FINERSPLAT: DISTILLING DIFFUSION PRIORS FOR FEED-FORWARD GAUSSIAN REFINEMENT

Anonymous authors
Paper under double-blind review

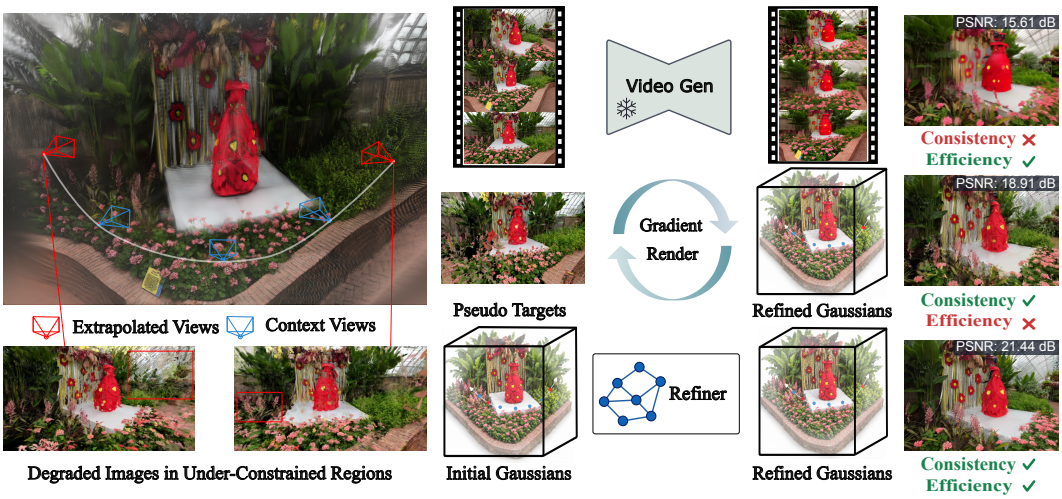


Figure 1: **Illustration. Left:** Blue cameras denote observed context views, whereas red cameras denote extrapolated target views, whose renderings are degraded by limited view coverage. **Right:** Video diffusion models restore these artifacts in 2D but lack explicit 3D representations and cannot support real-time rendering, while gradient-based Gaussian refinement requires costly per-scene optimization. In contrast, FinerSplat uses diffusion-restored images to guide feed-forward Gaussian refinement and pruning, producing high-quality, 3D-consistent renderings with efficient inference.

ABSTRACT

Feed-forward 3D Gaussian Splatting (3DGS) enables efficient reconstruction from sparse views, yet its single-pass prediction must determine both Gaussian attributes and primitive allocation without rendering feedback, often producing inaccurate and redundant primitives in under-constrained regions. Although generative priors can improve rendered images, transferring them into 3D Gaussian representations often requires costly per-scene optimization. We present FinerSplat, a feed-forward framework that casts refinement and compaction as coupled primitive-level decisions without per-scene optimization. It restores target-view renderings with a frozen diffusion model and lifts the resulting residuals into Gaussian-level observations through differentiable rendering contributions. An entropy-aware primitive-level Transformer predicts recurrent attribute updates and primitive-wise keep scores from these observations. Experiments on DL3DV and Waymo demonstrate that FinerSplat performs feed-forward refinement in 0.6 s and removes 30% of the initialized Gaussians while preserving rendering quality.

1 INTRODUCTION

3D reconstruction and novel view synthesis (NVS) are fundamental problems in computer vision, supporting a wide range of applications such as virtual reality, robotics, and autonomous driving Mildenhall et al. (2021); Barron et al. (2023); Miao et al. (2024; 2026). Recent feed-forward 3D Gaussian Splatting methods Szymanowicz et al. (2024); Charatan et al. (2024); Chen et al. (2024a); Tang et al. (2024) directly predict explicit Gaussian representations from sparse input views in a

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.

single network pass, enabling efficient reconstruction and real-time rendering. However, unlike per-scene optimization methods that iteratively refine representations through rendering feedback, feed-forward methods must finalize both Gaussian attributes and primitive allocation from limited observations in a single prediction stage. To ensure sufficient coverage under such ambiguity, they commonly instantiate an over-complete set of primitives. Consequently, this prediction formulation yields inaccurate primitives to correct and redundant ones to remove.

Existing efforts toward high-quality and compact Gaussian representations mainly follow two directions. One route augments feed-forward initialization with refinement modules, including structured fusion and alignment Zhang et al. (2026a); Jiang et al. (2025), feedback-guided iterative updates Chen et al. (2025; 2026), and density-control mechanisms Weng et al. (2025); Feng et al. (2026); Nam et al. (2025). While effective, these methods still derive their refinement signals primarily from observed views, leaving under-constrained regions insufficiently supervised. Another line introduces 2D generative priors to restore degraded renderings and synthesize plausible high-frequency details Chen et al. (2024b). However, such priors are often confined to image-space post-processing, while transferring them into the Gaussian representation typically requires costly per-scene iterative optimization de Lutio et al. (2026). Together, these limitations raise a key question: *How can we refine and compact feed-forward Gaussian representations in under-constrained regions while preserving feed-forward efficiency?*

In this paper, we propose FinerSplat, a feed-forward framework that transforms initial Gaussian predictions into high-quality and compact 3D representations. Our key insight is to cast refinement as coupled primitive-level decisions: each Gaussian receives an attribute update and a retention score based on restoration feedback and entropy-aware local context. Specifically, we render the initial Gaussian representation from target viewpoints that expose under-constrained regions and process the resulting images with a frozen diffusion restorer to obtain pseudo targets. At each refinement stage, residuals between the current renderings and these pseudo targets are aggregated into Gaussian-level feedback using differentiable rendering contributions. A primitive-level Transformer then combines this feedback with Gaussian attributes and entropy-aware local context to predict recurrent attribute updates and primitive-wise keep scores. By operating directly on 3D Gaussians, FinerSplat corrects inaccurate attributes and removes redundant primitives inherited from the over-complete initialization.

