A APPENDIX

We present our implementation details in Sec. A.1. Our training algorithm is shown in Sec. A.2. Further 3D asset generation results can be viewed in Sec. A.3, and additional comparisons to text-to-3D baseline methods are available in Sec. A.4. In Sec. A.5, we show additional image-to-3D reconstruction results and experiments with an image-guided hallucination task where we generate 3D assets that are hallucinated from the given input image (rather than reconstruction). Please refer to our video demo in the supplementary material for a comprehensive overview.

A.1 IMPLEMENTATION DETAILS

Model setup. Our approach is implemented based on a publicly available repository [2]. In this implementation, a NeRF is parameterized by a multi-layer perception (MLP), with instant-ngp (Müller et al., 2022) for positional encoding. To enhance photo-realism and enable flexible lighting modeling, instead of using Lambertian shading as employed in (Poole et al., 2022), we encode the ray direction using spherical harmonics and utilize it as an input to NeRF. Additionally, we incorporate a background network that predicts background color solely based on the ray direction. We employ a pre-trained SD model [3] as diffusion prior, as well as a pre-trained dense prediction model [4] to predict disparity maps.

Training setup. We use Adam (Kingma & Ba, 2015) with a learning rate of 10^{-2} for instant-ngp encoding, and 10^{-3} for NeRF weights. In practice, we choose total_iter as 10^4 iterations. The rendering resolution is 512×512. We employ DDIM (Song et al., 2021) with empirically chosen parameters $r = 0.25$, and $\eta = 1$ to accelerate training. We choose the hyper-parameters $\lambda_{\text{rgb}} = 0.1, \lambda_d = 0.1$, and $\lambda_{\text{zvar}} = 3$. Similar to prior work (Poole et al., 2022; Lin et al., 2023; Wang et al., 2023a), we use classifier-free guidance (Ho & Salimans, 2022) of 100 for our diffusion model.

A.2 TRAINING ALGORITHM

We present our training procedure in Algorithm 1. In step 5, either a single-step or multi-step denoising approach can be used to estimate the latent vector $\boldsymbol{z}$. Here, the multi-step denoising refers to the iterative denoising of $\hat{\boldsymbol{z}}_{\boldsymbol{t}}$, until $t = 0$.

Algorithm 1 Training Procedure

Input: A pre-trained SD Rombach et al. (2022) consisting of an encoder $\mathcal{E}$, a decoder $\mathcal{D}$, and a denoising autoencoder ϵ_ϕ; a rendering $\boldsymbol{x} = g(\theta)$; a latent vector $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x})$; a number of total training steps total_iter; range of the diffusion time steps $[t_{\max}, t_{\min}]$; a conditioning $\boldsymbol{y}$; scaling coefficients α_t and σ_t.

1: **for** iter = [0, total_iter] **do**
2: $t = t_{\max} - (t_{\max} - t_{\min})\sqrt{\frac{\text{iter}}{\text{total\_iter}}}$
3: $\boldsymbol{z}_t = \alpha_t \boldsymbol{z} + \sigma_t \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
4: Estimating noise $\hat{\boldsymbol{\epsilon}} = \epsilon_\phi(\boldsymbol{z}_{\boldsymbol{t}}; y, t)$
5: Estimating the latent vector $\hat{\boldsymbol{z}} = \frac{1}{\alpha_t}(\boldsymbol{z}_{\boldsymbol{t}} - \sigma_t \hat{\boldsymbol{\epsilon}})$ via either single- or multi-step denoising
6: Estimating the image $\hat{\boldsymbol{x}} = \mathcal{D}(\hat{\boldsymbol{z}})$
7: Compute the loss gradient $\nabla_\theta \mathcal{L}$ and update θ
8: **end for**

Return: θ

A.3 ADDITIONAL RESULTS OF TEXT-TO-3D GENERATION

We provide more generated 3D assets given text prompts in Fig. 10- 12.

[2] https://github.com/ashawkey/stable-dreamfusion/tree/main.
[3] We use the pre-trained SD in https://github.com/huggingface/diffusers.
[4] https://github.com/huggingface/transformers.

