3DDesigner: Towards Photorealistic 3D Object Generation and Editing with Text-guided Diffusion Models

Gang Li*†
Institute of Software, Chinese Academy of Sciences
University of Chinese Academy of Sciences
ucasligang@gmail.com

Heliang Zheng*
JD Explore Academy
zhengheliang@jd.com

Chaoyue Wang
JD Explore Academy
chaoyue.wang@outlook.com

Chang Li
JD Explore Academy
spacegoing@gmail.com

Changwen Zheng
Institute of Software, Chinese Academy of Sciences
changwen@iscas.ac.cn

Dacheng Tao
JD Explore Academy
dacheng.tao@gmail.com

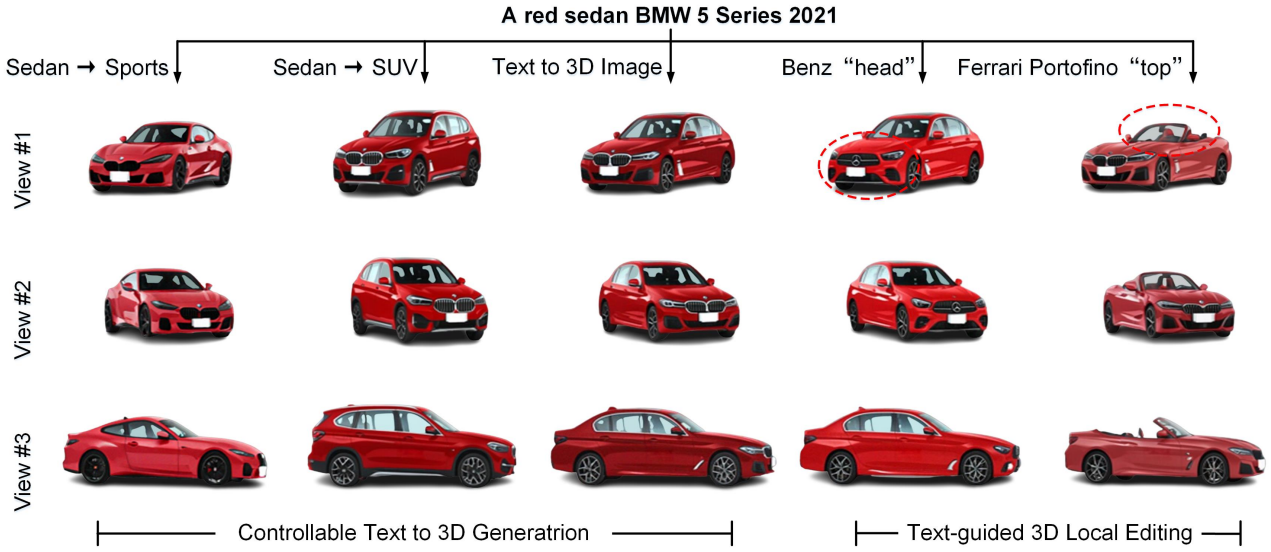

Figure 1. "3DDisigner" – Text guided 3D object generation and editing. Given a text, *e.g.,* "A red sedan BMW 5 series 2021", our method can 1) generate the corresponding 3D images, 2) create "SUV" and "Sports" 3D counterparts, and 3) support text-guided 3D local editing.

Abstract

*Text-guided diffusion models have shown superior performance in image/video generation and editing. While few explorations have been performed in 3D scenarios. In this paper, we discuss three fundamental and interesting problems on this topic. First, we equip text-guided diffusion models to achieve **3D-consistent generation**. Specifically, we integrate a NeRF-like neural field to generate low-resolution coarse results for a given camera view. Such results can provide 3D priors as condition information for the following diffusion process. During denoising diffusion, we further enhance the 3D consistency by modeling cross-view correspondences with a novel two-stream (corresponding to two different views) asynchronous diffusion process. Second, we study **3D local editing** and propose a two-step solution that can generate 360° manipulated results by editing an object from a single view. Step 1, we propose to perform 2D local editing by blending the predicted noises. Step 2, we conduct a noise-to-text inversion process that maps 2D blended noises into the view-independent text embedding space. Once the corresponding text embedding is obtained, 360° images can be generated. Last but not least, we extend our model to perform **one-shot novel view synthesis** by fine-tuning on a single image, firstly showing the potential of leveraging text guidance for novel view synthesis. Extensive experiments and various applications show the prowess of our 3DDesigner. Project page is available at https://3ddesigner-diffusion.github.io/.*

*Equal contribution.
†Work done during an internship at JD Explore Academy.

1. Introduction

Recently, text-guided diffusion models have revolutionized the field of image synthesis, showing excellent performance on high-fidelity, diverse and controllable generation [18, 22, 28]. Based on these models, various tools have been developed to empower humans to create rich and fantastical visual content. "AI artists" emerge, and imaginative paintings can be completed even in a few seconds. To further enable iterative refinement and fine-grained control, plenty of work has been proposed for image editing [1,5,12,13,15,27], making it possible to realize complex (*e.g.,* non-rigid) text-guided semantic edits by a simple text prompt. Moreover, text-guided diffusion models are also extended to complete video synthesis tasks [8, 10, 31, 35], and impressive results with the new state-of-the-art performance have been achieved on various dimensions, e.g, data efficiency, high-definition, video length, etc.

Real-world objects are 3D, and it usually requires multi-view information to represent every detail of an object. Meanwhile, creating a 3D asset requires very professional software and skills and may cost times of labor than creating its 2D counterpart. Thus a tool capable of 3D object generation would be critical for real-world applications. However, few explorations on text-guided diffusion models have been performed in 3D scenarios. CLIP-NeRF [36] integrate a CLIP [21] model to guide the rendering of a Neural Radiance Field (NeRF) [16], while without the generation capability of diffusion models, CLIP-NeRF cannot achieve flexible control and local editing. 3DiM [37] firstly makes a diffusion model work well for 3D novel view synthesis, while without text guidance, it is hard to perform novel object generation and manipulation. Dreamfusion [20] creatively adopt text-guided diffusion models as supervision to train a NeRF, but such a solution is not efficient, *i.e.,* an optimized model can only generate one specific object.