Our contributions are summarized as follows: (1) We introduce FinerSplat, a feed-forward framework that formulates Gaussian refinement and compaction as coupled primitive-level decisions without per-scene optimization; (2) we develop a primitive-level feedback mechanism that lifts diffusion restoration residuals into Gaussian-level observations and combines them with entropy-aware local context to predict primitive retention and recurrent attribute updates; and (3) extensive experiments on DL3DV and Waymo demonstrate improved rendering quality with more compact Gaussian representations.

2 Related Work

Feed-Forward 3D Gaussian Splatting: Feed-forward 3DGS predicts explicit Gaussian radiance fields from sparse images in a single forward pass Charatan et al. (2024); Liu et al. (2025); Smart et al. (2024). A common paradigm is pixel-aligned prediction, where Gaussians are instantiated from each input pixel using cost volumes Chen et al. (2024a); Yang et al. (2025), geometric priors Xu et al. (2025b); Song et al. (2025), or large-scale foundation models Lin et al. (2025); Ye et al. (2025). While efficient, dense image-grid prediction often introduces primitive redundancy and memory overhead. To improve primitive allocation, recent works adopt voxel-aligned Gaussians Ren et al. (2024); Wang et al. (2025b); Miao et al. (2025) or adaptive patch-level primitive prediction Zhang et al. (2026b); Moreau et al. (2026). Despite these advances, most feed-forward methods still treat a single-pass prediction as the final representation, limiting their ability to adapt Gaussian attributes and primitive allocation to scene- or task-specific objectives. In contrast, FinerSplat treats feed-forward Gaussians as an initial representation and learns an amortized refinement module that iteratively refines Gaussian attributes while pruning redundant primitives.

NVS with Diffusion Priors: Diffusion priors have recently been explored to improve novel view synthesis under sparse-view and extrapolated settings. One line of work uses pretrained video dif-

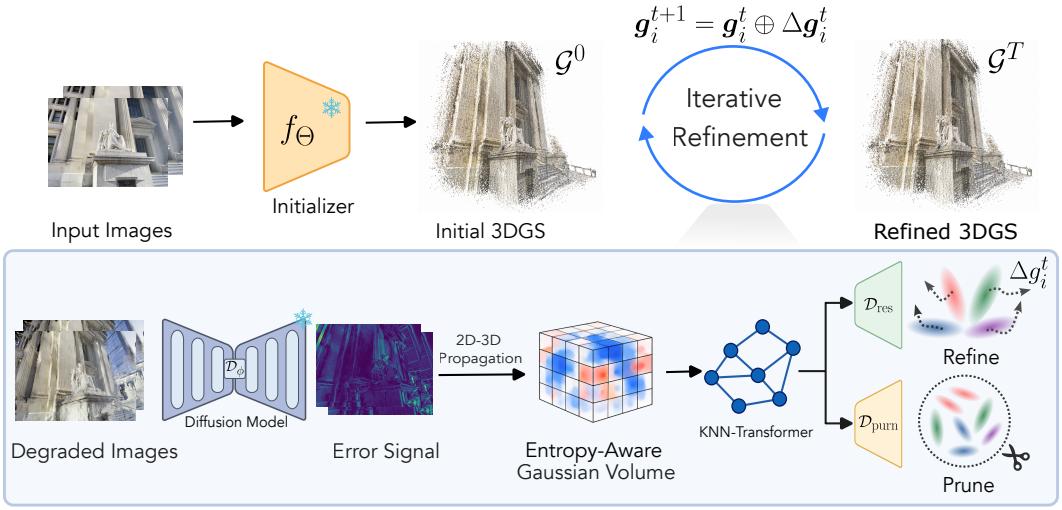


Figure 2: **Method overview.** Given sparse input images, a frozen feed-forward initializer predicts an initial 3D Gaussian representation. At each refinement stage, renderings of the current Gaussians are compared with diffusion-restored pseudo targets to construct 2D error features, which are propagated into an entropy-aware Gaussian volume. A local KNN-Transformer then fuses these cues to predict residual Gaussian updates and keep scores for pruning redundant primitives, producing a refined and compact 3DGS representation.

fusion models to directly synthesize or enhance novel views in both general Chen et al. (2024b); You et al. (2024) and driving scenes Yan et al. (2024); Wang et al. (2025a). In these methods, the primary outputs remain diffusion-generated images or videos rather than persistent refined 3D representations, and each new target trajectory requires costly diffusion sampling. Another line transfers diffusion priors into 3DGS through render-refine-distill pipelines, where diffusion models restore degraded renderings and the refined images serve as pseudo-supervision for per-scene Gaussian optimization Wu et al. (2025a); Yin et al. (2025); Liu et al. (2024); Wu et al. (2025b). More recent methods introduce uncertainty-aware guidance to selectively target unreliable regions and suppress undesired diffusion edits Zhou et al. (2026); Wang et al. (2025d). Although effective, these approaches rely on costly per-scene 2D-3D optimization to transfer diffusion priors into the Gaussian representation. FinerSplat instead amortizes this transfer with a feed-forward refinement module that lifts 2D diffusion residuals into Gaussian-level updates, producing a refined 3D representation without per-scene test-time optimization.

