How to Use Diffusion Priors under Sparse Views?

Qisen Wang,   Yifan Zhao,*   **Jiawei Ma,   Jia Li***
State Key Laboratory of Virtual Reality Technology and Systems, SCSE
Beihang University
{wangqisen, zhaoyf, majiawei, jiali}@buaa.edu.cn

Abstract

Novel view synthesis under sparse views has been a long-term important challenge in 3D reconstruction. Existing works mainly rely on introducing external semantic or depth priors to supervise the optimization of 3D representations. However, the diffusion model, as an external prior that can directly provide visual supervision, has always underperformed in sparse-view 3D reconstruction using Score Distillation Sampling (SDS) due to the low information entropy of sparse views compared to text, leading to optimization challenges caused by mode deviation. To this end, we present a thorough analysis of SDS from the mode-seeking perspective and propose Inline Prior Guided Score Matching (IPSM), which leverages visual inline priors provided by pose relationships between viewpoints to rectify the rendered image distribution and decomposes the original optimization objective of SDS, thereby offering effective diffusion visual guidance without any fine-tuning or pre-training. Furthermore, we propose the IPSM-Gaussian pipeline, which adopts 3D Gaussian Splatting as the backbone and supplements depth and geometry consistency regularization based on IPSM to further improve inline priors and rectified distribution. Experimental results on different public datasets show that our method achieves state-of-the-art reconstruction quality. The code is released at https://github.com/iCVTEAM/IPSM.

1   Introduction

Novel View Synthesis (NVS) [1, 2], *e.g.* Neural Radiance Fields (NeRF) [1, 3, 4] and recently emerged 3D Gaussian Splatting (3DGS) [2, 5, 6], requires dense training viewpoints for optimization, as demonstrated in prevailing works [7–9]. Indeed, NVS under sparse views has been an important and challenging task [10, 11, 7]. Due to the scarcity of viewpoints, most methods of 3D representation reconstruction often fall into over-fitting with sparse views, and cannot synthesize satisfactory novel views [9, 8, 12]. To address the optimization over-fitting problem under the sparse-view condition, current methods introduce external priors to supervise the optimization of reconstruction like CLIP [13] semantic information [7], monocular depth [11, 9], and diffusion visual priors [14–16]. However, although the diffusion model [17–21] as an external prior can provide stronger visual supervision than semantic and depth information, it often requires a significant amount of computational resources for *fine-tuning the diffusion prior* [16] or *pre-training encoders* [15] with external data. A few works have no fine-tuning and pre-training, but it is difficult to straightly extract diffusion prior knowledge to effectively supplement the missing visual information of sparse views [14].

Interestingly, although the diffusion model shows great potential in 3D generation tasks, *e.g.* text-to-3D [23], which benefit from the recent rapid development of score distillation techniques [23–26], Score Distillation Sampling (SDS) [23] shows little visual information guidance ability of the diffusion prior under sparse views and even takes an inhibitory effect on the baseline performance when the input views increase, as shown in Fig. 1. **The SDS dilemma** highlights that score distillation exhibits

---

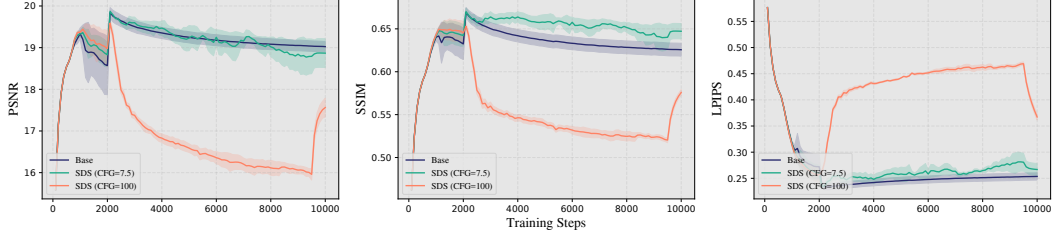*Correspondence should be addressed to Yifan Zhao and Jia Li. Website: https://cvteam.buaa.edu.cn

Figure 1: **Dilemma of SDS**. Average PSNR↑, SSIM↑, and LPIPS↓ of each iteration on the LLFF test dataset [22] with Base (without SDS), SDS (CFG=7.5), and SDS (CFG=100). The prior-added period starts from the 2K iteration and ends at the 9.5K iteration. The opacity is also reset at 2K. The details and final training results of SDS are shown in Sec. 4.4.

distinctive optimization characteristics across sparse input views. Consequently, SDS is NOT readily applicable for lifting visual supervision from diffusion priors under sparse views.

With the curiosity of the SDS dilemma in our mind, it can be recognized that the difference between *sparse views* and *text prompts* lies in the inline constraints sparse views bring. For the unsupervised invisible views, unlike text prompts, the ideal rendered image supervision information is not completely absent. Due to the consistency of the 3D geometry and structure, the information exists in the given sparse views, which we refer to the **inline priors**. Some researchers [15] attempt to implicitly encode the given input sparse views to guide the sampling trajectory of the diffusion model, thereby introducing inline priors. Nonetheless, owing to domain shifts between specific scenes and the diffusion prior, a significant amount of external 3D annotated data and computational resources are frequently necessitated for domain rectification [15]. To this end, a potentially viable approach is exploring the feasibility of adjusting the optimization objective of SDS by incorporating inline priors to facilitate efficient domain rectification without fine-tuning and pre-training.

In this paper, we conduct a comprehensive analysis of SDS from the perspective of mode-seeking. Intuitively, the optimization objective of SDS is to align the rendered image distribution with the target mode in the diffusion prior. However, due to the inherent suboptimality of the rendered image distribution under sparse views, SDS tends to deviate from the target mode, resulting in the SDS dilemma. To tackle this challenge, we present Inline Prior Guided Score Matching (IPSM), a method that rectifies the rendered image distribution by utilizing inline priors. IPSM leverages the rectified distribution to divide the optimization objective of SDS into two sub-objectives. The rectified distribution, as an intermediate state of the optimization objective, plays a role in controlling the mode-seeking direction, thereby suppressing mode deviation and promoting improvements in reconstruction. Moreover, we propose the pipeline IPSM-Gaussian, which combines IPSM with the efficient explicit 3D representation 3DGS for sparse-view 3D reconstruction. In addition to IPSM, IPSM-Gaussian integrates depth regularization to support inline priors and geometric consistency regularization to narrow the discrepancy between the rendered image distribution and the rectified distribution at the pixel level. Experimental results demonstrate that IPSM effectively leverages visual knowledge from the diffusion priors to improve sparse-view 3D reconstruction. The presented method achieves superior performance on publicly available datasets.

Overall, our contributions can be summarized as:

- *Analysis of SDS from mode-seeking perspective*. We present a comprehensive analysis of SDS optimization characteristics under sparse views, revealing that the mode deviation of SDS results in the optimization dilemma.

- *Rectified score distillation method for sparse views*. We propose Inline Prior Guided Score Matching (IPSM), which utilizes inline priors provided by sparse views to rectify rendered image distribution for controlling the direction of seeking the target mode.

- *Pipeline using IPSM based on 3DGS*. We present IPSM-Gaussian, a pipeline for sparse-view 3D reconstruction, which adopts IPSM for diffusion guidance, as well as depth and geometry regularization to boost the performance of IPSM. The experiments show that IPSM-Gaussian achieves state-of-the-art reconstruction quality on public datasets.

2

2 Related Works

Novel View Synthesis. Novel View Synthesis [27, 1, 2, 28–31] aims to synthesize invisible novel views given a set of images at seen viewpoints while preserving the geometric structure and appearance of the original 3D scene [32–37]. NeRF [1, 3, 4], as an implicit 3D representation, adopts volume rendering to establish an implicit mapping relationship from the positions and ray directions to colors using a Multi-Layer Perception (MLP). Although NeRF can achieve photographic-realistic rendering quality compared to traditional methods, its required training time and rendering speed are not satisfactory [1, 28]. Recently, 3DGS [2, 5, 6] has garnered attention from researchers by achieving high training speeds and real-time rendering capabilities through explicit modeling of 3D scenes using Gaussian point clouds and rasterization rendering [38–42]. To this end, we choose 3DGS instead of NeRF as the backbone of 3D representations and adopt it in subsequent experiments.

Sparse-view Novel View Synthesis. Although current training-based NVS techniques, *i.e.*NeRF [1] and 3DGS [2], can achieve satisfactory rendering quality in scenarios with dense input views, the quality of novel view synthesis significantly decreases under sparse views due to overfitting [43, 12, 44, 7, 11, 45]. To tackle this challenge, Yang *et al.*[8] leverage the optimization properties of MLP and employ annealing strategies for positional encoding [36] tailored to the characteristics of NeRF, but this cannot be directly applied to 3DGS. More broadly, some works [46, 47] leverage the intrinsic relationships between sparse views to augment the data required for model optimization, but this does not address the established condition of information deficiency. More works involve introducing external pre-trained priors as optimization guidance to supervise sparse-view 3D reconstruction. Jain *et al.*[10] introduce CLIP [13] to provide semantic guidance. Li *et al.*[9] propose global-local depth regularization with DPT [48] for geometric structure guidance. However, the aforementioned prior information cannot directly provide visual supervision for sparse-view NVS like diffusion priors.

Sparse-view Novel View Synthesis with Diffusion Priors. Although diffusion priors can provide more direct visual guidance, current works are limited by the mode deviation with using diffusion priors directly. Liu *et al.*[16] leverage diffusion models to progressively generate pseudo-observations at unseen views. Wu *et al.*[15] use PixelNeRF [49] to encode sparse inputs for guiding the trajectory of diffusion priors. Unlike score distillation techniques, these works either require fine-tuning the diffusion model for narrowing the mode range [16], or pre-training image encoders for guiding the direction of the target mode [15], both of which consume many resources [16, 15]. Xiong *et al.*[14] attempt to directly use SDS to extract the external visual prior of the diffusion model, but have to suppress its weighting, thus achieving limited effects. Although view-conditioned diffusion priors [50, 51] have emerged recently, different to helpness for 3D generation [50, 52], their guidance is still limited for sparse-view reconstruction, which is detailedly discussed in the Appendix. Therefore, *how to use diffusion priors* and *how to use score distillation* under sparse views without fine-tuning, pre-training, and the optimization dilemma shown in Fig. 1 have become crucial issues.

3 Method

With the phenomenon of the SDS dilemma shown in Fig. 1 in our mind, we have realized that SDS that works for text prompts does not work equally well for sparse views. Therefore, we attempt to analyze the disadvantages of SDS under sparse views and introduce inline constraints for effectively extracting visual guidance of diffusion priors without fine-tuning and pre-training. We start with the overview of 3DGS and also define the main symbols.

3.1 Overview of 3D Gaussian Splatting

Representation. The 3DGS models the 3D structure with a set of Gaussian points with positions μ_n, covariance matrix Σ_n, color c_n represented by Spherical Harmonic (SH) coefficients and opacity α_n. For each Gaussian point n, its 3D position follows

$$G(x) = e^{-\frac{1}{2}(x-\mu_n)^{\mathsf{T}}\Sigma_n^{-1}(x-\mu_n)},\tag{1}$$

where Σ_n can be represented by the scaling matrix S_n and the rotation matrix R_n

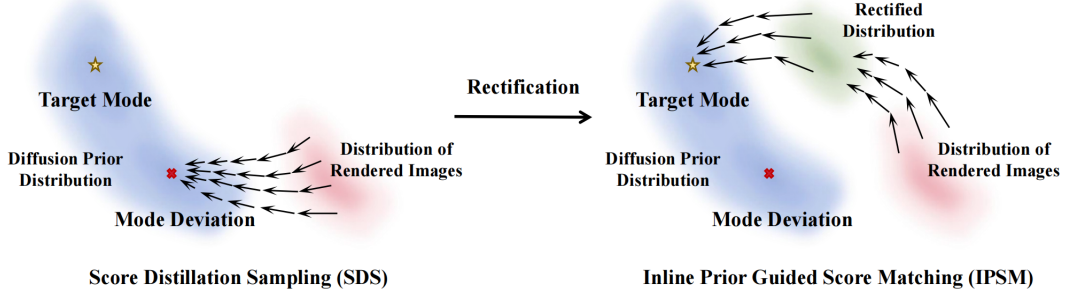$$\Sigma_n = R_n S_n S_n^{\mathsf{T}} R_n^{\mathsf{T}}.\tag{2}$$

Figure 2: Comparison of SDS and IPSM. Left: Tending to seek nearest mode, causing mode deviation. **Right:** Rectifying distribution to seek the target mode.

