# Free3D: Consistent Novel View Synthesis without 3D Representation

## Supplementary Material

The supplementary materials are organized as follows:

- A video to illuminate our work and the rendered videos.
- Introduction for the baseline diffusion model.
- Experiment details.
- Results for more single view NVS.

## A. Background: Diffusion Generators

In order to achieve sufficient generalization to operate in an *open-set* category setting, Free3D builds on a pre-trained 2D image generation, and specifically Stable Diffusion (SD) [51]. SD is a Latent Diffusion Model (LDM) trained on billions of text-image pairs from LAION-5B [56]. It consists of two stages. The first stage embeds the given image $x_0 \in \mathbb{R}^{H\times W\times 3}$ in a latent space $z \in \mathbb{R}^{\frac{H}{f}\times\frac{H}{f}\times c}$ through an autoencoder $\mathcal{E}: x \mapsto z$, paired with a decoder $\mathcal{D}: z \mapsto x$, which reconstructs the image ($x = \mathcal{D}\circ\mathcal{E}(x)$). The second stage uses diffusion to model the distribution $p(z|y)$ over such latent codes, where $y$ lumps any conditioning information (*e.g.*, text, image, or viewpoint). Diffusion involves a forward noising process that gradually perturbs the given latent $z_0 = z$ by adding the Gaussian noise $\epsilon$ in a Markovian fashion:

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{A.1}$$

producing a sequence $z_t, t = 1, \dots, T$, and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, $\alpha_t := 1 - \beta_t$ denote the noise strength at different steps. $\{\beta_t\}_{t=1}^{T}$ is a pre-defined variance schedule. Ultimately, $p(z_T|y)$ is approximately normal; we can thus easily sample $z_T$, and then go back to $z_0$ via the backward denoising process using the predicted noise:

$$z_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(z_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(z_t, t, y)\right) + \sigma_t\epsilon, \tag{A.2}$$

where $\epsilon_\theta$ is typically an UNet [16], and $\{\sigma_t\}_{t=1}^{T}$ is another control of noise $\epsilon$, which is also a pre-defined schedule corresponding to the schedule $\beta_t$ and introduces uncertainty for the synthesis of different views. Similar to the vanilla DDPM [24], SD uses the following training objective to optimize the UNet $\epsilon_\theta$:

$$\mathcal{L} = \mathbb{E}_{z_0, y, \epsilon\sim\mathcal{N}(0,I), t}\left[\|\epsilon - \epsilon_\theta(z_t, t, y)\|_2^2\right], \tag{A.3}$$

## B. Experiment Details

The Stable Diffusion (SD), originally trained for text-to-images generation, requires adaptation to suit image-conditional NVS tasks. Following Zero-1-to-3 [38], we utilize the image-to-image Stable Diffusion checkpoints[3]. Our baseline code is built upon the Zero-1-to-3 [38][4]. Hyper-parameters are configured in accordance with the default settings of the baseline code. The *ray conditioning normalisation (RCN)* is incorporated into each ResNet block within the diffusion Unet $\epsilon_\theta$, while the *pseudo-3D cross-attention* is introduced after the original CLIP-conditional cross-attention layer (as illustrated in Fig. 2).

Instead of directly providing $\boldsymbol{r}_{uv} = (\boldsymbol{o}\times\boldsymbol{d}_{uv}, \boldsymbol{d}_{uv})$ to the network for modulating the features, we embed them into higher-dimensional features, following the approach of NeRF [43] and LFN [60]. In particular, we employ the element-wise mapping $\boldsymbol{r} \mapsto [\boldsymbol{r}, \sin(f_1\pi\boldsymbol{r}), \cos(f_1\pi\boldsymbol{r}), \cdots, \sin(f_K\pi\boldsymbol{r}), \cos(f_K\pi\boldsymbol{r})]$, where $K$ is the number of Fourier bands, and $f_k$ is equally spaced to the sampling rate. In all experiments, $K$ is set as 6, leading to $78 = 2\times 3\times K_{\boldsymbol{o}} + 3 + 2\times 3\times K_{\boldsymbol{d}} + 3 = (6+6)\times 6 + 6$ dimensional features (as depicted in Fig. 2(a)).