Predict-Then-Refine Scheme: Predict-then-refine methods treat feed-forward Gaussian predictions as an initialization and improve the resulting representation through learned refinement. One line restructures or reallocates predicted Gaussians in 3D space through structured fusion Wang et al. (2025c); Jiang et al. (2025), geometry-aware anchor alignment Zhang et al. (2026a), or primitive-budget adaptation via densification, density control, and super-resolution Nam et al. (2025); Weng et al. (2025); Feng et al. (2026). Other methods exploit rendering feedback for iterative refinement, e.g., gradient guidance Chen et al. (2025) and observed-view residuals Xu et al. (2025a). Most closely related, GIFSplat Chen et al. (2026) incorporates generative priors for iterative Gaussian refinement but keeps the primitive set fixed. In contrast, FinerSplat jointly refines Gaussian attributes and prunes redundant primitives with entropy guidance.

3 Method

Overview: Our goal is to distill diffusion restoration priors into a feed-forward Gaussian refinement process that recurrently refines and compacts initialized Gaussians in under-constrained regions. Given N_c sparse posed context images $\{(\mathbf{I}_n, \mathbf{K}_n, \mathbf{T}_n)\}_{n=1}^{N_c}$, a feed-forward 3DGS model predicts an initial Gaussian set $\mathcal{G}^0$. We further consider a set of novel camera poses $\{\mathbf{T}_m\}_{m=1}^{N_t}$, whose renderings of $\mathcal{G}^0$ often expose artifacts in under-constrained regions due to limited view cov-

erage. Rather than optimizing each scene at test time, we convert diffusion restoration residuals into Gaussian-level observations and train a recurrent refiner to update and compact the initialized Gaussians in a feed-forward manner. We first introduce feed-forward Gaussian initialization in Sec. 3.1, then construct diffusion-guided cues in Sec. 3.2, followed by spatial Gaussian aggregation and the iterative fuse-and-refine network in Sec. 3.3.

3.1 PRELIMINARY

Feed-Forward Gaussian Initialization: 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) represents a scene as a collection of Gaussian primitives $\mathcal{G} = \{\mathbf{g}_i\}_{i=1}^{G}$. Each Gaussian consists of a center $\boldsymbol{\mu}_i$, an opacity α_i, a covariance matrix $\boldsymbol{\Sigma}_i$, and a view-dependent color $\mathbf{c}_i$ (computed by spherical harmonics). Given posed context images $\mathcal{I}_c = \{(\mathbf{I}_n, \mathbf{K}_n, \mathbf{T}_n)\}_{n=1}^{N_c}$, a feed-forward initializer predicts the initial Gaussian representation as

$$f_\Theta : \mathcal{I}_c \mapsto \mathcal{G}^0, \tag{1}$$

where $\mathcal{G}^0$ is predicted in a single forward pass. We adopt DepthSplat Xu et al. (2025b) as the initializer, although FinerSplat is agnostic to this choice. Since single-pass predictions may contain geometric errors and redundant primitives, we use $\mathcal{G}^0$ as the starting point for refinement.

3.2 DIFFUSION-GUIDED RESIDUAL CONSTRUCTION

Diffusion Pseudo Targets: We construct diffusion pseudo targets for novel views that expose under-constrained regions, where feed-forward Gaussian initialization is most fragile. For each target camera $(\mathbf{K}_m, \mathbf{T}_m)$, we use the differentiable Gaussian renderer $\mathcal{R}$ to render the initial representation and then restore it with a frozen single-step diffusion model:

$$\hat{\mathbf{I}}_m^0 = \mathcal{R}(\mathbf{K}_m, \mathbf{T}_m; \mathcal{G}^0), \qquad \tilde{\mathbf{I}}_m = \mathcal{D}_\phi(\hat{\mathbf{I}}_m^0). \tag{2}$$

The initial rendering reveals artifacts caused by insufficient geometric and appearance evidence. We instantiate $\mathcal{D}_\phi$ with DIFIX3D+ Wu et al. (2025a), whose restoration capability benefits from large-scale 2D diffusion priors. Unlike diffusion-based image enhancement that directly refine rendered images Chen et al. (2024b), $\tilde{\mathbf{I}}_m$ is not used as the final output. Instead, the pseudo target provides restoration cues that are distilled into 3D Gaussians in the following sections.

Error Signal Construction: Given the pseudo target $\tilde{\mathbf{I}}_m$, we construct feedback that guides the recurrent refiner toward incremental Gaussian updates. At recurrent stage t, we render the current Gaussian set $\mathcal{G}^t$ from the same target camera and compute the signed RGB residual:

$$\hat{\mathbf{I}}_m^t = \mathcal{R}(\mathbf{K}_m, \mathbf{T}_m; \mathcal{G}^t), \qquad \mathbf{e}_m^{t,\mathrm{rgb}} = \hat{\mathbf{I}}_m^t - \tilde{\mathbf{I}}_m. \tag{3}$$

While the RGB residual $\mathbf{e}_m^{t,\mathrm{rgb}} \in \mathbb{R}^{H\times W\times 3}$ directly captures per-pixel color discrepancy, it lacks neighborhood context and can be sensitive to exposure, white balance, and view-dependent effects. We therefore complement them with the feature-space residual $\mathbf{e}_m^{t,\mathrm{feat}} = \mathrm{MLP}(\psi(\hat{\mathbf{I}}_m^t) - \psi(\tilde{\mathbf{I}}_m))$. Here, ψ comprises the first three stages of an ImageNet-pretrained ResNet-18 He et al. (2016). The resulting 256-D feature difference captures a larger spatial context and is projected to 32 dimensions by a learnable MLP. We concatenate the two complementary residuals as

$$\mathbf{e}_m^t = [\mathbf{e}_m^{t,\mathrm{rgb}}, \mathbf{e}_m^{t,\mathrm{feat}}] \in \mathbb{R}^{H\times W\times 35}. \tag{4}$$

This joint error signal preserves precise color corrections while providing contextual features that are more robust to local appearance variation.

