

APPENDIX

A BROADER IMPACTS.

Our approach significantly expands the potential applications of pretrained rectified flow models. We demonstrate that a text-to-image model can be adapted for both image editing and text-to-3D generation tasks. However, since we utilize pretrained text-to-image rectified flow models as our foundational network, our methods might inherit the biases present in these networks.

B PROOF: UNDERSTANDING RFDS-REV FROM THE ANGLE OF EULER SAMPLING.

Euler sampling is one of the most fundamental and widely used sampling strategies in flow matching models (Albergo & Vanden-Eijnden, 2022; Lipman et al., 2022; Liu et al., 2024). A natural question arises: why does Euler Sampling succeed in generating high-quality 2D images, while the RFDS loss does not? Within the framework of a flow matching model, Euler sampling is defined as:

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \Delta t \mathbf{v}(\mathbf{x}_t, t). \quad (11)$$

In words, the image is generated by moving a small step each time given the predicted velocity. Given the \mathbf{x}_t predicted at each step, we can recover the original image \mathbf{x}_* by applying the definition of \mathbf{x}_t in Eq. 1:

$$\alpha_{t+\Delta t} \mathbf{x}'_* + \sigma_{t+\Delta t} \epsilon = \Delta t \mathbf{v}(\mathbf{x}_t, t) + \alpha_t \mathbf{x}_* + \sigma_t \epsilon. \quad (12)$$

If we re-arrange the above Equation and add $-\alpha_{t+\Delta t} \mathbf{x}_*$ on both side, we can have:

$$\alpha_{t+\Delta t} \mathbf{x}'_* - \alpha_{t+\Delta t} \mathbf{x}_* = \Delta t \mathbf{v}(\mathbf{x}_t, t) + \alpha_t \mathbf{x}_* + \sigma_t \epsilon - \sigma_{t+\Delta t} \epsilon - \alpha_{t+\Delta t} \mathbf{x}_*. \quad (13)$$

By dividing both side with Δt , we can have the final form of the updating rule of Euler sampler:

$$\frac{\alpha_{t+\Delta t} - \alpha_t}{\Delta t} \Delta \mathbf{x}_* = \mathbf{v}(\mathbf{x}_t, t) - \dot{\alpha}_t \mathbf{x}_* - \dot{\sigma}_t \epsilon. \quad (14)$$

The left side of the equation indicates the direction of \mathbf{x}_* updates. Notably, the right side of the equation is the same as the proposed RFDS loss (Eq. 8), with one key difference: in Euler Sampling, the noise ϵ is a fixed initial noise, whereas in RFDS, the noise is randomly sampled. However, in the context of 3D generation, sampling a fixed noise for the entire 3D scene is impractical because 3D optimization is inherently a stochastic process, and no fixed noise corresponds to every rendered view. Nevertheless, this “fixed noise” can be identified or learned. Our proposed RFDS-Rev addresses this by using iRFDS to perform image inversion on each rendered view, identifying the corresponding static noise ϵ and ultimately bridging the gap between RFDS and Euler sampling.

C PROOF: RFDS, iRFDS, RFDS-REV WITH DIFFUSION MODELS

RFDS is Identical to SDS When Expressed in Terms of Score Function

As proven in Stochastic Interpolants (Albergo et al., 2023), the velocity $\mathbf{v}(\mathbf{x}_t)$ can be expressed in terms of a score function \mathbf{s} learned with score-matching objective

$$\mathbf{v}(\mathbf{x}_t) = \frac{\sigma_t \mathbf{s}(\mathbf{x}_t) (\dot{\alpha}_t \sigma_t - \alpha_t \dot{\sigma}_t) + \alpha_t \mathbf{x}_t}{\alpha_t}. \quad (15)$$

For the detailed proof of this relation, we refer the readers to Albergo et al. (2023) and Ma et al. (2024).