In this paper, we study text-guided diffusion models for 3D object generation and manipulation (Figure 1). We originally discuss three fundamental and interesting problems. The first, and also the most important one, is to achieve **3D-consistent generation**. Our proposed model consists of a NeRF-based condition module and a two-stream asynchronous diffusion module. The NeRF-based condition module takes camera views and coarse text guidance (*e.g.,* "A red sedan BMW") and generates low-resolution coarse results for the given camera views as inputs. Such results can provide 3D priors as condition information for the following diffusion process. The two-stream asynchronous diffusion module is designed to further enhance the 3D consistency by modeling cross-view correspondences. Specifically, we jointly denoise two noised images of an object with different views (one view for each stream), where the cross-view feature interactions can encourage the two streams to generate images that are consistent with each other. During sampling, one of the two streams loads previously generated views to guide the generation of the current view. In this way, 360° consistent results can be obtained.

Second, we study **3D local editing**. To the best of our knowledge, we are the first to perform 360° manipulated results by editing an object from a single view. We achieve this in two steps. Step 1, 2D local editing. We propose a "noise blending" pipeline to edit a region of a generated image. Specifically, in each sampling step, we can obtain the predicted noise under the guidance of a given text, and we replace a specific region of this noise with the noise predicted by the target text. Thus, for example, a BMW series 5 car "body" with a Benz Class-E car "head" can be obtained. Step 2, we design a noise-to-text inversion process that maps 2D blended noises into the view-independent text embedding space. Once the corresponding text embedding is obtained, 360° images can be generated.

Last but not least, we find that our model can also be easily extended to perform **one-shot novel view synthesis** by simply fine-tuning on a single image. Impressive results show the potential of leveraging text guidance for novel view synthesis, and we hope our work can provide insights for the novel view synthesis community.

2. Related Work

Text-guided diffusion models. Diffusion models [9,32,34] have made tremendous progress recently, showing superiority on stable training, high-fidelity image/video synthesis, non-trivial semantic editing, multi-modality fusion, etc [1, 4, 5, 12, 18, 22, 26, 31, 35]. Meanwhile, billions of (text, image) pairs [29] enable the recent breakthroughs in text-to-image synthesis [23,24,26,28,41]. Thus not surprisingly, text-guided diffusion models have shown impressive performance and attracted lots of attention. In this work, we further explore the performance of text-guided diffusion models for 3D generation.

Once a text-guided diffusion model is trained, it can be further extended to perform image editing. GLIDE [18] formulate an in-painting task, where the text can guide the generation of the masked region. Blend Diffusion [1] and Repaint [15] spatially blend noised versions of the input image with the local text-guided diffusion latent at a progression of noise levels. Textual inversion [5] and DreamBooth [27] propose the concept of "personalizing generation", which inverts an object to a special text token. Imagic [12] propose a fine-tuning technique that enables various text-based semantic edits on a single real input image. In this work, we extend noisy image blending to noise blending and further conduct noise-to-text inversion for 3D local editing.

3D novel view synthesis. Since NeRF [16] was firstly proposed to learn implicit 3D representations for the task of novel view synthesis, plenty of variants have been proposed to improve the efficiency [17, 42], generalization [3, 11, 25,

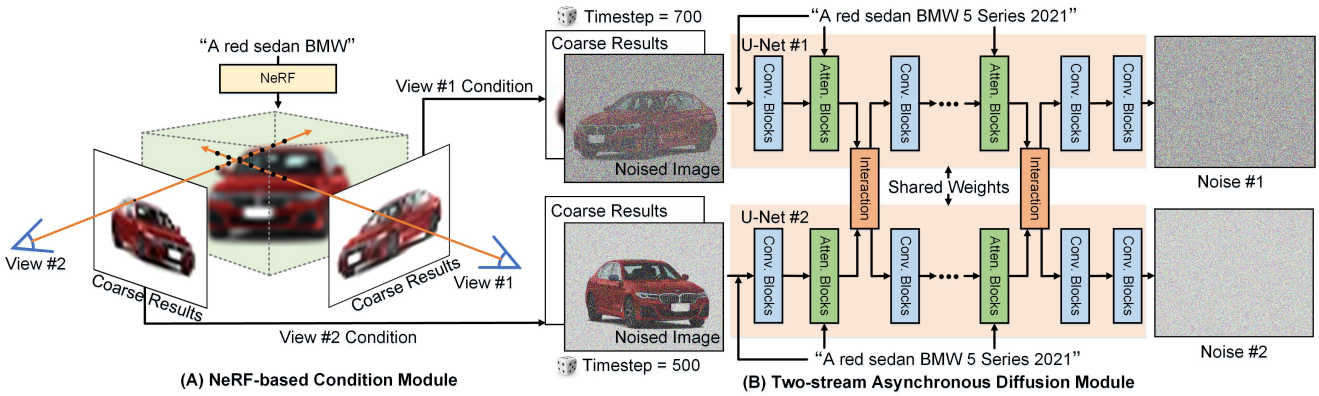

Figure 2. An illustration of our framework for text-guided 3D-consistent generation (training phase). (A) NeRF-based Condition Module, which takes <one coarse text, two camera views> pairs as inputs and generates low-resolution coarse results. The coarse results are resized and concatenated with noised images to provide conditions for denoising. (B) Two-stream Asynchronous Diffusion Module, which takes <one full text, two coarse results, two timesteps, two noised images> quadruples as inputs and predicts the added noises. Each stream is a vanilla text-guided diffusion model except for the feature interaction module after each attention block. Note that the timesteps are randomly generated and the parameters of these two streams are shared.

