SPLATREFINER: FAITHFUL AND EFFICIENT POST-HOC DIFFUSION REFINEMENT FOR SPARSE-VIEW NOVEL VIEW SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse-view novel view synthesis requires inferring content that a few input images leave unresolved. Many diffusion-assisted pipelines supply this content through reconstruction–generation cycles along predefined camera trajectories, making novel views costly to obtain. Refining renderings of a fixed Gaussian reconstruction directly into final outputs avoids these cycles. However, these renderings mix useful scene content with artifacts, and without 3D re-optimization, the refiner alone must remain faithful to the observed scene. To address this, we exploit information from both the reference views and the target view, during training and at inference. During training, VGGT features of the reference views condition denoising, and those of the ground-truth target view supervise intermediate representations. At inference, dual-condition guidance strengthens the contributions of the reference views and of the rendering of the target view. Motivated by these, we present SplatRefiner, a post-hoc diffusion refiner that restores renderings at arbitrary requested poses without predefined trajectories. On identical renderings, SplatRefiner outperforms strong baselines in most experiments while in 3 input view settings it runs 6.9× faster end-to-end with comparable PSNR using 2DGS, and 3.7× faster with higher PSNR using FSGS. Trained only on 2DGS renderings from DL3DV-10K, it improves PSNR, LPIPS, and FID over the renderings of 2DGS, 3DGS, and FSGS across three datasets, two unseen during training, with 3, 6, and 9 input views.

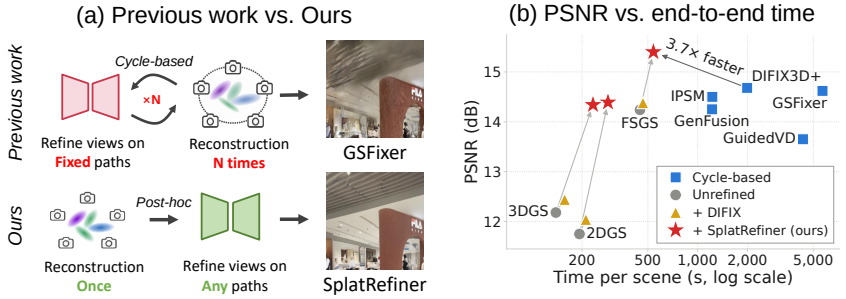

Figure 1: **(a)** Cycle-based methods repeatedly refine views on a fixed path and re-optimize the 3D scene, whose final renderings bypass the refiner and retain artifacts; SplatRefiner refines the requested views once, directly into final outputs. **(b)** With FSGS, SplatRefiner exceeds DIFIX3D+ in PSNR while running 3.7× faster (DL3DV-Evaluation, 3 views).

1 INTRODUCTION

Sparse-view novel view synthesis (NVS) requires both reconstructing what a few input images constrain and inferring what they leave unresolved, a capability needed whenever dense capture is impractical. Neural radiance fields and Gaussian splatting render densely captured scenes with high fidelity (Mildenhall et al., 2020; Kerbl et al., 2023), but sparse inputs leave geometry, visibility, and appearance ambiguous at unseen viewpoints. Regularization and learned priors narrow this ambigu-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ity without resolving it (Niemeyer et al., 2022; Zhu et al., 2024). Diffusion models offer what sparse observations lack: a learned prior over plausible image content (Ho et al., 2020; Rombach et al., 2022). Diffusion-assisted reconstruction exploits this prior by using generated views to supervise a 3D representation (Gao et al., 2024), often through reconstruction–generation cycles in which diffusion outputs at novel viewpoints are fed back into optimization (Wu et al., 2024; 2025b). Fitting one 3D representation to both captured and generated images encourages consistency between them, but does not ensure that a plausible generated view faithfully depicts the observed scene. Much recent work therefore controls where auxiliary views are generated, along predefined camera trajectories (Wu et al., 2025a; Yin et al., 2026) or at viewpoints selected by visibility (Zhong et al., 2025; Ni et al., 2026), and how their information enters optimization (Wang et al., 2024a; Liu et al., 2024).

Despite these advances, the cycles themselves remain expensive: they repeatedly generate auxiliary views and re-optimize the scene, typically spending tens of minutes to over an hour on a single scene (Figure 1(b)), yet their final renderings may still retain artifacts (Figure 1(a)). Directly refining renderings of a fixed reconstruction into final outputs avoids these cycles, and recent post-rendering refiners demonstrate the value of operating on rendered images (Wu et al., 2025a; Zhu et al., 2026; Dong et al., 2026). We pursue this approach as trajectory-free post-hoc refinement: a Gaussian reconstruction is fitted once to the input images—which we call reference views—and kept fixed, and the refiner restores its renderings at any requested target view. This setting poses two challenges. First, renderings of a sparse-view reconstruction mix useful scene content with artifacts, so the refiner must correct distortions and complete missing regions. Second, because its outputs are final and no re-optimization follows, the refiner alone must remain faithful to the observed scene, whereas diffusion priors favor plausibility and can introduce content that does not match it. If the renderings cannot be trusted on their own, where should a refiner draw the information it needs to stay faithful?

To address this challenge, we introduce SplatRefiner: a trajectory-free post-hoc diffusion refiner for sparse-view NVS (Figure 2). To remain faithful, a refiner can draw on two sources of information. The reference views directly observe the scene, and the target view is given by its rendering and, during training, also by its ground truth. SplatRefiner exploits both sources during training and at inference. During training, it conditions denoising on the reference views and supervises its intermediate representations with the ground-truth target view, using features of a frozen VGGT (Wang et al., 2025b). At inference, dual-condition guidance strengthens the contributions of the reference views and of the rendering.

To refine Gaussian splatting renderings, we train a multi-view diffusion model on pairs of renderings and their ground-truth images. Because its outputs are final, the model takes all reference views as input so that every observation of the scene is available during restoration. To serve as a refiner at any requested pose, it is trained on independently sampled target views rather than views along a fixed camera trajectory. To improve scene fidelity beyond what the rendering provides, the refiner additionally uses VGGT features of both sources. Reference-view features are injected into the encoder of the diffusion UNet through cross-attention and retained at inference, whereas ground-truth target features supervise decoder representations through a projector that is discarded after training; we choose these placements with a controlled analysis (Section 3). At inference, dual-condition guidance extrapolates each denoising prediction away from one computed without the reference views and then away from one computed without the rendering, so that denoising stays tied both to the observed scene and to the requested view.

Our experiments show that SplatRefiner restores renderings faithfully while avoiding the cost of reconstruction–generation cycles. On identical renderings, it outperforms DIFIX, the refiner of DIFIX3D+, on all four metrics across three datasets, three Gaussian reconstructions, and 3, 6, and 9 input views, with a single exception (SSIM with FSGS and 9 input views on Mip-NeRF 360). Against DIFIX3D+ itself, on DL3DV-Evaluation with 3 input views, SplatRefiner runs 6.9× faster end-to-end with comparable PSNR using 2DGS and 3.7× faster with higher PSNR using FSGS; it exceeds DIFIX3D+ in PSNR in every setting using FSGS on DL3DV-Evaluation and Mip-NeRF 360, and outperforms it on all four metrics using 2DGS on ScanNet++. Trained only on 2DGS renderings from DL3DV-10K, it also improves PSNR, LPIPS, and FID over 2DGS, 3DGS, and FSGS renderings across three datasets and 3, 6, and 9 input views.

Our contributions are as follows:

- **Trajectory-free post-hoc refinement.** We characterize post-hoc refinement of a fixed Gaussian reconstruction as restoring renderings at any requested pose, conditioned on all reference views, without reconstruction–generation cycles or 3D re-optimization.
- **Reference and target views at both stages.** SplatRefiner exploits the reference views and the target view during training, through VGGT-feature conditioning and supervision placed by a controlled analysis, and at inference, through dual-condition guidance.
- **Faithful and efficient.** SplatRefiner outperforms DIFIX on identical renderings in all but one evaluated setting, and matches or exceeds DIFIX3D+ in PSNR at 3.7–6.9× lower end-to-end cost, while being trained only on 2DGS renderings.

2 Related Work

2.1 Diffusion-Assisted Sparse-View Reconstruction

Diffusion priors for sparse-view reconstruction have progressed from image-based refinement (Wu et al., 2024; 2025a; Wang et al., 2024a) to video diffusion with improved multi-view consistency (Wu et al., 2025b; Liu et al., 2024; Wang et al., 2025a). Because generated views may still hallucinate content or drift in appearance and geometry, these methods restrict generation to selected viewpoints. Some methods sample views on predefined camera paths, such as interpolated, circular, or spiral trajectories around the reference views (Wu et al., 2024; Liu et al., 2024; Wu et al., 2025b; Yin et al., 2026). Others select viewpoints by hole- or visibility-aware criteria, exposing unreliable regions while avoiding severely under-constrained views (Zhong et al., 2025; Ni et al., 2026). Post-hoc refiners instead restore renderings of a fixed reconstruction into final outputs. GaussFusion (Zhu et al., 2026) refines renderings along smooth camera trajectories with a video generator conditioned on geometry buffers rendered from the reconstruction itself. One-Shot Refiner (Dong et al., 2026) likewise operates on rendered images but is trained jointly with the feed-forward reconstruction model it refines. DIFIX3D+ (Wu et al., 2025a) also refines the final views, but only after reconstruction–generation cycles that condition on a single nearby reference image and approach the target through interpolated poses. SplatRefiner belongs to the post-hoc family and keeps its outputs faithful by exploiting both the reference views and the target view, during training and at inference. Unlike trajectory-based refiners, it refines any requested set of views, which need not form a continuous camera path. Because it is trained separately from the reconstruction rather than jointly with it, the same refiner applies to renderings from different Gaussian reconstruction methods.

