# PIXGR: PIXEL-SPACE GENERATIVE DECODING FOR FAST FEED-FORWARD 3D RECONSTRUCTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We present PixGR, an efficient framework that integrates lightweight pixel-space generative modeling directly into feed-forward 3D Gaussian reconstruction. Rather than conditioning generation on an explicit and imperfect reconstruction, PixGR encodes the context observations into a 3D-aware scene latent and generatively reconstructs a global 3D Gaussian representation that jointly supports context and target views. We formulate this process with pixel-space flow matching, enabling generative supervision of 3D reconstruction through differentiable rendering without relying on compressed VAE latents. To retain feed-forward efficiency, PixGR encodes the input scene only once and performs iterative inference with a lightweight DiT decoder. Experiments demonstrate improved reconstruction over regression-based approaches in challenging sparse-view settings, particularly with widely spaced observations and view extrapolation, while maintaining sub-second inference and remaining competitive with substantially heavier pre-trained generative models. Code will be publicly available.

## 1 INTRODUCTION

Reconstructing 3D scenes from a few images is a long-standing and fundamental problem in computer vision. Recent feed-forward generalizable methods have made remarkable progress by directly regressing 3D scene representations from input observations (Charatan et al., 2024; Chen et al., 2024a; Huang & Mikolajczyk, 2025a; Xu et al., 2025; Ye et al., 2025b;a), as shown in Fig. 1a. These regression-based methods offer efficient single-pass inference and achieve impressive reconstruction quality in well-observed regions. However, under sparse-view observations, the underlying geometry and appearance of a 3D scene are weakly constrained, admitting multiple possible 3D explanations and introducing substantial reconstruction uncertainty. A deterministic regressor cannot explicitly model such ambiguity and instead commits to an averaged estimate, often resulting in blurry reconstructions or failures in occluded and unseen regions.

To address the high ambiguity of sparse-view reconstruction, generative modeling is introduced due to its ability to model a distribution over plausible scene explanations. Previous optimization-based approaches progressively densify sparse observations with generated novel views and distill them back into 3D (Wu et al., 2024; Sun et al., 2025; Wu et al., 2025; Yu et al., 2025b). To avoid their costly per-scene iterative optimization, feed-forward generative methods mainly follow two paradigms: (i) *generation then reconstruction* (Fig. 1b): it first generates novel views and subsequently uses a feed-forward network to regress a 3D Gaussian splatting (3DGS) representation (Kerbl et al., 2023) from them (Szymanowicz et al., 2025); (ii) *reconstruction then generation* (Fig. 1c): it first reconstructs 3DGS and then renders them at target viewpoints to condition a generative model for target-view refinement or synthesis (Chen et al., 2024b; Wewer et al., 2024; Stracquadanio et al., 2026).

However, these feed-forward generative approaches still face several limitations. First, they decouple generation and reconstruction, preventing the two stages from being jointly optimized within a unified 3D representation. For *generation then reconstruction*, inconsistencies across generated views can introduce artifacts into the subsequent 3D reconstruction. For *reconstruction then generation*, the errors in the initial reconstruction are propagated into the generative stage, while the refined target views cannot directly revise the feed-forward 3D reconstruction. Moreover, compared with regression-based methods, these approaches may lose fine-grained geometric and appearance



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.