30, 40], controllability [14, 36, 38], etc. For example, CLIP-NeRF [36] integrates CLIP [21] with NeRF and makes the colors and shapes can be controlled by text. Some recent literature also proposes to get rid of explicit geometric inductive biases (*e.g.,* introduced by volume rendering). 3DiM [37] firstly makes a diffusion model work well for 3D novel view synthesis. Moreover, Dreamfusion [20] creatively adopt text-guided diffusion models as supervision to train a NeRF. Although GAUDI [2] also integrates NeRF with diffusion models, the diffusion model here is only used to learn a controllable latent code, preventing their model from performing local editing. Compared to these works, we leverage text-guided diffusion models to achieve more controllable and editable 3D generation.

3. Method

In this section, we introduce the details of our proposed 3DDesigner, including 1) 3D-consistent generation, *i.e.,* learning a text-guided diffusion model that can generate images from different camera views with consistent content; 2) 3D local editing, *i.e.,* generating 360° manipulated results by editing an object from a single view; and 3) one-shot novel view synthesis, *i.e.,* synthesising 360° results based on a single image.

3.1. 3D-consistent generation

Figure 2 is an illustration of our framework for text-guided 3D-consistent generation (training phase). Specifically, we integrate a NeRF-like neural field to generate 3D-consistent coarse results, which are leveraged in the following denoising diffusion process in the form of image conditions. For the denoising diffusion process, we propose to jointly denoise two noised images of an object with different views. So that the cross-view feature interactions can encourage the two streams to generate images that are consistent with each other. For now, the two views' consistency can be achieved. To further obtain 360° consistent results, we adopt an autoregressive sampling strategy [37], where one stream loads previously generated views to guide the other stream to generate the current view.

NeRF-based condition module. The NeRF-based condition module takes <coarse text, camera view> pairs as inputs and generates coarse results. Note that we do not use the full-text guidance here to avoid suppressing the creative ability of the following denoising diffusion module (*i.e.,* making it a simple super-resolution module).

Learning MLP. We first calculate the 3D location and view direction of each camera ray according to the given camera view. The camera ray can be denoted as $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where $\mathbf{o}$ and $\mathbf{d}$ are the origin and direction of the ray, respectively. We use a multiple layer perceptual (MLP) to predict the density $\sigma(t)$ and RGB color $\mathbf{c}(t)$ of a given point in the NeRF implicit representation:

$$\sigma(t), \mathbf{c}(t) = MLP(\mathbf{r}(t), \mathbf{d}, \mathbf{y}_c) \tag{1}$$

where $\mathbf{y}_c$ denotes the coarse text embedding. In particular, we concatenate the position embedded of $\mathbf{r}(t)$ and the text embedding $\mathbf{y}_c$ before getting through the MLP.

Volume rendering. We can obtain coarse results by conducting volume rendering:

$$\mathbf{x}_c(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\mathbf{c}(t)dt \tag{2}$$

where t_n and t_f are the near and far bounds, respectively. $T(t) = exp(-\int_{t_n}^{t} \sigma(s)ds)$ handles occlusion, and $\mathbf{x}_c(\mathbf{r})$ indicates the RGB value of the coarse result $\mathbf{x}_c$ that rendered

by camera ray $\mathbf{r}$. In our experiments, we follow StyleNeRF [6], 1) approximating the volume rendering process to a GPU-friendly version that supports operations on a set of rays and 2) removing the view direction condition to suppress spurious correlations and dataset bias.

Two-stream asynchronous diffusion module. The two-stream asynchronous diffusion module takes <full text, coarse results, timesteps, noised images> quadruples as inputs and predicts the added noises.

Diffusion. Given two images of an object sampled from different views, $\mathbf{x}^1$ and $\mathbf{x}^2$, the diffusion processes (*i.e.,* adding noise) for these two images are independent, which can be denoted as:

$$\begin{aligned} q(\mathbf{x}_t^1|\mathbf{x}_0^1) &:= \mathcal{N}(\mathbf{x}_t^1; \sqrt{\bar{\alpha}_t^1}\mathbf{x}_0^1, (1-\bar{\alpha}_t^1)\mathbf{I}), \\ q(\mathbf{x}_t^2|\mathbf{x}_0^2) &:= \mathcal{N}(\mathbf{x}_t^2; \sqrt{\bar{\alpha}_t^2}\mathbf{x}_0^2, (1-\bar{\alpha}_t^2)\mathbf{I}), \end{aligned} \tag{3}$$

where t is the timestep, $\bar{\alpha}_t$ can be calculated by the variance schedule, and $\mathbf{I}$ is an identity matrix.

Posterior. Now let's move on to approximate the posterior (*i.e.,* reverse process), where all of the coarse results, cross-view information, and text guidance can be leveraged:

$$\begin{aligned} &p_\theta(\mathbf{x}_{t_1-1}^1|\mathbf{x}_{t_1}^1, \mathbf{x}_c^1, \mathbf{x}_{t_2}^2, \mathbf{x}_c^2, \mathbf{y}) := \\ &\mathcal{N}(\mu_\theta(\mathbf{x}_{t_1}^1, \mathbf{x}_c^1, \mathbf{x}_{t_2}^2, \mathbf{x}_c^2, \mathbf{y}), \Sigma_\theta(\mathbf{x}_{t_1}^1, \mathbf{x}_c^1, \mathbf{x}_{t_2}^2, \mathbf{x}_c^2, \mathbf{y})), \end{aligned} \tag{4}$$

where $\mathbf{y}$ is the full text, $\mathbf{x}_c$ is the coarse result obtained by Equation 2, t_1 and t_2 are randomly generated timesteps, μ_θ and Σ_θ are models that predict the mean and variance of the Gaussian distribution, respectively.