Rendering. For the 3D representation $\theta = \{\mu_n, \Sigma_n, c_n, \alpha_n\}$, we can optimize the trainable parameters θ through the following differentiable rendering function

$$x_0(\mathbf{p}) = \sum_{n=1}^{N} c_n \tilde{\alpha}_n \prod_{m=1}^{n-1} (1 - \tilde{\alpha}_m), \tag{3}$$

where $x_0(\mathbf{p})$ is the rendering color at pixel $\mathbf{p}$ of rendered image $\mathbf{x}_0$, and $\tilde{\alpha}_n$ are computed from the projected 2D Gaussians.

3.2 IPSM: Inline Prior Guided Score Matching

Review of Score Distillation Sampling. Intuitively, SDS tends to drive the rendered image distribution denoted with red color seeking the nearest mode of diffusion distribution denoted with blue color guided by text prompts. Specifically, we denote the rendered image at viewpoint $\mathbf{v}^j$ as $\mathbf{x}_0^j = g(\theta, \mathbf{v}^j)$, where $g(\theta, \cdot)$ is rendering function and θ is the 3D representation needed optimization. Without elaborating text prompts on the conditions for brevity, the posterior noisy distribution of rendered images is defined as

$$q_t^\theta(\mathbf{x}_t^j) \sim \mathcal{N}(\mathbf{x}_t^j; \sqrt{\bar{\alpha}_t}\mathbf{x}_0^j, (1 - \bar{\alpha}_t)\mathbf{I}). \tag{4}$$

The prevailing score distillation works start from minimizing the reverse KL divergence between the distribution of the noisy rendered images $q_t^\theta(\mathbf{x}_t^j)$ and the noisy real-world distribution $p_t^*(\mathbf{x}_t^j)$ represented by the pre-trained diffusion models, namely

$$\min_\theta \mathbb{E}_{t,\mathbf{v}_j} \left[\omega(t) D_{KL}(q_t^\theta(\mathbf{x}_t^j) \| p_t^*(\mathbf{x}_t^j)) \right], \tag{5}$$

which indicates the gradient of score distillation that

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\theta) \approx \mathbb{E}_{t,\epsilon,\mathbf{v}^j} \left[\omega(t)(\epsilon_*(\mathbf{x}_t^j, t) - \epsilon) \frac{\partial g(\theta, \mathbf{v}^j)}{\partial \theta} \right] = \mathbb{E}_{t,\epsilon,\mathbf{v}^j} \left[\frac{\omega(t)}{\gamma(t)}(\mathbf{x}_0^j - \hat{\mathbf{x}}_0^{j;*}) \frac{\partial g(\theta, \mathbf{v}^j)}{\partial \theta} \right], \tag{6}$$

where $\gamma(t) = \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}$ and $\mathbf{x}_0^j \sim q_0^\theta(\mathbf{x}_0^j)$, $\hat{\mathbf{x}}_0^{j;*} \sim p_0^*(\mathbf{x}_0^j)$. That is, for the given new viewpoints $\mathbf{v}_j$, the gradient $\nabla_\theta \mathcal{L}_{\text{SDS}}(\theta)$ considers the rendered image distribution of the 3D representation and drives it closer to the pre-trained diffusion prior.

Following [23, 53], we provide a further discussion of SDS. The optimization objective of Eq. 5 derives $q_t^\theta(\mathbf{x}_t^j)$ to the high-density region of $p_t^*(\mathbf{x}_t^j)$. Considering samples $\mathbf{m}^\mathcal{T}, \mathbf{m}^\mathcal{F}$ from two modes of $p_t^*(\mathbf{x}_t^j)$, where $\mathbf{m}^\mathcal{T}$ is from target mode and $\mathbf{m}^\mathcal{F}$ is from failure mode. $\mathbf{m}^\mathcal{F}$ is harmless for text-to-3D tasks due to the high information entropy properties of text prompts. However, for sparse-view 3D reconstruction, this leads the optimized 3D representation to be inconsistent with the given sparse images, thus causing optimization difficulties as shown in Fig. 2. Specifically, we denote the L2 distance of two samples as $\Gamma(\cdot, \cdot)$. We want $\sqrt{\bar{\alpha}_t}\mathbf{x}_0^j \approx \sqrt{\bar{\alpha}_t}\mathbf{m}^\mathcal{T}$ for any t, but the gap between two modes is unclear when t increases, $i.e. \Gamma(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^j, \sqrt{\bar{\alpha}_t}\mathbf{m}^\mathcal{T}) \approx \Gamma(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^j, \sqrt{\bar{\alpha}_t}\mathbf{m}^\mathcal{F})$, since $|\Gamma(\mathbf{x}_0^j, \mathbf{m}^\mathcal{T}) - \Gamma(\mathbf{x}_0^j, \mathbf{m}^\mathcal{F})|$ is not large enough for a small $\sqrt{\bar{\alpha}_t}$. This results in the mode aliasing for optimization and further affects the optimizing direction during training. To this end, the

Figure 3: **IPSM-Gaussian** obtains the inline prior within sparse views through inversely warping seen views to unseen pseudo views, thus modifying the rendered image distribution to the rectified distribution. Consequently taking the rectified distribution as the intermediate state, two sub-optimization objectives are utilized for controlling the optimization direction.

distribution of rendered images is not constrained to seeking the target mode, causing mode deviation. Therefore, we aim to construct a rectified distribution excluded failure mode using the inline prior from sparse views, whose sample $\mathbf{m}^{\mathcal{R}}$ provides $\mathbf{x}_0^j$ stable optimization guidance and amplifies the gap $|\Gamma(\mathbf{m}^{\mathcal{R}}, \mathbf{m}^{\mathcal{T}}) - \Gamma(\mathbf{m}^{\mathcal{R}}, \mathbf{m}^{\mathcal{F}})|$ so that $\Gamma(\sqrt{\bar{\alpha}_t}\mathbf{m}^{\mathcal{R}}, \sqrt{\bar{\alpha}_t}\mathbf{m}^{\mathcal{T}}) \ll \Gamma(\sqrt{\bar{\alpha}_t}\mathbf{m}^{\mathcal{R}}, \sqrt{\bar{\alpha}_t}\mathbf{m}^{\mathcal{F}})$, and the rectified distribution is served as the bridge between $\mathbf{x}_0^j$ and $\mathbf{m}^{\mathcal{T}}$ to control the mode-seeking direction.

Inline Prior. Different from text-to-3D tasks, sparse views can achieve geometry consistency guidance of novel views through camera pose transformation, namely the inline prior we mentioned in Sec. 1. Therefore, we aim to utilize the additional visual information of sparse views compared to text prompts to correct the erroneous tendency of SDS optimization. Specifically, we sample a set of random pseudo viewpoints $\mathbf{v}^j$ around the seen views $\mathbf{v}^i$. Given the ground-truth image $\mathbf{I}_0^i$ at the seen viewpoint $\mathbf{v}^i$, we formulate the transforming function $\psi(\mathbf{I}_0^i; \mathbf{D}^j, \mathbf{R}^{j \to i})$ which inversely warps image $\mathbf{I}_0^i$ from viewpoint $\mathbf{v}^i$ to $\mathbf{v}^j$. $\mathbf{R}^{j \to i}$ represents the relative pose transformation between two viewpoints, and $\mathbf{D}^j$ is the alpha-blending rendered depth at viewpoint $\mathbf{v}^j$ following

$$D^j(\mathbf{p}) = \sum_{n=1}^{N} d_n \tilde{\alpha}_n \prod_{m=1}^{N-1} (1 - \tilde{\alpha}_m), \tag{7}$$

where d_n is the z-buffer of the n-th Gaussian. During transformation, each pixel location $\mathbf{p}^j$ at the pseudo viewpoint $\mathbf{v}^j$ is warped to the pixel location $\mathbf{p}^{j \to i}$ at the seen viewpoint $\mathbf{v}^i$, and $\mathbf{p}^{j \to i}$ can be represented by

$$\mathbf{p}^{j \to i} \sim \mathbf{K}\mathbf{R}^{j \to i} D^j(\mathbf{p}^j)\mathbf{K}^{-1}\mathbf{p}^j, \tag{8}$$

where $\mathbf{K}$ is the camera intrinsic parameter. Then, we can obtain the warped image $I_0^{i \to j}(\mathbf{p}^j)$ using inverse warping with the nearest sampling operator

$$I_0^{i \to j}(\mathbf{p}^j) = \mathrm{Sampler}(\mathbf{I}_0^i, \mathbf{p}^{j \to i}). \tag{9}$$

However, this direct inverse warping may lead to warping distortion due to erroneous geometry. Following [46], we tackle it through the generated consistency mask with an error threshold τ

$$M^{i \to j}(\mathbf{p}^j) = \mathrm{Mask}(\|D^j(\mathbf{p}^j) - D^{i \to j}(\mathbf{p}^j)\|_1 < \tau), \tag{10}$$

where $D^{i \to j}(\mathbf{p}^j) = \mathrm{Sampler}(\mathbf{D}^i, \mathbf{p}^{j \to i})$ like Eq. 9. Eq. 10 ensures the filterability of erroneous geometry using the difference between the warped depth of the seen viewpoint and the depth of the pseudo viewpoint. In practice, the warped image $\mathbf{I}_0^{i \to j}$ and its accompanying mask $\mathbf{M}^{i \to j}$ are served

as the *inline geometry consistency prior* to guide external diffusion prior scene specialization. The intuitive explanation of inline priors can be found in Appendix B.7.