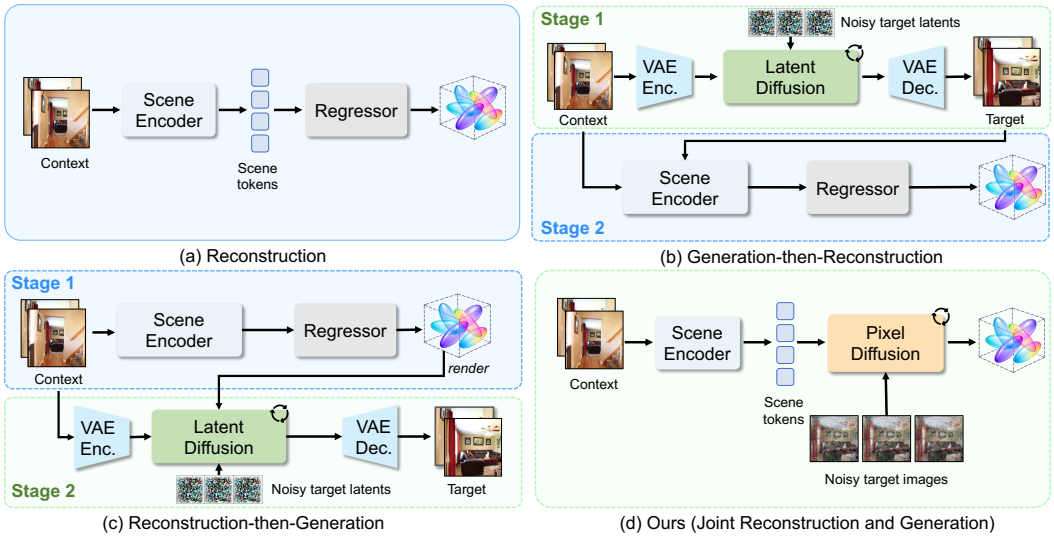



Figure 1: Comparison of feed-forward reconstruction and generative pipelines. (a) Deterministic reconstruction cannot model scene uncertainty. Generative methods address this through (b) generation then reconstruction or (c) reconstruction then generation, but introduce error propagation, information loss, and higher cost. (d) PixG3R instead integrates lightweight pixel-space generation directly into 3D reconstruction, predicting a global 3D Gaussian representation at each sampling step for cross-view-consistent reconstruction.

fidelity due to generation in a compressed VAE latent space (Rombach et al., 2022a), while substantially sacrificing inference efficiency because of their reliance on heavyweight generative models (Rombach et al., 2022a; Blattmann et al., 2023; Wan et al., 2025).

To address these limitations, we introduce PixGR, an end-to-end trainable framework that integrates lightweight pixel-space generative modeling directly into feed-forward 3D Gaussian reconstruction. PixGR consists of a deterministic scene encoder and a generative decoder. Shown in Fig. 1d, instead of encoding context observations into an explicit 3D reconstruction that may already contain errors, the scene encoder produces a 3D-aware latent, deferring uncertain 3D decisions to generative decoding. Conditioned on this latent and noisy target views, the generative decoder predicts at each sampling step a global 3D Gaussian representation that supports both the observed context views and evolving target views, directly shaping a cross-view-consistent 3D scene. To avoid the representation bottleneck of latent-space generation, PixGR performs flow matching directly in pixel space (Li & He, 2025), where renderings of the predicted 3D Gaussians serve as denoised target predictions, allowing the generative objective to directly supervise 3D reconstruction through differentiable rendering. For efficient generative inference, PixGR adopts a lightweight DiT (Peebles & Xie, 2023) decoder, while the input scene is encoded only once and cached throughout sampling.

Experiments show that this lightweight probabilistic formulation substantially improves reconstruction quality under high uncertainty while retaining sub-second inference. PixGR outperforms regression-based approaches in challenging settings, including widely spaced input views and view extrapolation, while remaining competitive with methods that rely on much heavier pretrained generative models for extrapolated views. Our main contributions are summarized as follows:

- We introduce PixGR, a highly efficient and end-to-end trainable framework that formulates generalizable feed-forward 3D reconstruction in a probabilistic manner by integrating lightweight generative modeling directly into 3D Gaussian reconstruction.
- We develop pixel-space flow formulation conditioned on a 3D-aware scene latent, avoiding compressed latent-space bottlenecks and enabling direct supervision of 3D reconstruction through differentiable rendering.
- Extensive experiments show that PixGR consistently improves over regression-based reconstruction in challenging sparse-view settings, while remaining competitive with much heavier generative approaches and achieving sub-second inference.