Propagating the Error to Gaussians: To make the diffusion-restored supervision actionable for Gaussian refinement, we lift the image-space error signal back to 3D Gaussian primitives. This step is non-trivial because our pseudo targets are rendered from arbitrary target cameras rather than context views. Unlike pixel-aligned feed-forward settings, where each primitive has a deterministic correspondence to an image pixel Zhang et al. (2026a); Xu et al. (2025a), we rely on differentiable rasterization weights to associate image residuals with the Gaussians that contribute to each pixel. Specifically, for a target camera, let $w_i^m(\mathbf{p})$ denote the contribution weight of Gaussian $\mathbf{g}_i$ to pixel

$\mathbf{p}$ during rendering. Given the image-space residual $\mathbf{e}_m^t$ from each of N_t target views, we compute the Gaussian-level error observation by weighted aggregation:

$$\mathbf{e}_i^t = \frac{1}{N_t} \sum_{m=1}^{N_t} \frac{\sum_{\mathbf{p}} w_i^m(\mathbf{p}) \mathbf{e}_m^t(\mathbf{p})}{\sum_{\mathbf{p}} w_i^m(\mathbf{p}) + \epsilon}. \tag{5}$$

Local Color Differential Entropy: Feed-forward 3DGS initializers densely predict pixel-aligned Gaussians to ensure scene coverage, which can introduce substantial redundancy in low-information regions. To provide the refiner with a local cue for identifying such redundancy, we characterize information richness in the 3D Gaussian space using color entropy. Specifically, we voxelize the initialized Gaussians using a regular grid of resolution R_{vox}^3, where each non-empty voxel defines a local spatial unit. For each non-empty voxel V_j, we compute the per-channel variance $\boldsymbol{\sigma}_j^2 \in \mathbb{R}^3$ of the Gaussian base RGB colors and estimate its local color differential entropy under a diagonal Gaussian approximation:

$$\eta_j = \frac{1}{2} \sum_{c=1}^{3} \log\left(\sigma_{j,c}^2 + \epsilon\right), \tag{6}$$

where ϵ is a numerical stability term. The entropy values are normalized to $[0, 1]$. Each Gaussian then inherits the entropy value of its containing voxel, denoted by η_i, which remains fixed throughout recurrent refinement. In this way, each Gaussian is augmented with a continuous spatial descriptor that captures the local appearance complexity of its surrounding 3D region and complements its Gaussian attributes and propagated rendering-error observations. We therefore construct the input feature for the i-th Gaussian at recurrent stage t as

$$\mathbf{z}_i^t = [\mathbf{g}_i^t, \mathbf{e}_i^t, \eta_i], \tag{7}$$

where $\mathbf{g}_i^t$ denotes the current Gaussian attributes, $\mathbf{e}_i^t$ is the propagated Gaussian-level error feature, and η_i is the local color differential entropy descriptor. The resulting representation $\mathbf{z}_i^t$ is subsequently used for local Gaussian fusion.

3.3 Iterative Fuse-and-Refine Network

Local Gaussian Fusion: Rendering artifacts can arise from geometric misalignment or appearance inconsistency among neighboring Gaussians, making independent primitive-wise updates insufficient. To aggregate such local 3D context, we adopt a KNN Transformer as the fusion backbone. At each recurrent stage t, we construct the neighborhood $\mathcal{N}_k^t(i)$ by querying the k nearest neighbors of the current Gaussian center $\boldsymbol{\mu}_i^t$ in 3D Euclidean space. The backbone performs neither downsampling nor hierarchical pooling, thereby maintaining a direct correspondence between its input and output Gaussians. Specially, we first map the Gaussian input feature $\mathbf{z}_i^t$ into the Transformer feature space using a learnable linear projection, obtaining $\mathbf{x}_i^{0,t}$. The backbone then applies L local Transformer blocks with the same architecture. For $l = 0, \ldots, L-1$, Gaussian i attends only to its KNN neighbors:

$$\mathbf{x}_i^{l+1,t} = \mathcal{T}_{\text{KNN}}\left(\mathbf{x}_i^{l,t}, \{\mathbf{x}_j^{l,t}\}_{j \in \mathcal{N}_k^t(i)}\right). \tag{8}$$

Here, $\mathcal{T}_{\text{KNN}}$ denotes the common pre-norm block structure, comprising local attention, residual connections, and an MLP. After L blocks, the fused feature $\mathbf{h}_i^t = \mathbf{x}_i^{L,t}$ aggregates the current Gaussian state with geometry, appearance, error, and entropy cues from its local neighborhood. The resulting feature is subsequently used by the residual-update and primitives pruning.

Residual Gaussian Update: Given the fused feature $\mathbf{h}_i^t$, we predict residual updates to the current Gaussian attributes rather than regressing their absolute values. This residual formulation allows the refiner to incrementally correct the geometry and appearance errors remaining in the current Gaussian representation. At stage t, the residual decoder combines the fused neighborhood context $\mathbf{h}_i^t$ with the stage input feature $\mathbf{z}_i^t$:

$$\Delta \mathbf{g}_i^t = \mathcal{D}_{\text{res}}\left([\mathbf{h}_i^t, \mathbf{z}_i^t]\right), \qquad \mathbf{g}_i^{t+1} = \mathbf{g}_i^t \oplus \Delta \mathbf{g}_i^t. \tag{9}$$

Here, $\mathcal{D}_{\text{res}}$ denotes the residual decoder and $\oplus$ denotes parameter-specific residual composition; applying the resulting updates to all Gaussians yields $\mathcal{G}^{t+1} = \{\mathbf{g}_i^{t+1}\}_{i=1}^G$. The fusion backbone and

residual decoder are shared across recurrent stages. After each update, the resulting Gaussian representation is rendered again to construct the next-stage residual observation. This recurrent feedback enables iterative feed-forward Gaussian refinement without per-scene gradient optimization.