By substituting this relation into RFDS (Eq. 8) and considering the relation $\mathbf{s} = \frac{-\epsilon_\phi}{\sigma_t}$, we directly obtain the same equation as the SDS loss

$$\nabla_{\theta} \mathcal{L}_{\text{rfds}}(\phi, \mathbf{x}, \epsilon, t) \simeq \mathbb{E} \left[w(t) \underbrace{(\epsilon_\phi(\mathbf{x}_t) - \epsilon)}_{\text{Score Residual}} \underbrace{\frac{\partial \mathbf{x}}{\partial \theta}}_{\text{Generator Jacobian}} \right]. \quad (16)$$

iRFDS Expressed in Terms of Score Function

Similarly, we can derive the iRFDS in terms of score function

$$\nabla_{\epsilon} \mathcal{L}_{\text{iRFDS}}(\phi, \mathbf{x}, \epsilon, t) \simeq \mathbb{E} \left[w'(t) \underbrace{(\epsilon_{\phi}(\mathbf{x}_t) - \epsilon)}_{\text{Score Residual}} \right]. \quad (17)$$

After the two formulas are derived, we can naturally arrive at RFDS-Rev.

D IMPLEMENTATION DETAILS

RFDS and RFDS-Rev for 3D generation. We use the 3D model implicit model Instant-NGP (Müller et al., 2022) as the 3D backbone. Each 3D model is optimized for 15000 steps. We use a CFG of 50 for all 3D experiments and 2D toy experiments. The model is optimized with a resolution of 256 for the first 5000 steps and then 500 for the final 10000 steps. The experiments are carried out on NVIDIA A6000 GPUs. On InstaFlow, generating a single 3D scene takes approximately 30 minutes of wall-clock time with RFDS-Rev and 20 minutes with the RFDS baseline. On SD3, the same task requires around 1 hour with RFDS-Rev and 40 minutes with the RFDS baseline. We use $w(t) = 1$ on SD3 and $w(t) = -1$ on InstaFlow. Choosing iRFDS step size is important to achieve high quality generation with RFDS-Rev. For InstaFlow, we use a stepsize 1. For SD3, we observe that a stepsize of $1 - \sigma_t$ produce reasonable results.

iRFDS for image inversion and editing. The inversion starts from a randomly sampled Gaussian noise. We optimize the noise for 1000 steps using iRFDS and CFG 1 with a learning rate of 3×10^{-3} . To facilitate effective noise optimization, we add one additional loss to enforce the noise follows a Gaussian distribution. Specifically, we add a loss to enforce the mean and variance of the current noise to be zero and one respectively. After the noise is optimized, we change the caption to the target caption and run the forward flow for 5 steps using CFG 1.5. We use $w'(t) = -1$ on SD3 and $w'(t) = 1$ on InstaFlow.

E ADDITIONAL RELATED WORKS

Diffusion as plug-and-play priors. Our work is greatly motivated by the development of diffusion-based priors. The earliest work of diffusion prior (Graikos et al., 2022) requires to backpropagate through the diffusion U-Net. Dreamfusion (Poole et al., 2022) recognizes that such a backpropagate process greatly hurts the performance of diffusion priors. By ignoring the U-Net Jacobian, the SDS loss proposed in Dreamfusion can be used for 3D generation and image editing. However, the initial version of the SDS loss suffers greatly from the lack of details and diversity. DDS loss (Hertz et al., 2023) proposes an improved version of the SDS to improve image editing by taking the difference between the current SDS and the source image SDS. VSD loss (Wang et al., 2023) improves the SDS loss for 3D generation problems. Specifically, it first trains a LoRA of the current 3D model and then takes the difference between the diffusion SDS and LoRA SDS. Although lots of methods have been proposed for diffusion models, this is the first work that studies how to effectively use rectified flow as priors.

F ALGORITHM FOR RFDS AND iRFDS

We list the detailed algorithm for RFDS and iRFDS in Algorithm 2 and Algorithm 3.

G ADDITIONAL ABLATION EXPERIMENTS

Ablation Results on CFG. The scale of classifier-free guidance (CFG) (Ho & Salimans, 2022) plays a crucial role in diffusion-based methods, such as SDS and VSD. We observe a very similar phenomenon with the rectified flow priors. As demonstrated in Fig. 8, both of the proposed methods require a CFG greater than 10 to learn reasonable shapes. However, when the CFG becomes excessively large, the 3D objects generated by RFDS exhibit over-saturated colors. In contrast, RFDS-Rev remains robust to large CFG values, even when the CFG exceeds 2000.