Optimization. In practice, we follow the widely used variational lower-bound re-weighing to learn the posterior. Specifically, we generate samples $\{\mathbf{x}_{t_1}^1, \mathbf{x}_{t_2}^2\}$ by applying Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to $\{\mathbf{x}_0^1, \mathbf{x}_0^2\}$ using Equation 3. Then we train a model ϵ_θ to predict the added noise:

$$L = \mathbb{E}_{\mathbf{y}, \mathbf{x}_0^1, \mathbf{x}_0^2, t_1, t_2, \epsilon} \parallel \epsilon_\theta(\mathbf{x}_{t_1}^1, \mathbf{x}_c^1, \mathbf{x}_{t_2}^2, \mathbf{x}_c^2, \mathbf{y}) - \epsilon \parallel . \tag{5}$$

Architecture. We follow the design of multi-stream U-Nets that are widely obtained in video generation [10, 31] and 3DiM [37]. Specifically, the two-stream asynchronous diffusion module consists of two U-Nets and several feature interaction modules. The interaction module can be the temporal convolution/attention used in Make-A-Video [31] or the cross-attention used in 3DiM [37] (the latter performs better on our task). The coarse results obtained by Equation 2 are resized and concatenated with noised images obtained by Equation 3. The text embedding is integrated by attention blocks, and the timestep is specified by adding position embedding into each convolution block [9].

Sampling. The cross-view feature interaction can encourage the two streams to generate images that are consistent with each other in the sampling phase. Note that we adopt different timesteps to generate different-level noises. Our insight is to make the "cleaner" (condition) stream to guide the generation of the "noisier" (generation) stream. In particular, we adopt the DDIM [33] sampling process:

$$\mathbf{x}_{t_1-1}^1 = \sqrt{\alpha_{t_1-1}^1}\left(\frac{\mathbf{x}_{t_1}^1 - \sqrt{1-\alpha_{t_1}^1}\hat{\epsilon}_{t_1}}{\sqrt{\alpha_{t_1}^1}}\right) + \sqrt{1-\alpha_{t_1-1}^1}\hat{\epsilon}_{t_1}, \tag{6}$$

where $\hat{\epsilon}_{t_1} = \epsilon_\theta(\mathbf{x}_{t_1}^1, \mathbf{x}_c^1, \mathbf{x}_{t_1-\Delta t}^2, \mathbf{x}_c^2, \mathbf{y})$ is the predicted noise, $\Delta t \geq 0$ is a hyper-parameter to adjust the noise level of the two streams. We further follow 3DiM [37] to adopt an autoregressive sampling strategy, where the second stream loads previously generated views to guide the generation of the current view. In the training phase, two timesteps are randomly sampled, and in the sampling phase, we set Δt to be a constant (except for the first view, where $\Delta t = 0$).

3.2. 3D local editing

The model introduced above makes 3D-consistent visual content creation as simple as preparing a text. Moreover, to further achieve iterative refinement and fine-grained control, 3D local editing is required. 3D local editing takes a generated image (actually, the corresponding text), a single view mask that indicates the region to be manipulated, and a target text that guides the manipulation as inputs. The output would be manipulated images from different views. In the following, we first introduce our "noise blending" for 2D local editing. After that, we introduce a noise-to-text inversion process that enables our model to generate other views of the manipulated object. Figure 3 shows an illustration to make the editing process clearer.

Noise blending for editing a single view image. We conduct spatial blending over the predicted noises to perform local editing for a given camera view. Formally, in each sampling step, we have:

$$\begin{aligned} \hat{\epsilon}_{t-1}^b &= (1-\mathbf{M}) \odot \hat{\epsilon}_{t-1}^o + \mathbf{M} \odot \hat{\epsilon}_{t-1}^n, \\ \hat{\epsilon}_{t-1}^o &= \epsilon_\theta(\mathbf{x}_t^b, \mathbf{y}^o), \\ \hat{\epsilon}_{t-1}^n &= \epsilon_\theta(\mathbf{x}_t^b, \mathbf{y}^n), \end{aligned} \tag{7}$$

where $\hat{\epsilon}_{t-1}^b$ is the blended noise at timestep $t-1$, $\mathbf{M}$ is the user-provided mask that indicates the region to be manipulated, $\odot$ denotes the Hadamard product (*i.e.,* element-wise product), $\hat{\epsilon}_{t-1}^o$ and $\hat{\epsilon}_{t-1}^n$ are the noise predicted based on the original $\mathbf{y}^o$ and new (target) text $\mathbf{y}^n$, respectively. ϵ_θ is the trained diffusion model (we omit the coarse results and the other view for simplicity), $\mathbf{x}_t^b$ is the image that sampled by Equation 6 with the blended noise in the last timestep.

Noise-to-text inversion for generating 3D manipulated images. To generate other views of the manipulated object, our insight is to invert the blended noise into the view-independent text space. Specifically, we *fix the diffusion*

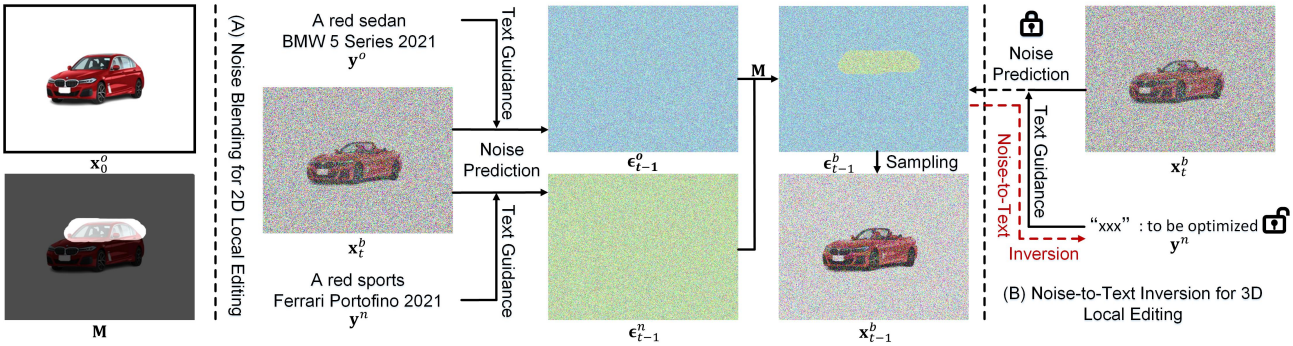

Figure 3. An illustration of 3D local editing. We propose to blend noises in each sampling step to achieve 2D local editing and conduct noise-to-text inversion to generate 3D manipulated images. The notations are explained in Sec. 3.2.

