSNR ↑	SSIM ↑	LPIPS↓	ID↑	nvFID ↑	nvID↑
Opt.	$\mathcal{W}$	14.98	0.65	0.42	0.54	60.67	0.50
	$\mathcal{W}+$	16.62	0.71	0.34	0.74	51.23	0.66
	Tri.	17.52	0.76	**0.12**	**0.92**	185.6	0.50
Pred.	$\mathcal{W}$	14.55	0.59	0.54	0.28	68.66	0.26
Ours		**21.86**	**0.78**	0.26	0.82	**27.76**	**0.72**

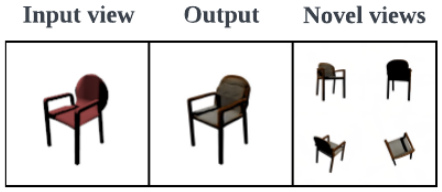


Figure 10: One failure case for conditional generation tasks on ShapeNet Chairs. While Control3Diff is always able to generate high-fidelity 3D objects, it sometimes fails to recover the texture information from the input view even with guided diffusion.

B.4. Baseline Details

GAN Inversion Our primary focus is to compare our approach with prevalent 2D GAN inversion methods, such as the direct optimization scheme introduced by [43], which inverts real images into the $\mathcal{W}$ space. Additionally, we examine a related method that extends to the $\mathcal{W}+$ space [1] and directly optimizes the tri-plane, denoted as *Tri.*. The implementation is based on EG3D-projector [¶]. We initialize all methods with the average w derived from the dataset. For the optimization process, we employ the LPIPS loss [102] and utilize the Adam optimizer [45], conducting 400 optimization steps for each image. Additionally, we utilize the encoder proposed by [47] to directly estimate the w values from images. We employ their pretrained model.

Pix2Pix3D [19] We directly utilize the pretrained checkpoints provided by authors [‖].

Pix2NeRF [8] We utilize the values provided by the authors for our analysis. However, due to the absence of released models and quantitative results, our comparison is limited to the ShapeNet chair dataset.

C. Additional Quantitative Results

Inversion on ShapeNet We include additional quantitative results for ShapeNet Cars & Chairs in Table 4. For both cases, we follow the standard evaluation protocol which takes a fixed input view (typically view 64) as input control, and render from all other cameras. Evaluation is conducted on the test sets. As the results shown in Table 4, while the proposed approach significantly improves over the existing 3D-GAN inversion baselines, and achieves high scores on perceptual scores such as LPIPS and FID, it has a clear gap compared to PixelNeRF in term of PSNR. The primary reason for this discrepancy is that PixelNeRF utilizes multi-view supervision during training, whereas our method relies solely on single-view information. Consequently, PixelNeRF can achieve improved performance in certain aspects. In contrast, our GAN-based approach demonstrates both enhanced 3D consistency and sharper outputs, which contribute to the lower FID and LPIPS scores.

Additional Results on CelebA-HQ To fully validate the generality of the proposed method, we conduct additional 3D inversion experiments on out-of-distribution (OOD) face data. As shown in Table 3, we directly apply the model trained from the FFHQ tri-plane space onto CelebA-HQ, and report the single-view inversion performance. Although tested OOD, the proposed Control3Diff performs stably and achieves larger gains against standard inversion baselines.

D. Additional Qualitative Results

3D Inversion & SR We show additional qualitative results of Control3Diff across datasets for both the 3D inversion (Figs. 11 to 13) and super-resolution (Fig. 14) applications.

Seg-to-3D Editing Fig. 16 presents an application of our method which supports progressive 3D editing based on 2D segmentation maps.

Text-to-3D Editing The conditional diffusion of Control3Diff also supports interactive editing given text prompt, as demonstrated in Fig. 17.