Algorithm 2: The RFDS Algorithm.

```

1 Initialize the learnable parameter  $\theta$ 
2 while Not Converge do
3   Sample random timestep  $t$ 
4   Sample random noise  $\epsilon$ 
5   Optimize  $\theta$  with  $\epsilon$  based on RFDS
    (Eq. 8)
6 RETURN  $\theta$ 

```

Algorithm 3: The iRFDS Algorithm.

```

1 Initialize the learnable parameter  $\epsilon$ 
2 Get initial image  $x$ 
3 while Not Converge do
4   Sample random timestep  $t$ 
5   Optimize  $\epsilon$  using fixed  $x$  based on
    iRFDS (Eq. 10)
6 RETURN  $\epsilon$ 

```



Figure 8: Ablation experiments of classifier free guidance scale on text-to-3D generation. Prompt: A DSLR image of a hamburger.

RFDS-Rev vs. RFDS-VSD. As mentioned in the main text, some of the existing methods aimed at improving the diffusion models can be used directly on rectified flow based methods. We explore to combine VSD with the baseline RFDS, denoted as RFDS-VSD. Specifically, we train a rectified flow LoRA model based on the current rendered images and then calculate the gradients by taking the difference between RFDS and RFDS-LoRA following the VSD setting. Results are shown in Fig. 9. Our experiments, conducted using the InstaFlow backbone, demonstrate that RFDS-Rev produces significantly better results compared to RFDS-VSD, despite RFDS-VSD requiring more computational resources. Notably, implementing VSD on the SD3 model presents significant challenges for most currently available commercial GPUs due to its requirement for fine-tuning the base model, which demands excessive GPU memory.



Figure 9: Ablation experiments of RFDS-Rev vs. RFDS-VSD. Top: 2D case. Bottom: 3D case.

H COMPARISON OF CONVERGENCE SPEED.

As listed in Fig. 10, we observe that our rectified flow based methods lead to much faster convergence speed when doing text-to-3D generation.

I COMPARISON OF THE COMPUTATIONAL COST.

We evaluate the computational costs by examining the number of forward and backward passes of the diffusion or rectified flow network required in 1 optimization iteration. Results are list in Table. 3. The RFDS baseline has the same computational demands as the SDS loss. Due to the calculation of CFG (Ho & Salimans, 2022), they both require two forward passes. RFDS-Rev requires only one additional forward pass, whereas VSD needs two additional forward passes and one additional costly backward pass.

Table 3: Computational cost of one iteration based on the number of forward and backward passes of the network.

Method		SDS Poole et al. (2022)	VSD Wang et al. (2023)	DDS Hertz et al. (2023)	RFDS	iRFDS	RFDS-Rev
Category	-	Diffusion	Diffusion	Diffusion	Rectified Flow	Rectified Flow	Rectified Flow
Computation	Forward	2	4	2	2	1	3
	Backward	0	1	0	0	0	0

J MORE RESULTS OF IMAGE INVERSION AND EDITING USING iRFDS.

We show more results of 2D editing in Fig. 11.

K IMPROVE THE PERFORMANCE OF iRFDS BY INCORPORATING NOISE INTO THE INTERMEDIATE GENERATION STEP.

As discussed in the main text, the performance of our proposed iRFDS can be further enhanced by integrating the learned noise into an intermediate flow generation step. A visual comparison of this approach is presented in Fig. 12.

L COMPARISON OF IMAGE RECONSTRUCTION BETWEEN iRFDS AND NULL-INVERSION

A visual comparison of the image reconstruction ability between our proposed iRFDS and null-inversion is presented in Fig. 13.