model and optimize a text embedding $\mathbf{y}^b$ to fit the blended noise in each sampling step:

$$L_{inv} = \Sigma_t \parallel \hat{\epsilon}_{t-1}^{b^*} - \epsilon_\theta{}^*(\mathbf{x}_t^{b^*}, \mathbf{y}^b) \parallel, \tag{8}$$

where $*$ indicates the fixed variables. In practice, we find that if the manipulated region is small, the optimization tends to be dominated by the original noise $\hat{\epsilon}_t^o$. Thus we extend the above Equation by adding a loss weight λ to the manipulated region:

$$\begin{aligned} L_{inv} =& \Sigma_t \| (1-\mathbf{M}) \odot (\hat{\epsilon}_{t-1}^{o^*} - \epsilon_\theta{}^*(\mathbf{x}_t^{b^*}, \mathbf{y}^b)) \\ &+ \lambda \mathbf{M} \odot (\hat{\epsilon}_{t-1}^{n^*} - \epsilon_\theta{}^*(\mathbf{x}_t^{b^*}, \mathbf{y}^b)) \|. \end{aligned} \tag{9}$$

Please refer to Equation 7 for the explanation of notations.

Discussion and comparison. *Local editing.* Our "noise blending" pipeline is similar to Blended-diffusion [1], which conducts spatially blending over the noisy image to edit a specific region of a given image. We propose to adapt the manipulation space from the noisy image space to the noise space, thus we can further invert the blended noise to a text embedding (as introduced in noise-to-text inversion) to achieve 3D local editing. *Inversion.* The high-level insight of our noise-to-text inversion process is somehow similar to Textual inversion [5], which learns a special text token with a pretrained diffusion model (fixed) to represent a specific concept (from 3-5 images). While an essential difference is that instead of following the fine-tuning paradigm, we learn the text embedding by fitting a whole DDIM sampling process, *i.e.,* learning to fit every step that generates the manipulated object. Figure 7 shows the effectiveness of our method. *An alternative solution.* An alternative solution for 3D local editing would be generating a 2D manipulated image and further synthesising other views. 1) As discussed above, with a fixed diffusion model, image-to-text is much more difficult than noise-to-text inversion. 2) Fine-tuning diffusion models (Sec. 3.3) can achieve one-shot novel view synthesis but not efficient. Since every time to fit an object, the diffusion model needs to be adjusted.

3.3. One-shot novel view synthesis

Here we conduct an extensive study on an interesting question, *i.e., How does our text-guided 3D generation model perform on the novel view synthesis task?* In particular, one-shot novel view synthesis aims to synthesise novel views of an object based on a single given image. We find our model can be easily extended to conduct one-shot novel view synthesis by fine-tuning on the given image by Equation 5. Specifically, we add noise to the given image and fine-tune our model to predict the added noise. Note that during fine-tuning, we set the two views of our model to be the same as the given view and the timesteps are also set to be the same, *i.e.,* $\Delta t = 0$. The results and further discussions can be found in Sec. 4.4.

4. Experiments

4.1. Experiment setup

In this section, we present the experimental details of our approach. We outline the datasets, implementation details, baseline methods and evaluation metrics.

Datasets. We collect a 3D car dataset, which contains around 3k car models with corresponding texts (*i.e.,* model names). Each model is rendered into 30 images from different camera views (12°/360° per view) and augmented by around 4 different colors. Some models are rendered into 60 images with two different pitch angles. We also train a classification model on CompCars [39] dataset to predict the color and type of each car model. To this end, we can obtain a 3D car dataset with 400k images with texts, *e.g.,* "A red sedan BMW 5 series 2021". We will make this dataset publicly released. For text-guided 3D generation, we create a testing set where the texts are novel compositions of the color, types, and model names. For novel view synthesis, the training and testing set contains different car models.

Implementation details. We use Pytorch [19] as our codebase and the overall training of our model costs 14 days on 8 NVIDIA A100 GPUs. The model size is around 600Mb. During inference, we adopt DDIM [33] sampling with 100

Figure 4. Our 3DDsigner can perform (A) fine-grained 3D generation, (B) semantic meaningful interpolation in the text embedding space, and (C) controllable generation that can change car types to create counterparts of real-world car models.

Table 1. Quantitative Evaluation on text-guided 3D generation. 3DiM*: our extended version to support text-guided generation.

Method	Components	Consistency		Quality	Controllability
		PSNR ↑	SSIM ↑	FID ↓	Acc ↑(%)
CLIP-NeRF [36]	NeRF	**35.10**	**0.95**	62.97	10.90
3DiM* [37]	Diffusion	29.13	0.89	*35.46*	*60.13*
Ours	NeRF+Diffusion	*30.84*	*0.93*	**18.77**	**71.33**

steps. Note that the NeRF-based condition model only needs to run one time (instead of 100), thus the inference time is around 2 times longer than traditional single-stream diffusion models. The loss weight λ in Equation 9 is experimentally set to be 3.