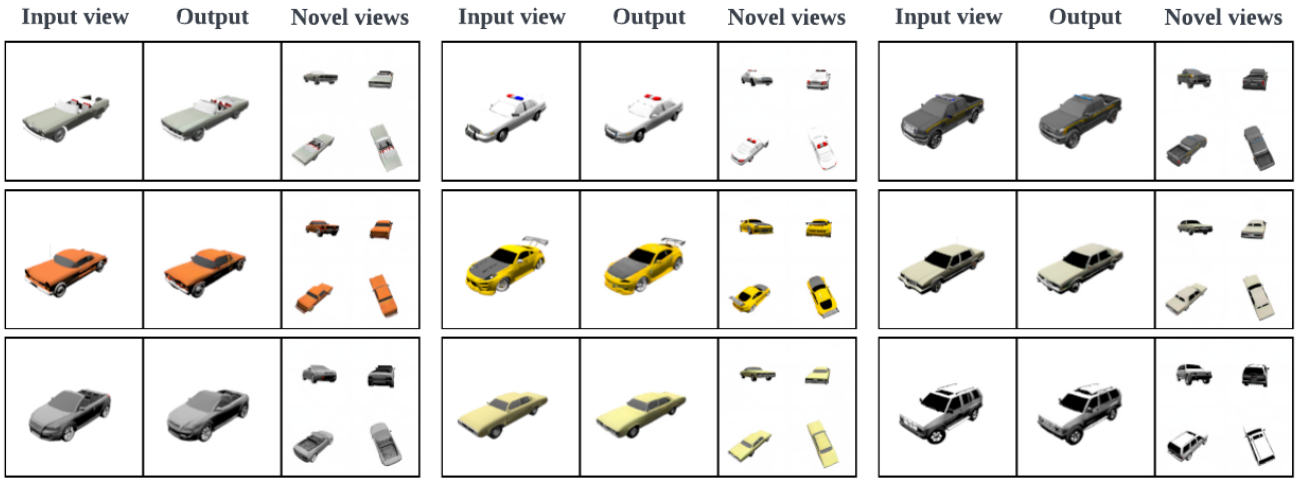
Shape-to-3D Fig. 15 presents qualitative results on this task, demonstrating that the generated images can semantically ensure the preservation of identity. However, the color exhibits constant fluctuations. The current control mechanisms are unable to effectively disentangle factors such as lighting.

E. Limitations and Future Work

Our method has two major limitations. First, while learning from the latent space of GANs allows us effectively learn controllable diffusion models for 3D, it also brings the drawbacks that GANs commonly have. For instance, a common artifact of the adversarial training is that the learned space

[¶] https://github.com/oneThousand1000/EG3D-projector

[‖] https://github.com/dunbar12138/pix2pix3D



(a) ShapeNet Cars

(b) ShapeNet Chairs

Figure 11: Qualitative results on *3D inversion* for ShapeNet Cars and Chairs.

Table 4: Quantitative comparison on single-image view synthesis on ShapeNet. *Models have multi-view supervision during training, while our methods including standard optimization-based 3D GAN-inversion baselines are trained with single-view information only.

(a) ShapeNet-Cars

	ShapeNet Cars			
	PSNR ↑	SSIM ↑	LPIPS↓	FID ↓
PixelNeRF [98]*	**23.17**	**0.89**	0.146	59.24
3DiM [96]*	21.01	0.57	–	8.99
Opt. $\mathcal{W}$	17.89	0.85	0.124	33.15
Opt. $\mathcal{W}+$	19.23	0.86	0.106	17.95
Opt. *Tri.*	14.85	0.63	0.461	319.8
Ours	21.13	**0.89**	**0.090**	**8.86**

(b) ShapeNet-Chairs

	ShapeNet Chair			
	PSNR ↑	SSIM ↑	LPIPS↓	FID ↓
PixelNeRF [98]*	**23.72**	**0.90**	0.128	38.49
3DiM [96]*	17.05	0.53	–	**6.57**
Pix2NeRF [8]	18.14	0.84	-	14.31
Opt. $\mathcal{W}$	18.28	0.86	0.110	10.96
Opt. $\mathcal{W}+$	19.30	0.87	0.099	12.70
Opt. *Tri.*	14.11	0.64	0.412	237.4
Ours	20.16	0.89	**0.090**	9.76

typically has mode collapse, which in turn affects the 3D diffusion learning that it may not cover full data space. In our experiments, we particularly noticed this collapsing effect on synthetic datasets with complex geometries such as ShapeNet (see Fig. 10). As the future work, this issue can be potentially eased by jointly training the diffusion prior

Figure 12: Qualitative results on *3D inversion* for AFHQ-cat. The input images are randomly sampled from the AFHQ training set.

with the 3D-GAN, and including additional image reconstruction loss. Moreover, comparing to pure encoder-based approaches [51], the iterative nature of the diffusion models generally has a slower generation process. However, our methods can be easily integrated with existing works for speed-up diffusion models [79]. We leave this exploration as future work.