A.4 ADDITIONAL COMPARISONS TO THE BASELINE METHODS

We present additional comparisons to the baseline methods in Fig 13- 18, following the rendering settings used in ProlificDreamer (Wang et al., 2023b).

Specifically, in Fig. 13, we present results only using NeRF representation, comparing them to two baseline methods, namely ProlificDreamer (Wang et al., 2023b) and DreamFusion (Poole et al., 2022). In this case, no fine-tuning stage for 3D asset generation is applied in these baseline methods as illustrated in Fig. 13; our method allows the generation of high-fidelity details and natural colors through only a *single-stage* optimization. We observe flickering issues and improper geometries when using only the NeRF representation in ProlificDreamer (Wang et al., 2023b). In contrast, our method consistently provides view and geometry-consistent results without flickering.

In Fig. 17, we present additional visual results, comparing them to the baseline methods, including ProlificDreamer (Wang et al., 2023b), Fantasia3D Chen et al. (2023b), Magic3D (Lin et al., 2023) and DreamFusion (Poole et al., 2022). In this case, the baseline methods (Wang et al., 2023b; Lin et al., 2023) employ the full training pipeline, which includes NeRF representation followed by fine-tuning.

Additional comparisons with Fantasia3D (Chen et al., 2023b) and Magic3D (Lin et al., 2023) are shown in Fig. 14- 16, and comparisons with DreamFusion (Poole et al., 2022) in Fig. 18.

In Fig. 19, we integrate the z-variance loss into ProlificDreamer. We observe that incorporating the z-variance loss results in sharper textures. In Fig.20, we present the results of ProlificDreamer both without and with our proposed method, which includes the z-variance loss, the image-space loss, and the square root time-step annealing schedule. From the results, our method enhances the baseline approach, enabling it to generate superior renderings with detailed textures.

A.5 ADDITIONAL RESULTS OF IMAGE-TO-3D RECONSTRUCTION

In Fig. 21, we present additional image-to-3D reconstruction results. Additionally, we conduct image-guided 3D hallucination experiments. Specifically, we execute image-to-3D reconstruction at early training iterations, and then optimize the NeRF representation only using our proposed distillation loss, omitting the image reconstruction loss. We show these results in Fig. 22.

Figure 10: **Additional 3D asset generation results with the corresponding normal map given text prompts (below each object).**

Figure 11: **Additional 3D asset generation results with the corresponding normal map given text prompts (below each object).**

Figure 12: **Additional 3D asset generation results with the corresponding normal map given text prompts (below each object).**

Figure 13: **Additional visual comparisons using NeRF representation only**. We compare visually with the baseline methods, ProlificDreamer (Wang et al., 2023b) and DreamFusion (Poole et al., 2022), specifically after the first training stage. In this case, 3D assets are represented by NeRF, with no additional fine-tuning applied in the baselines.

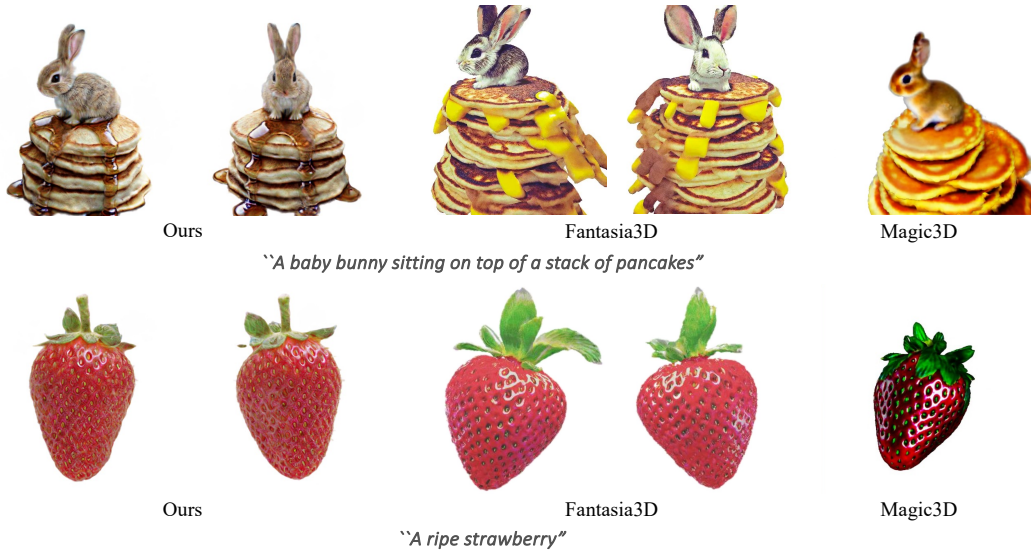


Figure 14: **Additional visual comparisons** with Fantasia3D (Chen et al., 2023b) and Magic3D (Lin et al., 2023).

