DiSR-NeRF: Diffusion-Guided View-Consistent Super-Resolution NeRF

Supplementary Material

A. Limitations

One limitation of DiSR-NeRF is the limited upscaling factor due to the use of the Stable Diffusion ×4 Upscaler which is designed for 4× super-resolution. In future work, we can consider applying cascaded diffusion models to achieve higher SR upscaling factors on low-resolution NeRFs.

B. Pseudocode

Algorithm 1 and Algorithm 2 shows the pseudocode for I3DS and RSD, respectively.

Algorithm 1 I3DS

```
Input: LR NeRF $\omega_{lr}$, LR images $I_{lr}$, training poses $P_{tr}$
Output: SR NeRF $\omega_{sr}$
$\omega = \omega_{lr}$
for stage_iter = $[0, \text{max\_stage\_iter}]$ do
    //upscaling-stage
    $x_0 = \text{RenderImage}(\omega, P_{tr})$
    $x_0 = \text{InterpolateX4}(x_0)$
    $z_0 = \text{VaeEncode}(x_0)$
    $z'_0 = \text{RSD}(z_0, I_{lr})$
    $x'_0 = \text{VaeDecode}(z'_0)$
    $I_{tr} = x'_0$
    //synchronization-stage
    for sync_iter = $[0, \text{max\_sync\_iter}]$ do
        $r_o, r_d, c_{tr} = \text{SampleRays}(I_{tr}, P_{tr})$
        $\hat{c} = \text{RenderRays}(r_o, r_d)$
        Take gradient descent step on $\nabla_\omega \|\hat{c} - c_{tr}\|$
        $\omega_{old} \leftarrow \omega$
    end for
end for
return $\omega_{sr} = \omega$
```

Algorithm 2 RSD

```
Input: Latent $z_0$, text prompt embeddings $y_{text}$, noise level $y_{noise\_level}$, min timestep $t_{min}$, max timestep $t_{max}$, LR images $I_{lr}$
Output: Refined latent residuals $h_\theta$
$h_\theta = 0$                                                    ▷ Same shape as $z_0$
for sr_iter = $[0, \text{max\_sr\_iter}]$ do
    $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
    $y = y_{text} + y_{noise\_level} + I_{lr}$
    $t = t_{max} - (t_{max} - t_{min})\frac{\text{sr\_iter}}{\text{max\_sr\_iter}}$
    $z'_0 = z_0 + h_\theta$
    $z'_t = \sqrt{\bar{\alpha}_t} z'_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$                ▷ Eq. (1)
    $z'_{t-1} = \sqrt{\bar{\alpha}_{t-1}} z'_0 + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon$        ▷ Eq. (1)
    $\epsilon_\phi(z'_t, y, t) = \text{UNet}(z'_t, y, t)$
    $\hat{z}'_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(z'_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\phi(z'_t, y, t)\right) + \sigma_t\epsilon$ ▷ Eq. (3)
    Take gradient descent step on $\nabla_\theta \|z'_{t-1} - \hat{z}'_{t-1}\|$
    $h^{old}_\theta \leftarrow h_\theta$
end for
return $h_\theta$
```

C. Relating SDS to RSD

We show the relation of our RSD to the existing SDS loss here. As discussed in [5], SDS can be reformulated into a difference of latent vector residuals $\mathbf{z}_0$ and $\hat{\mathbf{z}}_0$, where $\mathbf{z}_0$ is the latent vector under optimization and $\hat{\mathbf{z}}_0$ is the one-step denoised estimate of $\mathbf{z}$.

Starting with the SDS objective proposed in [3]:

$$\nabla_\theta \mathcal{L}_{\text{SDS}} = \gamma(t)(\hat{\epsilon}_\phi(\mathbf{z}_t, y, t) - \epsilon)\frac{\partial \mathbf{z}}{\partial \theta},$$

we substitute

$$\hat{\epsilon}_\phi(\mathbf{z}_t, y, t) = \frac{1}{\sqrt{1 - \bar{\alpha}_t}}(\mathbf{z}_t - \sqrt{\bar{\alpha}_t}\hat{\mathbf{z}}_0)$$

from the reconstruction equation

$$\hat{\mathbf{z}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{z}_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\phi(\mathbf{z}_t, y, t)),$$

and also substitute $\epsilon = \frac{1}{\sqrt{1-\bar{\alpha}_t}}(\mathbf{z}_t - \sqrt{\bar{\alpha}_t}\mathbf{z}_0)$ from the forward noising process in Eq. (1). This gives us:

$$= \gamma(t)\left(\frac{1}{\sqrt{1 - \bar{\alpha}_t}}(\mathbf{z}_t - \sqrt{\bar{\alpha}_t}\hat{\mathbf{z}}_0) - \frac{1}{\sqrt{1 - \bar{\alpha}_t}}(\mathbf{z}_t - \sqrt{\bar{\alpha}_t}\mathbf{z}_0)\right)\frac{\partial \mathbf{z}}{\partial \theta},$$

which reduces to:

$$= \gamma(t)\left(\frac{1}{\sqrt{1 - \bar{\alpha}_t}}\right)\left(-\sqrt{\bar{\alpha}_t}\hat{\mathbf{z}}_0 + \sqrt{\bar{\alpha}_t}\mathbf{z}_0\right)\frac{\partial \mathbf{z}}{\partial \theta},$$

and finally

$$\nabla_\theta \mathcal{L}_{\text{SDS}} = \gamma(t)\left(\frac{\sqrt{\bar{\alpha}_t}}{\sqrt{1 - \bar{\alpha}_t}}\right)\left(\mathbf{z}_0 - \hat{\mathbf{z}}_0\right)\frac{\partial \mathbf{z}}{\partial \theta}. \tag{1}$$