Figure 13: Qualitative results on *3D inversion* for FFHQ. **Due to concerns about individual consent**, all the input faces are synthesized and manually selected from a pre-trained StyleGAN3 [40] checkpoint. We perform exactly the same pre-processing procedure as EG3D [11] over these synthetic images, which re-centers the faces and estimates the camera positions.

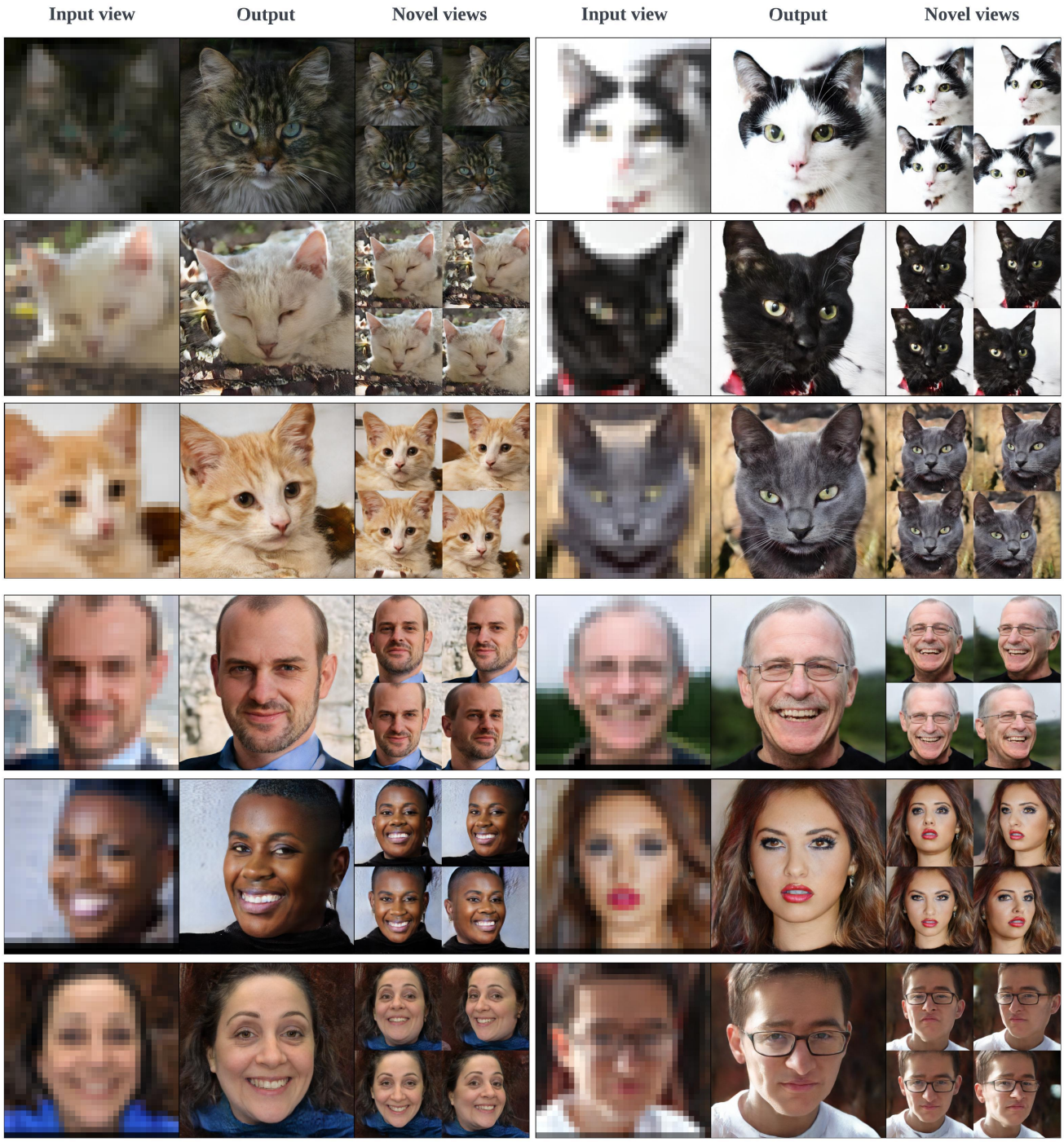


Figure 14: Qualitative results on *3D super-resolution* tasks for AFHQ-cat and FFHQ.