2.2 Conditioning, Supervision, and Guidance in Diffusion Models

External information can enter a diffusion model as a denoising condition, as supervision of its intermediate representations, or through guidance at sampling time. Conditioning-based approaches inject pretrained features through embedding-level conditioning (Wei et al., 2023; Li et al., 2023) or adapter modules (Ye et al., 2023; Mou et al., 2024; Zhang et al., 2023). In sparse-view reconstruction, GSFixer (Yin et al., 2026) conditions on VGGT features through cross-attention, but only from two reference views at a time and along a predefined path inside reconstruction–generation cycles. Alignment-based approaches instead supervise intermediate diffusion features with foundation-model features (Yu et al., 2025), distilling a pretrained representation into the diffusion model through a simple auxiliary loss and substantially accelerating its training. Follow-up works adapt this alignment to UNet backbones (Tian et al., 2025), emphasize the spatial structure of the target features (Singh et al., 2026), and extend it to physics- and geometry-aware video generation (Zhang et al., 2025; Wu et al., 2026). These works use either conditioning or alignment, whereas we use both for the two sources available in post-hoc refinement: reference-view features condition denoising, and ground-truth target features supervise intermediate representations. At inference, classifier-free guidance (Ho & Salimans, 2022) extrapolates a conditional prediction away from an unconditional one, and InstructPix2Pix (Brooks et al., 2023) applies separate guidance scales to its image and text conditions. Our dual-condition guidance applies separate scales to the reference views and the rendering, the two inputs that carry scene information in post-hoc refinement.

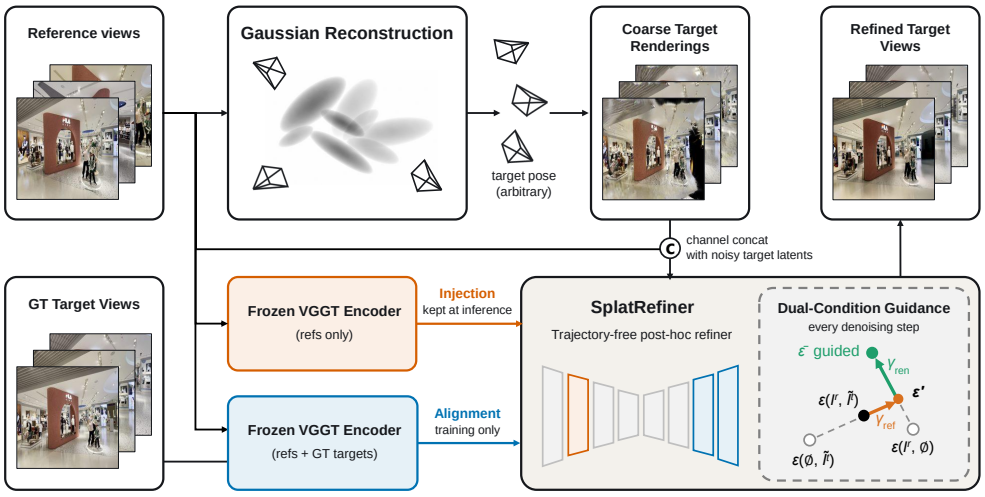

Figure 2: Overview of SplatRefiner. A Gaussian reconstruction is fitted once to the reference views and kept fixed, and its renderings at arbitrary requested poses are refined into final outputs. Frozen VGGT features of the reference views are injected into the UNet encoder and kept at inference (orange), whereas those of the ground-truth target views supervise the decoder during training only (blue). At inference, dual-condition guidance further strengthens the contributions of the reference views and of the rendering.

3 Analysis

3.1 Target-Feature Alignment for a Post-Hoc Refiner

Alignment supervises intermediate UNet features with VGGT features of the target images, whose ground truth is available only during training. Let I^t denote the target images and let a_j be the VGGT feature of target view j; these features serve only as alignment targets and never as denoising conditions. At a UNet site k, an MLP P_{ψ_k} projects the feature h_j^k of target view j into the VGGT feature space, and the site is aligned by

$$\mathcal{L}_{\text{align}}^{(k)} = \frac{1}{N_t} \sum_{j=1}^{N_t} d\big(P_{\psi_k}(h_j^k),\ a_j\big), \tag{1}$$

where d is the mean token-wise cosine distance over a common spatial grid. The loss is added to the denoising objective with weight λ_{align}, and the projector is discarded at inference. Most existing recipes instantiate Eq. (1) once, at a single site aligned to the final VGGT layer. We instead follow VGGT's geometry head, which reads layers 4, 11, 17, and 23 and fuses them coarse-to-fine: we refer to these as VGGT L1–L4 from shallow to deep, denote the four resolution levels of the UNet encoder (Enc) and decoder (Dec) as L1–L4 from fine to coarse, pair VGGT Lk with UNet Lk (Figure 6), and average Eq. (1) over the chosen sites.

The remainder of this section follows one protocol. All configurations share the training setup of Section 5.1 and are evaluated on the 6-view validation split (20 scenes, one seed) at the 50k-iteration checkpoint. We report differences from the refiner without VGGT (Figure 3).

Target supervision favors the decoder. Figure 3(a) reports the gains of alignment alone over the refiner without VGGT. Existing recipes align a single encoder site to the final VGGT layer (L1†); in our refiner this placement gains only $+0.04$ dB and degrades LPIPS ($+1.2 \times 10^{-3}$). Adding encoder sites does not help: level-matched alignment at all four encoder sites gains $+0.03$ dB and still does not improve LPIPS, so encoder-side alignment is ineffective regardless of the number of sites. Moving the targets to the decoder is what makes alignment work: the two sites at Dec L1–L2 gain $+0.06$ dB, and extending to all four decoder sites (Dec L1–L4) doubles this to $+0.12$ dB, with LPIPS improved rather than degraded (-1.4×10^{-3}). Level-matched decoder alignment therefore beats the last-layer encoder recipe on both metrics ($+0.08$ dB and -2.6×10^{-3} LPIPS over L1†).

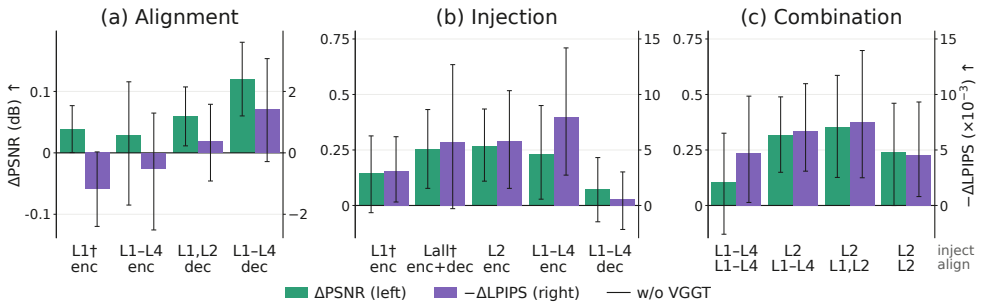

Figure 3: Where and how many sites to attach the VGGT features: alignment alone (a), injection alone (b), or both (c). DL3DV-Validation, 20 scenes, 6 input views, 50k steps. Bars are the change from the w/o-VGGT baseline (black line): ΔPSNR (left axis, dB) and $-\Delta$LPIPS (right axis, $\times 10^{-3}$), both better upwards; error bars are 95% paired bootstrap intervals across scenes. Vertical scales differ across panels. L1–L4 index UNet levels from finest to coarsest, as in Figure 6, each paired with the VGGT layer of the same index; † marks sites fed with the final VGGT layer instead of the level-matched one: L1† is the single encoder site of prior alignment recipes, and Lall† follows prior cross-attention recipes, which condition every transformer block (16 in our UNet).

Target supervision thus favors the decoder, although the gains from alignment alone remain modest (at most $+0.12$ dB). We attribute this to what the two sides represent: encoder features are still close to the noisy input, whereas decoder features represent the denoised prediction and thus correspond more naturally to the clean-target features that alignment supervises. The same conclusion holds on the held-out 53-scene evaluation split (Appendix A.3).

3.2 Reference-Feature Injection for a Post-Hoc Refiner

A post-hoc refiner relies on the reference views to separate reliable content from reconstruction errors and to fill missing regions. We therefore use their VGGT features in a second role, as denoising conditions, and search their placement as we did for alignment. Let I^r denote the reference views and b their VGGT features at one layer, concatenated over references along the token dimension. At a UNet site k, the feature h_j^k of target view j queries the reference tokens through cross-attention, added as a residual:

$$\tilde{h}_j^k = h_j^k + \mathrm{MHA}_{\phi_k}\big(\mathrm{LN}(h_j^k)W_Q,\ \mathrm{LN}(b)W_K,\ \mathrm{LN}(b)W_V\big), \tag{2}$$

where the output projection of MHA_{ϕ_k} is zero-initialized following ControlNet (Zhang et al., 2023), so that the residual starts at zero. The update is applied to target views only, and the module is retained at inference. We use the same VGGT-to-UNet pairing as for alignment, placing decoder sites at the first block of each level so that Enc and Dec sites match in resolution and channel width (Table 4), and compare the same placements as for alignment: the single site L1†, injection at all four Enc or all four Dec sites, and additionally the paired single site at Enc L2.