Primitive Pruning: Pixel-aligned feed-forward 3DGS densely instantiates Gaussians from input pixels, often introducing substantial redundancy. Rather than reducing redundancy through image-space primitive sampling Zhang et al. (2026b), we directly operate on the refined 3D Gaussians based on their spatial context. Specifically, at the first recurrent stage ($t = 0$), the keep-score head predicts a keep score s_i for each primitive:

$$s_i = \mathcal{D}_{\text{keep}}\left([\mathbf{h}_i^0, \mathbf{z}_i^0]\right). \tag{10}$$

The local color entropy explicitly characterizes spatial information richness, helping the keep-score head identify redundant Gaussians in regions with limited appearance variation. Given a predefined pruning ratio ρ, we prune the lowest-scoring $\lfloor \rho G \rfloor$ Gaussians by setting their opacities to zero, using a straight-through estimator (STE) for gradient propagation during training. To reduce information loss, we merge these low-scoring Gaussians into retained representatives before removal, yielding a compact representation for subsequent refinement. More details are provided in the supplementary.

3.4 Loss Function

We train the model with the overall objective $\mathcal{L} = \mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{prune}}$.

Photometric Loss: We supervise the recurrent refiner by rendering the current Gaussian representation at each recurrent stage. Notably, the diffusion-restored pseudo targets are used to construct Gaussian-level residual observations rather than as supervision. For each rendered view, the rendering loss ℓ_{render} combines an L_2 loss and LPIPS, with LPIPS evaluated at $0.5\times$ spatial resolution. We supervise all T recurrent stages with progressively larger weights toward later stages:

$$\mathcal{L}_{\text{photo}} = \sum_{t=0}^{T-1} \gamma^{T-1-t} \sum_{v=1}^{V} \ell_{\text{render}}\left(\hat{\mathbf{I}}_v^{t+1}, \mathbf{I}_v\right), \tag{11}$$

where $\mathbf{I}_v$ denotes the corresponding ground-truth image during training, and $\gamma = 0.9$ places greater emphasis on later refinement stages.

Pruning Loss: The pruning loss encourages the global keep-score ranking to reflect the rendering contribution of each Gaussian. To guide this ranking, we derive a deletion prior p_i from the accumulated rendering contribution of each Gaussian. As a result, Gaussians with lower rendering contributions receive larger p_i, indicating a stronger tendency to be pruned. The pruning loss is defined as

$$\mathcal{L}_{\text{prune}} = \frac{1}{G} \sum_{i=1}^{G} \left(\lambda_{\text{p}} p_i s_i + \lambda_{\text{a}} s_i^2\right). \tag{12}$$

The first term suppresses the keep scores of low-contribution Gaussians, encouraging them to rank lower under global pruning, while the second term regularizes the score magnitude to stabilize training.

4 Experiment

4.1 Experimental Setup

Datasets and Metrics: We train FinerSplat on 1000 scenes from the DL3DV-10K dataset Ling et al. (2024) and evaluate it on the official DL3DV benchmark split, which contains diverse real-world scenes. For driving scenarios, we train FinerSplat on the Waymo Open Dataset Sun et al. (2020) and evaluate it under both view interpolation and extrapolation settings. We use the same experimental protocol for both datasets: four images are used to initialize the scene representation, four additional views are used for diffusion-based restoration, and the remaining images are reserved for evaluation. On DL3DV and the interpolation setting of Waymo, we evaluate novel view

Model	Category	RGB			DM Type	NumGS	ReconTime
		PSNR↑	SSIM↑	LPIPS↓			
DIFIX3D+	Opt.	17.20	0.405	0.461	Image	409K	8m10s
Depthsplat + DIFIX3D+	Opt.	21.24	0.623	0.201	Image	458K	3m26s
MVSplat-360	feed-forward	16.74	0.438	0.410	Video	-	5min21s
AnySplat	feed-forward	15.62	0.385	0.448	N/A	417K	0.62s
ReSplat	feed-forward	20.12	0.598	0.359	N/A	28K	0.67s
Ours	feed-forward	21.49	0.667	0.219	Image	321K	0.61s

Table 1: Quantitative comparison on DL3DV. All methods are trained on the DL3DV training sequences and evaluated on the same test set.



Figure 3: Qualitative comparison on the DL3DV dataset.

synthesis using PSNR, SSIM, and LPIPS. For extrapolated views on Waymo, where paired ground-truth images are unavailable, we report FID and KID. We additionally report the number of retained Gaussians and end-to-end reconstruction time to evaluate model efficiency.

Baselines: On DL3DV, we compare against DIFIX3D+ Wu et al. (2025a) as a per-scene optimization baseline. We additionally evaluate DIFIX3D+ with DepthSplat Xu et al. (2025b) initialization instead of its original SfM initialization. We also compare with MVSplat360 Chen et al. (2024b), AnySplat Jiang et al. (2025), and ReSplat Xu et al. (2025a) as feed-forward baselines. We use their official implementations and released checkpoints. On Waymo, we compare against FSGS Zhu et al. (2023), a representative few-shot 3DGS baseline, as well as StreetCrafter Yan et al. (2024) and FreeVS Wang et al. (2025a), two diffusion-based methods designed for extrapolated-view synthesis in driving scenes.