# 2 Related Work

## 2.1 Feed-forward 3D Regression

Recent advancements in feed-forward 3D regression aim to directly regress 3D representations in a single forward pass (Fig. 1a) (Charatan et al., 2024; Chen et al., 2024a; Wang et al., 2024; Huang & Mikolajczyk, 2025a; Wang et al., 2025; Xu et al., 2025; Ye et al., 2025b;a), significantly improving efficiency compared to per-scene optimization pipelines (Schonberger & Frahm, 2016; Kerbl et al., 2023). Among these approaches, methods based on 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) are particularly relevant to our work due to their strong rendering efficiency and ability to produce high-quality novel view synthesis. Within this paradigm, prior work explores a variety of strategies, including incorporating geometric inductive biases (Charatan et al., 2024; Chen et al., 2024a), leveraging pretrained depth priors (Xu et al., 2025), enabling pose-free inputs (Huang & Mikolajczyk, 2025a;b; Ye et al., 2025b;a), and adopting self-supervised training schemes (Zhao et al., 2025; Huang et al., 2026). Despite these advances, feed-forward regression-based methods lack the ability to model uncertainty, often producing blurry reconstructions or failures in occluded and unseen regions.

## 2.2 3D Generation

A line of work learns generative models directly in native 3D representations, including point clouds (Nichol et al., 2022), voxel grids (Müller et al., 2023), triplanes (Shue et al., 2023), and 3D Gaussians (Zhang et al., 2024a), or in structured 3D latent spaces (Lin et al., 2025a; Xiang et al., 2025). These methods model 3D distributions directly, but typically rely on large-scale 3D supervision and are mostly developed for bounded object-centric domains.

To exploit 2D generative priors, many methods follow *generation then reconstruction* (Fig. 1b): they first synthesize novel views using image or video diffusion (Chan et al., 2023; Liu et al., 2023a; Gao et al., 2024; Hollein et al., 2024; Long et al., 2024; Sargent et al., 2024; Shi et al., 2024; Wu et al., 2024; He et al., 2025; Sun et al., 2025; Yu et al., 2025b;a; Liu et al., 2026), and then recover 3D through per-scene optimization (Mildenhall et al., 2021; Kerbl et al., 2023) or feed-forward reconstruction (Hong et al., 2024; Tang et al., 2024; Xu et al., 2024; Zhang et al., 2024b; Szymanowicz et al., 2025). Although effective, per-scene optimization is computationally expensive, while feed-forward variants remain sensitive to inconsistencies and artifacts across generated views, which can propagate into the reconstructed geometry. Another paradigm follows *reconstruction then generation* (Fig. 1c): an initial 3D representation is first reconstructed and then rendered to condition a generative model (Chen et al., 2024b; Wewer et al., 2024; Stracquadanio et al., 2026). Such designs generate novel views conditioned on a potentially imperfect 3D reconstruction and cannot directly revise errors in the initial geometry.

More tightly coupled approaches instead embed differentiable rendering into the generative process (Anciukevicius et al., 2023; Xu et al., 2023; Anciukevicius et al., 2024; Cai et al., 2025), but focus mainly on synthetic object-centric scenarios and repeatedly process context observations during generation. In contrast, PixGR targets real-world scene reconstruction and performs pixel-space flow matching with a lightweight decoder conditioned on a cached 3D-aware scene latent, providing consistent scene guidance across sampling steps while enabling efficient iterative 3D reconstruction.

## 2.3 Diffusion and Flow Matching

Diffusion models (Ho et al., 2020; Song et al., 2021a) and flow matching (Albergo & Vanden-Eijnden, 2023; Lipman et al., 2023; Liu et al., 2023b) have emerged as fundamental frameworks in modern generative modeling, both learning probability flows from a simple initial distribution to a target data distribution. While latent-space methods now dominate high-resolution image generation (Rombach et al., 2022b; Peebles & Xie, 2023; Ma et al., 2024), early diffusion models operated in pixel space with computationally expensive U-Net backbones (Ronneberger et al., 2015; Ho et al., 2020; Dhariwal & Nichol, 2021; Nichol & Dhariwal, 2021; Song et al., 2021b). Recent works revisit pixel-space modeling with Vision Transformers (ViT) (Dosovitskiy et al., 2021; Chen et al., 2025a;b; Li & He, 2025; Yu et al., 2025c; Lu et al., 2026); we are particularly inspired by JiT (Li & He, 2025), which predicts denoised images that are hypothesized to lie on a low-dimensional manifold (Zhu, 2005).

# 3 PRELIMINARIES

PixGR builds upon the flow-based paradigm (Albergo & Vanden-Eijnden, 2023; Lipman et al., 2023; Liu et al., 2023b) and uses 3D Gaussians as the scene representation. We therefore briefly review the necessary background on pixel-space flow matching, its multi-view extension, and the 3D Gaussian representation before presenting our full method.