Figure 15: Qualitative results on *Shape-to-3D* for FFHQ. These images can semantically ensure the preservation of identity; however, the color exhibits constant fluctuations. The current control mechanisms are unable to effectively disentangle factors such as lighting.

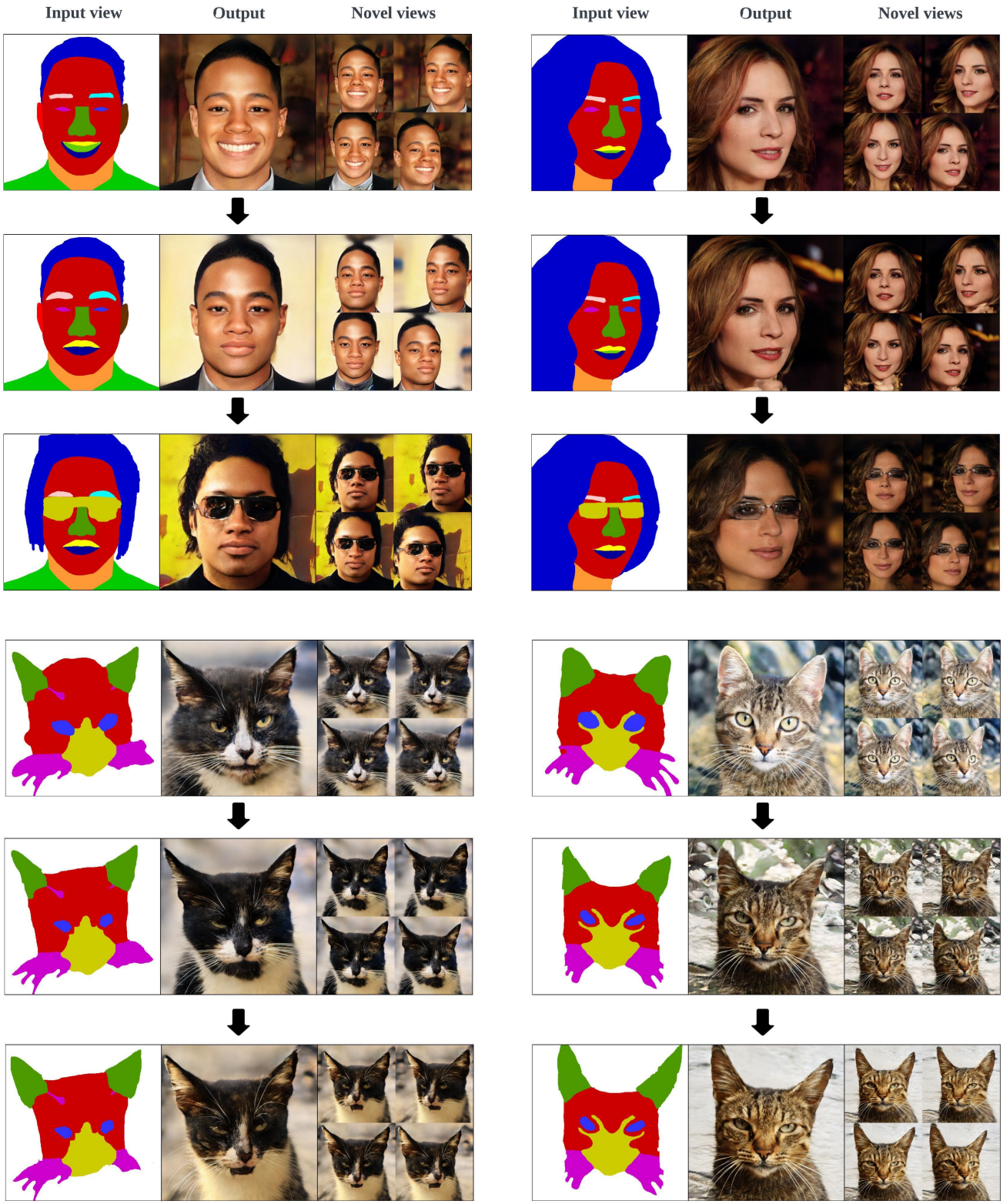


Figure 16: Progressive editing of *Seg-to-3D* synthesis. The input seg-maps are interactively edited. To achieve that, we fix the initial tri-plane noise and use DDIM [88] to obtain diffusion samples.