4.2 Main Results

DL3DV Datasets: Tab. 1 and Fig. 3 present quantitative and qualitative comparisons on the DL3DV benchmark split, respectively. FinerSplat achieves 21.49 dB PSNR, exceeding AnySplat and ReSplat by 5.87 and 1.37 dB, while requiring only 0.61 s for reconstruction. These results suggest that diffusion priors can improve reconstruction in under-constrained regions substantially. As shown in Fig. 3, FinerSplat better preserves fine vegetation details and thin structures, whereas the baselines produce blurry renderings in these regions. Notably, ReSplat uses a fixed $16\times$ subsampling in image space during initialization, but its LPIPS drops at 0.359. In contrast, FinerSplat prunes 30% of its initialized Gaussians and achieves an LPIPS of 0.219 with 321K Gaussians.

We also report diffusion-based baselines in Tab. 1. MVSplat360 uses video diffusion to refine rendered views, but its outputs show noticeable color shifts relative to the observed images as shown in

Method	Interpolation			Extrapolation @ 3 m	
	PSNR↑	SSIM↑	LPIPS↓	FID↓	KID↓
FreeVS	18.09	0.654	0.398	143.35	0.098
StreetCrafter	21.13	**0.668**	0.236	105.11	0.064
FSGS	21.10	0.616	0.322	148.90	0.132
Ours	**21.17**	0.655	**0.227**	**92.77**	**0.050**

Table 2: Quantitative comparison on the Waymo dataset. All metrics are averaged over five novel scenes.

Method	PSNR↑	SSIM↑	LPIPS↓
w/o Error Feedback	21.67	0.677	0.222
w/o Entropy	22.48	0.694	0.226
RGB Error Only	22.81	0.693	0.216
Context-View Feedback	22.65	0.698	0.214
Full Model	23.12	0.717	0.202

Table 3: Ablation study of different components.



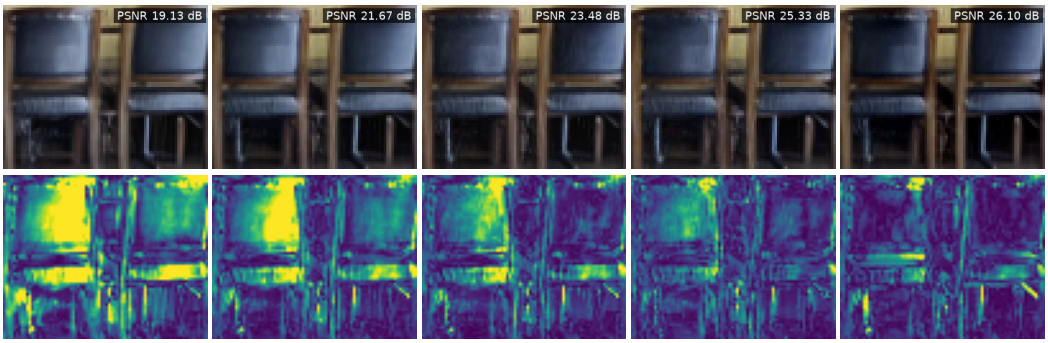
Figure 4: Qualitative comparison on Waymo under the 3 m lane-shift extrapolation setting.

Fig. 3. DIFIX3D+ only achieves 17.20 dB PSNR, as its SfM initialization becomes unreliable under sparse inputs. Replacing SfM with DepthSplat improves PSNR to 21.24 dB, but alternating image restoration with per-scene Gaussian optimization still requires expensive optimization (3 min 26 s). In contrast, FinerSplat learns across scenes to translate diffusion cues into Gaussian updates, reaching better PSNR and SSIM without per-scene optimization.

Waymo Dataset: We further evaluate FinerSplat on street scenes under both view interpolation and extrapolation settings, as reported in Table 2. For interpolated views, FinerSplat achieves the best PSNR (21.17 dB) and LPIPS (0.227), with a comparable SSIM. Under the challenging 3 m lane-shift extrapolation setting, FinerSplat achieves the best FID and KID scores of 92.77 and 0.050. As shown in Fig. 4, FreeVS and StreetCrafter rely on video diffusion models without explicit 3D representations, resulting in vehicle deformation and appearance shifts. Although FSGS uses Metric depth Hu et al. (2024) as supervision to regularize the scene geometry, it achieves comparable performance to FinerSplat on interpolated views but produces noticeable ghosting under extrapolation. In contrast, FinerSplat preserves the clear lane markings and achieves higher fidelity in both interpolated and extrapolated views.

4.3 Ablation Study and Analysis

Ablation of Method Components: For efficient ablation, we evaluate all variants on the same subset of DL3DV, as shown in Tab. 3. We first remove all error feedback and directly predict Gaussian residuals from the initial Gaussian attributes using the KNN-Transformer. PSNR drops from 23.12 to 21.67 dB, indicating that Gaussian attributes alone provide limited guidance for correcting reconstruction errors. Removing local color entropy increases LPIPS from 0.202 to 0.226, suggesting that the entropy cue helps distinguish informative regions from redundant primitives. We then remove feature error while retaining RGB error. This 0.31 dB drop in PSNR indicates that feature residuals provide complementary cues beyond pixel-level RGB differences. Finally, restrict-



(a) w/o Error Feedback (b) w/o Entropy (c) RGB Error Only (d) Context-View Feedback (e) Full Model

Figure 5: Qualitative Component Ablation on DL3DV. We visualize RGB crops (top row) and the corresponding error maps (bottom row) for different ablation variants.

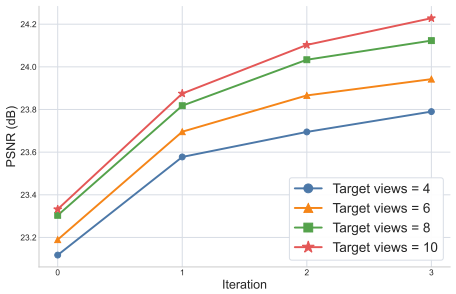


Figure 6: Effect of target views and refinement iterations on PSNR.