Figure 15: **Additional visual comparisons** with Fantasia3D (Chen et al., 2023b)

Figure 16: **Additional visual comparisons** with Fantasia3D (Chen et al., 2023b)

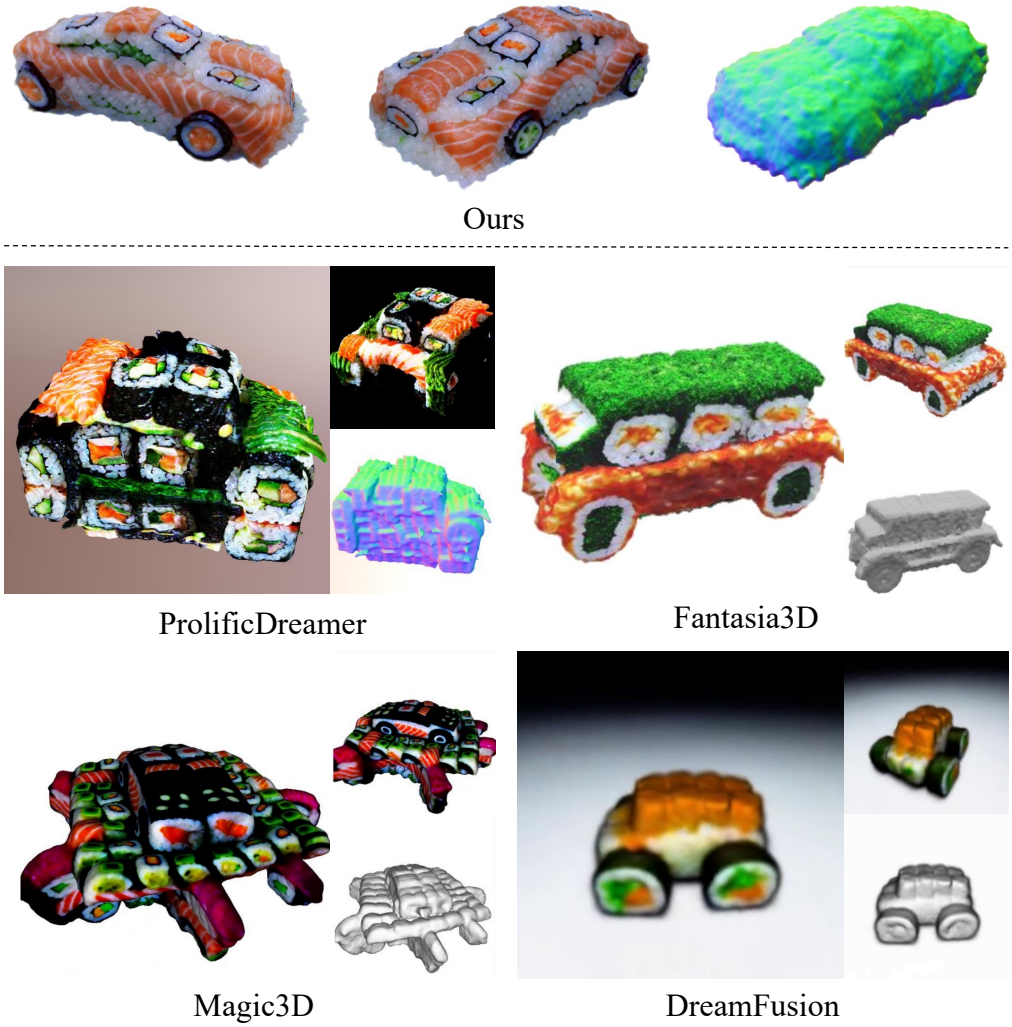
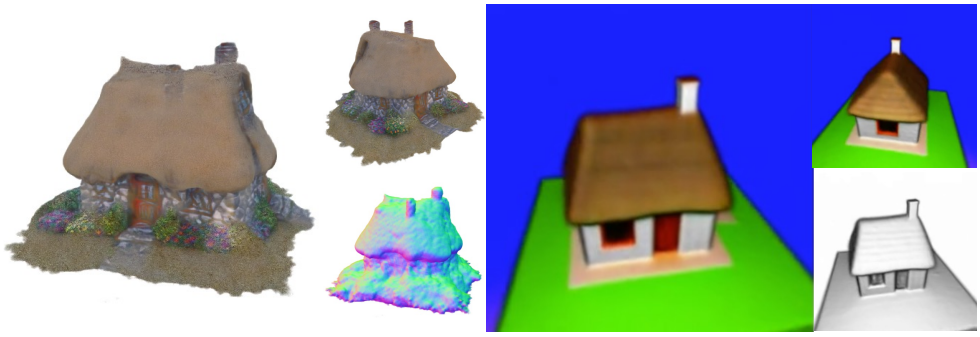