Inline Prior Guided Score Matching. Using score distillation directly in the case of sparse views overlooks the inline geometry consistency prior within the sparse views themselves, which is fundamentally different from text-to-3D. To this end, we rectify the distribution denoted with green color from $q_0^\theta(\mathbf{x}_0^j)$ to $\tilde{q}_0^{\theta,\phi}(\mathbf{x}_0^j|\mathbf{M}^{i\to j}\odot\mathbf{I}_0^{i\to j},\mathbf{M}^{i\to j})$ using the inline prior. As shown in Fig. 3, we utilize the warped masked image $\mathbf{I}_0^{i\to j}$ from the seen viewpoints to guide the sampling trajectory of $\hat{\mathbf{x}}_0^{j;\phi}\sim\tilde{q}_0^{\theta,\phi}(\mathbf{x}_0^j|\mathbf{M}^{i\to j}\odot\mathbf{I}_0^{i\to j},\mathbf{M}^{i\to j})$, thus introducing the inline geometry consistency prior to the score distillation. So our optimization objective is changed to minimizing (1) the KL divergence between the noisy rendered image distribution $q_t^\theta(\mathbf{x}_t^j)$ and the noisy rectified distribution $\tilde{q}_t^{\theta,\phi}(\mathbf{x}_t^j)$; (2) the KL divergence between the noisy rectified distribution $\tilde{q}_t^{\theta,\phi}(\mathbf{x}_t^j)$ and the noisy diffusion prior distribution $p_t^*(\mathbf{x}_t^j)$ represented by the pre-trained diffusion models, namely

$$\min_\theta\left\{\eta_r\mathbb{E}_{t,c}\left[\omega(t)D_{KL}(q_t^\theta(\mathbf{x}_t^j)\|\tilde{q}_t^{\theta,\phi}(\mathbf{x}_t^j))\right]+\mathbb{E}_{t,c}\left[\omega(t)D_{KL}(\tilde{q}_t^{\theta,\phi}(\mathbf{x}_t^j)\|p_0^*(\mathbf{x}_t^j))\right]\right\},\qquad(11)$$

where η_r is the adjustment parameter of the two sub-optimization objectives. In practice, we introduce an inpainting diffusion model $\boldsymbol{\epsilon}_\phi(\mathbf{x}_t^j,t,\mathbf{M}^{i\to j}\odot\mathbf{I}_0^{i\to j},\mathbf{M}^{i\to j})$, which shares the same VAE-feature domain with the pre-trained diffusion model $\boldsymbol{\epsilon}_*(\mathbf{x}_t^j,t)$ representing the real data distribution. So we have the rectified gradient of score distillation

$$\begin{aligned}\nabla_\theta\mathcal{L}_{\text{IPSM}}(\theta)\approx&\eta_r\mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{v}^j}\left[\frac{\omega(t)}{\gamma(t)}(\mathbf{x}_0^j-\hat{\mathbf{x}}_0^{j;\phi})\frac{\partial g(\theta,\mathbf{v}^j)}{\partial\theta}\right]+\mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{v}^j}\left[\frac{\omega(t)}{\gamma(t)}(\hat{\mathbf{x}}_0^{j;\phi}-\hat{\mathbf{x}}_0^{j;*})\frac{\partial g(\theta,\mathbf{v}^j)}{\partial\theta}\right]\\=&\eta_r\mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{v}^j}\left[\omega(t)(\boldsymbol{\epsilon}_\phi(\mathbf{x}_t^j,t,\mathbf{M}^{i\to j}\odot\mathbf{I}_0^{i\to j},\mathbf{M}^{i\to j})-\boldsymbol{\epsilon})\frac{\partial g(\theta,\mathbf{v}_j)}{\partial\theta}\right]\\&+\mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{v}^j}\left[\omega(t)(\boldsymbol{\epsilon}_*(\mathbf{x}_t^j,t)-\boldsymbol{\epsilon}_\phi(\mathbf{x}_t^j,t,\mathbf{M}^{i\to j}\odot\mathbf{I}_0^{i\to j},\mathbf{M}^{i\to j}))\frac{\partial g(\theta,\mathbf{v}^j)}{\partial\theta}\right].\end{aligned}$$
$$(12)$$

Consequently, the IPSM regularization can be represented as

$$\mathcal{L}_{\text{IPSM}}=\eta_r\underbrace{\mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{v}^j}\left[\|\omega(t)(\boldsymbol{\epsilon}_\phi-\boldsymbol{\epsilon})\|_2^2\right]}_{\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_1}}+\underbrace{\mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{v}^j}\left[\|\omega(t)(\boldsymbol{\epsilon}_*-\boldsymbol{\epsilon}_\phi)\|_2^2\right]}_{\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_2}}.\qquad(13)$$

3.3 Training Details

Depth Regularization. In the warping process, it can be observed that the rendered depth influences pixel mapping relations, which is detailed in Sec. 4. Therefore, it is necessary to incorporate monocular depth estimation prior to supervising rendered depth, thus providing the correct inline prior. We use the Pearson Correlation to provide depth regularization, which can be represented as

$$\text{Corr}(\mathbf{D}_r,\mathbf{D}_m)=\frac{\text{Cov}(\mathbf{D}_r,\mathbf{D}_m)}{\sqrt{\text{Var}(\mathbf{D}_r)\text{Var}(\mathbf{D}_m)}}.\qquad(14)$$

Given the rendered depth $\mathbf{D}_r^i$, monocular depth $\mathbf{D}_m^i$ from the input view $\mathbf{I}_0^i$ at the seen view $\mathbf{v}^i$, and the rendered depth $\mathbf{D}_r^j$, monocular depth $\mathbf{D}_m^j$ from the rendered image $\mathbf{x}_0^j$ at the unseen view $\mathbf{v}^j$, we take the depth regularization as

$$\mathcal{L}_{\text{depth}}=\eta_d\|\text{Corr}(\mathbf{D}_r^i,\mathbf{D}_m^i)\|_1+\|\text{Corr}(\mathbf{D}_r^j,\mathbf{D}_m^j)\|_1,\qquad(15)$$

where η_d serves as the weight to balance the supervision of seen views and pseudo-unseen views.

Geometry Consistency Regularization. In Eq. 13, we introduce $\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_1}$ for providing guidance to minimize the reverse KL divergence between the rendered image and rectified distribution. In practice, we not only supervise from the diffusion feature domain but also provide stronger guidance by directly adding masked L1 loss of $\mathbf{x}_0^j$ and $\mathbf{I}_0^{i\to j}$, which is denoted as the geometry consistency regularization and can be represented as

$$\mathcal{L}_{\text{geo}}=\|\mathbf{M}^{i\to j}\odot(\mathbf{x}_0^j-\mathbf{I}_0^{i\to j})\|_1.\qquad(16)$$

Table 1: **Quantitative comparisons with other methods.**

Methods	Setting	LLFF [22]				DTU [54]			
		SSIM↑	LPIPS↓	PSNR↑	AVGE↓	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
SRF [55]	Trained on DTU	0.250	0.591	12.34	0.313	0.671	0.304	15.32	0.171
PixelNeRF [49]		0.272	0.682	7.93	0.461	0.695	0.270	16.82	0.147
MVSNeRF [56]		0.557	0.356	17.25	0.171	0.769	0.197	18.63	0.113
SRF ft. [55]	Fine-tuned per Scene	0.436	0.529	17.07	0.203	0.698	0.281	15.68	0.162
PixelNeRF ft. [49]		0.438	0.512	16.17	0.217	0.710	0.269	18.95	0.125
MVSNeRF ft. [56]		0.584	0.327	17.88	0.157	0.769	0.197	18.54	0.113
Mip-NeRF [57]	Based on NeRF Optimized per Scene	0.351	0.495	14.62	0.246	0.571	0.353	8.68	0.323
DietNeRF [10]		0.370	0.496	14.94	0.240	0.633	0.314	11.85	0.243
RegNeRF [7]		0.587	0.336	19.08	0.149	0.745	0.190	18.89	0.112
FreeNeRF [8]		0.612	0.308	19.63	0.134	0.787	0.182	19.92	0.098
SparseNeRF [12]		0.624	0.328	19.86	0.127	0.769	0.201	19.55	0.102
3DGS [2]	Based on 3DGS Optimized per Scene	0.456	0.385	14.97	0.208	0.795	0.178	15.06	0.136
FSGS [58]		0.682	0.248	20.43	0.108	0.825	0.145	17.69	0.101
DNGaussian [9]		0.591	0.294	19.12	0.132	0.790	0.176	18.91	0.102
DNGaussian †[9]		0.687	0.228	19.94	0.109	-	-	-	-
Ours		0.702	0.207	20.44	0.101	0.856	0.121	19.99	0.077
		±0.001	±0.001	±0.08	±0.001	±0.001	±0.001	±0.10	±0.001

†: Using SfM initialization same as 3DGS, FSGS and Ours for fair comparisons.

Total Training Objectives. Overall, our training objectives can be divided into three parts: 1) The direct supervision $\mathcal{L}_1$ and $\mathcal{L}_{ssim}$ of the sparse input views, which are inherited from the vanilla 3DGS; 2) The supervision $\mathcal{L}_{IPSM}$ provided by diffusion priors using IPSM; 3) The supervision of depth and vision information $\mathcal{L}_{depth}$ and $\mathcal{L}_{geo}$ to support the inline priors and provide low-level inline guidance. The total training loss function can be summarized as

$$\mathcal{L} = \lambda_1 \mathcal{L}_1 + \lambda_{ssim} \mathcal{L}_{ssim} + \lambda_{depth} \mathcal{L}_{depth} + \lambda_{geo} \mathcal{L}_{geo} + \lambda_{IPSM} \mathcal{L}_{IPSM}. \qquad (17)$$

More training details are shown in the Appendix A.2.

4 Experiments

4.1 Experiments Settings

Datasets and Metrics. We evaluate our method on the LLFF [22] and DTU dataset [54]. The LLFF dataset involves 8 forward-facing scenes and we select 3 training views following prevailing works [8, 7]. On the DTU dataset, we choose the 15 testing scenes, and 3 training views whose IDs are 25, 22, and 28, following RegNeRF [7]. Following prevailing works [7–9] to focus on the object-of-interest for the DTU dataset, we also remove the background with the mask of objects when evaluating. Aligning with the protocol of baselines, we apply the downsampling rate of 8 and 4 on the LLFF and DTU datasets respectively. We evaluate the reconstruction quality using SSIM [59], LPIPS [60], and PSNR. Following DNGaussian [9] and FreeNeRF [8], we also report AVGE for a comprehensive evaluation of the reconstruction quality. The AVGE is calculated by the geometric mean of $\sqrt{1 - \text{SSIM}}$, LPIPS, and MSE $= 10^{-\text{PSNR}/10}$. The experiments are conducted 3 times and we report the mean and standard deviation. More details about datasets, *e.g.*the sparsity of training views and train-test split protocols, can be found in Appendix A.1.

Implementation details. Our method is built on 3DGS instead of NeRF due to the advantages of 3DGS on high training speed and real-time rendering. Following prevailing works [8, 9], the camera poses are known before optimization. The initialized point clouds are estimated by Structure from Motion (SfM) [61] only using the given sparse input views. The total training process involves 10K iterations for experiments on all datasets. The guidance of pseudo views starts from 2K iteration and ends at 9.5K iteration. Following FSGS, we introduce the proximity-guided Gaussian unpooling operation [58] and retain the high tolerance for large Gaussian points without size thresholds. For the score distillation methods, we randomly select one of 3 training views to generate BLIP-based
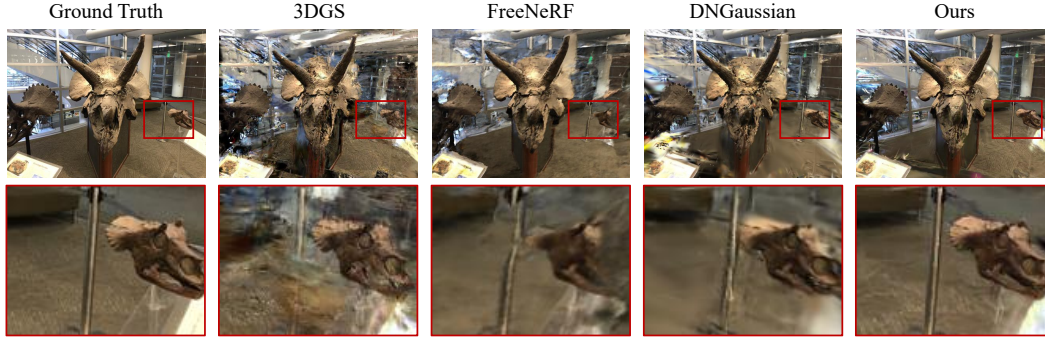
Figure 4: **Qualitative comparison on the LLFF dataset**.

[62] text prompts. Background priors are introduced on DTU for accurately reconstructing the object-of-interest. All experimental results are obtained on a single RTX 3090. More training details and experimental environments can be found in Appendix A.2 and A.3.

Baselines. Following prevailing works, we compare our method with the state-of-the-art methods, *i.e.* SRF [55], PixelNeRF [49], MVSNeRF [56], Mip-NeRF [57], DietNeRF [10], RegNeRF [7], FreeNeRF [8], SparseNeRF [12], the vanilla 3DGS [2], FSGS [58] and DNGaussian [9] as our baselines. Except for the reproduced results of the 3DGS [2] on the LLFF dataset, and 3DGS [2] and FSGS [58] on the DTU dataset, the rest are based on the values reported. Since the original DNGaussian uses random initialization, while other 3DGS methods use SfM [61], we also report the provided LLFF results of using SfM [61]. Reproduction details can be found in the Appendix A.4.