Reference conditioning favors the encoder. Figure 3(b) reports the gains of injection alone over the refiner without VGGT. Prior cross-attention recipes inject the final VGGT layer into every transformer block (Lall†, $+0.25$ dB in our refiner). Following the level-matched pairing of the alignment study, we instead inject at single sites; the level-matched site alone (L2, $+0.27$ dB) matches the all-block recipe with one sixteenth of the injection parameters. Among single encoder sites, the level-matched pair (L2) roughly doubles the gain of the last-layer single site (L1†) on both metrics. Injecting at all four encoder sites gives no further PSNR gain ($+0.23$ vs. $+0.27$ dB) and only a small additional LPIPS gain (-7.9 vs. -5.8×10^{-3}). Decoder-side injection is the weakest placement ($+0.07$ dB, LPIPS essentially unchanged). Reference conditioning therefore favors the encoder. We attribute this to the capacity left for integration: reference evidence injected early can be integrated by all subsequent layers, whereas evidence injected in the decoder has fewer layers left to act on. The same conclusion holds on the held-out 53-scene evaluation split (Appendix A.3).

3.3 Standalone Properties Do Not Carry Over to the Combination

Since target supervision favors the decoder and reference conditioning the encoder, the two can be combined at their preferred positions. Figure 3(c) reports combinations that inject at Enc sites and align at Dec sites, varying the number of sites on each side. Combining everything is the worst option: injection at all four Enc sites with alignment at all four Dec sites gains only $+0.11$ dB, no more than alignment alone ($+0.12$ dB) and 0.16 dB below injection alone at Enc L2; the injection gain is forfeited. Reducing injection to the single level-matched site restores it: with alignment at all four Dec sites the combination gains $+0.32$ dB, and with alignment at Dec L1–L2 it gains $+0.35$ dB and -7.6×10^{-3} LPIPS. On the alignment side, dropping to Dec L2 alone loses 0.11 dB, while four sites are no better than two: the advantage of four sites observed for alignment alone does not reappear once injection is present. In both respects, what each mechanism does alone is not preserved when the two are combined, so the number of sites has to be chosen jointly rather than inherited from the standalone optima. The same conclusion holds on the held-out 53-scene evaluation split (Appendix A.3). We therefore select the final configuration from the validation curves in Figure 8. Among the candidates, injection at Enc L2 with alignment at Dec L1–L2 is the highest in PSNR and SSIM at every checkpoint from 10k to 50k, and we take its 40k checkpoint, the peak of that trajectory.

4 Method

4.1 Refinement from a Fixed Reconstruction

A post-hoc refiner must meet two conditions: its outputs are final, and it must accept any requested poses rather than a predefined trajectory. To meet them, we train a multi-view diffusion model to restore renderings into final outputs, using pairs of renderings and their ground-truth images. Let $I^r = \{I^r_i\}_{i=1}^{N_r}$ be the reference views and $\mathcal{G}$ a Gaussian reconstruction fitted once to them and kept fixed. For requested target poses $\{\pi^t_j\}_{j=1}^{N_t}$, we render the target views and refine them as

$$\widetilde{I}^t = \{\mathrm{Render}(\mathcal{G}, \pi^t_j)\}_{j=1}^{N_t}, \qquad \widehat{I}^t = F_{\theta,\phi}(I^r, \widetilde{I}^t), \tag{3}$$

where $F_{\theta,\phi}$ is the refiner, and θ and ϕ denote the parameters of the diffusion UNet and of reference-feature injection. For each target view, the latent of its rendering is concatenated with the noisy target latent along the channel dimension.

Because its outputs are final, the refiner should restore each target view from every available observation of the scene. We therefore give it all reference views as input, with their latents concatenated with the target latents along the frame dimension. To accept any requested poses, its target views are sampled at random during training rather than along a predefined trajectory. We train with $N_r \in \{3, 6, 9\}$ reference views per scene and $N_t = 10$ target views sampled from the remaining views of the scene, which need not form a continuous camera sequence.

4.2 Learning and Final Image Formation

During training, the refiner uses VGGT features of both the reference views and the target view, at the sites selected in Section 3.3: reference-feature injection at Enc L2 and target-feature alignment at Dec L1–L2. VGGT is run twice. It processes the reference views alone to provide b in Eq. (2), and the reference and ground-truth target views jointly to provide a_j in Eq. (1). Processing the target views together with the reference views gives the target features multi-view context, and these features serve only as alignment targets.

We optimize the UNet parameters θ, the injection module ϕ, and the alignment projectors $\{\psi_k\}_{k=1}^{2}$, keeping VGGT and the reconstructions fixed, with

$$\mathcal{L} = \mathcal{L}_{\mathrm{denoise}} + \lambda_{\mathrm{align}}\, \mathcal{L}_{\mathrm{align}}, \tag{4}$$

where $\mathcal{L}_{\mathrm{denoise}}$ is the diffusion objective on the target latents and $\mathcal{L}_{\mathrm{align}}$ averages Eq. (1) over the two decoder sites. Cubic timestep sampling emphasizes high-noise timesteps (Appendix A.4).

At inference, the alignment projectors are discarded and injection is retained. The refiner is conditioned on two inputs with different roles: the renderings $\widetilde{I}^t$ provide the target-view structure but

carry reconstruction errors, whereas the reference views I^r are the only direct observations of the scene. We therefore guide the denoising with respect to each input with *dual-condition guidance*, an extrapolation in the spirit of classifier-free guidance (Ho & Salimans, 2022). At each denoising step,

$$\varepsilon' = \varepsilon_{\theta,\phi}(I^r, \widetilde{I}^t) + \gamma_{\text{ref}}\big(\varepsilon_{\theta,\phi}(I^r, \widetilde{I}^t) - \varepsilon_{\theta,\phi}(\emptyset, \widetilde{I}^t)\big), \qquad \bar{\varepsilon} = \varepsilon' + \gamma_{\text{ren}}\big(\varepsilon' - \varepsilon_{\theta,\phi}(I^r, \emptyset)\big), \quad (5)$$

where the noisy latents and timestep are omitted. The two extrapolations are applied in sequence: the first (γ_{ref}) moves the prediction away from one that omits the reference views and their injected features, and the second (γ_{ren}) moves the result away from one that omits the renderings. The second is the classifier-free guidance of the base model, whose rendering-free prediction is learned in See3D pretraining by dropping the visual condition with probability 0.1 (Ma et al., 2025); neither omitted-input prediction is trained during our training, and no training stage omits the references. The refined views $\widehat{I}^t$ are decoded from the final denoised latents.

5 Experiments

5.1 Experimental Setup

Datasets. For training, we use 300 scenes from DL3DV-10K, which covers diverse indoor and outdoor environments, and 20 scenes from DL3DV-Benchmark for validation, which we refer to as DL3DV-Validation (Ling et al., 2024). For each scene, we uniformly sample a 9-view reference set and derive the 3- and 6-view settings by uniform subsampling. We optimize a 2DGS model for 7k iterations using each reference set and render independently sampled target views from the resulting Gaussian representation, which serve as degraded inputs to SplatRefiner.

For evaluation, we use three real-world datasets with diverse scene characteristics: 53 scenes from DL3DV-Evaluation (Ling et al., 2024), six indoor scenes from ScanNet++ (Yeshwanth et al., 2023), and seven indoor and outdoor scenes from Mip-NeRF 360 (Barron et al., 2022). For DL3DV-Evaluation and Mip-NeRF 360, we follow the LLFF evaluation protocol (Mildenhall et al., 2019) by sorting frames by image index and reserving every 8th frame as an evaluation view. The remaining frames are used to construct sparse-view reference sets using the same reference-view sampling strategy as in training. For ScanNet++, following G4Splat (Ni et al., 2026), we use all views except the selected reference views as evaluation views.

Baselines. We compare SplatRefiner with diffusion-assisted sparse-view reconstruction methods and with direct post-hoc refinement of Gaussian reconstructions. Diffusion-assisted baselines include IPSM (Wang et al., 2024a), GenFusion (Wu et al., 2025b), G4Splat (Ni et al., 2026), GuidedVD (Zhong et al., 2025), GSFixer (Yin et al., 2026), and DIFIX3D+ (Wu et al., 2025a), all of which rely on iterative reconstruction–generation cycles. Notably, GSFixer also leverages VGGT features, while DIFIX3D+ further refines the final rendered views using diffusion, similar to SplatRefiner. For direct post-hoc refinement, we use 2DGS (Huang et al., 2024), 3DGS (Kerbl et al., 2023), and FSGS (Zhu et al., 2024) as reconstruction methods and compare their renderings before and after refinement. As a post-hoc refiner baseline, we apply DIFIX, the refiner of DIFIX3D+, to the same renderings as a standalone module.