Figure 17: **Additional visual comparisons with the baseline methods**, including ProlificDreamer (Wang et al., 2023b), Fantasia3D Chen et al. (2023b), Magic3D (Lin et al., 2023) and DreamFusion (Poole et al., 2022). In this case, the baseline methods (Wang et al., 2023b; Lin et al., 2023) employ the full training pipeline, which includes NeRF representation followed by fine-tuning.

Ours DreamFusion

``A ladybug"

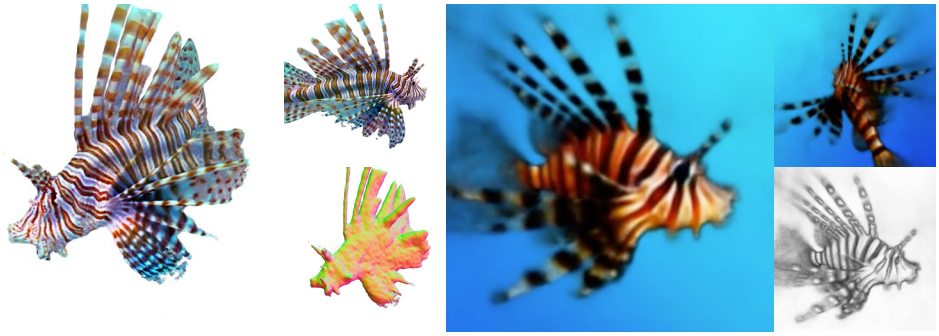


Ours DreamFusion

``A 3d model of an adorable cottage with a thatched roof"

Ours DreamFusion

``Michelangelo style statue of dog reading news on a cellphone"



Ours DreamFusion

``A lionfish"

Figure 18: **Additional visual comparisons** with DreamFusion (Poole et al., 2022).

Figure 19: **Visual results of incorporating the z-variance loss to ProlificDreamer Wang et al. (2023b) throughout the training process.** We show rendered results w/o (left) and $w/$ (right) the z-variance loss after 4K, 7K and 10K training iterations.

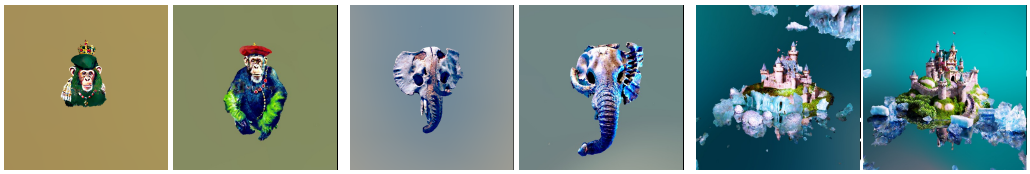


Figure 20: **The baseline results, ProlificDreamer (Wang et al., 2023b), without and with our proposed method**. This includes the z-variance loss, the image-space loss, and the square root time-step annealing schedule.