4.2 Comparison with Other Methods

LLFF. The quantitative results on the LLFF dataset [22] are shown in Tab. 1. Our method shows significant improvement and achieves the best reconstruction quality among state-of-the-art methods under multi-metric evaluation. For the NeRF-based methods, SSIM of our method is improved by $+12.5\%$ compared to SparseNeRF [12], and LPIPS is improved by $+32.79\%$ compared to FreeNeRF [8], which are the state-of-the-art in the NeRF-based methods respectively. For the 3DGS-based methods, the AVGE of our method is improved by $+6.48\%$ and $+7.34\%$ compared to the state-of-the-art FSGS [58] and DNGaussian †[9] respectively. Note that the vanilla DNGaussian uses random initialization, but the 3DGS, FSGS, and our method use SfM initialization. Thus, we also report the provided results of SfM-initialized DNGaussian which is denoted by †. The qualitative results are shown in Fig. 4. Due to the lack of external priors, 3DGS [2] and FreeNeRF [8] show the optimization tendencies of 3D representations themselves, which are high-frequency artifacts and low-frequency smoothness respectively. Although DNGaussian [9] using external depth prior can suppress artifacts, it only uses coarse-grained depth guidance and lacks fine-grained visual guidance, so the rendered image lacks high-frequency information. Our approach achieves improvements in both visual and geometric quality.

DTU. Similar performances of the quantitative results on the DTU dataset [54] are shown in Tab. 1. The AVGE of our method is improved by $+23.76\%$ compared to FSGS [58] and $+21.43\%$ compared to FreeNeRF [8]. Note that DNGaussian [9] does not provide the corresponding parameter settings for using SfM [61] initialization on the DTU dataset [54]. The qualitative results are shown in Fig. 5. SparseNeRF [12]

Figure 5: **Qualitative comparison on DTU**.

and DNGaussian [9], which only use depth priors, cannot obtain guidance on visual texture details, causing optimization difficulties. Our IPSM-Gaussian using diffusion priors can obtain textured details of reconstruction close to the Ground Truth.
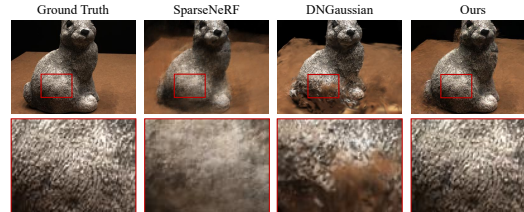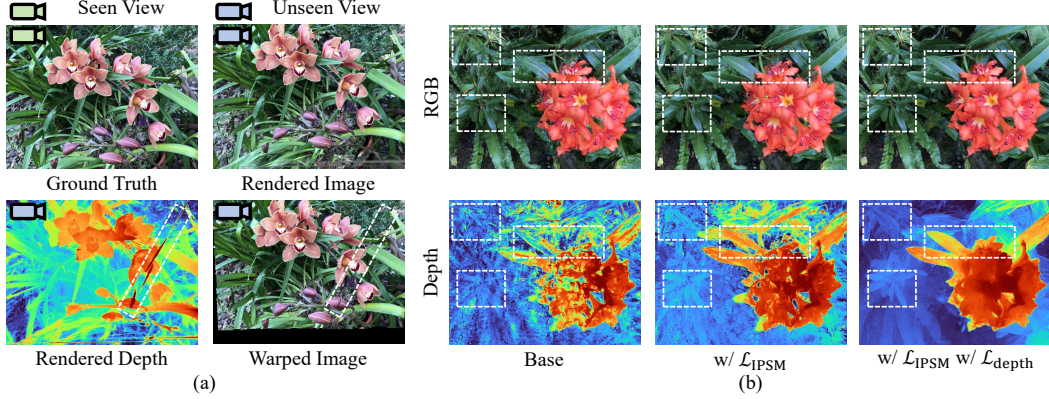
Details of reported experimental results are shown in Appendix B.3. More rendered novel views and qualitative comparisons can be found in the Appendix B.8.

Table 2: **Ablation Study** on the LLFF dataset with 3-views setting.

| w/ $\mathcal{L}_{\text{IPSM}}$ | | w/ $\mathcal{L}_{\text{depth}}$ | w/ $\mathcal{L}_{\text{geo}}$ | SSIM↑ | LPIPS↓ | PSNR↑ | AVGE↓ |
w/ $\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_1}$	w/ $\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_2}$						
				0.625 ±0.008	0.254 ±0.007	19.00 ±0.12	0.125 ±0.003
✓				0.636 ±0.004	0.245 ±0.003	19.22 ±0.02	0.121 ±0.001
✓	✓			0.670 ±0.001	0.229 ±0.002	19.60 ±0.11	0.113 ±0.001
✓	✓	✓		0.697 ±0.002	0.211 ±0.001	20.20 ±0.03	0.104 ±0.001
✓	✓	✓	✓	0.702 ±0.001	0.207 ±0.001	20.44 ±0.08	0.101 ±0.001

Figure 6: **(a)** Impact of depth error on the inline prior. **(b)** Ablation of IPSM and depth regularizations.

4.3 Ablation Study

We conduct detailed ablations of regularization terms on the LLFF dataset [22] shown in Tab. 2. We can notice that the first two regularization terms, *i.e.* IPSM and depth, provide significant improvements. The first three lines demonstrate the promoting effect of our proposed IPSM on the reconstruction quality of 3D representations, *e.g.* using IPSM boosts 9.8% on the LPIPS and 9.6% on the AVGE compared to the Base. It is worth noting that since the inline prior requires an accurate rendering depth from the unseen perspective shown in Eq. 8. The impact of depth error on inline priors is shown in Fig. 6 (a). However, the diffusion priors, as a kind of visual supervision, cannot provide direct depth geometry guidance, so an additional external depth prior needs to be introduced, which can support the accuracy of inline prior to further provide performance improvements. In Fig. 6 (b), we show the visual and geometry improvements of IPSM and depth regularization. The last line in Tab. 2 introduces the geometry consistency regularization for providing pixel-wise guidance, which shows a steady improvement. More additional ablations are detailed in the Appendix B.4.

4.4 Comparison to SDS

As shown in Fig. 1, SDS guidance is hard to provide effective supervision but tends to hinder reconstruction due to the mode deviation we have analyzed. Due to the too-strong semantic visual supervision of SDS(CFG=100), the performance increases significantly in the final 500 iterations after the 2K-9.5K prior-added period instead. In this section, we report the final evaluated performance comparison of *Base* (without any regularization), *w/ SDS(CFG=7.5)*, *w/ SDS(CFG=100)*, and *w/ IPSM(CFG=7.5)* in Tab 3. Except for SDS (CFG=7.5), which can provide a limited improvement in structural similarity compared to the Base, the other performances show a downward trend, which is colored by blue. However, IPSM can provide considerable improvements in multiple metrics which are colored by red. It is supposed to be noted that all the experiments of SDS shown in Fig. 1 and Tab. 3 are under the same experimental setting. We also present the qualitative comparison of SDS. As shown in Fig. 7 (a), the guidance of SDS will produce the imaginary reconstruction caused by mode deviation when using the diffusion prior directly. This property is reasonable and acceptable in text-to-3D generation tasks, but it fails in specific scene reconstructions limited by sparse views. As shown in Fig. 7 (b), we can observe that SDS will also produce large floaters during optimization,
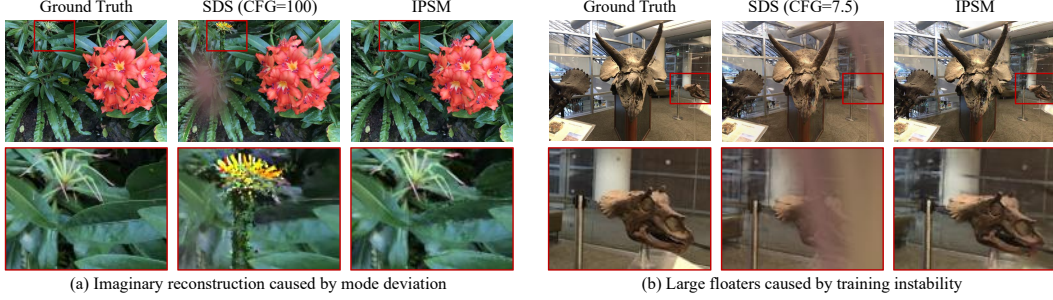
(a) Imaginary reconstruction caused by mode deviation

(b) Large floaters caused by training instability

Figure 7: Qualitative comparison with SDS.

Table 3: **Comparison to SDS** on the LLFF dataset with 3-views setting.

Setting	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
Base	0.625 (+0.00%)	0.254 (+0.00%)	19.00 (+0.00%)	0.125 (+0.00%)
w/ SDS(CFG=7.5)	0.647 (+3.52%)	0.267 (-5.12%)	18.80 (-1.05%)	0.128 (-2.40%)
w/ SDS(CFG=100)	0.576 (-7.84%)	0.367 (-44.49%)	17.53 (-7.74%)	0.162 (-29.60%)
w/ IPSM(CFG=7.5)	**0.670 (+7.20%)**	**0.229 (+9.84%)**	**19.60 (+3.16%)**	**0.113 (+9.60%)**

which indicates the characteristic of its training instability since SDS overlooks the inline prior of sparse views and is hard to provide stable guidance towards target mode.

The experiments are conducted 3 times reporting the average results, and use the weight of 2.0 and the VAE encoder same as IPSM for fair comparisons. Since the feature domains of Stable Diffusion and Stable Diffusion Inpainting are identical, using the original VAE of Stable Diffusion shows similar performance, which is reported in the Appendix B.2. We have also analyzed the training instability of SDS additionally in Appendix B.1. Furthermore, we discuss the effects of using view-conditioned diffusion prior for SDS in Appendix B.6.

5 Conclusions and Limitations

In this paper, we start by revisiting the phenomenon where SDS not only fails to improve optimization in sparse-view 3D reconstruction but degrades performance. We present a comprehended analysis of SDS from a mode-seeking perspective. Based on these observations and analyses, we propose Inline Prior Guided Score Matching (IPSM), which utilizes the sparse-view input as the inline prior to rectifying the rendered image distribution. IPSM utilizes the rectified distribution as an intermediate state to decompose the mode-seeking optimization objective of SDS for controlling the optimization direction of mode-seeking to suppress mode deviation. We further propose the pipeline IPSM-Gaussian, which selects 3DGS as the backbone and incorporates IPSM with depth and geometry regularization for boosting IPSM. Experimental results on different public datasets show that our method achieves state-of-the-art reconstruction quality compared to other current methods.

The limitation of our method is that the rectified distribution needs to match the same feature space as the diffusion prior, which restricts the range of inpainting models used for the rectified distribution, thereby limiting the scalability and performance of our method. An alternative improvement could be substituting the pre-trained inpainting models with fine-tuning the diffusion prior like VSD. However, it would further increase the computational complexity of the method. We leave it as our future work.

Acknowledgement

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[2] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–14, 2023.

[3] A. Rabby and C. Zhang, "Beyondpixels: A comprehensive review of the evolution of neural radiance fields," *arXiv preprint arXiv:2306.03000*, 2023.

[4] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li, "Nerf: Neural radiance field in 3d vision, a comprehensive review," *arXiv preprint arXiv:2210.00379*, 2022.

[5] G. Chen and W. Wang, "A survey on 3d gaussian splatting," *arXiv preprint arXiv:2401.03890*, 2024.

[6] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, "3d gaussian as a new vision era: A survey," *arXiv preprint arXiv:2402.07181*, 2024.

[7] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5480–5490, 2022.

[8] J. Yang, M. Pavone, and Y. Wang, "Freenerf: Improving few-shot neural rendering with free frequency regularization," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8254–8263, 2023.