**Pixel-Space Flow Matching with $\boldsymbol{x}$-prediction.** Consider a data distribution $\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})$, a noise distribution $\boldsymbol{\epsilon} \sim p_{\text{noise}}(\boldsymbol{\epsilon})$ (e.g., $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$), and a timestep $t \in [0, 1]$. During training, a noisy sample $\boldsymbol{z}_t$ is interpolated using a linear schedule as: $\boldsymbol{z}_t = t\,\boldsymbol{x} + (1-t)\,\boldsymbol{\epsilon}$, which leads to $\boldsymbol{z}_t \sim p_{\text{data}}$ when $t$=1. A flow velocity $\boldsymbol{v}$ is defined as the time-derivative of $\boldsymbol{z}$, that is $\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{\epsilon} = (\boldsymbol{x} - \boldsymbol{z}_t)/(1-t)$. Following $\boldsymbol{x}$-prediction in JiT (Li & He, 2025), the network directly predicts the denoised image $\boldsymbol{x}_\theta$, with the predicted velocity $\boldsymbol{v}_\theta = (\boldsymbol{x}_\theta - \boldsymbol{z}_t)/(1-t)$. Then the velocity loss can be defined as:

$$\mathcal{L} = \mathbb{E}_{t,\boldsymbol{x},\boldsymbol{\epsilon}} \|\boldsymbol{v}_\theta(\boldsymbol{z}_t, t) - \boldsymbol{v}\|^2 = \mathbb{E}_{t,\boldsymbol{x},\boldsymbol{\epsilon}} \frac{1}{(1-t)^2} \|\boldsymbol{x}_\theta(\boldsymbol{z}_t, t) - \boldsymbol{x}\|^2, \tag{1}$$

which is equivalent to a time-weighted squared $\ell_2$ image reconstruction loss. Given the predicted $\boldsymbol{v}_\theta$, sampling is done by solving an ordinary differential equation (ODE) $\boldsymbol{z}_t$ (Albergo & Vanden-Eijnden, 2023; Lipman et al., 2023; Liu et al., 2023b):

$$d\boldsymbol{z}_t/dt = \boldsymbol{v}_\theta(\boldsymbol{z}_t, t), \tag{2}$$

starting from $\boldsymbol{z}_0 \sim p_{\text{noise}}$ and ending at $t = 1$. In practice, it can be approximately solved using numerical solvers such as the Euler and Heun methods.

**Multi-View Flow Matching with $\boldsymbol{x}$-prediction.** The $\boldsymbol{x}$ distribution in pixel-space flow matching is the single image distribution. We extend it to model *joint* distribution of multi-view images $\mathcal{I} = \{\boldsymbol{I}^i\}_{i=1}^V$, representing $V$ observations of the same 3D scene from viewpoints $\mathcal{P} = \{\boldsymbol{P}^i\}_{i=1}^V$. The forward process diffuses every view independently under a shared noise schedule:

$$\mathcal{I}_t = \left\{ t\,\boldsymbol{I} + (1-t)\,\boldsymbol{\epsilon}_{\boldsymbol{I}} \;\middle|\; \boldsymbol{I} \in \mathcal{I} \right\}, \tag{3}$$

where $\boldsymbol{\epsilon}_{\boldsymbol{I}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is independent Gaussian noise sampled per view.

**3D Gaussian Representation.** We represent a scene using pixel-aligned 3D Gaussian primitives. For $V$ views of resolution $H \times W$, the Gaussian representation is defined as $\boldsymbol{G} \in \mathbb{R}^{V \times H \times W \times K}$, where $K$ is the parameter dimension. Each primitive is parameterized by a center $\boldsymbol{\mu} \in \mathbb{R}^3$, rotation quaternion $\boldsymbol{r} \in \mathbb{R}^4$, scale $\boldsymbol{s} \in \mathbb{R}^3$, opacity $\alpha \in \mathbb{R}$, and spherical harmonics coefficients $\boldsymbol{c} \in \mathbb{R}^k$.