ρ	NumGS	PSNR↑	SSIM↑	LPIPS↓	ΔPSNR↑
0%	458K	22.70	**0.717**	**0.242**	–
10%	412K	**22.71**	0.716	0.242	+0.01
20%	367K	22.70	0.716	0.245	+0.00
30%	321K	**22.71**	**0.717**	0.243	**+0.01**
40%	275K	22.65	0.714	0.249	-0.05
50%	229K	22.49	0.708	0.258	-0.21

Table 4: Effect of Gaussian pruning. NumGS is reported in thousands (K), and ΔPSNR is relative to $\rho = 0\%$.

ing error feedback to context views reduces PSNR to 22.65 dB, demonstrating that feedback from diffusion-restored target views provides more effective guidance for under-constrained regions than context-view errors.

Analysis of Iterative Refinement: We further analyze the effects of recurrent refinement and the number of target views. As shown in Fig. 6, increasing the number of recurrent steps consistently improves PSNR across all view settings, with the largest gain from $t = 0$ to $t = 1$. The improvement becomes smaller at later stages, indicating that early feedback corrects the dominant Gaussian errors. Increasing the number of target views further improves PSNR at every stage. At $t = 3$, using 10 target views achieves 24.23 dB, compared with 23.79 dB using 4 target views, as additional views provide richer refinement cues for under-constrained regions. In our experiments, we use 4 target views and set $T = 2$ by default to balance rendering quality and efficiency.

Analysis of Gaussian Pruning Ratio: We investigate the effect of the Gaussian pruning ratio ρ in Tab. 4. We train FinerSplat with $\rho = 30\%$ and vary ρ from 0% to 50% during inference. Increasing ρ from 0% to 30% reduces the Gaussian count from 458K to 321K while maintaining PSNR (22.70 vs. 22.71 dB). Performance remains stable up to $\rho = 30\%$, indicating that the learned keep scores provide a reliable ranking across pruning ratios. Further increasing ρ removes more Gaussians but degrades rendering quality.

5 Conclusion

We presented FinerSplat, a feed-forward framework for refining and compacting 3D Gaussian representations initialized from sparse views. FinerSplat lifts diffusion priors into Gaussian-level feedback and employs an entropy-aware primitive-level Transformer to recurrently update Gaussian attributes and remove redundant primitives. Experiments on DL3DV and Waymo demonstrate improved rendering quality and compact Gaussian representations without per-scene optimization. Current limitations include the inability to recover primitives absent from the initialization and potential hallucinations inherited from DIFIX restorations. Future work will explore adaptive primitive generation and hallucination-aware diffusion guidance to address these limitations.

AI USE STATEMENT

In this work, we used ChatGPT to improve the readability and English writing of the manuscript, and to assist in writing and editing a small portion of the code used for visualization. We did not use generative AI tools for tasks requiring mandatory disclosure under the ICLR AI policy. All AI-assisted text and code were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023. 1

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19457–19467, 2024. 1, 2

Tianyu Chen, Wei Xiang, Kang Han, Yu Lu, Di Wu, Gaowen Liu, and Ramana Rao Kompella. Gifsplat: Generative prior-guided iterative feed-forward 3d gaussian splatting from sparse views. *arXiv preprint arXiv:2602.22571*, 2026. 2, 3

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*, 2024a. 1, 2

Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. In *Advances in Neural Information Processing Systems*, pp. 107064–107086, 2024b. 2, 3, 4, 7

Yun Chen, Jingkang Wang, Ze Yang, Sivabalan Manivasagam, and Raquel Urtasun. G3r: Gradient guided generalizable reconstruction. In *European Conference on Computer Vision*, pp. 305–323. Springer, 2025. 2, 3

Riccardo de Lutio, Tobias Fischer, Yen-Yu Chang, Yuxuan Zhang, Jay Zhangjie Wu, Xuanchi Ren, Tianchang Shen, Katarina Tothova, Zan Gojcic, and Haithem Turki. Artifixer: Enhancing and extending 3d reconstruction with auto-regressive diffusion models. *arXiv preprint arXiv:2603.00492*, 2026. 2

Xiang Feng, Xiangbo Wang, Tieshi Zhong, Chengkai Wang, Yiting Zhao, Tianxiang Xu, Zhenzhong Kuang, Feiwei Qin, Xuefei Yin, and Yanming Zhu. Sr3r: Rethinking super-resolution 3d reconstruction with feed-forward gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. 2, 3

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. 4

Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10579–10596, 2024. 8

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics*, 44(6):1–16, 2025. 2, 3, 7

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 4

Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. 2

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024. 6

Tianqi Liu, Guangcong Wang, Shoukang Hu, Liao Shen, Xinyi Ye, Yuhang Zang, Zhiguo Cao, Wei Li, and Ziwei Liu. Mvsgaussian: Fast generalizable gaussian splatting reconstruction from multi-view stereo. In *European Conference on Computer Vision*, pp. 37–53. Springer, 2025. 2

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*, pp. 133305–133327, 2024. 3

Sheng Miao, Jiaxin Huang, Dongfeng Bai, Weichao Qiu, Bingbing Liu, Andreas Geiger, and Yiyi Liao. Efficient depth-guided urban view synthesis. *arXiv preprint arXiv:2407.12395*, 2024. 1

Sheng Miao, Jiaxin Huang, Dongfeng Bai, Xu Yan, Hongyu Zhou, Yue Wang, Bingbing Liu, Andreas Geiger, and Yiyi Liao. Evolsplat: Efficient volume-based gaussian splatting for urban view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11286–11296, 2025. 2