[9] J. Li, J. Zhang, X. Bai, J. Zheng, X. Ning, J. Zhou, and L. Gu, "Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20775–20785, 2024.

[10] A. Jain, M. Tancik, and P. Abbeel, "Putting nerf on a diet: Semantically consistent few-shot view synthesis," in *International Conference on Computer Vision (ICCV)*, pp. 5885–5894, 2021.

[11] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12882–12891, 2022.

[12] G. Wang, Z. Chen, C. C. Loy, and Z. Liu, "Sparsenerf: Distilling depth ranking for few-shot novel view synthesis," in *International Conference on Computer Vision (ICCV)*, pp. 9065–9076, 2023.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

[14] H. Xiong, S. Muttukuru, R. Upadhyay, P. Chari, and A. Kadambi, "Sparsegs: Real-time 360 {\deg} sparse view synthesis using gaussian splatting," *arXiv preprint arXiv:2312.00206*, 2023.

[15] R. Wu, B. Mildenhall, P. Henzler, K. Park, R. Gao, D. Watson, P. P. Srinivasan, D. Verbin, J. T. Barron, B. Poole, *et al.*, "Reconfusion: 3d reconstruction with diffusion priors," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21551–21561, 2024.

[16] X. Liu, J. Chen, S.-H. Kao, Y.-W. Tai, and C.-K. Tang, "Deceptive-nerf/3dgs: Diffusion-generated pseudo-observations for high-quality sparse-view reconstruction," in *ECCV 2024 Workshop on Wild 3D: 3D Modeling, Reconstruction, and Generation in the Wild*, 2024.

[17] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning (ICML)*, pp. 2256–2265, 2015.

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 6840–6851, 2020.

[19] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations (ICLR)*, 2021.

[20] F. Bao, C. Li, J. Zhu, and B. Zhang, "Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models," in *International Conference on Learning Representations (ICLR)*, 2021.

[21] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations (ICLR)*, 2021.

[22] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.

[23] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," in *International Conference on Learning Representations (ICLR)*, 2023.

[24] G. Qian, J. Mai, A. Hamdi, J. Ren, A. Siarohin, B. Li, H.-Y. Lee, I. Skorokhodov, P. Wonka, S. Tulyakov, *et al.*, "Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors," in *International Conference on Learning Representations (ICLR)*, 2024.

[25] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu, "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023.

[26] Y. Liang, X. Yang, J. Lin, H. Li, X. Xu, and Y. Chen, "Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6526, 2024.

[27] S. Avidan and A. Shashua, "Novel view synthesis in tensor space," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1034–1040, IEEE, 1997.

[28] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[29] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5501–5510, 2022.

[30] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision (ECCV)*, pp. 333–350, 2022.

[31] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10318–10327, 2021.

[32] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7210–7219, 2021.

[33] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8248–8258, 2022.

[34] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-nerf: Structured view-dependent appearance for neural radiance fields," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5481–5490, 2022.

[35] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5470–5479, 2022.

[36] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *International Conference on Computer Vision (ICCV)*, pp. 5741–5751, 2021.

[37] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 15651–15663, 2020.

[38] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, "Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21634–21643, 2024.

[39] Z. Shao, Z. Wang, Z. Li, D. Wang, X. Lin, Y. Zhang, M. Fan, and Z. Wang, "Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1606–1616, 2024.

[40] Z. Li, Z. Zheng, L. Wang, and Y. Liu, "Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19711–19722, 2024.

[41] Y. Jiang, Z. Shen, P. Wang, Z. Su, Y. Hong, Y. Zhang, J. Yu, and L. Xu, "Hifi4g: High-fidelity human performance rendering via compact gaussian splatting," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19734–19745, 2024.

[42] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," in *International Conference on Learning Representations (ICLR)*, 2023.

[43] S. Seo, Y. Chang, and N. Kwak, "Flipnerf: Flipped reflection rays for few-shot novel view synthesis," in *International Conference on Computer Vision (ICCV)*, pp. 22883–22893, 2023.

[44] J. Song, S. Park, H. An, S. Cho, M.-S. Kwak, S. Cho, and S. Kim, "Därf: Boosting radiance fields from sparse input views with monocular depth adaptation," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023.

[45] N. Somraj, A. Karanayil, and R. Soundararajan, "Simplenerf: Regularizing sparse input neural radiance fields with simpler solutions," in *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–11, 2023.

[46] M.-S. Kwak, J. Song, and S. Kim, "Geconerf: Few-shot neural radiance fields via geometric consistency," in *International Conference on Machine Learning (ICML)*, pp. 18023–18036, 2023.

[47] D. Chen, Y. Liu, L. Huang, B. Wang, and P. Pan, "Geoaug: Data augmentation for few-shot nerf with geometry constraints," in *European Conference on Computer Vision (ECCV)*, pp. 322–337, 2022.

[48] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *International Conference on Computer Vision (ICCV)*, pp. 12179–12188, 2021.

[49] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4578–4587, 2021.

[50] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in *International Conference on Computer Vision (ICCV)*, pp. 9298–9309, 2023.

[51] K. Sargent, Z. Li, T. Shah, C. Herrmann, H.-X. Yu, Y. Zhang, E. R. Chan, D. Lagun, L. Fei-Fei, D. Sun, *et al.*, "Zeronvs: Zero-shot 360-degree view synthesis from a single image," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9420–9429, 2024.

[52] Y. Kant, A. Siarohin, M. Vasilkovsky, R. A. Guler, J. Ren, S. Tulyakov, and I. Gilitschenski, "invs: Repurposing diffusion inpainters for novel view synthesis," in *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–12, 2023.

[53] B. Tang, J. Wang, Z. Wu, and L. Zhang, "Stable score distillation for high-quality 3d generation," *arXiv preprint arXiv:2312.09305*, 2023.

[54] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, "Large scale multi-view stereopsis evaluation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 406–413, 2014.

[55] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, "Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7911–7920, 2021.

[56] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo," in *International Conference on Computer Vision (ICCV)*, pp. 14124–14133, 2021.

[57] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *International Conference on Computer Vision (ICCV)*, pp. 5855–5864, 2021.

[58] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang, "Fsgs: Real-time few-shot view synthesis using gaussian splatting," in *European Conference on Computer Vision (ECCV)*, pp. 145–163, 2025.

[59] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[60] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

[61] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113, 2016.

[62] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International Conference on Machine Learning (ICML)*, pp. 12888–12900, 2022.

[63] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

[64] J. Zhang, J. Li, X. Yu, L. Huang, L. Gu, J. Zheng, and X. Bai, "Cor-gs: sparse-view 3d gaussian splatting via co-regularization," in *European Conference on Computer Vision (ECCV)*, pp. 335–352, 2025.

[65] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny, "Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction," in *International Conference on Computer Vision (ICCV)*, pp. 10901–10911, 2021.

[66] B. Xiao and S.-C. Kang, "Development of an image data set of construction machines for deep learning object detection," *Journal of Computing in Civil Engineering*, vol. 35, no. 2, p. 05020005, 2021.

[67] B. Xiao, Y. Wang, and S.-C. Kang, "Deep learning image captioning in construction management: a feasibility study," *Journal of Construction Engineering and Management*, vol. 148, no. 7, p. 04022049, 2022.

[68] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: learning view synthesis using multiplane images," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–12, 2018.

[69] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 25278–25294, 2022.

Appendix

A Experimental Details

A.1 Datasets Details

LLFF Dataset. The LLFF dataset [22] is a forward-facing dataset, which contains 8 challenging scenes. Following FreeNeRF [8] and DNGaussian [9], we select every 8th image for testing and evenly sample the remaining images for 3 input views. Following DNGaussian [9], we downsample the resolutions of images to $8\times$ for both training and testing. In Tab. 4, we report the level of sparsity for intuitive exhibition. The Original Training Views means the number of training views for the original dense-view NVS, and the Sparsity of 3 Views means the ratio of 3 input sparse views to the Original Training Views.

Table 4: **Level of sparsity in the input views of the LLFF dataset**.

Dataset	Sparsity	Fer.	Flo.	For.	Hor.	Lea.	Orc.	Roo.	Tre.	AVG
	Total Views	20	34	42	62	26	25	41	55	38.125
LLFF	Original Training Views	17	29	36	54	22	21	35	48	32.750
	Test Views	3	5	6	8	4	4	6	7	5.375
	Sparsity of 3 Views	17.65%	10.34%	8.33%	5.56%	13.64%	14.29%	8.57%	6.25%	9.16%

DTU Dataset. The DTU dataset [54] contains 124 scenes in total. PixelNeRF [49] and MVSNeRF [56] split the DTU dataset [54] into 88 training scenes for pre-training and 15 testing scenes for per-scene fine-tuning. Following RegNeRF [7], FreeNeRF [8], and DNGaussian [9], we only use the selected 15 testing scenes for optimization. The IDs of testing scenes are: 8, 21, 30, 31, 34, 38, 40, 41, 45, 55, 63, 82, 103, 110, and 114. For each scene optimization with the 3-view setting, the IDs of images served as sparse views for training are 25, 22, and 28. The IDs of images that served as testing novel views for evaluation are 1, 2, 9, 10, 11, 12, 14, 15, 23, 24, 26, 27, 29, 30, 31, 32, 33, 34, 35, 41, 42, 43, 45, 46, 47. Following FreeNeRF [8] and DNGaussian [9], all metrics for the evaluation of the DTU dataset are computed with the object mask. Following DNGaussian [9], we use the estimated pose which is exactly the same as DNGaussian [9]. Following RegNeRF [34], we downsample the resolutions of images to $4\times$ for both training and testing.

A.2 Training Details

SfM Initialization. Following FSGS [58], we use SfM [61] initialization with 3 input sparse views only for the 3D Gaussian points initialization. However, sometimes SfM will fail when using sparse input images. In practice, scan 30 and scan 110 of the DTU dataset cannot extract enough features for initial point cloud prediction, so we only perform random initialization on these two scenes. We perform SfM [61] initialization on the remaining scenes of the DTU dataset [54] and all scenes of the LLFF dataset [22]. It is supposed to be noted that SfM [61] initialization will significantly improve the final reconstruction quality, so random initialization of these two scenarios will not improve our final performance but must be dealt with due to factual limitations.

Gaussian Unpooling. Following FSGS [58], we introduce the operation of Gaussian unpooling for filling the spaces uniformly and geometry fitting. The Gaussian unpooling determines whether to add a new Gaussian point by calculating the K-nearest neighbor graph structure of the Gaussian point and its corresponding Euclidean distance metric ($K = 3$ in practice). The SH coefficients of newly densified Gaussian points are set to 0. In this paper, both experiments of our method, corresponding ablations, and explorations on SDS using different diffusion priors adopt this operation.

Gaussian Size Threshold. The vanilla 3DGS [2] filters out Gaussian points with excessively large sizes, but in the case of sparse views, discarding these large-sized Gaussian points can lead to poor fitting of low-frequency regions during the optimization process. Following FSGS [58] and DNGaussian [9], we have eliminated this Gaussian point size filtering operation, which significantly enhances the performance of sparse-view 3D reconstruction.

Training Strategy. Following FSGS [58], the maximum degree of SH coefficients is set to 3, and we level up the SH degree every 500 iterations. The total training iterations is 10K for all datasets. Following FSGS [58], we introduce the warm-up period of 500 iterations for the beginning of the pseudo views supervision, *i.e.*the 2K iteration, and we reduce the weight of the depth regularization of seen views to 0.001 after the end of pseudo views supervision, *i.e.*the 9.5K iteration.

Background Prior. Following FreeNeRF [8] and DNGaussian [9] on the optimization prior based on pixel value (*i.e.*FreeNeRF: white and black background prior; DNGaussian: strategic masking of black backgrounds), we introduce the mask for the white and black background served as an additional prior on the DTU dataset [54] for the selection of previous work [7–9] in accurately reconstructing the object-of-interest [7]. Specifically, we mask the values of images that are less than $30/255 \approx 0.1176$ (Following DNGaussian [9], the vertical scan rectangles are also introduced to reduce mask of black regions), and larger than 0.99 for L1 losses of all scenarios on the DTU dataset.