We can interpret this formulation of SDS as an optimization objective within the $\mathbf{z}_0$ space. With this formulation of SDS, RSD can be interpreted as a *renoised* variant of SDS. Specifically, the conversion entails applying the forward noising process in Eq. (1) to both $\mathbf{z}_0$ and $\hat{\mathbf{z}_0}$ towards time $t - 1$ which derives the RSD objective:

$$\mathcal{L}_{RSD} = \|\mathbf{z}_{t-1} - \hat{\mathbf{z}}_{t-1}\|\frac{\partial \mathbf{z}_{t-1}}{\partial \theta},$$

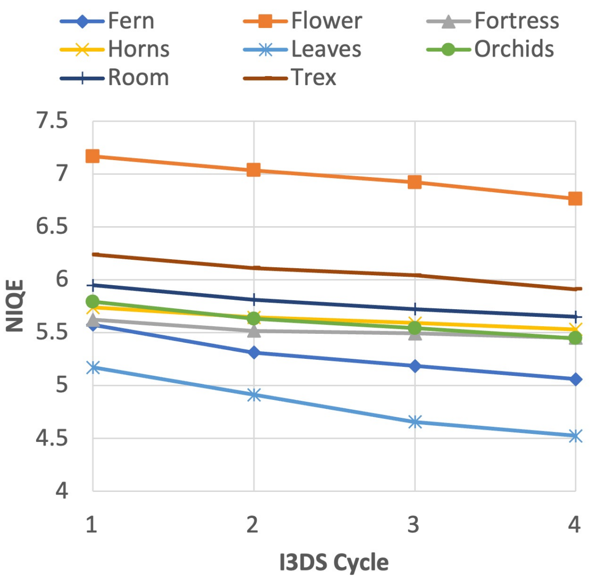

Figure 1. NIQE scores over successive I3DS cycles for scenes in LLFF dataset.

In practice, $\hat{\mathbf{z}}_{t-1}$ can be obtained directly from $\hat{\boldsymbol{\epsilon}}_\phi(\mathbf{z}_t, y, t)$ using the DDPM denoising equation in Eq. (3) instead of renoising $\hat{\mathbf{z}}_0$.

D. I3DS Convergence

In Fig. 1 we plot a graph of the NIQE scores DiSR-NeRF on LLFF scenes over successive I3DS cycles. Across all scenes, we see that NIQE improves with increasing I3DS cycles. This validates that the I3DS framework indeed enables NeRF $\boldsymbol{\omega}$ to converge onto high quality, view-consistent details.

Methods	NeRF-Synthetic		LLFF	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
NGP	31.95	0.959	18.02	0.617
SDx4	28.94	0.935	14.31	0.519
NeRF-SR [4]	**32.86**	**0.962**	**18.27**	**0.633**
DreamFusion [3]	26.21	0.920	14.11	0.502
IN2N [1]	28.62	0.936	14.49	0.521
DiSR-NeRF (Ours)	31.05	0.948	17.01	0.580

Table 1. PSNR and SSIM scores between DiSR-NeRF and the baselines.

E. PSNR & SSIM Scores

We report PSNR & SSIM scores in Tab. 1. DiSR-NeRF still achieves highest similarity scores among prior-based methods (DiSR-NeRF, IN2N, DreamFusion-SDS, SD×4).

Optimization Time (Hrs)↓					
NGP	SD×4	NeRF-SR[4]	DreamFusion[3]	IN2N[1]	DiSR-NeRF (Ours)
0.25	0.35	0.30	8.00	5.00	6.00

Table 2. Comparison of optimization times.

F. Optimization Time

We show optimization times in Tab. 2. DiSR-NeRF's optimization time is longer due to the RSD optimization, but is still faster than DreamFusion-SDS due to I3DS's segregation of upscaling and synchronization stages.

G. Implementation Details

NeRF Backbone. In our DiSR-NeRF implementation, we use Instant-NGP [2] as our default NeRF backbone due to its fast training and rendering speed. We also utilize dynamic ray sampling to increase ray count when the occupancy grid is sufficiently pruned. This optimizes GPU memory usage for faster convergence.

Patch Sampling. In our RSD upscaling stage, we sample uniform random crops of 128×128 resolution in latent space, which corresponds to an image patch of 512×512 resolution. Compared to full image optimization, we find empirically that patch cropping at 128×128 resolution offers the fastest optimization speed and best upscaling performance across all scenes.

Text Prompt. For the text conditioning, we use a fixed text prompt *”⟨subject⟩, high resolution, 4K, photo”* for all scenes and all patches, where the subject tag is replaced with the scene name defined in the dataset.

Learning Rate. We use a constant learning rate of $1e-2$ for all learnable parameters.

I3DS. In the I3DS training regime, we use 5000 upscaling steps with a batch size of 16 patches, followed by 20,000 NeRF training steps. We repeat this two stage cycle for 4 iterations in total.

References

[1] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2

[2] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 2

[3] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 1, 2

[4] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. Nerf-sr: High-quality neural radiance fields using supersampling. *arXiv*, 2021. 2

[5] Junzhe Zhu and Peiye Zhuang. Hifa: High-fidelity text-to-3d with advanced diffusion guidance, 2023. 1