**Training Details.** Our model was trained on $4\times$ A40 48GB GPUs in two stages: **i)** We first finetuned the model with RCN, utilizing a batch size of 256 for 3 days on random camera viewpoints, enhancing the pose accuracy for target views. **ii)** Subsequently, the pseudo-3D cross-attention was finetuned on the 4 nearest views, employing a batch size of 192 for 2 days. In the second stage, different views from one instance were perturbed by adding noise from the same time step $t$.

In an alternative approach during the first stage, we initially attempted to jointly train the pseudo-3D cross-attention with random camera viewpoints. However, the performance is worse than the configuration $\mathbb{D}$. We believe this is because the camera viewpoints have a large gap along these random views in the rendered datasets, making it harder to calculate the similarity across these frames. In all experiments, we use AdamW with a learning rate of $10^{-5}$ for the old parameters in the original diffusion Unet $\epsilon_\theta$ and a $10\times$ larger learning rate for new parameters, namely the parameters for RCN and Pseudo-3D cross-attention.

**Inference Details.** At the testing phase, we configure the diffusion model with a sampling step set to $T = 50$. The computational time for rendering a novel view using our proposed Free3D is approximately 3 seconds, utilizing an A6000 GPU. For a fair comparison, all models are evaluated on the same A6000 GPU employing the same batch size of 4. This batch size is chosen due to the operational constraints of syndreamer [39], which can only run such a

[3] https://huggingface.co/spaces/lambdalabs/stable-diffusion-image-variations

[4] https://github.com/cvlab-columbia/zero123

small size. Additionally, we also utilize the CFG with a scale $s = 3$ to guide the rendering for each target view.

**360° Video Rendering.** To render a 360° video, we establish a circle trajectory by uniformly subdividing the azimuth $\phi$ into discrete intervals of $\frac{2\pi}{50} = 7.2°$, while the elevation $\theta$ and the distance $z$ remain fixed. For each 3D instance, we replicate the same latent variable $z_T$ over 50 frames, which can minimize temporal flickering across different views. Additionally, we also set the parameter $\sigma_t$ in Eq. (A.2) to zero, thereby further mitigating uncertainty introduced by varying noise patterns.

## C. More Visual Results

**More results on Objaverse NVS.** In Figs. C.1 and C.2, we present more visual comparisons on `Objaverse` datasets [15] that given one input image and the target viewpoint, all models render the target novel view. This is an extension of Fig. 4 in the main paper.

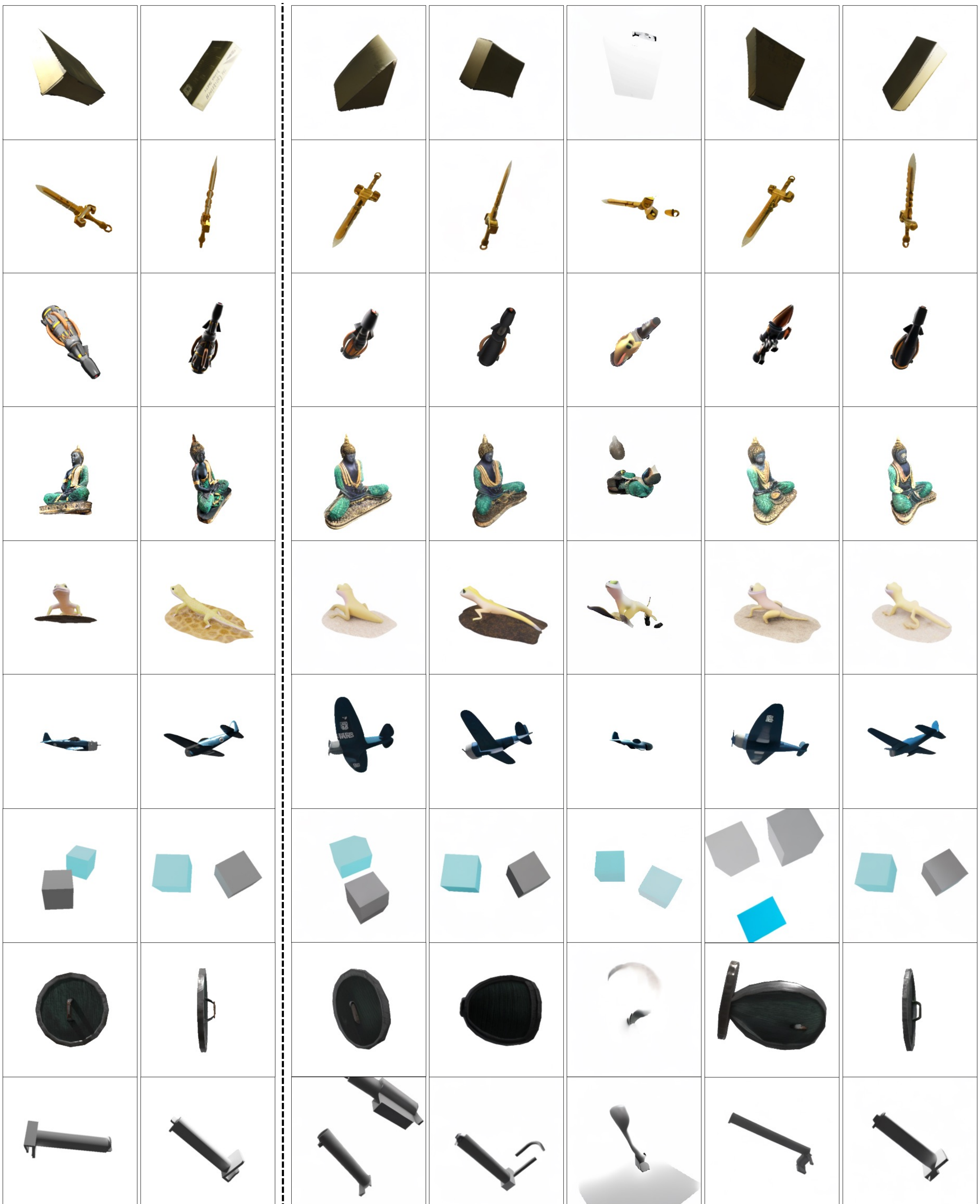
Here, all examples shown come from the corresponding `test-set` following the split, as in Zero-1-to-3 [38]. These examples are good evidence that our Free3D is suitable for *open-set* categories NVS, where it can generate semantically reasonable content with visually realistic appearances across various categories. More importantly, compared to existing state-of-the-art methods, the Free3D provides better results with a more precise pose for the target novel view. This observation suggests that the RCN is able to provide better viewpoint perception for the NVS.