Gaussian Points Controlling. The opacity of Gaussian points would be reset at the 2K iteration. The opacity would not be reset for the following iterations on the LLFF dataset [22] and the opacity would be reset every 1K iterations for the following iterations on the DTU dataset [54] due to the easy over-fitting property associated with large view differences. Besides, the Gaussian points are densified every 100 iterations and pruned every 500 iterations for all datasets.

Text Prompts. For the experiment of score distillation methods, we randomly selected one of the 3 training images and used BLIP [62] to extract the corresponding text prompts. For fair comparisons, the text prompts corresponding to each scene on all datasets of all score distillation methods relying on text prompts are identical.

A.3 Hyper-parameters

For the inline prior, the mask threshold τ is set to 0.3 for IPSM regularization and 0.1 for the geometry consistency regularization, since the latter is at pixel level thus requiring more strict constraints. For the diffusion priors guidance, the weight λ_{IPSM} of IPSM regularization $\mathcal{L}_{\text{IPSM}}$ is set to 2.0 for all datasets, and the parameter η_r for controlling $\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_1}$ and $\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_2}$ is set to 0.1 for all datasets. The parameter η_d for controlling the depth guidance of seen views and pseudo unseen views is set to 0.1 for all datasets. On the LLFF dataset [22], the weight λ_{depth} of depth regularization $\mathcal{L}_{\text{depth}}$ is set to 0.5 and the weight λ_{geo} of the geometry consistency regularization $\mathcal{L}_{\text{geo}}$ is set to 2.0. λ_{ssim} is set to 0.2 and $\lambda_1 = 1 - \lambda_{\text{ssim}}$ following 3DGS [2]. On the DTU dataset [54], following DNGaussian [9], we reduce λ_1 to 0.4 (*i.e.*increase λ_{ssim} to 0.6), and at the same time reduce λ_{depth} and λ_{geo}, both of which are multiplied by 0.1.

A.4 Reproduction of Baselines

3DGS. We use the vanilla 3DGS [2] for reproduction on the LLFF [22] and DTU [54] dataset. We do not make any other changes except for the necessary operations to render depth and convert dense views to sparse-view training. In addition, all our experiments use the rasterizer of FSGS [58] to ensure fairness, although this rasterizer has the same function as the rasterizer of 3DGS [2]. Meanwhile, the reported results of 3DGS [2] are also obtained with the SfM [61] initialization which is the same as ours.

FSGS. We use the official code of FSGS [58] to reproduce the results on the DTU dataset [54]. Since FSGS [58] performed experiments on the LLFF dataset [22], we report the results provided by FSGS [58]. Since FSGS [58] does not conduct experiments on the DTU dataset [54], we reproduce it and add the white & black mask prior to it, which is the same as ours and detailed in Appendix A.2. On the DTU dataset [54], we adopt the hyper-parameters of FSGS [58] on the MipNeRF-360 dataset [35] (*i.e.*the weight of depth regularization on the pseudo views is 0.03, the weight of depth regularization on the seen views is 0.05, and the supervision interval on the pseudo unseen views is 10) because we observe that the selected hyper-parameters are more suitable for non-forward-facing datasets and can achieve better performance than directly using the hyper-parameters on the LLFF dataset [22]. The reproduction of FSGS [58] on the DTU dataset [54] is also enhanced by the SfM [61] initialization same as ours.

DNGaussian. The original DNGaussian [9] does not use SfM [61] initialization. However, directly changing the random initialization to SfM [61] initialization without changing the hyper-parameters

Table 5: **Training instability of SDS**. Detailed data of the reported results in the main manuscript.

Setting	Experiments	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
Base	Exp. 1	0.619	0.260	18.89	0.128
	Exp. 2	0.622	0.256	18.95	0.126
	Exp. 3	0.636	0.244	19.18	0.121
	Mean ±Std.	0.625 ±0.008	0.254 ±0.007	19.00 ±0.12	0.125 ±0.003
w/ SDS(CFG=7.5)	Exp. 1	0.645	0.262	18.91	0.126
	Exp. 2	0.637	0.282	18.29	0.136
	Exp. 3	0.659	0.255	19.20	0.121
	Mean ±Std.	0.647 ±0.009	0.267 ±0.012	18.80 ±0.38	0.128 ±0.006
w/ SDS(CFG=100)	Exp. 1	0.571	0.375	17.20	0.167
	Exp. 2	0.577	0.364	17.62	0.160
	Exp. 3	0.578	0.362	17.76	0.158
	Mean ±Std.	0.576 ±0.003	0.367 ±0.006	17.53 ±0.24	0.162 ±0.004
w/ IPSM(CFG=7.5)	Exp. 1	0.669	0.231	19.55	0.114
	Exp. 2	0.670	0.229	19.50	0.114
	Exp. 3	0.672	0.227	19.76	0.111
	Mean ±Std.	**0.670 ±0.001**	**0.229 ±0.002**	**19.60 ±0.11**	**0.113 ±0.001**

makes it difficult to provide sufficient performance improvement due to the incompatibility of a series of hyper-parameters such as the learning rate. Therefore, we only report the results provided by DNGaussian [9] using SfM [61] initialization on the LLFF dataset [22] and do not report the reproduced results of directly using the original random initialization hyper-parameters with SfM [61] initialization on the DTU dataset [54].

A.5 Experimental Environments and Computing Resources

All the experiments are conducted on a single RTX 3090 with CUDA 11.3. The training time of IPSM-Gaussian is about 1 hour on the RTX 3090, which is mainly due to the inference time of the diffusion model itself. For pseudo view supervision from 2K to 9.5K iteration, we need to perform two inferences of the diffusion model in each iteration to calculate $\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_1}$ and $\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_2}$.

B Additional Experimental Results

B.1 Training Instability of SDS

In practice, it can be noticed that using SDS directly can produce training instability, which is shown in Tab. 5. Using SDS [23] causes more performance differences between independent experiments, *i.e.* training instability. The standard deviation of 3 independent experiments using SDS (CFG=7.5) is about 9 times that of IPSM on the SSIM, about 6 times that of IPSM on the LPIPS, and about 3 times that of IPSM on the PSNR. The standard deviation of 3 independent experiments using SDS (CFG=100) is about 3 times that of IPSM on the SSIM, about 3 times that of IPSM on the LPIPS, and about 2 times that of IPSM on the PSNR. This is because SDS [23], as a score distillation technique guided by text-prompt semantics, overlooks the inline priors present in the sparse-view 3D reconstruction task from a limited number of input viewpoints. Owing to the high information entropy inherent in the text, it is hard for SDS to provide stable guidance of diffusion priors towards the target mode during training, leading to instability in the final reconstruction quality.

To further illustrate the training instability of SDS [23], additional experiments of SDS [23] on the LLFF dataset [22] are conducted 3 times, which is shown in Tab. 6. It is supposed to be noted that the

Table 6: **Training instability of SDS**. Detailed data of the additional re-conducted results of SDS.

Setting	Experiments	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
w/ SDS(CFG=7.5)	Exp. 1	0.643	0.272	18.78	0.129
	Exp. 2	0.635	0.282	18.30	0.136
	Exp. 3	0.661	0.251	19.31	0.120
	Mean ±Std.	0.647 ±0.011	0.268 ±0.013	18.80 ±0.41	0.128 ±0.007
w/ SDS(CFG=100)	Exp. 1	0.575	0.369	17.77	0.159
	Exp. 2	0.573	0.369	17.58	0.162
	Exp. 3	0.583	0.361	18.03	0.154
	Mean ±Std.	0.577 ±0.004	0.367 ±0.004	17.79 ±0.19	0.158 ±0.003
w/ IPSM(CFG=7.5)	Exp. 1	0.671	0.228	19.63	0.113
	Exp. 2	0.673	0.227	19.68	0.112
	Exp. 3	0.670	0.230	19.55	0.113
	Mean ±Std.	**0.671 ±0.001**	**0.228 ±0.001**	**19.62 ±0.05**	**0.113 ±0.001**

Table 7: **Comparison to SDS** on the LLFF dataset 3-views setting with different VAE settings.

Setting	VAE Setting	Experiments	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
Base	-	Mean ±Std.	0.625 ±0.008	0.254 ±0.007	19.00 ±0.12	0.125 ±0.003
w/ SDS (CFG=7.5)	Same VAE	Mean ±Std.	0.647 ±0.009	0.267 ±0.012	18.80 ±0.38	0.128 ±0.006
	Origin VAE	Exp. 1	0.667	0.240	19.27	0.118
		Exp. 2	0.646	0.271	18.95	0.127
		Exp. 3	0.618	0.328	17.53	0.153
		Mean ±Std.	0.644 ±0.020	0.279 ±0.037	18.58 ±0.75	0.133 ±0.015
w/ SDS (CFG=100)	Same VAE	Mean ±Std.	0.576 ±0.003	0.367 ±0.006	17.53 ±0.24	0.162 ±0.004
	Origin VAE	Exp. 1	0.578	0.364	17.97	0.156
		Exp. 2	0.568	0.374	17.55	0.163
		Exp. 3	0.567	0.377	17.52	0.164
		Mean ±Std.	0.571 ±0.005	0.372 ±0.005	17.68 ±0.21	0.161 ±0.004
w/ IPSM (CFG=7.5)	-	Mean ±Std.	**0.670 ±0.001**	**0.229 ±0.002**	**19.60 ±0.11**	**0.113 ±0.001**

reported results of SDS [23] in the main manuscript are **NOT** out of the re-conducted experiments. In 3 re-conducted experiments, we can still observe the instability exhibited by SDS compared to our method. The standard deviation of 3 independent experiments using SDS (CFG=7.5) is about 11 times that of IPSM on the SSIM, about 13 times that of IPSM on the LPIPS, and about 8 times that of IPSM on the PSNR. The standard deviation of 3 independent experiments using SDS (CFG=100) is about 4 times that of IPSM on the SSIM, about 4 times that of IPSM on the LPIPS, and about 4 times that of IPSM on the PSNR.

B.2 SDS with Different VAE

For a fair comparison, we report the results of SDS on the LLFF dataset [22] using the VAE same to us, *i.e.* the VAE of Stable Diffusion Inpainting v1-5 [63], in the main manuscript. To demonstrate the reconstruction quality of SDS [23] in more detail, we report the experimental results of using the

Table 8: **Detials of quantitative comparisons** with other methods.

Methods	Experiments	LLFF				DTU			
		SSIM↑	LPIPS↓	PSNR↑	AVGE↓	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
Ours	Exp. 1	0.703	0.207	20.55	0.100	0.857	0.119	20.13	0.076
	Exp. 2	0.702	0.207	20.40	0.101	0.854	0.121	19.92	0.078
	Exp. 3	0.701	0.208	20.38	0.101	0.855	0.121	19.93	0.078
	Mean	0.702	0.207	20.44	0.101	0.856	0.121	19.99	0.077

Table 9: **Details of ablation study** on the LLFF dataset with 3-views setting.

w/ $\mathcal{L}_{\text{IPSM}}$		w/ $\mathcal{L}_{\text{depth}}$	w/ $\mathcal{L}_{\text{geo}}$	Experiments	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
w/ $\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_1}$	w/ $\mathcal{L}_{\text{IPSM}}^{\mathcal{G}_2}$							
				Mean	0.625	0.254	19.00	0.125
✓				Exp. 1	0.638	0.243	19.23	0.120
✓				Exp. 2	0.641	0.242	19.24	0.120
✓				Exp. 3	0.631	0.249	19.18	0.122
✓				Mean	0.636	0.245	19.22	0.121
✓	✓			Mean	0.670	0.229	19.60	0.113
✓	✓	✓		Exp. 1	0.697	0.210	20.23	0.103
✓	✓	✓		Exp. 2	0.699	0.211	20.22	0.103
✓	✓	✓		Exp. 3	0.695	0.212	20.16	0.104
✓	✓	✓		Mean	0.697	0.211	20.20	0.104
✓	✓	✓	✓	Mean	0.702	0.207	20.44	0.101

original VAE, *i.e.* the VAE of Stable Diffusion v1-5 [63], which is shown in Tab. 7. The experiments employing the original VAE, *i.e.* the VAE of Stable Diffusion v1-5, are also independently repeated thrice. It can be observed that VAE of Stable Diffusion v1-5 and VAE of Stable Diffusion Inpainting v1-5 exhibit nearly identical performances, with the VAE of Stable Diffusion v1-5 (CFG=7.5) even demonstrating greater instability. These experiments further elucidate the mode deviation issue and training instability problem in SDS [23].