Implementation details. We build SplatRefiner on See3D (Ma et al., 2025) as the base diffusion model. We optimize the model using AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a weight decay of 10^{-2}, and a constant learning rate of 1×10^{-5}. We set the alignment loss weight to $\lambda_{\text{align}} = 1$, and the guidance strengths to $\gamma_{\text{ref}} = 0.5$ and $\gamma_{\text{ren}} = 2$, the latter being the default guidance scale of the base model. For the analysis, all configurations are evaluated on the 6-view validation setting and trained for 50k iterations. For all other diffusion-specific settings, we follow the original See3D configuration. All experiments are conducted on a single NVIDIA H200 GPU with 143 GB memory.

Metrics. We evaluate rendering quality using PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and FID (Heusel et al., 2017). These standard NVS metrics directly assess whether diffusion outputs can serve as final target-view renderings without subsequent 3D optimization.

5.2 Main Results

Direct refinement enables high-quality NVS at a fraction of the cost. Table 1 compares rendering quality and computational efficiency on DL3DV-Evaluation and ScanNet++ with 3 input views. Methods that use diffusion only within reconstruction–generation cycles reach competitive PSNR and SSIM but retain noise and artifacts in their final renderings (Figure 4), and their cycles take 1222–5680s per scene. SplatRefiner instead refines the renderings of a fixed reconstruction in a single refinement stage. With FSGS, it achieves the highest PSNR and SSIM on DL3DV-Evaluation and the second-best LPIPS and FID, behind only DIFIX3D+, in 540s ($3.7\times$ faster than DIFIX3D+'s 1991s); with 2DGS, it reaches comparable PSNR (14.39 vs. 14.68) in 287s ($6.9\times$ faster). Because refinement adds about 90s regardless of the reconstruction, the end-to-end cost is set by the reconstruction, which can be chosen to trade speed for quality without retraining the refiner. On the unseen ScanNet++, SplatRefiner with 2DGS inputs achieves the best scores on all four metrics among all compared methods, including GSFixer and DIFIX3D+, and it remains competitive with DIFIX3D+ with 6 and 9 input views and on Mip-NeRF 360 (Appendix A.6). Where DIFIX3D+ scores comparably, its outputs can still alter fine structures and details of the observed scene, which SplatRefiner preserves (Figure 4; more examples in Appendix A.8).

Table 1: Quantitative comparison on DL3DV-Evaluation and ScanNet++ with 3 input views. Bold and underlined values indicate the best and second-best results, ranked separately for methods with and without reconstruction–generation cycles.

Method	DL3DV-Evaluation							ScanNet++			
	PSNR↑	SSIM↑	LPIPS↓	FID↓	3D Opt. Iterations↓	Refinement Passes↓	Time↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
Diffusion-assisted reconstruction											
IPSM (Wang et al., 2024a)	14.50	**0.480**	0.489	116.54	10k	7499	1226s	<u>13.11</u>	<u>0.630</u>	<u>0.499</u>	259.15
GenFusion (Wu et al., 2025b)	14.25	<u>0.475</u>	0.493	119.85	7k	6	1222s	10.90	0.554	0.528	262.60
G4Splat (Ni et al., 2026)	9.09	0.201	0.644	154.70	28k	3	2402s	9.65	0.404	0.588	249.81
GuidedVD (Zhong et al., 2025)	13.65	0.420	0.519	124.60	20k	37	4329s	11.24	0.525	0.523	<u>231.93</u>
GSFixer (Yin et al., 2026)	<u>14.62</u>	0.474	<u>0.481</u>	<u>116.52</u>	30k	3	5680s	11.77	0.577	0.526	283.08
Diffusion-assisted reconstruction + refinement											
DIFIX3D+ (Wu et al., 2025a)	**14.68**	0.461	**0.390**	**27.24**	60k	30	1991s	**14.40**	**0.663**	**0.402**	**68.26**
Direct post-hoc refinement											
2DGS (Huang et al., 2024)	11.75	0.332	0.539	129.33	7k	0	193s	9.23	0.402	0.558	224.97
+ DIFIX	12.04	0.331	0.493	46.63	7k	1	211s	9.52	0.431	0.514	100.02
+ SplatRefiner (Ours)	<u>14.39</u>	0.439	<u>0.428</u>	<u>31.14</u>	7k	1	287s	**14.43**	**0.665**	**0.385**	**57.91**
3DGS (Kerbl et al., 2023)	12.18	0.357	0.534	131.29	10k	0	139s	8.67	0.379	0.561	226.15
+ DIFIX	12.44	0.352	0.486	43.52	10k	1	157s	8.87	0.399	0.527	99.97
+ SplatRefiner (Ours)	14.34	0.435	0.429	31.22	10k	1	233s	13.84	0.650	0.410	<u>58.96</u>
FSGS (Zhu et al., 2024)	14.24	<u>0.468</u>	0.507	129.99	10k	0	448s	12.91	0.619	0.503	247.05
+ DIFIX	14.38	0.441	0.440	41.43	10k	1	467s	13.20	0.609	0.460	112.03
+ SplatRefiner (Ours)	**15.40**	**0.481**	**0.393**	**30.93**	10k	1	540s	<u>14.04</u>	<u>0.659</u>	<u>0.404</u>	59.35

Post-hoc refinement can be faithful across reconstructions and datasets. To isolate what the refiner itself contributes, we compare SplatRefiner with another post-hoc refiner on identical inputs. DIFIX, the refiner of DIFIX3D+, can likewise be applied directly to renderings, so we evaluate it as a standalone module on the same Gaussian renderings. Without the preceding cycles, DIFIX falls well behind the full DIFIX3D+ pipeline (e.g., 12.04 vs. 14.68 PSNR with 2DGS on DL3DV-Evaluation) and leaves structural distortions and unfilled regions (Figure 5). SplatRefiner outperforms DIFIX on all four metrics across both datasets and all three reconstruction methods, correcting these distortions and completing the missing content. The advantage extends to 6 and 9 input views and to Mip-NeRF 360, with a single exception (SSIM with FSGS and 9 input views on Mip-NeRF 360; Appendix A.6). Although trained only on 2DGS renderings from DL3DV-10K, SplatRefiner also improves the unrefined 2DGS, 3DGS, and FSGS renderings on all four metrics on both evaluation datasets with 3 input views, suggesting that it learns to restore renderings rather than to undo artifacts specific to 2DGS.

Ablations. We ablate the components of SplatRefiner and the grouping of target views at inference. Table 2 starts from a base refiner trained on renderings and adds the components cumulatively (DL3DV-Evaluation, 3 input views, 2DGS). The VGGT features of the reference and target views

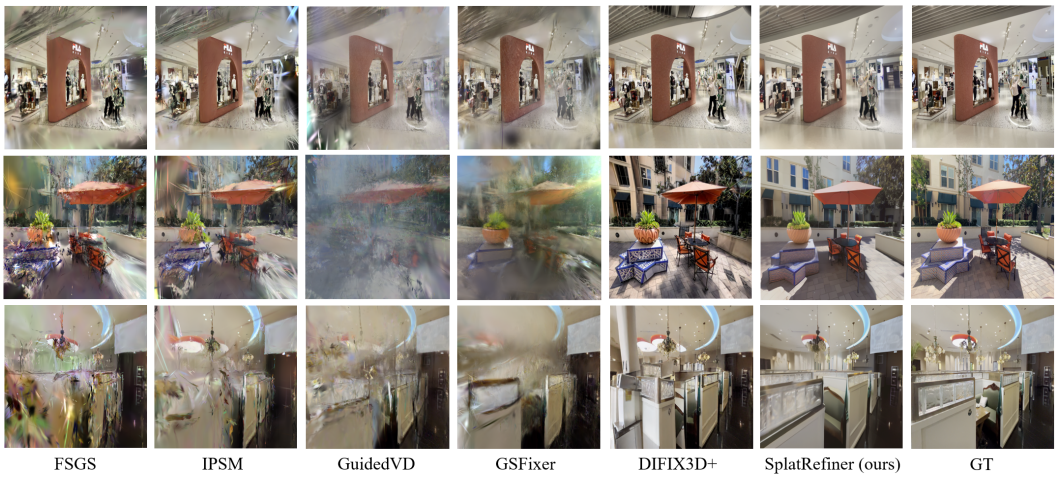

Figure 4: Qualitative comparison of sparse-view novel view synthesis results.



Figure 5: Qualitative comparison with DIFIX under direct post-hoc refinement. Red and green boxes highlight artifact removal and missing-region completion, respectively.

improve all four metrics, adding 0.34 dB PSNR. Reference guidance adds a further 0.56 dB, with gains in SSIM and LPIPS at a small cost in FID. Since every row already uses the base model's guidance on the rendering, this gain comes from the reference term alone. Appendix A.5 decomposes the improvement from the pretrained base model. Table 3 replaces consecutive grouping of the requested views with random grouping (6 input views). PSNR drops by only 0.20 dB and SSIM by 0.009, while FID improves, so refinement does not depend on the views forming a continuous camera path.

Table 2: Component ablation (DL3DV-Evaluation, 3 views, 2DGS), added cumulatively to a base refiner trained on renderings.

Variant	PSNR↑	SSIM↑	LPIPS↓	FID↓
Base refiner	13.49	0.418	0.450	31.77
+ VGGT features	13.83	0.424	0.441	**29.50**
+ reference guidance	**14.39**	**0.439**	**0.428**	31.14

Table 3: Consecutive vs. random grouping of target views (DL3DV-Evaluation, 6 views, 2DGS).