# 4 METHOD

As shown in Fig. 2, PixGR is a fast feed-forward generative reconstruction model that aims to produce a unified 3D Gaussian representation covering both context and target viewpoints. Let $\mathcal{I}_{\text{cxt}} = \{\boldsymbol{I}^i_{\text{cxt}}\}_{i=1}^{V_{\text{cxt}}}$ denote a set of $V_{\text{cxt}}$ clean context images with corresponding camera parameters $\mathcal{P}_{\text{cxt}} = \{\boldsymbol{P}^i_{\text{cxt}}\}_{i=1}^{V_{\text{cxt}}}$, and let $\mathcal{P}_{\text{tgt}} = \{\boldsymbol{P}^j_{\text{tgt}}\}_{j=1}^{V_{\text{tgt}}}$ denote the camera parameters of $V_{\text{tgt}}$ target views.

## 4.1 RECONSTRUCTION-BASED DENOISING

**Training.** Inspired by (Xu et al., 2023), we leverage 3D Gaussian reconstruction and rendering to achieve multi-view image denoising. During training, we construct noisy target images $\mathcal{I}^{(t)}_{\text{tgt}} = \{\boldsymbol{I}^{j,(t)}_{\text{tgt}}\}_{j=1}^{V_{\text{tgt}}}$ by corrupting clean target images $\mathcal{I}_{\text{tgt}} = \{\boldsymbol{I}^j_{\text{tgt}}\}_{j=1}^{V_{\text{tgt}}}$ with Gaussian noise at timestep $t$ following Eq. (3). Our model, denoted as $\mathcal{F}(\cdot)$, outputs pixel-aligned 3D Gaussians $\boldsymbol{G}$ given the clean context images, noisy target images, and their camera parameters:

$$\boldsymbol{G} = \mathcal{F}\Big(\mathcal{I}_{\text{cxt}},\, \mathcal{P}_{\text{cxt}},\, \mathcal{I}^{(t)}_{\text{tgt}},\, \mathcal{P}_{\text{tgt}},\, t\Big), \tag{4}$$

Denoising $\mathcal{I}^{(t)}_{\text{tgt}}$ is performed by rendering $\boldsymbol{G}$ at viewpoints $\mathcal{P}_{\text{tgt}}$, yielding the noise-free prediction $\mathcal{I}^{(1)}_{\text{tgt}}$. This formulation is equivalent to $\boldsymbol{x}$-prediction in JiT, allowing the velocity objective in Eq. 1 to

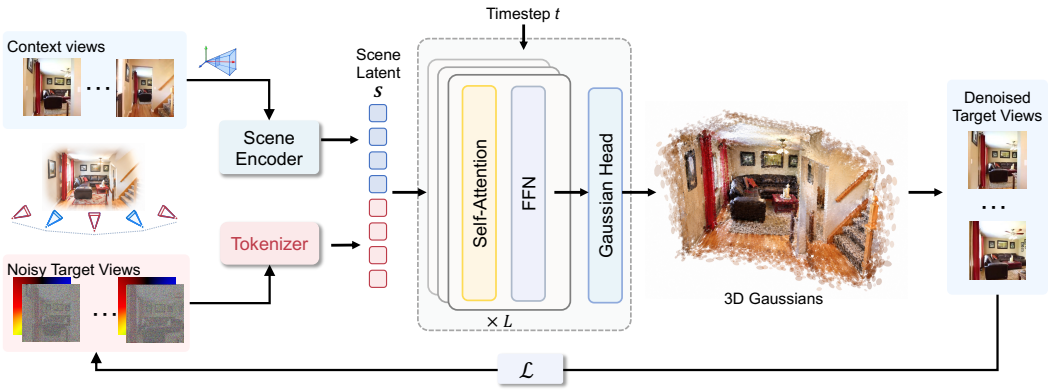



Figure 2: PixGR Overview. Multi-view posed context images are encoded once into a latent representation, conditioned on which a lightweight generative decoder reconstructs pixel-aligned 3D Gaussians given the noisy posed target inputs. The Gaussian decoder is a lightweight DiT-style architecture with global self-attention. The decoder predicts a global 3D Gaussian representation, whose renderings provide denoised target views and supervision through differentiable rendering.