B.3 Details of Reported Experimental Results

The details of the reported experimental results in the main manuscript are shown in Tab. 8 and Tab. 9. Tab. 8 shows the details corresponding to the mean and standard deviation of our method as described in Tab. 1 in the main manuscript, obtained from 3 independent experiments with 3-views setting on the LLFF dataset [22] and DTU dataset [54], respectively. Tab. 9 shows the details corresponding to the mean and standard deviation as described in Tab. 2 in the main manuscript, obtained from 3 independent experiments with 3-views setting on the LLFF dataset. Note that the individual results of the first and third row are shown in Tab. 5. The individual results of the last row are shown in Tab. 8.

B.4 Additional Ablation Results

To supplement more complete experimental results, we provide an additional ablation study using 3 views on the LLFF and DTU dataset in Tab. 10 and Tab. 11 respectively. We can see that $\mathcal{L}_{\text{depth}}$ presents a strong prior for optimization since it directly provides the 3D geometric guidance on 3D representations. Notably, although both $\mathcal{L}_{\text{geo}}$ and $\mathcal{L}_{\text{IPSM}}$ utilize re-projection techniques to introduce the 2D visual prior information of the sparse views to promote optimization, $\mathcal{L}_{\text{IPSM}}$ achieves satisfactory performance comparable to direct 3D guidance of $\mathcal{L}_{\text{depth}}$ as shown in Tab. 10 and Tab. 11. At the same time, it is difficult for $\mathcal{L}_{\text{geo}}$ to promote optimization independently without the assistance of other regularizations.

Besides, in repeated experiments, we also notice that both IPSM and depth regularization can promote the stability of training of 3D Gaussians. As shown in Tab. 10, both IPSM and depth

Table 10: **Additional ablation study** on the LLFF dataset with 3-views setting.

Setting	Experiments	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
Base	Mean ±Std.	0.625 ±0.008	0.254 ±0.007	19.00 ±0.12	0.125 ±0.003
Base + $\mathcal{L}_{\text{depth}}$	Exp. 1	0.687	0.212	20.08	0.105
	Exp. 2	0.690	0.210	20.18	0.104
	Exp. 3	0.687	0.212	20.10	0.105
	Mean ±Std.	0.688 ±0.001	0.211 ±0.001	20.12 ±0.04	0.105 ±0.001
Base + $\mathcal{L}_{\text{geo}}$	Exp. 1	0.651	0.235	19.35	0.117
	Exp. 2	0.643	0.240	19.14	0.120
	Exp. 3	0.661	0.225	19.55	0.113
	Mean ±Std.	0.652 ±0.007	0.233 ±0.006	19.35 ±0.17	0.117 ±0.003
Base + $\mathcal{L}_{\text{IPSM}}$	Mean ±Std.	0.670 ±0.001	0.229 ±0.002	19.60 ±0.11	0.113 ±0.001
Ours	Mean ±Std.	0.702 ±0.001	0.207 ±0.001	20.44 ±0.08	0.101 ±0.001

Table 11: **Additional ablation study** on the DTU dataset with 3-views setting.

Setting	Experiments	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
Base	Exp. 1	0.836	0.134	19.11	0.087
	Exp. 2	0.836	0.135	18.86	0.089
	Exp. 3	0.837	0.134	19.39	0.085
	Mean ±Std.	0.836 ±0.001	0.134 ±0.001	19.12 ±0.22	0.087 ±0.002
Base + $\mathcal{L}_{\text{depth}}$	Exp. 1	0.849	0.122	19.77	0.079
	Exp. 2	0.853	0.121	19.92	0.078
	Exp. 3	0.852	0.121	19.77	0.079
	Mean ±Std.	0.851 ±0.001	0.122 ±0.001	19.82 ±0.07	0.079 ±0.001
Base + $\mathcal{L}_{\text{geo}}$	Exp. 1	0.835	0.135	19.28	0.086
	Exp. 2	0.833	0.137	18.86	0.090
	Exp. 3	0.837	0.134	19.41	0.085
	Mean ±Std.	0.835 ±0.001	0.135 ±0.001	19.18 ±0.23	0.087 ±0.002
Base + $\mathcal{L}_{\text{IPSM}}$	Exp. 1	0.853	0.122	19.67	0.080
	Exp. 2	0.852	0.123	19.80	0.079
	Exp. 3	0.850	0.125	19.34	0.083
	Mean ±Std.	0.852 ±0.001	0.123 ±0.001	19.60 ±0.19	0.080 ±0.002
Ours	Mean ±Std.	0.856 ±0.001	0.121 ±0.001	19.99 ±0.10	0.077 ±0.001

regularization can greatly suppress the fluctuation of reconstruction results in structural similarity and perception evaluation quality, *i.e.* SSIM and LPIPS. However, unlike depth prior, IPSM has a limited suppression effect on the fluctuations of the pixel-level evaluation, *i.e.* PSNR, which is consistent with the randomness of the fluctuations of the baseline as shown in Tab. 10 and Tab. 11. This is because the depth prior participates in optimization throughout the training process (namely [0, 10K] iterations), while IPSM only participates in optimization in [2K, 9.5K] iterations. Due to the significant randomness of 3DGS itself under sparse views [64] (especially in more difficult scenarios in DTU compared to LLFF), the optimization of 3DGS itself in the first 2K training iterations may collapse in some scenarios, *e.g.* scan 103, 30, 82, which in turn affects the optimization guidance of the regularization term in subsequent optimizations. Even so, IPSM has a very significant improvement in SSIM and LPIPS compared to $\mathcal{L}_{\text{geo}}$ which also uses re-projection technology, and is comparable to direct 3D guidance of depth prior as shown in Tab. 11.

B.5 Additional Experiments with Different Input Views

More input views. Experimental results using more input views can further explore the robustness of our method when working with sparse views. We provide additional experimental results under 6

Table 12: **Quantitative comparisons with 6 input views** on the LLFF dataset.

Method	Pretrain	Experiments	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
Zip-NeRF *	-	-	0.764	0.221	20.71	0.097
RegNeRF * [7]	-	-	0.760	0.243	23.09	0.084
DiffusioNeRF *	✓	-	0.775	0.235	23.60	0.079
FreeNeRF * [8]	-	-	0.773	0.232	23.72	0.078
SimpleNeRF * [45]	-	-	0.737	0.296	23.05	0.091
ReconFusion * [15]	✓	-	0.815	0.152	**24.25**	0.063
3DGS # [2]	-	-	0.699	0.226	20.63	0.108
DNGaussian # [9]	-	-	0.755	0.198	22.18	0.088
Ours	-	Exp. 1	0.818	0.135	23.98	0.061
	-	Exp. 2	0.819	0.135	23.95	0.061
	-	Exp. 3	0.818	0.135	23.91	0.062
	-	Mean ±Std.	**0.818 ±0.001**	**0.135 ±0.001**	23.94 ±0.03	**0.061 ±0.001**

*: results reported in ReconFusion [15].
\#: results reported in DNGaussian [9].

and 9 input views on the LLFF dataset in Tab. 12 and Tab. 13 respectively. Notably, our method uses exactly the same parameters as the LLFF dataset with 3 views for training. For the **6 input views**, as shown in Tab. 12, we achieve an improvement of 11.18% on LPIPS compared to ReconFusion [15]. It is supposed to be noted that ReconFusion [15] requires additional computational resources for pre-training an encoder with external data as we demonstrated in the main manuscript. Excluding methods that require additional resources for pre-training, our method achieves improvements of 7.94%, 8.34%, 31.82%, 30.68% on PSNR, SSIM, LPIPS, and AVGE respectively, compared to DNGaussian [9], which is the state-of-the-art method based on the 3DGS [2]. For the **9 input views**, similar to the experimental results of 6 input views, our method still outperforms all state-of-the-art methods on SSIM, LPIPS, and AVGE scores and achieves comparable results on PSNR. As shown in Tab. 13, compared to 3DGS-based DNGaussian [9], we achieve improvements of 8.46%, 8.50%, 38.33%, 33.77% on PSNR, SSIM, LPIPS, and AVGE respectively.

Less input views. To evaluate extreme circumstances, *e.g.* opposite views and extrapolation scenarios, we construct corresponding data and conduct experiments with the state-of-the-art method DNGaussian [9]. For the **two opposite input views**, we select 2 opposite views of each scene on the MipNeRF-360 dataset, i.e. the IDs of training views of each scene: 2, 26 of bicycle; 22, 151 of bonsai; 57, 185 of counter; 1, 57 of garden; 14, 171 of kitchen; 2, 79 of room; 26, 34 of stump. The test views are selected every 8th image following Mip-NeRF. The quantitative comparisons with state-of-the-art method DNGaussian [9] are shown in Tab. 14. It can be seen that our method outperforms DNGaussian [9] and our model achieves improvements of 21.90%, 18.86% on average PSNR and AVGE scores respectively. For the **extrapolation scenarios**, We select 2 views on 0 and 90 degrees of each scene on the MipNeRF-360 dataset, i.e. IDs: 2, 14 of bicycle; 22, 248 of bonsai; 57, 145 of counter; 1, 15 of garden; 14, 37 of kitchen; 2, 291 of room; 26, 28 of stump. The test views are selected on the 180 degrees, i.e. IDs: 26 of bicycle; 151 of bonsai; 185 of counter; 57 of garden; 171 of kitchen; 79 of room; 34 of stump. The quantitative results similar to opposite views are shown in Tab. 14. It can be seen that our method outperforms the state-of-the-art method DNGaussian [9] and our model achieves improvements of 27.27%, 22.57% on average PSNR and AVGE scores respectively. We can notice that although our method is improved compared to DNGaussian, in fact, current sparse-view reconstruction methods (including our method) cannot successfully reconstruct extreme cases. We leave it as our future work.

Table 13: **Quantitative comparisons with 9 input views** on the LLFF dataset.

Method	Pretrain	Experiments	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
Zip-NeRF *	-	-	0.830	0.166	23.63	0.067
RegNeRF * [7]	-	-	0.820	0.196	24.84	0.065
DiffusioNeRF *	✓	-	0.807	0.216	24.62	0.069
FreeNeRF * [8]	-	-	0.820	0.193	25.12	0.063
SimpleNeRF * [45]	-	-	0.762	0.286	23.98	0.082
ReconFusion * [15]	✓	-	0.848	0.134	**25.21**	0.054
3DGS # [2]	-	-	0.697	0.230	20.44	0.108
DNGaussian # [9]	-	-	0.788	0.180	23.17	0.077
Ours	-	Exp. 1	0.854	0.113	25.02	0.051
	-	Exp. 2	0.856	0.111	25.20	0.050
	-	Exp. 3	0.856	0.110	25.19	0.050
	-	Mean ±Std.	**0.855 ±0.001**	**0.111 ±0.001**	25.13 ±0.08	**0.051 ±0.001**

*: results reported in ReconFusion [15].
\#: results reported in DNGaussian [9].

Table 14: **Quantitative comparisons** with 2 views on the MipNeRF-360 dataset.