Grouping	PSNR↑	SSIM↑	LPIPS↓	FID↓
Consecutive	**16.76**	**0.520**	**0.330**	19.97
Random	16.56	0.511	0.336	**19.01**

6 Conclusion

We presented SplatRefiner, a trajectory-free post-hoc diffusion refiner for sparse-view novel view synthesis. It restores renderings of a fixed Gaussian reconstruction at arbitrary requested poses directly into final outputs, without reconstruction–generation cycles or 3D re-optimization. To keep these outputs faithful, SplatRefiner exploits both the reference views and the target view. During training, VGGT features of the reference views condition denoising and those of the ground-truth target view supervise intermediate representations. At inference, dual-condition guidance strengthens the contributions of the reference views and the rendering. On identical renderings, it outperforms DIFIX in all but one evaluated setting; against DIFIX3D+, it reaches comparable or higher PSNR at 3.7–6.9× lower end-to-end cost. Trained only on 2DGS renderings, it also transfers across Gaussian reconstructions and datasets. Appendix A.10 discusses its limitations.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code implementation, the statistical analysis of our experimental results (including the paired bootstrap confidence intervals reported in Figures 3 and 7), feedback on our experimental design and analysis protocol, the generation of plots and figures, and drafting and editing the manuscript for clarity and readability. All research ideas, methodological and experimental design choices, and scientific interpretations were developed by the authors. All AI-assisted work was reviewed and verified by the authors: every reported number was produced by executing our code rather than generated by a model. The authors take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, Section 4 describes the post-hoc refinement formulation, the reference-feature injection and target-feature alignment mechanisms, and the dual-condition guidance used at inference. Section 5.1 details the datasets, reference-view sampling, Gaussian reconstruction settings, base diffusion model, optimization hyperparameters, guidance strengths, and hardware. Appendices A.1, A.2, and A.4 specify the VGGT–UNet level pairing and candidate sites, the projector and cross-attention architectures, and the timestep sampling scheme. Appendix A.9 documents the consistency and correspondence protocols, including the excluded scene. Our source code, configuration files, and supporting materials will be publicly released upon publication.

REFERENCES

Mohammad Asim, Christopher Wewer, Thomas Wimmer, Bernt Schiele, and Jan Eric Lenssen. MEt3R: Measuring multi-view consistency in generated images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Yitong Dong, Qi Zhang, Minchao Jiang, Zhiqiang Wu, Qingnan Fan, Ying Feng, Huaqi Zhang, Hujun Bao, and Guofeng Zhang. One-shot refiner: Boosting feed-forward novel view synthesis via one-step diffusion. In *AAAI Conference on Artificial Intelligence*, 2026.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: Create anything in 3D with multi-view diffusion models. In *Advances in Neural Information Processing Systems*, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42, 2023.

Dongxu Li, Junnan Li, and Steven C. H. Hoi. BLIP-Diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *Advances in Neural Information Processing Systems*, 2023.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Xi Liu, Chaoyi Zhou, and Siyu Huang. 3DGS-Enhancer: Enhancing unbounded 3D gaussian splatting with view-consistent 2D diffusion priors. In *Advances in Neural Information Processing Systems*, 2024.

Baorui Ma, Huachen Gao, Haoge Deng, Zhengxiong Luo, Tiejun Huang, Lulu Tang, and Xinlong Wang. You see it, you got it: Learning 3D creation on pose-free videos at scale. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics*, 2019.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2I-Adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI Conference on Artificial Intelligence*, 2024.

Junfeng Ni, Yixin Chen, Zhifei Yang, Yu Liu, Ruijie Lu, Song-Chun Zhu, and Siyuan Huang. G4Splat: Geometry-guided gaussian splatting with generative prior. In *International Conference on Learning Representations*, 2026.

Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? In *International Conference on Learning Representations*, 2026.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-REPA: Aligning diffusion U-Nets to ViTs. In *Advances in Neural Information Processing Systems*, 2025.

Hanyang Wang, Fangfu Liu, Jiawei Chi, and Yueqi Duan. VideoScene: Distilling video diffusion model to generate 3D scenes in one step. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025b.

Qisen Wang, Yifan Zhao, Jiawei Ma, and Jia Li. How to use diffusion priors under sparse views? In *Advances in Neural Information Processing Systems*, 2024a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3D vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. ELITE: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3D representation for consistent world modeling. In *International Conference on Learning Representations*, 2026.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. DIFIX3D+: Improving 3D reconstructions with single-step diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a.

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. ReconFusion: 3D reconstruction with diffusion priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Sibo Wu, Congrong Xu, Binbin Huang, Andreas Geiger, and Anpei Chen. GenFusion: Closing the loop between reconstruction and generation via videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025b.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Xingyilang Yin, Qi Zhang, Jiahao Chang, Ying Feng, Qingnan Fan, Xi Yang, Chi-Man Pun, Huaqi Zhang, and Xiaodong Cun. GSFixer: Improving 3D gaussian splatting with reference-guided video diffusion priors. In *International Conference on Machine Learning*, 2026.

Jason J. Yu, Fereshteh Forghani, Konstantinos G. Derpanis, and Marcus A. Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. VideoREPA: Learning physics for video generation through relational alignment with foundation models. In *Advances in Neural Information Processing Systems*, 2025.

Yingji Zhong, Zhihao Li, Dave Zhenyu Chen, Lanqing Hong, and Dan Xu. Taming video diffusion prior with scene-grounding guidance for 3D gaussian splatting from sparse inputs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Liyuan Zhu, Manjunath Narayana, Michal Stary, Will Hutchcroft, Gordon Wetzstein, and Iro Armeni. GaussFusion: Improving 3D reconstruction in the wild with a geometry-informed video generator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, 2024.

A Appendix

A.1 VGGT–UNet Pairing and Site Configurations

We derive the multi-layer correspondence from the design of VGGT's own geometry head. The DPT head that predicts depth and point maps does not read the final layer alone: it takes intermediate features from layers 4, 11, 17, and 23 and resizes them by $4\times$, $2\times$, $1\times$, and $1/2\times$ into a four-scale pyramid that is fused starting from the deepest layer. We transfer this ordering to the diffusion UNet, which has a four-level resolution hierarchy of its own.

Table 4: Candidate sites of the diffusion UNet and their paired VGGT layers. Level k pairs UNet Lk (finest L1 to coarsest L4) with the k-th layer read by VGGT's geometry head. Alignment sites are the last block of each level, so decoder features are read after upsampling; injection sites on the decoder side are the first block of each level, matching the encoder sites in resolution and channel width. L1† is the REPA-style single site that pairs encoder level L1 with the final VGGT layer.

		UNet encoder		UNet decoder			VGGT		
	Level	Ch.	Res.	Ch.	Res. (align)	Res. (inject)	Layer	Ch.	Res.
Multi-layer	L1	320	64×64	320	64×64	64×64	4	2048	37×37
	L2	640	32×32	640	64×64	32×32	11	2048	37×37
	L3	1280	16×16	1280	32×32	16×16	17	2048	37×37
	L4	1280	8×8	1280	16×16	8×8	23	2048	37×37
Single-layer	L1†	320	32×32	–	–	–	23	2048	37×37

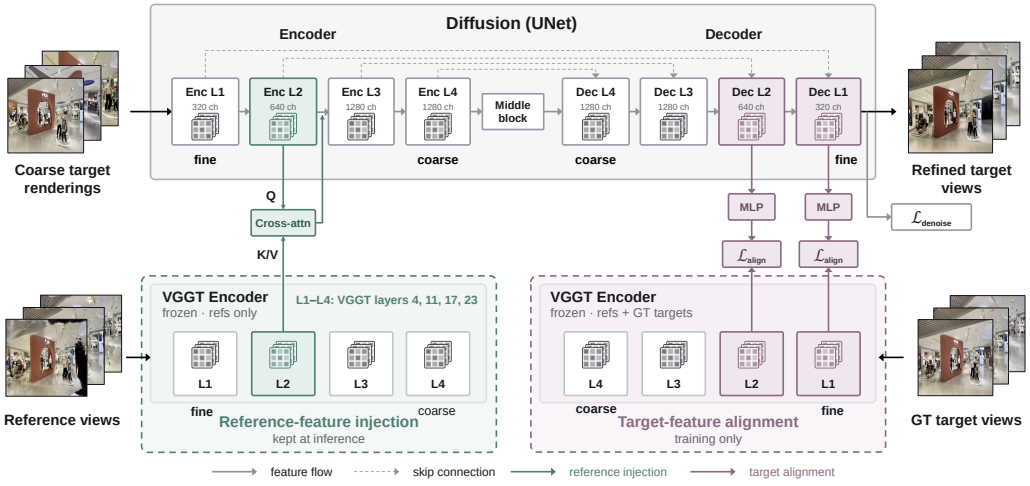

Figure 6: VGGT–diffusion connections explored in our analysis. Candidate injection and alignment sites are shown across UNet levels, with the highlighted paths indicating the final configuration.