Baselines. Since there are few works exploring text-guided 3D object generation, we re-implement and compare two related models on our task to discuss the effectiveness and limits of NeRF and diffusion components. CLIP-NeRF [36] leverages text features and NeRF to achieve multi-modal 3D object manipulation, we remove the CLIP branch and jointly learn a text embedding and a NeRF on our dataset. 3DiM [37] is a diffusion model designed for novel view synthesis, we extend their model to a text-guided version.

Figure 5. Visualization and comparison in terms of text-guided 3D generation. Our model can generate high-fidelity images that are more controllable than CLIP-NeRF [36] and better 3D-consistent than extended 3DiM [37].

Evaluation metric. We evaluate our model from three dimensions, *i.e.,* 3D consistency, image quality, and controllability. Specifically, 1) we follow 3DiM [37] to adopt a 3D consistency scoring, which can evaluate the 3D-consistency of multi-view images without requiring ground truth images. 2) We adopt FID [7] to evaluate the image quality. Moreover, 3) we use a pre-trained classification model [39] to evaluate whether the types of the generated images are consistent with the given text.

4.2. 3D-consistent generation

Gallery. Figure 4 shows the visualization results of our 3DDesigner from three aspects, i.e., fine-grained 3D generation, semantic meaningful interpolation, and controllable generation. In particular, our model is capable of generating very fine-grained 3D objects (*e.g.*, different series of BMW as shown in Figure 4 (A)), showing the large capacity of our model and making it more practical for real-world scenarios. Moreover, we also find that our model can produce semantically meaningful interpolations as shown in Figure 4 (B), which opens a convenient gateway to create novel car models. Last but not least, our 3DDesigner shows great performance in terms of controllability, *e.g.*, we can effectively change the type of cars from sedan to sports and SUV as shown in Figure 4 (C). More visualization results shown in Appendix.

Comparison. To the best of our knowledge, we are the first to extend text-guided diffusion models to achieve photorealistic 3D-consistent generation. Thus there is no available previous work that can be compared directly. To discuss the effectiveness and limits of NeRF and diffusion compo-

Figure 6. Ablation study on the two-stream asynchronous diffusion module: (a) single-stream diffusion, (b) synchronistic timesteps, (c) clean condition, and (d) ours. The red circles locate 3D-inconsistent parts. [Best viewed with zoom in]

Table 2. Ablation study (*i.e.,* different Δt) of the asynchronous diffusion process. Although multi-stream U-Nets are also adopted in Make-A-Video [31] and 3DiM [37], they use synchronistic timesteps and clean conditions, respectively.

Method	Δt	PSNR↑	SSIM↑
Single-stream	–	28.56	0.87
Synchronistic timesteps	0	30.62	0.92
Clean condition	1000	30.68	0.91
Ours	200	**30.84**	**0.93**

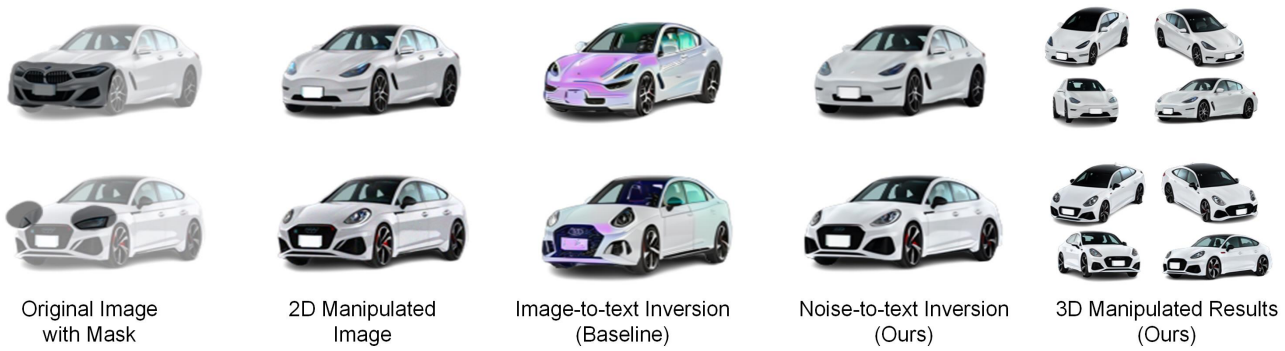

Figure 7. Visualization of 3D local editing. Noise-to-text inversion outperforms image-to-text inversion, and our model can generate 360° results given a single view mask.

Figure 8. Visualization and comparison on one-shot novel view synthesis. Leveraging text guidance can significantly improve the performance of novel view synthesis.

nents, we re-implement and extend CLIP-NeRF [36] and 3DiM [37] to our task. It can be observed from Table 1 that CLIP-NeRF [36] performs best on 3D consistency, however, lacking diffusion model makes it fail to achieve promising image quality and controllability. Adding GAN loss may improve the visual quality, while the controllability is hard to be improved in their design. Note that 3DiM originally cannot perform text-guided generation, thus we integrate 3DiM with text guidance [18]. Compared to our model, the extended 3DiM: 1) lacks of the NeRF-based condition module and 2) adopts a different two-stream diffusion process. The results show that our model can significantly improve 3D consistency, image quality, and controllability. Moreover, visualization results in Figure 5 also show the superior performance of our model.

Ablation study. The comparison to extended 3DiM [37] in Table 1 has shown the effectiveness of our NeRF-based condition model. Here we take a closer look at our proposed two-stream asynchronous diffusion module, where a cleaner condition stream is designed to guide a generation stream. The results in Table 2 show that 1) without the two-stream architecture to learn cross-view correspondence, the 3D consistency drops seriously; 2) without asynchronous timesteps, the conditioned stream is as noisy as the generation stream, leading to poor guidance; 3) conditioned on a clean image cannot well leverage the guidance due to the large gap of noise level. Figure 6 shows that our proposed two-stream asynchronous diffusion can generate more consistent results.