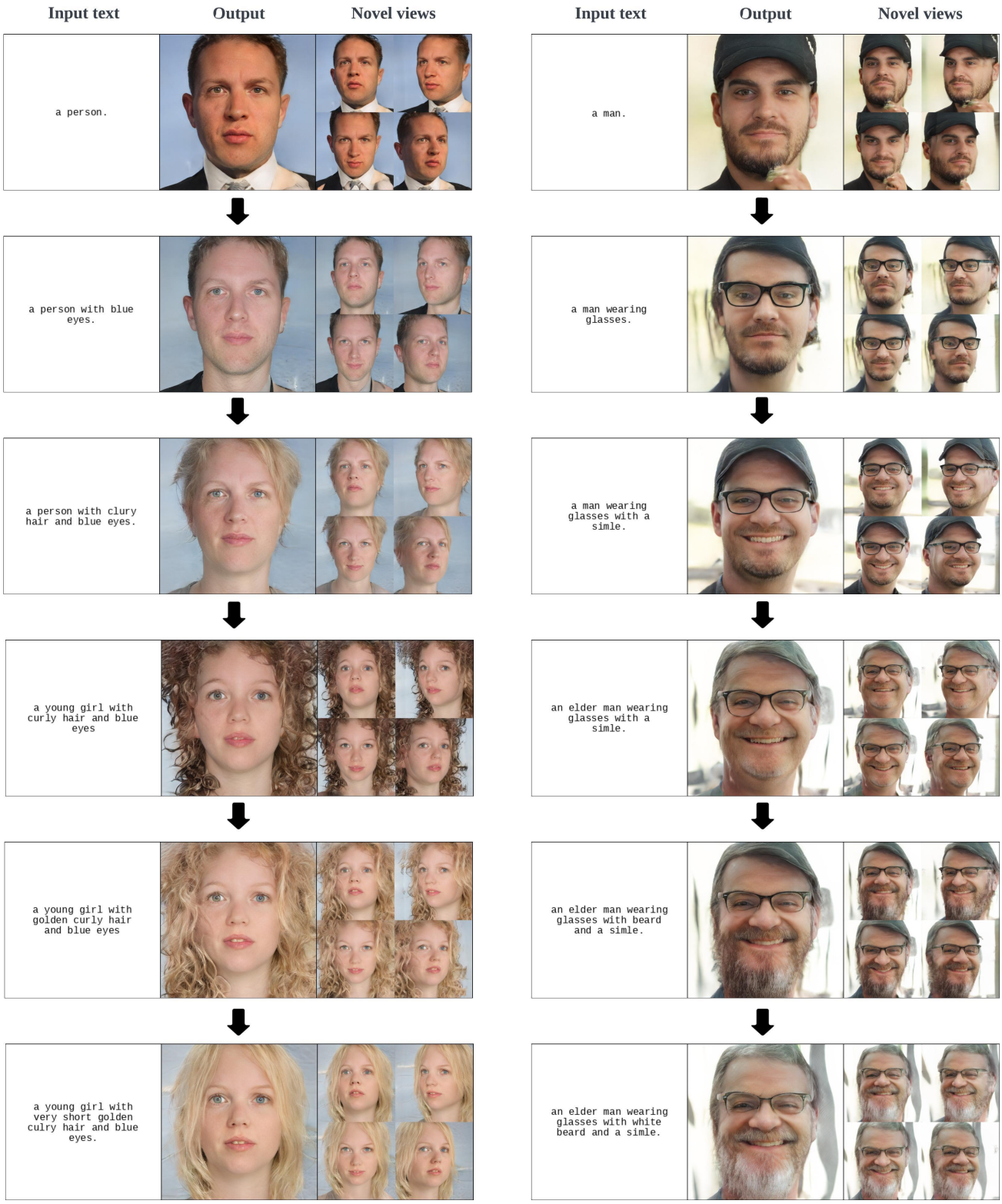


Figure 17: Progressive editing of *Text-to-3D* synthesis. The text prompts will be first transformed to normalized CLIP embeddings, which the diffusion model directly condition on. To achieve that, we fix the initial tri-plane noise and use DDIM [88] to obtain diffusion samples.