Table 4 lists the resulting sites for alignment and injection. The UNet consists of four encoder levels, a middle block, and four decoder levels; the middle block is not used as a candidate site. Because the encoder runs fine-to-coarse and the decoder coarse-to-fine, the same assignment pairs shallow VGGT layers with the early encoder levels and with the late decoder levels. For alignment, each site is the last block of its level: downsampling blocks are excluded on the encoder side, while on the decoder side the upsampler belongs to that last block, so decoder features are read after upsampling. For injection, the encoder sites are unchanged, whereas decoder sites are taken at the first block of each decoder level, so that injected features are processed by the level's remaining blocks; this also makes the decoder sites skip-connected counterparts of the encoder sites, with identical resolution and channel width. The single-site baseline, denoted L1† in the main text, follows REPA: we take the output of encoder level L1, whose 32×32 resolution is closest to the VGGT grid, and pair it with the final VGGT layer.

A.2 Alignment and Injection Details

Following the VGGT DPT decoder design, we use the full 2048-dimensional VGGT representation, formed by concatenating the frame-level and global features, as the alignment target. Each selected VGGT feature is bilinearly resized to the spatial resolution of its paired UNet representation before alignment, analogous to how the DPT decoder resamples intermediate features to the resolutions of the stages that consume them. For each alignment site k, a three-layer MLP projector P_{ψ_k} maps the corresponding UNet representation to the 2048-dimensional VGGT feature space. The projector consists of two 2048-dimensional hidden layers with SiLU activations, followed by a final linear layer with an output dimension of 2048. The same 2048-dimensional representation is used for reference-feature injection, where the query, key, and value projections in Eq. (2) are each a single linear layer applied after LayerNorm, and the output projection is zero-initialized (Section 3.2).

A.3 Placement Analysis: Held-Out Evaluation and Final Selection

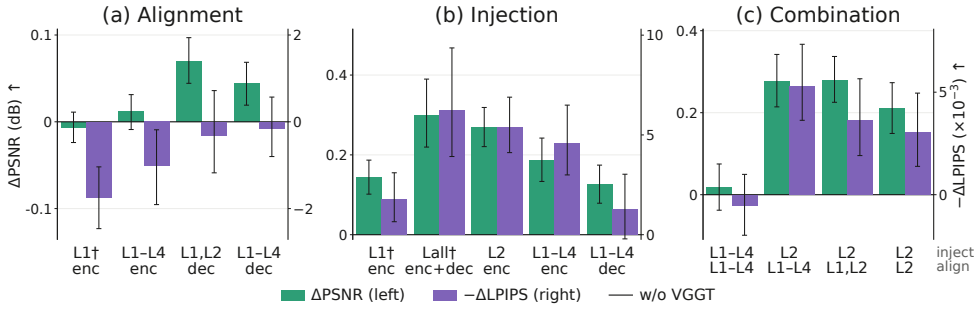

Figure 7: Placement analysis on the evaluation set, in the same format as Figure 3: alignment alone (a), injection alone (b), or both (c). DL3DV-Evaluation, 53 scenes, 6 input views, 50k steps; bars are the change from the w/o-VGGT baseline and error bars are 95% paired bootstrap intervals across scenes.

Figure 7 repeats the placement analysis of Section 3 on the held-out DL3DV-Evaluation split (53 scenes, 6 input views, 50k iterations), with the same configurations and format as Figure 3. The conclusions drawn on the validation split carry over. For alignment, encoder-side sites bring no PSNR gain regardless of their number, only decoder-side alignment improves PSNR, and level-matched decoder alignment again outperforms the REPA-style site (L1†) on both metrics. For injection, encoder sites outperform decoder sites; the level-matched single site (L2) matches the all-block recipe (Lall†) in PSNR and LPIPS and clearly exceeds the REPA-style site (L1†), and injecting at all four encoder sites brings no further PSNR gain. For the combination, combining all sites remains the worst option, reducing injection to the level-matched single site restores the gain, dropping to a single alignment site loses part of it, and four alignment sites are no better than two. With more than twice as many scenes, the evaluation split yields narrower intervals than the validation split.

Single-checkpoint comparisons between the strongest candidates are of the same order as the variation of one configuration across checkpoints, so we select the final configuration from full training curves. Figure 8 tracks the candidates of Figure 3 on the validation split every 10k iterations. Injection at Enc L2 with alignment at Dec L1–L2 is the highest in PSNR and SSIM at every checkpoint from 10k to 50k, while its LPIPS stays on par with injection alone. We therefore adopt this configuration and take its 40k checkpoint, where its PSNR and SSIM peak (16.69 dB and 0.495) and its LPIPS is lowest (0.349).

A.4 Diffusion Noise-Level Sampling

For cubic sampling, we first sample $u \sim \mathcal{U}(0, 1)$ and transform it as

$$t = \left\lfloor t_{\min} + (1 - u^3)(t_{\max} - t_{\min}) \right\rfloor . \tag{6}$$

This transformation biases samples toward larger timesteps. Before integer rounding, the expected normalized timestep increases from 0.5 under uniform sampling to 0.75 under cubic sampling, thereby placing greater emphasis on the high-noise regime.

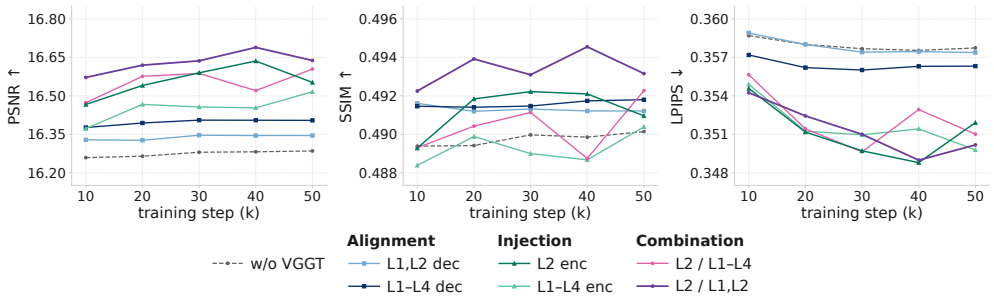

Figure 8: Validation curves of the candidate placements over training (DL3DV-Validation, 20 scenes, 6 input views), evaluated every 10k iterations. The dashed line is the refiner without VGGT, combination labels read injection sites / alignment sites.

A.5 Component Ablation

Table 5 decomposes the improvement on DL3DV-Evaluation with 3 input views and 2DGS renderings. Each row in the lower block adds one component to the row above. Applied to the same renderings without training, the pretrained See3D reduces FID by 37% (129.33 to 81.32) but raises PSNR by only 0.36 dB. DIFIX shows the same pattern on these renderings, with FID reduced by 64% and PSNR raised by 0.29 dB (Table 1). The two models differ in architecture, conditioning, and training, so this behavior is not specific to one model. Used off the shelf, both move the renderings toward the distribution of real images but hardly closer to the ground truth at the pixel level.

Fidelity instead comes from what SplatRefiner adds. Training on renderings, i.e., training See3D on our data with the training scheme of Sections 4.1 and 5.1 but without the VGGT features, adds 0.86 dB. Replacing the uniform timestep sampling and tiled VAE decoding of the base model's code with cubic timestep sampling (Appendix A.4) and full-resolution VAE decoding adds a further 0.52 dB. The VGGT features improve all four metrics and adds another 0.34 dB, and reference guidance add 0.56 dB at a small cost in FID. Together, these three additions improve on training on renderings alone by 1.42 dB in PSNR, 0.038 in SSIM, and 0.033 in LPIPS, and account for 62%, 59%, 32%, and 4% of the total improvement over See3D in PSNR, SSIM, LPIPS, and FID, respectively. The closer a metric is to pixel-level agreement with the ground truth, the more it depends on these additions, whereas distributional realism is almost entirely inherited from the diffusion prior and training on renderings.

Table 5: Component ablation (DL3DV-Evaluation, 3 views, 2DGS). Components are added cumulatively; every row uses the base model's guidance on the rendering, and ref. guidance adds the reference term.

Variant	PSNR↑	SSIM↑	LPIPS↓	FID↓
2DGS	11.75	0.332	0.539	129.33
See3D (pretrained)	12.11	0.375	0.531	81.32
Training on renderings	12.97	0.401	0.461	33.08
+ sampling & decoding	13.49	0.418	0.450	31.77
+ VGGT features	13.83	0.424	0.441	29.50
+ reference guidance	14.39	0.439	0.428	31.14

A.6 Additional Quantitative Results

Tables 6 and 7 report results with six and nine input views on DL3DV-Evaluation and ScanNet++, and Table 8 reports results on Mip-NeRF 360 with three, six, and nine input views. Although trained only on 2DGS renderings from DL3DV-10K, SplatRefiner improves PSNR, LPIPS, and FID over the unrefined 2DGS, 3DGS, and FSGS renderings in every setting of these tables. SSIM also

improves except for FSGS at six and nine views on DL3DV-Evaluation and Mip-NeRF 360 (decreasing by up to 0.017 and 0.080, respectively), where the unrefined FSGS renderings already have the highest SSIM among the three reconstructions; this is consistent with the limitation discussed in Appendix A.10. On DL3DV-Evaluation and Mip-NeRF 360, FSGS renderings yield the highest PSNR after refinement: SplatRefiner with FSGS exceeds DIFIX3D+ in PSNR at every view count, attains the highest PSNR among all methods at six views on DL3DV-Evaluation, and is second in PSNR only to GSFixer at every view count on Mip-NeRF 360. DIFIX3D+ retains the best LPIPS and FID on these two datasets, with SplatRefiner second, except at nine views on Mip-NeRF 360, where SplatRefiner with FSGS is best in both. On ScanNet++, 2DGS and 3DGS instead surpass FSGS on all four metrics after refinement at both view counts, although the unrefined FSGS renderings have the highest SSIM. There, SplatRefiner with 2DGS exceeds DIFIX3D+ on all four metrics and, among all methods, achieves the best results on all four metrics at six views and the best PSNR, LPIPS, and FID at nine views. Across the three tables, SplatRefiner with any reconstruction method attains lower FID than every method in the diffusion-assisted reconstruction group, and it outperforms DIFIX applied to the same renderings in PSNR, LPIPS, and FID in every setting.