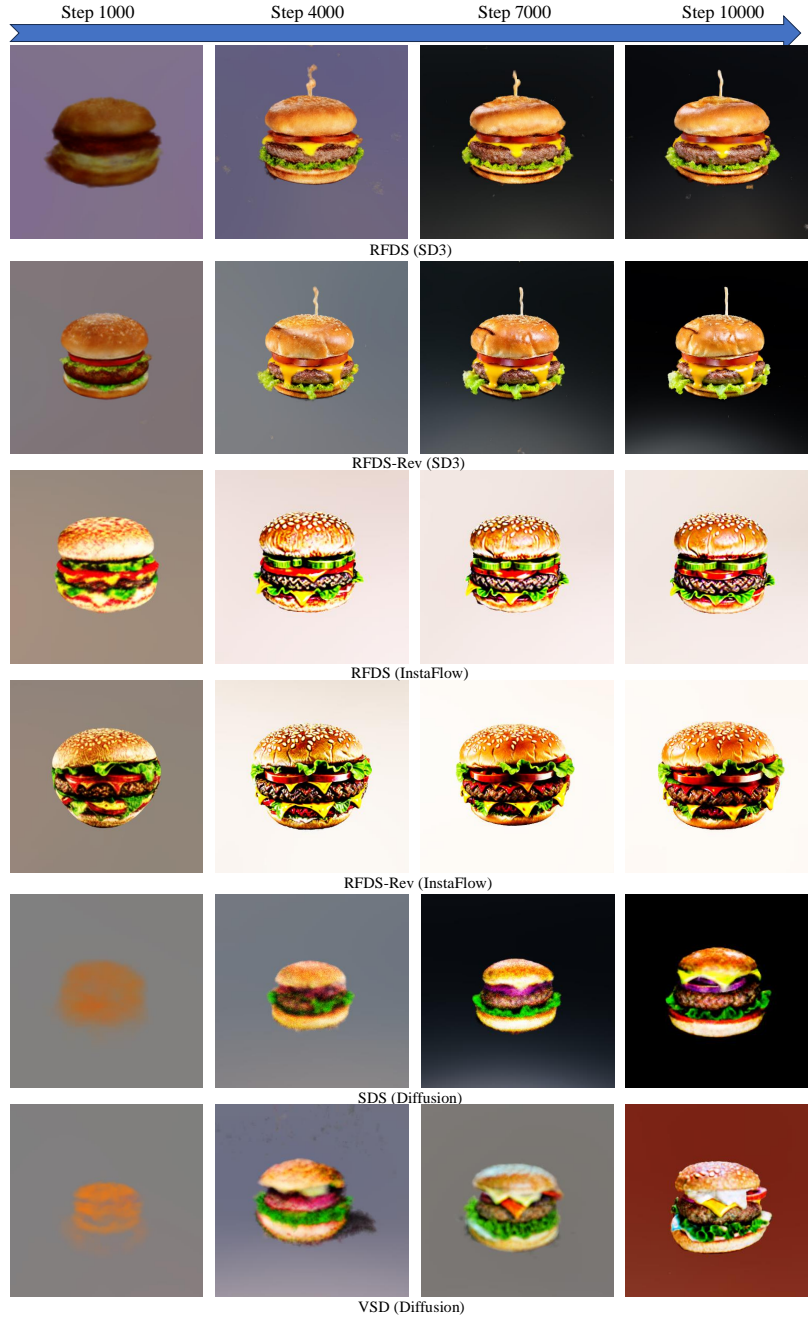


Figure 10: Comparison of convergence speed in 3D generation. Caption: A DSLR image of a hamburger. The 3D model is trained with the same learning rate. We observe that the rectified flow based methods converge much faster compared with diffusion-based methods.

M MORE RESULTS OF TEXT-TO-3D GENERATION

We show more qualitative results of text-to-3D generation in Fig. 14, Fig. 15, Fig. 16 and Fig. 17.