Metric	Methods	Experiments	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
Opposite Views	DNGaussian	Exp. 1	0.142	0.705	10.53	0.387
		Exp. 2	0.141	0.704	10.49	0.388
		Exp. 3	0.142	0.705	10.49	0.388
		Mean ±Std.	0.142 ±0.001	0.705 ±0.001	10.50 ±0.02	0.387 ±0.001
	Ours	Exp. 1	0.243	0.677	12.85	0.313
		Exp. 2	0.245	0.675	12.78	0.314
		Exp. 3	0.242	0.678	12.77	0.315
		Mean ±Std.	**0.243 ±0.001**	**0.677 ±0.001**	**12.80 ±0.04**	**0.314 ±0.001**
Extrapolation Scenarios	DNGaussian	Exp. 1	0.075	0.734	9.89	0.417
		Exp. 2	0.063	0.739	9.67	0.426
		Exp. 3	0.081	0.736	9.81	0.419
		Mean ±Std.	0.073 ±0.007	0.736 ±0.002	9.79 ±0.09	0.421 ±0.004
	Ours	Exp. 1	0.267	0.707	12.61	0.322
		Exp. 2	0.266	0.711	12.44	0.326
		Exp. 3	0.258	0.712	12.33	0.330
		Mean ±Std.	**0.264 ±0.004**	**0.710 ±0.002**	**12.46 ±0.11**	**0.326 ±0.003**

B.6 Additional Evaluation and Discussion of View-conditioned Diffusion Priors

It is worth noting that the ***SDS mentioned before are all based on the 2D diffusion priors***. A natural idea is that we can use the 3D diffusion prior with the vanilla SDS to promote sparse-view 3D reconstruction without designing a complex method to extract 3D visual knowledge from the 2D diffusion prior. In this section, we discuss using view-conditioned 3D diffusion priors with SDS to improve the reconstruction quality under sparse views. We conduct experiments on the LLFF dataset with 3 views using view-conditioned 3D diffusion priors to evaluate their visual guidance of them. Specifically, we use the 3D prior, *i.e.* Zero-1-to-3 [50] and ZeroNVS [51], and their default CFG to optimize the 3D scene under sparse views through the vanilla SDS. We also use the same backbone and weights as IPSM. Besides, we explore the effect of warmup operation for the SDS regularization of 3D priors.

Table 15: **Quantitative experimental results** using view-conditioned diffusion priors on the LLFF dataset with 3-views setting.

Setting	Experiments	SSIM↑	LPIPS↓	PSNR↑	AVGE↓
Base	Mean ±Std.	0.625 ±0.008	0.254 ±0.007	19.00 ±0.12	0.125 ±0.003
SD, CFG=7.5	Mean ±Std.	0.647 ±0.009	0.267 ±0.012	18.80 ±0.38	0.128 ±0.006
SD, CFG=100	Mean ±Std.	0.576 ±0.003	0.367 ±0.006	17.53 ±0.24	0.162 ±0.004
ISD($i.e.\mathcal{L}_{\mathrm{IPSM}}^{\mathcal{G}_1}$), CFG=7.5	Mean ±Std.	0.636 ±0.004	0.245 ±0.003	19.22 ±0.02	0.121 ±0.001
Zero-1-to-3, CFG=3.0	Exp. 1	0.566	0.361	17.65	0.160
	Exp. 2	0.576	0.354	17.70	0.158
	Exp. 3	0.577	0.351	18.00	0.153
	Mean ±Std.	0.573 ±0.005	0.355 ±0.004	17.78 ±0.15	0.157 ±0.003
Zero-1-to-3, CFG=3.0 w/ WarmUp	Exp. 1	0.584	0.344	17.81	0.154
	Exp. 2	0.576	0.349	17.87	0.155
	Exp. 3	0.575	0.361	17.79	0.158
	Mean ±Std.	0.578 ±0.004	0.352 ±0.007	17.82 ±0.03	0.156 ±0.001
ZeroNVS, CFG=7.5	Exp. 1	0.639	0.289	19.12	0.129
	Exp. 2	0.633	0.292	19.15	0.129
	Exp. 3	0.641	0.286	19.40	0.125
	Mean ±Std.	0.638 ±0.003	0.289 ±0.003	19.22 ±0.12	0.128 ±0.002
ZeroNVS, CFG=7.5 w/ WarmUp	Exp. 1	0.647	0.281	19.22	0.126
	Exp. 2	0.643	0.283	19.29	0.126
	Exp. 3	0.644	0.282	19.30	0.126
	Mean ±Std.	0.645 ±0.001	0.282 ±0.001	19.27 ±0.04	0.126 ±0.001
IPSM(Ours), CFG=7.5	Mean ±Std.	**0.670 ±0.001**	**0.229 ±0.002**	**19.60 ±0.11**	**0.113 ±0.001**

As shown in Tab. 15, the first three rows and the last row are the experimental results mentioned before. The fourth line shows the result of using the Inpainting Stable Diffusion model (ISD) with inline priors to assist SDS, which is actually the ablation result of $\mathcal{L}_{\mathrm{IPSM}}^{\mathcal{G}_1}$ in the ablation experiment shown in Tab. 2. We can notice that both Zero-1-to-3 and ZeroNVS can only provide limited visual guidance and may even hinder reconstruction compared to the Baseline. Besides, using ZeroNVS [51] is superior compared to using Zero-1-to-3 [50] since the former utilizes 3D annotated scene data for fine-tuning while Zero-1-to-3 only uses 3D objects dataset for fine-tuning. However, although ZeroNVS [51] as 3D prior can achieve stunning results in single-view reconstruction for inferring 3D structure from an unlabeled 2D image [51], it still cannot boost the sparse-view reconstruction quality as IPSM since the ZeroNVS guidance does not exploit inline priors for sparse views which is different from the single-view setting.

Currently, 3D diffusion priors already have a certain ability to represent the 3D world. However, as reported experimental results in Tab. 15, 3D diffusion priors still cannot provide a significant boost on different 3D scene datasets, since the scarcity of 3D annotation data used to fine-tune 3D diffusion priors exists. Specifically, ZeroNVS [51] fine-tuned on a mixture million-level dataset consisting of CO3D [65], ACID [66, 67], and RealEstate10K [68]. But, Stable Diffusion [63] and its inpainting version are trained on billion-level LAION-5B [69]. With the additional conducted experiments, we notice that there is still an objective fact that 3D training data for 3D diffusion models is scarce. How to efficiently construct high-fidelity 3D data, or how to use 2D data knowledge to complement the training of 3D diffusion prior remains a core challenge in this field.

B.7 Intuitive Explanation of Inline Priors

To visually demonstrate the effect of inline priors for rectification on the diffusion prior more intuitively, we show the inline priors along with their associated visual content in Fig. 8 as a intuitive supplement to our motivation. Note that we choose a relatively tight depth error threshold to better illustrate the potential of the rectified distribution. The first column shows the input sparse seen-view
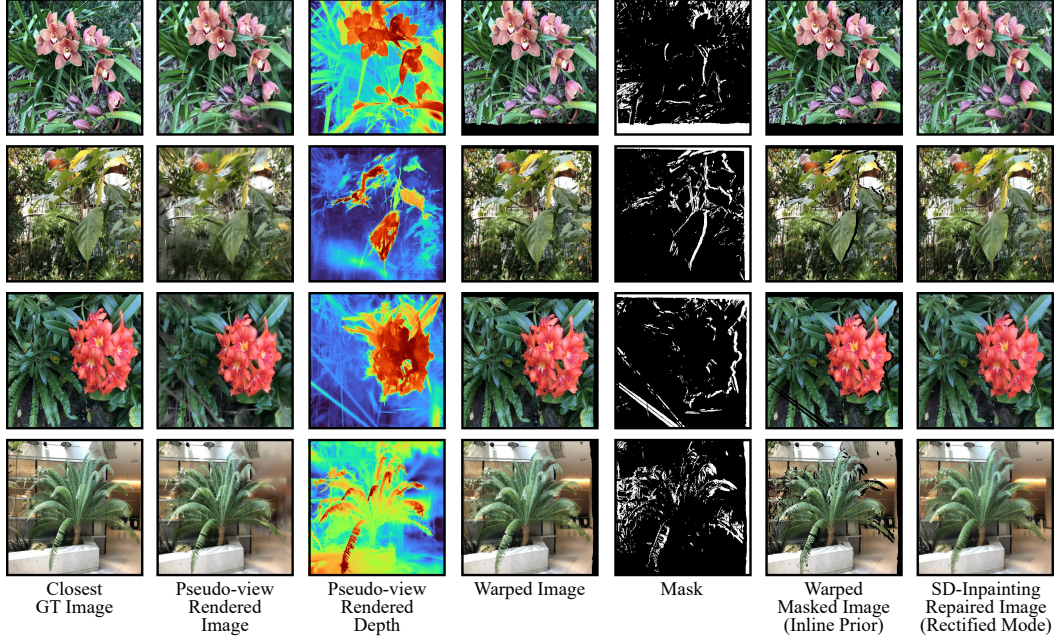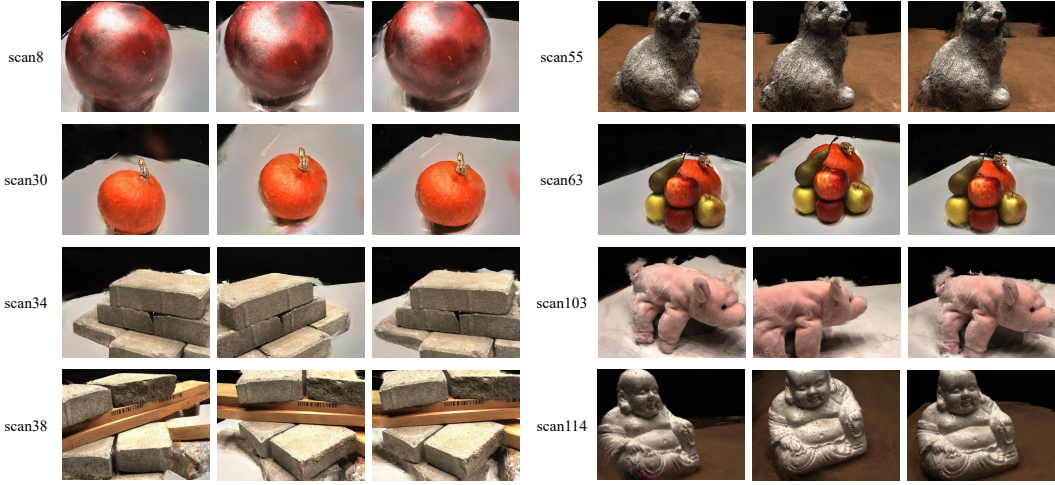
Figure 8: **Intuitive explanation of the inline priors**.

Figure 9: **Examples of novel view synthesis from our method with 3 input views on the DTU dataset**.

image; the second column includes the rendering images of the pseudo unseen view which is sampled around the seen view (Eq. 3); the third column presents the rendering depths corresponding to the pseudo view (Eq. 7); the fourth column consists of the warped images obtained based on the pose transformation relationships with the seen-view image, pseudo-view rendering depth (Eq. 9); the fifth column depicts masks derived from the depth differences (Eq. 10); the sixth column displays the masked warped images, known as inline priors, which integrate visual inline information from the seen view to the pseudo unseen views, thereby laying the foundation for subsequent rectification of the diffusion prior; and the seventh column intuitively exhibits images obtained through 25-step sampling using noise-added rendering images served as latents and inline priors served as conditions with Stable Diffusion Inpainting v1-5 [63], representing the rectified mode of the corresponding scenes in the rectified distribution.

Figure 10: **Examples of novel view synthesis from our method with 3 input views on the LLFF dataset**.

B.8   More Qualitative Results

We present additional examples of rendered images in the test set shown in Fig. 9 and Fig. 10. The examples of rendering results are obtained from the DTU dataset [54] and the LLFF dataset [22] with 3 training views. More qualitative results can be found in our **supplementary video**.