Table 6: Quantitative comparison on DL3DV-Evaluation with 6 and 9 input views. Bold and underlined values indicate the best and second-best results, ranked separately for methods with and without reconstruction–generation cycles.

Method	6 views				9 views			
	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
Diffusion-assisted reconstruction								
IPSM (Wang et al., 2024a)	17.10	0.586	0.393	67.96	19.17	<u>0.660</u>	0.339	50.01
GenFusion (Wu et al., 2025b)	17.17	<u>0.589</u>	0.381	<u>59.20</u>	**19.25**	**0.666**	0.307	<u>38.41</u>
G4Splat (Ni et al., 2026)	10.06	0.235	0.626	136.26	10.44	0.247	0.619	129.18
GuidedVD (Zhong et al., 2025)	15.36	0.487	0.444	83.67	16.84	0.541	0.389	62.37
GSFixer (Yin et al., 2026)	**17.38**	**0.590**	<u>0.367</u>	59.91	<u>19.22</u>	<u>0.660</u>	<u>0.300</u>	41.27
Diffusion-assisted reconstruction + refinement								
DIFIX3D+ (Wu et al., 2025a)	<u>17.21</u>	0.577	**0.301**	**16.93**	19.04	0.647	**0.251**	**13.33**
Direct post-hoc refinement								
2DGS (Huang et al., 2024)	13.84	0.416	0.465	87.16	15.78	0.497	0.402	62.49
+ DIFIX	14.12	0.421	0.412	27.05	15.95	0.498	0.351	19.92
+ **SplatRefiner (Ours)**	<u>16.76</u>	0.520	<u>0.330</u>	19.97	18.30	0.573	<u>0.279</u>	16.66
3DGS (Kerbl et al., 2023)	14.16	0.422	0.465	90.81	15.75	0.481	0.415	70.61
+ DIFIX	14.36	0.428	0.410	25.27	15.86	0.489	0.359	19.93
+ **SplatRefiner (Ours)**	16.56	0.511	0.337	<u>19.78</u>	17.95	0.559	0.289	<u>16.46</u>
FSGS (Zhu et al., 2024)	16.63	**0.562**	0.437	92.73	<u>18.36</u>	**0.622**	0.396	80.34
+ DIFIX	16.65	0.537	0.361	26.53	18.25	0.596	0.313	20.63
+ **SplatRefiner (Ours)**	**17.75**	<u>0.559</u>	**0.305**	**19.38**	**19.17**	<u>0.605</u>	**0.259**	**15.94**

A.7 EFFECT OF SAMPLING STEPS AND REFERENCE GUIDANCE STRENGTH

Table 9(a) varies the number of sampling steps on DL3DV-Evaluation with 6 input views and 2DGS renderings. Without reference guidance (unguided), reducing the steps from 50 to 4 raises PSNR by 0.41 dB and SSIM by 0.021 but worsens LPIPS from 0.343 to 0.363 and FID from 19.89 to 34.42, the usual trade-off between distortion and perceptual quality. Reference guidance relaxes this trade-off rather than moving along it. With 50 steps, it reaches a higher PSNR (16.76) than any unguided setting (at most 16.68). Its FID is nearly the best unguided value (19.97 vs. 19.89), and its LPIPS (0.330) is the best in the table. Fidelity thus improves without sacrificing realism. However, guidance needs enough steps to improve fidelity. With 8 steps, it improves all four metrics over the unguided 8-step setting and gives the highest PSNR (16.90). With 4 steps, it instead lowers PSNR by 0.23 dB and SSIM by 0.022 and worsens LPIPS, although FID still improves. The default setting (50 steps with guidance) gives the best LPIPS and near-best FID, while 8 guided steps are a faster alternative when PSNR and SSIM matter more, at the cost of perceptual quality (FID 22.01 vs. 19.97).

Table 9(b) varies the guidance strength γ with 50 steps. Most of the benefit is already obtained at $\gamma = 0.25$. Relative to $\gamma = 0$, it raises PSNR by 0.41 dB, out of the 0.49 dB gained at the default $\gamma = 0.5$, while leaving FID unchanged (19.88 vs. 19.89). For $\gamma \in [0.25, 1.0]$, PSNR varies by less

Table 7: Quantitative comparison on ScanNet++ with 6 and 9 input views. Bold and underlined values indicate the best and second-best results, ranked separately for methods with and without reconstruction–generation cycles.

Method	6 views				9 views			
	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
Diffusion-assisted reconstruction								
IPSM (Wang et al., 2024a)	14.81	0.698	0.447	217.92	16.52	0.732	0.397	180.77
GenFusion (Wu et al., 2025b)	13.69	0.658	0.453	211.94	16.34	0.719	0.385	153.02
G4Splat (Ni et al., 2026)	11.65	0.496	0.548	222.26	11.76	0.508	0.539	219.05
GuidedVD (Zhong et al., 2025)	<u>16.02</u>	<u>0.702</u>	<u>0.401</u>	<u>170.21</u>	**18.72**	**0.752**	<u>0.338</u>	<u>129.26</u>
GSFixer (Yin et al., 2026)	14.97	0.681	0.438	222.46	18.02	<u>0.747</u>	0.352	159.52
Diffusion-assisted reconstruction + refinement								
DIFIX3D+ (Wu et al., 2025a)	**16.24**	**0.708**	**0.335**	**49.65**	<u>18.26</u>	0.746	**0.276**	**42.24**
Direct post-hoc refinement								
2DGS (Huang et al., 2024)	12.72	0.573	0.447	173.65	16.70	0.683	0.358	133.03
+ DIFIX	12.97	0.589	0.397	67.73	16.80	0.690	0.307	54.50
+ **SplatRefiner (Ours)**	**17.02**	**0.716**	**0.292**	**42.36**	**19.43**	**0.751**	**0.235**	**35.12**
3DGS (Kerbl et al., 2023)	12.97	0.592	0.439	168.74	16.96	0.687	0.360	132.89
+ DIFIX	13.15	0.599	0.392	64.82	16.99	0.692	0.309	52.89
+ **SplatRefiner (Ours)**	<u>16.68</u>	<u>0.707</u>	<u>0.302</u>	<u>43.72</u>	<u>18.77</u>	<u>0.738</u>	<u>0.250</u>	<u>35.61</u>
FSGS (Zhu et al., 2024)	14.93	0.677	0.459	213.09	16.79	0.719	0.421	203.74
+ DIFIX	15.26	0.671	0.399	81.84	17.39	0.716	0.343	72.06
+ **SplatRefiner (Ours)**	16.05	0.693	0.336	45.63	18.17	0.728	0.274	38.33

Table 8: Quantitative comparison of sparse-view novel view synthesis methods on Mip-NeRF 360.

Method	3 views				6 views				9 views			
	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
Diffusion-assisted reconstruction												
IPSM (Wang et al., 2024a)	14.33	0.324	0.560	241.77	16.40	<u>0.458</u>	0.467	163.38	<u>18.55</u>	<u>0.551</u>	0.404	116.53
GenFusion (Wu et al., 2025b)	13.38	0.265	0.555	230.58	15.91	0.428	0.440	148.39	17.97	0.527	0.367	105.95
G4Splat (Ni et al., 2026)	11.48	0.177	0.643	269.43	12.63	0.226	0.629	239.54	13.10	0.257	0.605	221.17
GuidedVD (Zhong et al., 2025)	13.24	0.279	0.605	252.31	15.41	0.357	0.532	181.56	16.71	0.400	0.478	148.58
GSFixer (Yin et al., 2026)	**15.68**	**0.385**	<u>0.545</u>	<u>204.65</u>	**18.11**	**0.501**	<u>0.430</u>	<u>125.03</u>	**19.64**	**0.568**	<u>0.356</u>	<u>92.10</u>
Diffusion-assisted reconstruction + refinement												
DIFIX3D+ (Wu et al., 2025a)	<u>14.97</u>	<u>0.328</u>	**0.465**	**111.07**	<u>16.86</u>	0.431	**0.385**	**79.56**	18.12	0.497	**0.342**	**69.27**
Direct post-hoc refinement												
2DGS (Huang et al., 2024)	12.07	0.205	0.611	249.30	14.31	0.291	0.531	184.69	15.72	0.354	0.475	145.41
+ DIFIX	12.48	0.204	0.570	156.05	14.51	0.282	0.487	111.25	15.83	0.338	0.440	91.71
+ **SplatRefiner (Ours)**	<u>14.50</u>	0.315	<u>0.515</u>	**112.16**	<u>16.71</u>	0.376	<u>0.420</u>	**81.87**	18.11	0.413	<u>0.374</u>	71.18
3DGS (Kerbl et al., 2023)	12.15	0.214	0.613	264.14	14.15	0.279	0.549	199.33	15.62	0.329	0.496	168.75
+ DIFIX	12.49	0.208	0.569	160.20	14.42	0.277	0.494	111.87	15.74	0.324	0.451	91.01
+ **SplatRefiner (Ours)**	14.17	0.308	0.529	129.19	16.36	0.366	0.435	88.51	17.79	0.402	0.384	<u>70.07</u>
FSGS (Zhu et al., 2024)	14.23	<u>0.324</u>	0.583	256.57	16.37	**0.447**	0.506	181.55	<u>18.49</u>	**0.531**	0.449	147.56
+ DIFIX	14.33	0.287	0.525	151.66	16.33	0.399	0.434	102.65	18.07	<u>0.471</u>	0.383	84.25
+ **SplatRefiner (Ours)**	**15.09**	**0.339**	**0.493**	<u>119.33</u>	**17.61**	<u>0.413</u>	**0.387**	<u>82.30</u>	**19.17**	0.451	**0.336**	**66.01**

than 0.1 dB and SSIM and LPIPS by at most 0.002. FID rises only slightly, though monotonically, from 19.88 to 20.29. The results therefore do not hinge on tuning γ, and the default lies within this flat region. Guidance only re-weights the model's own predictions with and without the references, so it adds no new information. Its early saturation suggests that a small correction suffices, i.e., the model already exploits most of the reference evidence it receives through its inputs and the injected VGGT features.