**More results on OmniObject3D and GSO NVS.** In Figs. C.3 and C.4, we show additional comparison results on `OmniObject3D` [73] and `GSO` [17] datasets, respectively. This is an extension of Fig. 5 in the main paper, which demonstrates the generalizability of our Free3D on unseen datasets encompassing various categories.

As can be seen from these results, although the baseline Zero-1-to-3 [38] provides visually realistic appearances for all objects, the content is *not* always reasonable, and the pose is inaccurate in many cases. This indicates the global language token embedding with elevation $\theta$, azimuth $\phi$, and distance $z$ is *not* so precise for the network to interpret and utilize the camera viewpoints. While the Zero123-XL [14] and consistent123 [69] enhance the quality by training on a larger dataset and employing multi-view diffusion, respectively, they do *not* directly deal with the camera pose perception. In contrast, our Free3D leverages the *per-pixel* ray conditioning as well as the modulating, which significantly improves the pose perception accuracy.

**(a)** Input View **(b)** Target View **(c)** Zero123[38] **(d)** Zero123-XL[14] **(e)** SyncDreamer[39] **(f)** Consistent123[69] **(g)** Ours Free3D

Figure C.1. **Qualitative comparisons on `Objaverse` dataset.** Given the exact target pose, the proposed Free3D significantly improves the pose precision compared to existing state-of-the-art methods.



**(a)** Input View **(b)** Target View **(c)** Zero123[38] **(d)** Zero123-XL[14] **(e)** SyncDreamer[39] **(f)** Consistent123[69] **(g)** Ours Free3D

Figure C.2. **Qualitative comparisons on `Objaverse` dataset.** Given the exact target pose, the proposed Free3D significantly improves the pose precision compared to existing state-of-the-art methods.


Figure C.3. **Qualitative comparisons on `OminiObject3D` dataset.** Given the exact target pose, the proposed Free3D significantly improves the pose precision compared to existing state-of-the-art methods.


**(a)** Input View **(b)** Target View **(c)** Zero123[38] **(d)** Zero123-XL[14] **(e)** SyncDreamer[39] **(f)** Consistent123[69] **(g)** Ours Free3D

Figure C.4. **Qualitative comparisons on GSO dataset.** Given the exact target pose, the proposed Free3D significantly improves the pose precision compared to existing state-of-the-art methods.