be applied directly to target views. We additionally supervise the model with image reconstruction losses on context and auxiliary novel views to further enforce geometric consistency, and apply perceptual supervision (Zhang et al., 2018) to all views. The following loss is applied to all views:

$$\mathcal{L} = \mathbb{E}_{(\boldsymbol{I},\boldsymbol{P})}\left[\omega(\boldsymbol{I},t)\,\|\mathcal{R}(\boldsymbol{G},\boldsymbol{P})-\boldsymbol{I}\|^2 + \lambda_{\text{lpips}}\,\text{LPIPS}(\mathcal{R}(\boldsymbol{G},\boldsymbol{P}),\boldsymbol{I})\right], \tag{5}$$

where $\omega(\boldsymbol{I},t) = (1-t)^{-2}$ for target views $\boldsymbol{I} \in \mathcal{I}_{\text{tgt}}$ and $\omega(\boldsymbol{I},t) = 1$ otherwise, $\mathcal{R}(\cdot)$ denotes the differentiable renderer, and $\lambda_{\text{lpips}}$ is set to 0.1.This allows PixGR to be supervised by 2D image losses in an end-to-end manner. The training process is summarized in Algorithm 1.

**Inference.** At inference, the target images are initialized from pure Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$, and both the noisy target images and the 3D Gaussians are jointly iterated through the ODE process in Eq. (2). The final Gaussians are returned as the 3D output. The full procedure is detailed in Algorithm 2.

### 4.2 Architecture

Our model $\mathcal{F}$ consists of a deterministic scene encoder $\mathcal{E}$ and a generative Gaussian decoder $\mathcal{D}$:

$$\boldsymbol{S} = \mathcal{E}(\mathcal{I}_{\text{cxt}},\,\mathcal{P}_{\text{cxt}})\,, \quad \boldsymbol{G} = \mathcal{D}\Big(\boldsymbol{S},\,\mathcal{I}_{\text{tgt}}^{(t)},\,\mathcal{P}_{\text{tgt}},\,t\Big)\,, \tag{6}$$

where $\boldsymbol{S} \in \mathbb{R}^{V_{\text{cxt}}\times P\times d}$ denotes the latent scene representation encoded from context views, with $P$ patch tokens per view and feature dimension $d$.

**Scene Encoder.** The scene encoder $\mathcal{E}$ can in principle be instantiated with standard vision transformers (Dosovitskiy et al., 2021), as shown in Sec. 5.2.4. Inspired by (Szymanowicz et al., 2026), we establish and initialize $\mathcal{E}$ as the transformer backbone of VGGT (Wang et al., 2025), which is pre-trained for 3D reconstruction under explicit 3D supervision, to provide a 3D-aware scene latent for target-view denoising.

Following VGGT (Wang et al., 2025), all cameras are expressed relative to the first context image $\boldsymbol{I}_{\text{cxt}}^1$. The camera parameters for each context view are parameterized as: $\boldsymbol{P}_{\text{cxt}}^i = [\mathbf{q}_i,\,\mathbf{t}_i,\,\mathbf{k}_i]$, where $\mathbf{q}_i \in \mathbb{R}^4$ is the rotation quaternion, $\mathbf{t}_i \in \mathbb{R}^3$ is the camera translation, and $\mathbf{k}_i \in \mathbb{R}^2$ denotes the horizontal and vertical fields-of-view. The camera parameters are projected to 1024-dimensional embeddings using a 2-layer MLP and added to VGGT's learnable camera tokens. We concatenate the patch tokens from the final local and global attention layers along the channel dimension, then project them to dimension $d$ with a linear layer followed by LayerNorm (Ba et al., 2016). The resulting scene latent $\boldsymbol{S}$ encodes geometry and appearance information from the context observations and serves as the conditioning signal for the generative decoder. Crucially, $\mathcal{E}$ is independent of the flow timestep $t$ and can therefore be computed once and cached throughout the ODE solve, amortizing the cost of the scene encoder across sampling steps.

**Generative Gaussian Decoder.** The decoder $\mathcal{D}$ maps observed context views and noisy target views at timestep $t$ to pixel-aligned 3D Gaussians. Following (Xu et al., 2023; Zhang et al., 2024b), each camera $\boldsymbol{P}^j$ is encoded as a Plücker ray map. The corresponding image is concatenated channel-wise with its Plücker ray map, resulting in a 9-channel posed image that provides per-pixel camera conditioning. After patchification and linear projection, each target view is represented as a sequence of patch tokens. The full set of target tokens is stacked as $\boldsymbol{T} \in \mathbb{R}^{V_{\text{tgt}} \times P \times d}$.