A.8 Qualitative Comparison with DIFIX3D+

Figure 9 compares SplatRefiner with DIFIX3D+ on ScanNet++ with 6 input views, where SplatRefiner refines 2DGS renderings. Red boxes mark the compared regions. SplatRefiner restores the structures and details that DIFIX3D+ alters, consistent with the quantitative results on ScanNet++ in Table 7.

Table 9: Effect of sampling steps and reference guidance strength on DL3DV-Evaluation with 6 input views and 2DGS renderings. Bold marks the best value in each column. Time is the refinement time per scene. All settings in (b) use 50 sampling steps. The default setting (50 steps with guidance, $\gamma = 0.5$) is the last row of (a) and the third row of (b).

(a) Sampling steps

Guidance	Steps	PSNR↑	SSIM↑	LPIPS↓	FID↓	Time↓
w/o	4	16.68	**0.530**	0.363	34.42	52s
	8	16.58	0.520	0.347	24.70	56s
	50	16.27	0.509	0.343	**19.89**	104s
w/	4	16.45	0.508	0.378	31.82	**51s**
	8	**16.90**	0.525	0.334	22.01	56s
	50	16.76	0.520	**0.330**	19.97	119s

(b) Guidance strength

γ	PSNR↑	SSIM↑	LPIPS↓	FID↓
0 (w/o)	16.27	0.509	0.343	19.89
0.25	16.68	0.519	0.331	**19.88**
0.5 (default)	16.76	**0.520**	**0.330**	19.97
0.75	**16.77**	0.519	**0.330**	20.15
1.0	16.74	0.518	0.332	20.29

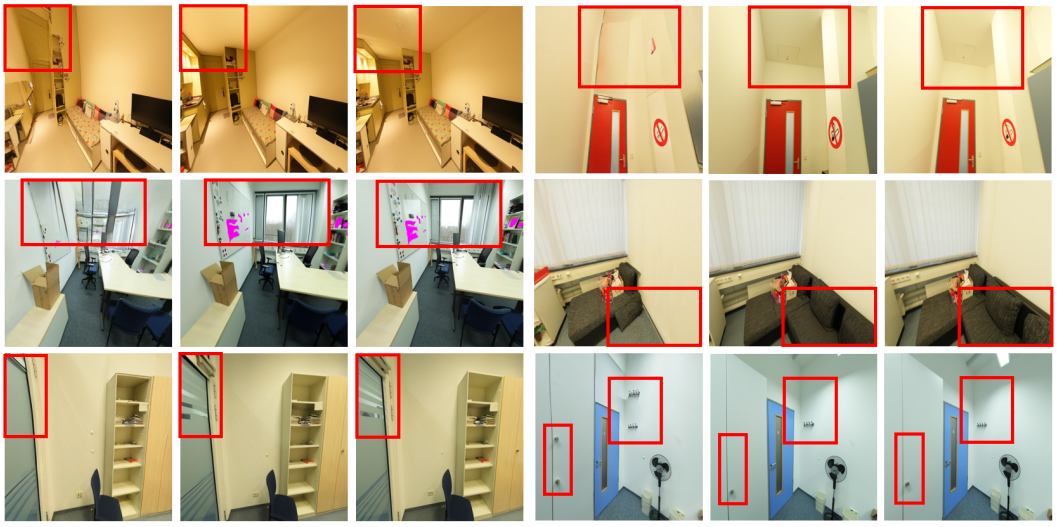

Figure 9: Qualitative comparison with DIFIX3D+ on ScanNet++. DIFIX3D+ (a) extends the wall texture into ceilings, (b) distorts or truncates large objects, and (c) loses fine details or duplicates small objects, whereas SplatRefiner restores them.

A.9 Cross-view Consistency and Ground-truth Correspondence

Beyond per-view fidelity, we compare the refiners on how consistent their outputs are across views and how closely they correspond to the ground truth and to the reference images. All methods receive the same 2DGS renderings with 6 input views (Table 10). All metrics in this section are computed over 52 of the 53 scenes. The excluded scene has a degenerate intrinsic matrix and no method reconstructs it: there, even the ground-truth views fail the epipolar test.

Cross-view consistency. We use TSED (Yu et al., 2023), which calls a pair of views consistent when enough correspondences are found between them, and their median symmetric epipolar distance, under the given camera poses, falls below a threshold. We also use MEt3R (Asim et al., 2025), which measures the feature discrepancy between two views after aligning them through 3D reconstruction. For each scene, test views are sorted by frame index, and every pair of consecutive test views forms an evaluation pair; all methods are evaluated on the same pairs. All images are downsampled to 256×256, the resolution used in the original TSED evaluation, at which its 2-pixel threshold is defined. For TSED, we follow the official implementation with its default thresholds and report the percentage of consistent pairs, together with the mean number of SIFT correspondences per pair. For MEt3R, we use the official implementation with the DUSt3R (Wang et al., 2024b) backbone and the earlier view of each pair as the first input; lower values indicate higher consis-

tency. Ground-truth images are evaluated in the same way and serve as a reference, since imperfect poses, matching, and reconstruction keep even real images short of the ideal value of either metric. DIFIX lowers TSED below that of the unrefined renderings (73.8 vs. 77.5) and yields fewer correspondences (109.7 vs. 127.0), although it improves MEt3R (0.285 vs. 0.305). SplatRefiner improves MEt3R (0.227) and the number of correspondences (158.1) by a clear margin while matching the unrefined renderings on TSED (78.5 vs. 77.5).

Table 10: Cross-view consistency and correspondence with the ground truth.

	Cross-view consistency			Ground-truth correspondence			
	TSED@2 ↑	SIFT matches ↑	MEt3R ↓	Corr. ↑	≤ 2 px ↑	Median disp. ↓	Ref. ratio ↑
Ground truth	91.9	184.7	0.220	–	–	–	–
2DGS	77.5	127.0	0.305	948	687	9.21	0.594
DIFIX	73.8	109.7	0.285	1123	714	10.70	0.782
SplatRefiner (ours)	**78.5**	**158.1**	**0.227**	**1686**	**1146**	**7.93**	**0.910**

Correspondence with ground truth and references. To measure how faithfully each method reproduces the observed scene, we match each refined view with LoFTR (Sun et al., 2021) against (i) the ground-truth image at the same pose and (ii) the six reference images. For (i), we report the mean number of correspondences (confidence ≥ 0.5), how many of them lie within 2 px of their ground-truth position, and the median displacement between matched points, which would be zero for a perfect reconstruction and is computed only for views with at least 16 correspondences; for (ii), we report the number of correspondences to the references relative to that of the ground-truth image, which equals one when the refined view shares as much content with the references as the ground truth does. DIFIX increases the correspondences with the ground truth (1123 vs. 948) and the reference ratio (0.782 vs. 0.594), but while it adds 175 correspondences, the number lying within 2 px increases by only 27 (714 vs. 687), and the median displacement rises (10.70 vs. 9.21 px). SplatRefiner adds 738 correspondences and increases the number within 2 px by 459 (1146 vs. 687), while raising the reference ratio to 0.910 without degrading the displacement (7.93 vs. 9.21 px).

A.10 Limitations and future work

SplatRefiner has two main limitations. First, it keeps the reconstruction fixed, and its rendering is the only input that specifies the requested view. When the rendering shows wrong content, the requested view cannot be inferred, and SplatRefiner produces a plausible but different scene (Figure 10, top), whereas DIFIX3D+, which re-optimizes the reconstruction, recovers the overall layout. Second, its training objective supervises the denoising of target latents without directly constraining the decoded images, so nothing explicitly enforces agreement with the observed scene. Even when the scene can be inferred from a blurred rendering and the reference images, SplatRefiner can over-smooth fine texture that DIFIX3D+ recovers more sharply (Figure 10, bottom). Constraints that enforce agreement with the reference images, such as a reprojection loss, could correct such errors while preserving the realism and efficiency of post-hoc refinement.



Figure 10: Failure cases.