Table 3. Quantitative Evaluation on one-shot novel view synthesis

Method	PSNR ↑	SSIM ↑
3DiM [37]	32.53	0.93
Ours	**33.14**	**0.94**

4.3. 3D local editing

We propose to extend the 2D editing method (*i.e.,* noisy image blending [1]) to noise blending, thus the blended noise can further be inverted into a view-independent text embedding for 3D local editing. The motivation of such a design is that inverting noises in a specific DDIM sampling process is much easier than inverting an image. It can be observed from the third column of Figure 7 that obvious artifacts appear in image-to-text inversion results. The last two columns of Figure 7 show that our proposed method can generate high-fidelity 360° manipulated images with a single view mask.

4.4. One-shot novel view synthesis

Inspired by recent advances in fine-tuning text-guided diffusion models for image inversion [5, 12], we extend our text-guided 3D generation model to perform novel view synthesis. Given an object that the model has nerve seen, we fine-tune our model by adding noise to a single-view image of the object and conduct denoising. After that, we can obtain novel view images by changing the input camera pose. Table 3 and Figure 8 shows the qualitative and quantitative results, respectively. It can be observed that our model can synthesis more realistic and consistent novel view images.

5. Conclusion

In this paper, we introduced 3DDesigner, a text-guided diffusion model which is capable of generating and manipulating 3D objects. To our best knowledge, the proposed 3DDesigner is the first text-guided generative model that can perform 3D-consistent generation, 3D local editing, and one-shot novel view synthesis together. Benefitting from the proposed NeRF-based Condition Module, Two-stream Asynchronous Diffusion Module, and novel diffusion training, sampling&blending strategies, our 3DDesigner outperforms existing methods and achieves highly realistic, detailed, and controlled 3D object generation.

Limitations. As the price of highly detailed 3D object generation and manipulation, the proposed 3DDesigner requires training data that contains multi-view images of a single object. It limits the application scenarios of 3DDesigner to some extent. Due to this paper being an academic exploration, we only test 3DDesigner on a single "car&text description" dataset. Yet we believe that our 3DDesigner will be used in more and more emerging datasets accompanied by the rapid development of 3D novel view synthesis methods (*e.g.,* NeRF-based methods). In addition, we will continue focusing on relaxing the requirements of training data and improving the generation quality.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, pages 18208–18218, 2022. 2, 5, 8

[2] Miguel Angel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, et al. GAUDI: A neural architect for immersive 3D scene generation. *arXiv preprint arXiv:2207.13751*, 2022. 3

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, pages 14124–14133, 2021. 2

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 2

[5] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2, 5, 9

[6] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3D aware generator for high-resolution image synthesis. In *ICLR*, 2021. 4

[7] 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. *NeurIPS*, 30, 2017. 7

[8] 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. 2

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2, 4

[10] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *NeurIPS*, 2022. 2, 4

[11] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. GeoNeRF: Generalizing NeRF with geometry priors. In *CVPR*, pages 18365–18375, 2022. 2

[12] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022. 2, 9

[13] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, pages 2426–2435, 2022. 2

[14] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-NeRF: Editable feature volumes for scene rendering and manipulation. *arXiv preprint arXiv:2204.10850*, 2022. 3

[15] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, pages 11461–11471, 2022. 2

[16] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2

[17] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 41(4):102:1–102:15, July 2022. 2

[18] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2, 8

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch:

An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 5

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

[21] 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 *ICML*, pages 8748–8763. PMLR, 2021. 2, 3

[22] 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. 2

[23] 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. 2

[24] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831. PMLR, 2021. 2

[25] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. LOLNeRF: Learn from one look. In *CVPR*, pages 1558–1567, 2022. 2

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2

[27] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022. 2

[28] 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. 2

[29] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Theo Coombes, Cade Gordon, Aarush Katta, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: laion-5b: A new era of open large-scale multi-modal datasets, 2022. 2

[30] Prafull Sharma, Ayush Tewari, Yilun Du, Sergey Zakharov, Rares Ambrus, Adrien Gaidon, William T Freeman, Fredo Durand, Joshua B Tenenbaum, and Vincent Sitzmann. Seeing 3D objects in a single image via self-supervised static-dynamic disentanglement. *arXiv preprint arXiv:2207.11232*, 2022. 2

[31] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2, 4, 7

[32] 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. 2

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 4, 5

[34] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32, 2019. 2

[35] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022. 2

[36] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. CLIP-NeRF: Text-and-image driven manipulation of neural radiance fields. In *CVPR*, pages 3835–3844, 2022. 2, 3, 7, 8

[37] 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. 2, 3, 4, 7, 8

[38] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *ICCV*, pages 13779–13788, 2021. 3

[39] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, pages 3973–3981, 2015. 5, 7

[40] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021. 2

[41] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 2

[42] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *TOG*, 40(6):1–18, 2021. 2

Appendix

A. More visualization results.

3D-consistent generation. We show more generation results in Figure 9 by sampling images with equally spaced views. It can be observed that our 3DDesigner achieves photorealistic 3D-consistent generation. We strongly recommend readers to watch the video in our supplementary material for better visualization.

Local Editing. Figure 10 shows our proposed 3DDesigner can generate high-fidelity 360° manipulated images with a single view mask.

Novel view synthesis. Given an object that the model has nerve seen, we fine-tune our model by adding noise to a single-view image of the object and conduct denoising. After that, we can obtain novel view images by changing the input camera pose. Figure 11 shows some results on novel view synthesis. It can be observed that our model can synthesise more realistic and consistent novel view images.

B. Ablation study.

Figure 12 shows the comparison of different inputs for the NeRF-based condition module, i.e., full text and coarse text. It can be observed that full-text inputs cannot change the car types (e.g., "SUV").