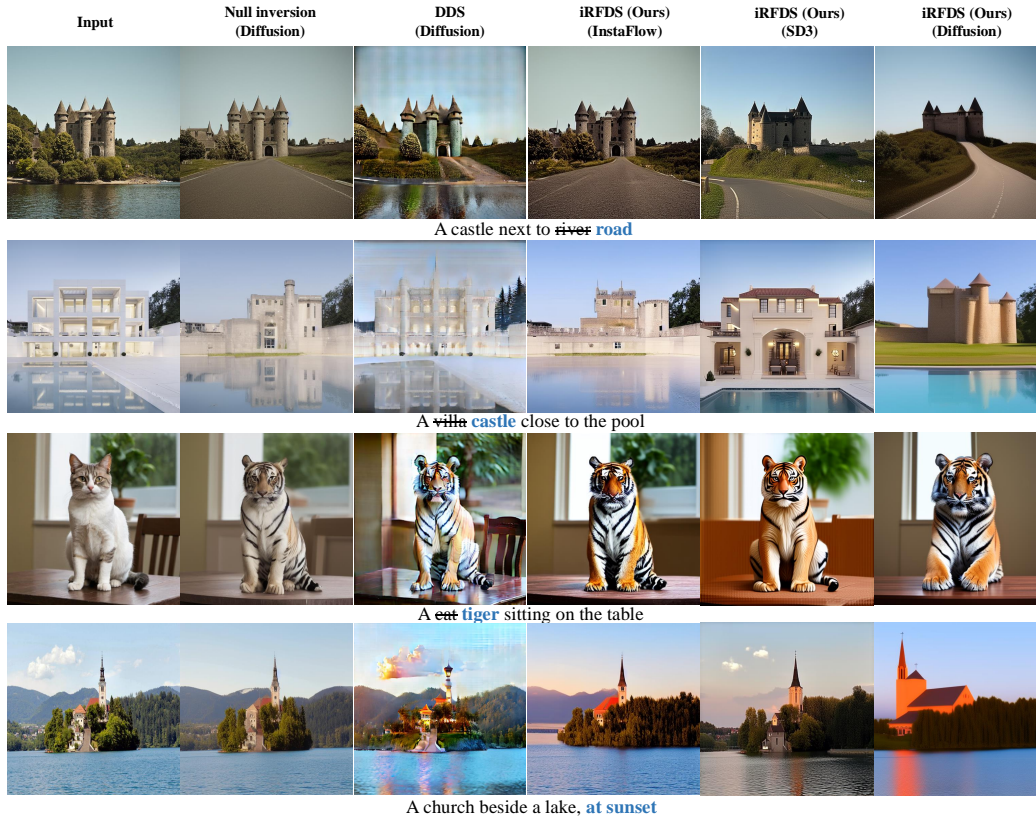


Figure 11: More results on 2D editing.

N IRFDS USER STUDY DETAILS

The user study is carried out with google doc. The users are ask to select the best editing results from the 4 methods. A screenshot of the user study is shown in Fig 18.

O MORE COMPARISON WITH OTHER STATE-OF-THE-ART 3D GENERATION METHODS

We further compare our proposed method (RFDS-Rev + SD3) with other state-of-the-art 3D generation methods, including LucidDreamer (Liang et al., 2024), DreamCraft3D (Sun et al., 2023), and GaussianDreamer (Yi et al., 2023). The results are presented in Fig 19. Our findings demonstrate that our proposed method is versatile, as it can be applied to both NeRF and 3DGS backbones. Our method achieves highly competitive performance across different settings.