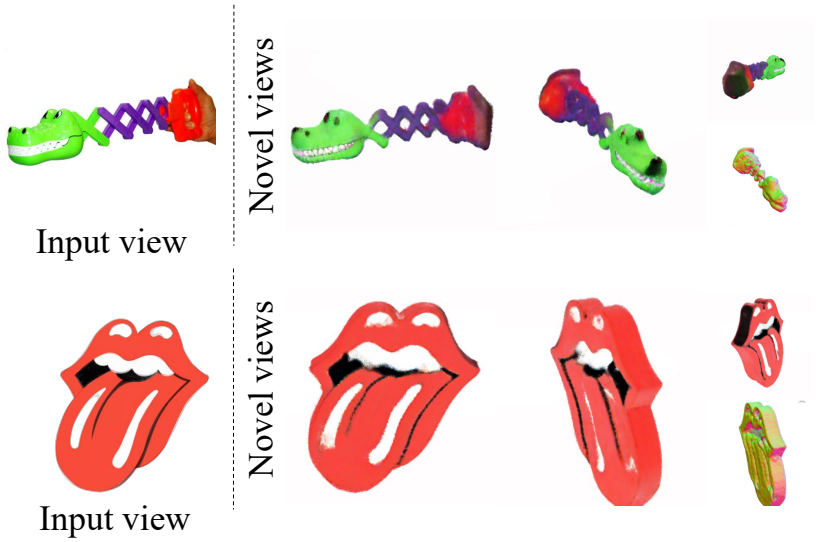


Figure 21: **Additional results of image-to-3D reconstruction.**

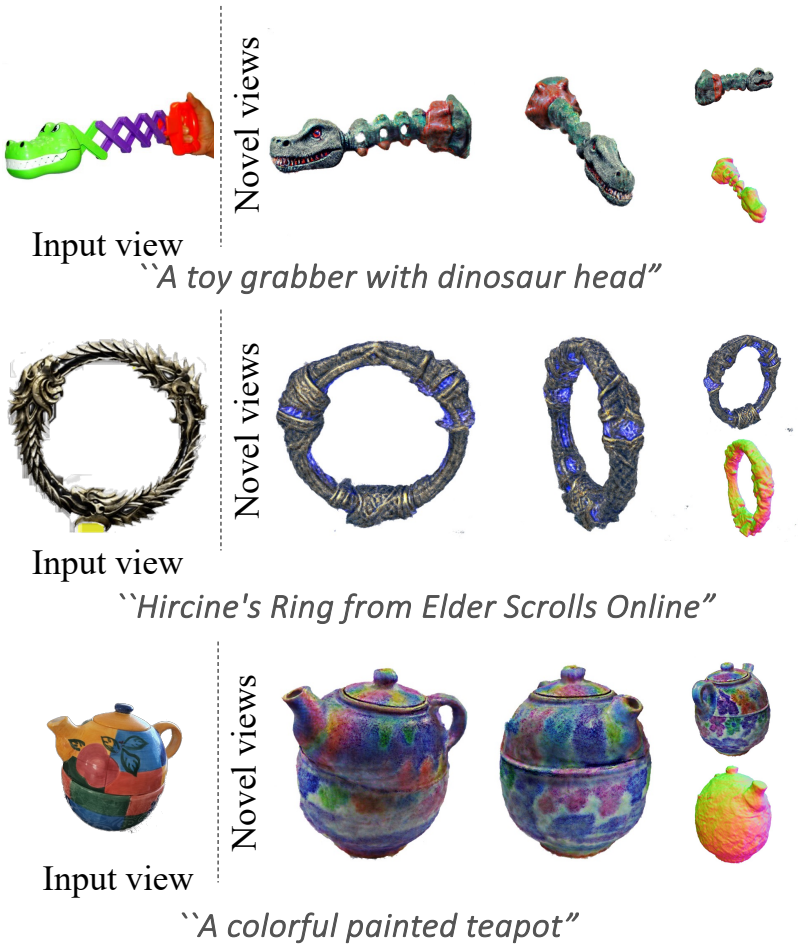


Figure 22: **Visual results of image-guided 3D hallucination.** We **hallucinate** the 3D asset from a single given image using the prompt below the object.