"Benz Class E 2021 red sedan"--> "Benz Class E 2021 red sports"

"Havel H7 2019 black SUV"-->"Havel H7 2019 black sedan"

“Jetta va3 2019 blue sedan”--> “Jetta va3 2019 blue SUV”

“Skoda Xinrui 2019 white sedan”-->“Skoda Xinrui 2019 white SUV”

“Benz Class E 2021 red sedan”--> “Benz Class E 2021 red SUV”

Figure 9. Visualization on 3D-consistent generation.

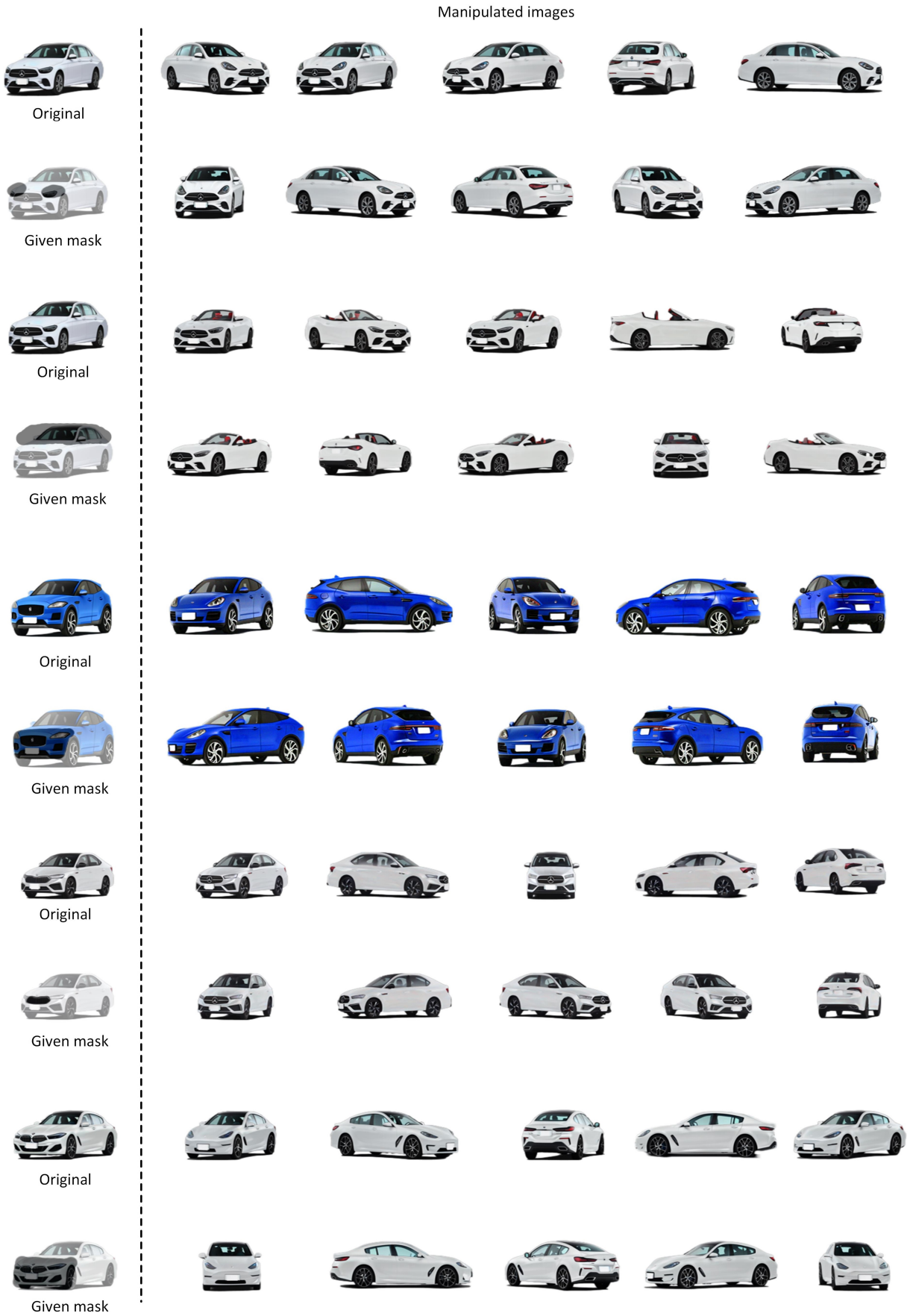

Figure 10. Visualization on text-guided 3D local editing.

Figure 11. Visualization on one-shot novel view synthesis.

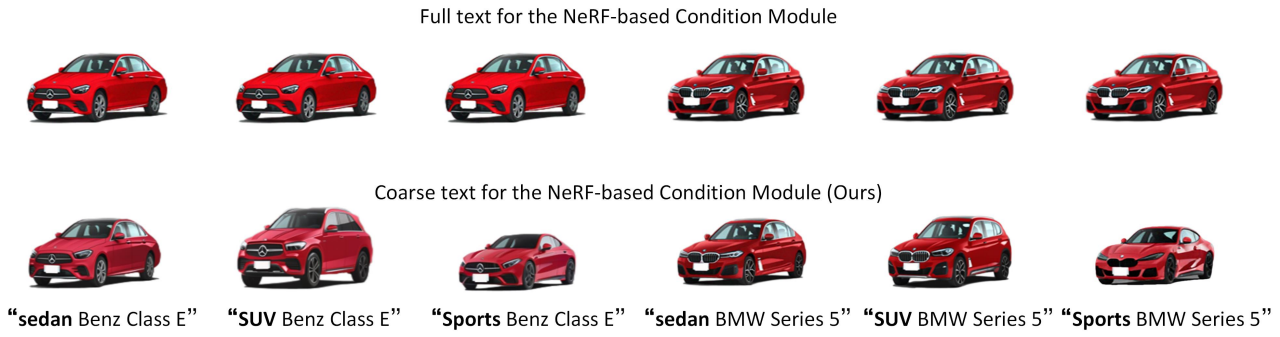

Figure 12. Ablation study on the inputs of NeRF-based condition Module with full text and coarse text.