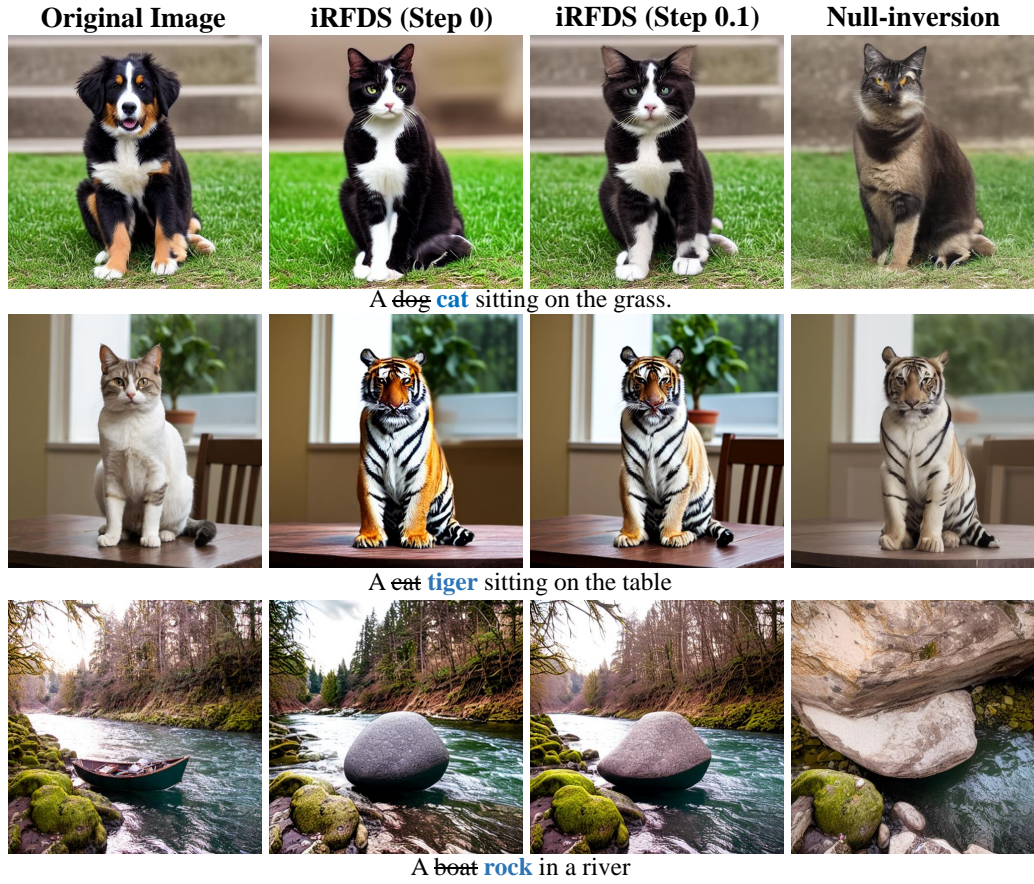


Figure 12: Inserting the iRFDS learned noise into the intermediate generation step improves background and color consistency.

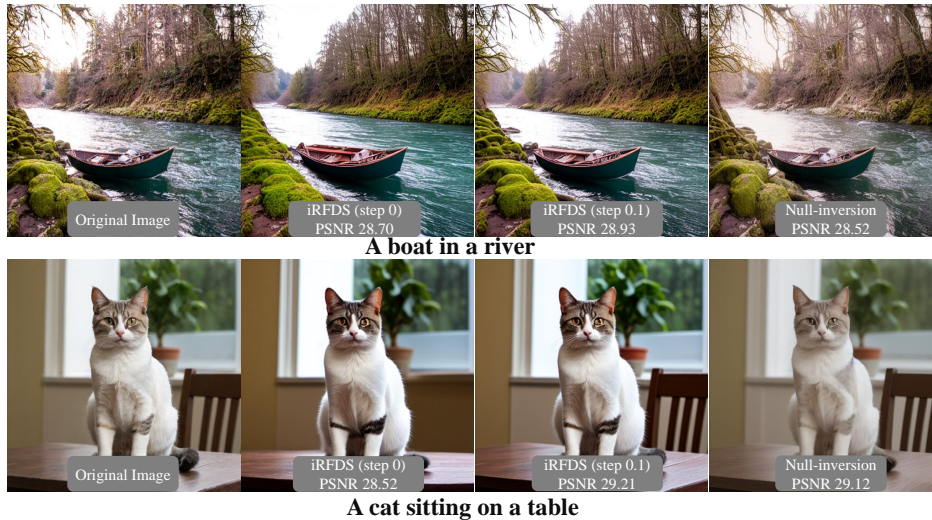


Figure 13: Comparison of image reconstruction.