Sheng Miao, Sijin Li, Pan Wang, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Evolsplat4d: Efficient volume-based gaussian splatting for 4d urban scene synthesis. *arXiv preprint arXiv:2601.15951*, 2026. 1

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. *Communications of the ACM*, 65(1):99–106, 2021. 1

Arthur Moreau, Richard Shaw, Michal Nazarczuk, Jisu Shin, Thomas Tanay, Zhensong Zhang, Songcen Xu, and Eduardo Pérez-Pellitero. Off the grid: Detection of primitives for feed-forward 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. 2

Seungtae Nam, Xiangyu Sun, Gyeongjin Kang, Younggeun Lee, Seungjun Oh, and Eunbyung Park. Generative densification: Learning to densify gaussians for high-fidelity generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26683–26693, 2025. 2, 3

Xuanchi Ren, Jiahui Huang, Xianfang Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Scube: Instant large-scale scene reconstruction using voxsplats. In *Advances in Neural Information Processing Systems*, 2024. 2

Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024. 2

Qi Song, Chenghong Li, Haotong Lin, Sida Peng, and Rui Huang. Adgaussian: Generalizable gaussian splatting for autonomous driving via multi-modal joint learning. *arXiv preprint arXiv:2504.00437*, 2025. 2

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020. 6

Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10208–10217, 2024. 1

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pp. 1–18. Springer, 2024. 1

Qitai Wang, Lue Fan, Yuqi Wang, Yuntao Chen, and Zhaoxiang Zhang. Freevs: Generative view synthesis on free driving trajectory. In *International Conference on Learning Representations*, 2025a. 3, 7

Weijie Wang, Yeqing Chen, Zeyu Zhang, Hengyu Liu, Haoxiao Wang, Zhiyuan Feng, Wenkang Qin, Zheng Zhu, Donny Y. Chen, and Bohan Zhuang. Volsplat: Rethinking feed-forward 3d gaussian splatting with voxel-aligned prediction. *arXiv preprint arXiv:2509.19297*, 2025b. 2

Yiming Wang, Lucy Chai, Xuan Luo, Michael Niemeyer, Manuel Lagunas, Stephen Lombardi, Siyu Tang, and Tiancheng Sun. Learning efficient fuse-and-refine for feed-forward 3d gaussian splatting. In *Advances in Neural Information Processing Systems*, 2025c. 3

YuAn Wang, Xiaofan Li, Chi Huang, Wenhao Zhang, Hao Li, Bosheng Wang, Xun Sun, and Jun Wang. Faithfusion: Harmonizing reconstruction and generation via pixel-wise information gain. *arXiv preprint arXiv:2511.21113*, 2025d. 3

Yijia Weng, Zhicheng Wang, Songyou Peng, Saining Xie, Howard Zhou, and Leonidas J. Guibas. Gaussianlens: Localized high-resolution reconstruction via on-demand gaussian densification. *arXiv preprint arXiv:2509.25603*, 2025. 2, 3

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26024–26035, 2025a. 3, 4, 7

Sibo Wu, Congrong Xu, Binbin Huang, Andreas Geiger, and Anpei Chen. Genfusion: Closing the loop between reconstruction and generation via videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6078–6088, 2025b. 3

Haofei Xu, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Resplat: Learning recurrent gaussian splatting. *arXiv preprint arXiv:2510.08575*, 2025a. 3, 4, 7

Haofei Xu, Chuanxia Zheng, Lingzhi Li, Tat-Jen Cham, Jianfei Cai, Hamid Rezatofighi, and Andreas Geiger. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16453–16463, 2025b. 2, 4, 7

Yunzhi Yan, Zhen Xu, Haotong Lin, Haian Jin, Haoyu Guo, Yida Wang, Kun Zhan, Xianpeng Lang, Hujun Bao, Xiaowei Zhou, and Sida Peng. Streetcrafter: Street view synthesis with controllable video diffusion models. *arXiv preprint arXiv:2412.13188*, 2024. 3, 7

Jiawei Yang, Jiahui Huang, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Apoorva Sharma, Maximilian Igl, Peter Karkus, Danfei Xu, Boris Ivanovic, Yue Wang, and Marco Pavone. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. *arXiv preprint arXiv:2501.00602*, 2025. 2

Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. Yonosplat: You only need one model for feedforward 3d gaussian splatting. *arXiv preprint arXiv:2511.07321*, 2025. 2

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. *arXiv preprint arXiv:2508.09667*, 2025. 3

Meng You, Zhiyu Zhu, Hui Liu, and Junhui Hou. Nvs-solver: Video diffusion model as zero-shot novel view synthesizer. *arXiv preprint arXiv:2405.15364*, 2024. 3

Xiaoxue Zhang, Xiaoxu Zheng, Yixuan Yin, Tiao Zhao, Kaihua Tang, Michael Bi Mi, Zhan Xu, and Dave Zhenyu Chen. Anchorsplat: Feed-forward 3d gaussian splatting with 3d geometric priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18924–18933, 2026a. 2, 3, 4

Zicheng Zhang, Xiangting Meng, Ke Wu, and Wenchao Ding. Sparsesplat: Towards applicable feed-forward 3d gaussian splatting with pixel-unaligned prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b. 2, 6

Hongyu Zhou, Zisen Shao, Sheng Miao, Pan Wang, Dongfeng Bai, Bingbing Liu, and Yiyi Liao. Freefix: Boosting 3d gaussian splatting via fine-tuning-free diffusion models. *arXiv preprint arXiv:2601.20857*, 2026. 3

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00451*, 2023. 7