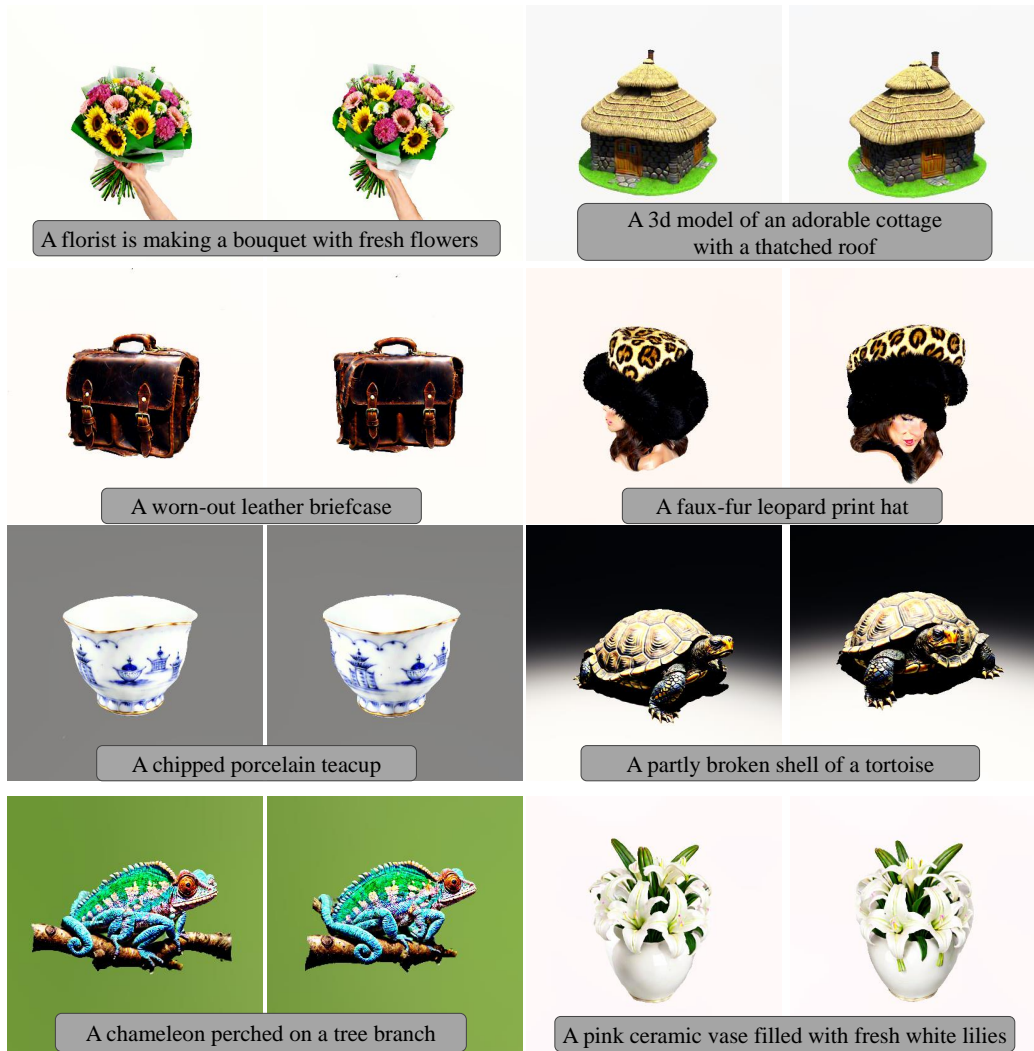
Text-to-3D Generation with RFDS-Rev

Figure 14: More results on text-to-3D generation. Model:SD3



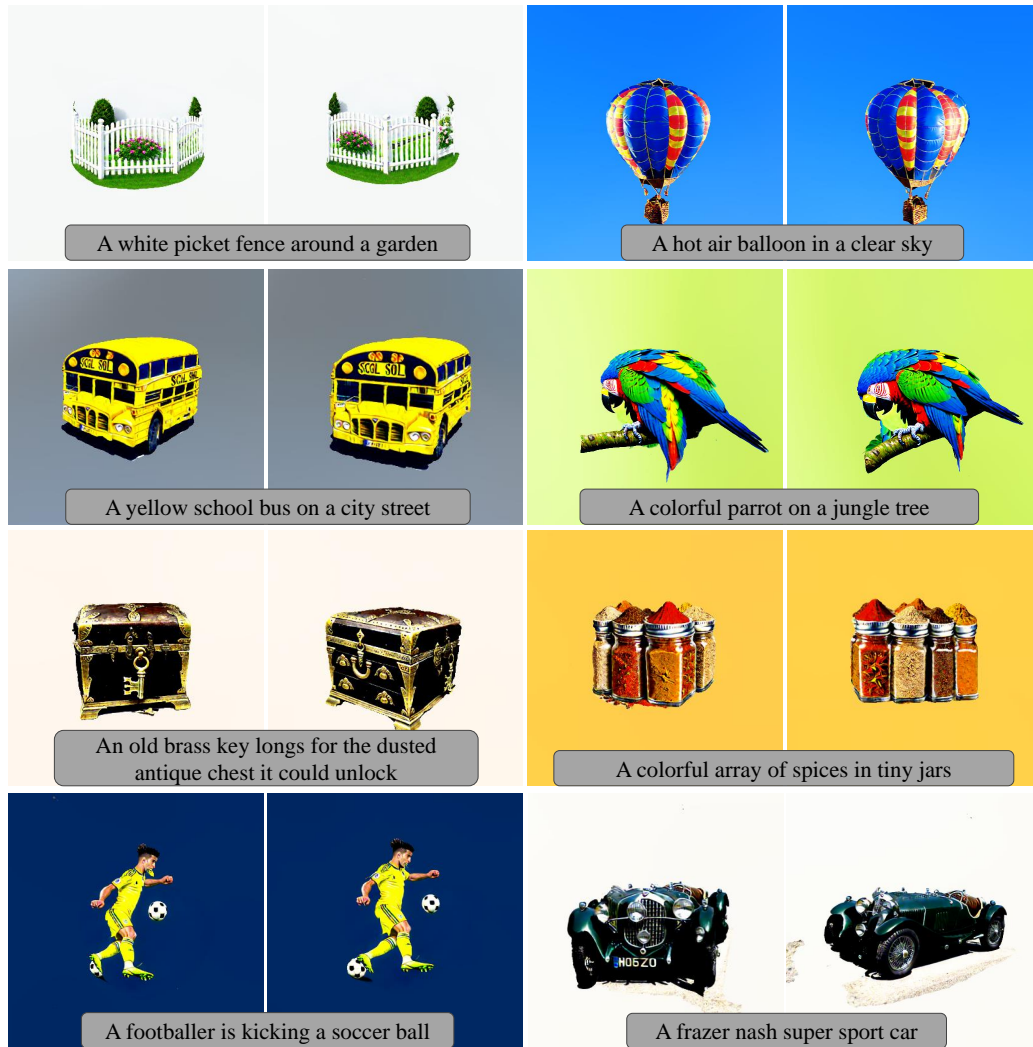
Text-to-3D Generation with RFDS-Rev

Figure 15: More results on text-to-3D generation. Model:SD3



Text-to-3D Generation with RFDS-Rev

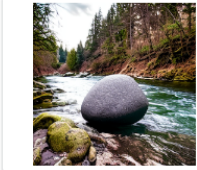
Figure 16: More results on text-to-3D generation. Model:InstaFlow



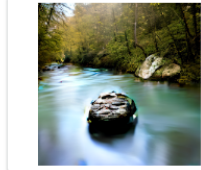
Text-to-3D Generation with RFDS-Rev

Figure 17: More results on text-to-3D generation. Model:InstaFlow

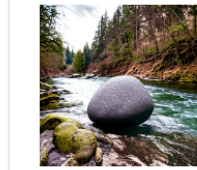
Given the source image "a boat in a river," which method has better semantic correspondence when changing "boat" to "rock"?



☐ Option 1



☐ Option 2



☐ Option 1



☐ Option 2



☐ Option 3



☐ Option 4



☐ Option 3



☐ Option 4



☐ Option 5



☐ Option 5

Figure 18: User study page

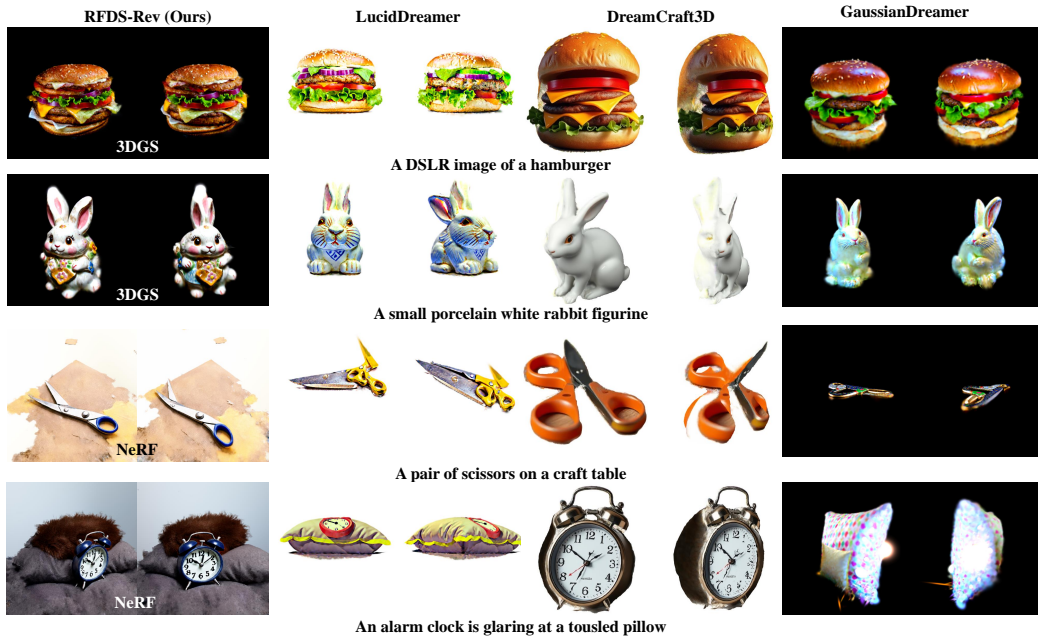


Figure 19: Comparison with other state-of-the-art 3D generation methods