The scene latent $\boldsymbol{S}$ and target tokens $\boldsymbol{T}$ are concatenated along the view dimension to form a unified token sequence $\boldsymbol{F} \in \mathbb{R}^{V \times P \times d}$, where $V = V_{\text{cxt}} + V_{\text{tgt}}$. This sequence is processed by a series of DiT blocks Peebles & Xie (2023) with AdaLN-Zero conditioning on timestep $t$. Following Yu et al. (2025c), we replace LayerNorm with RMSNorm Zhang & Sennrich (2019) and apply 2D RoPE Su et al. (2024) in all attention layers. Self-attention is performed globally over the entire token sequence $\boldsymbol{F}$, facilitating cross-view information aggregation and enforcing multi-view consistency. The refined tokens $\boldsymbol{F}$ are projected to Gaussian parameters via a linear layer and unpatchified to produce pixel-aligned Gaussians $\boldsymbol{G}$. Specifically, we project the tokens to per-view Gaussian parameters and depth maps, with the depth maps lifted to 3D Gaussian centers using the corresponding context and target camera parameters.

Overall, the architecture conditions lightweight global generative decoding with 3D-aware scene features, and is trained end-to-end with Eq. (5) to produce cross-view-consistent 3D Gaussians while denoising target views across sampling steps.

# 5 Experiments

## 5.1 Experimental Setup

**Implementation Details.** PixGR is implemented in PyTorch and trained on A100 GPUs at $256 \times 256$ resolution, using AdamW with linear warmup and cosine decay. The logit-normal noise schedule is set to $\mu_t = -0.8$, $\sigma_t = 0.8$, biasing training toward higher noise levels. The generative decoder is 12 DiT blocks with hidden dimension 768 and is trained from scratch. At inference, we use an Euler sampler. Further details are provided in Sec. B.

**Metrics.** We employ standard novel view synthesis (NVS) metrics: PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and FID (Heusel et al., 2017), capturing pixel-level accuracy, structural similarity, perceptual quality, and distributional alignment, respectively.

**Datasets.** We train and evaluate on RealEstate10K (RE10K) (Zhou et al., 2018) and outdoor DL3DV (Ling et al., 2024), and further conduct out-of-domain evaluation on the outdoor ACID dataset (Liu et al., 2021) in Sec. C.

**Baselines.** We compare against two categories of methods. (1) Feed-forward regression-based 3DGS methods, including the pose-required MVSplat (Chen et al., 2024a), DepthSplat (Xu et al., 2025), and YoNoSplat (Ye et al., 2025a), as well as the pose-free AnySplat (Jiang et al., 2025). (2) Generative methods, including *generation-then-reconstruction* method Bolt3D (Szymanowicz et al., 2025), and *reconstruction-then-generation* methods MVSplat360 (Chen et al., 2024b) and latentSplat (Wewer et al., 2024), ReconSplat (Stracquadanio et al., 2026). Additional baseline comparisons are provided in Sec. C, while Sec. B summarizes the geometric and generative priors used by each method and reports their parameter counts.

**Target-view selection.** Unless otherwise specified, PixGR takes three target keyframe viewpoints as queries and generates a shared 3D Gaussian representation conditioned on the scene latent. We render the resulting representation at all evaluation viewpoints for metric computation.

## 5.2 Results

### 5.2.1 View Interpolation

Tab. 1 shows view interpolation on the 2-view RE10K and 6-view DL3DV benchmarks, where the novel cameras lie between the context viewpoints. For RE10K, we use the three evaluation viewpoints directly as target keyframes. For DL3DV, we uniformly sample three target keyframes within the range spanned by the context views, and render the resulting 3D representation from all

Table 1: Quantitative comparisons of 3D reconstruction performance on view interpolation. Results are reported on 2-view RE10K and 6-view DL3DV. PixGR consistently outperforms prior methods across datasets. **Best** and second-best results are highlighted.

| Method | RE10K | | | | DL3DV | | | |
|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
| *Regression-based Methods* | | | | | | | | |
| MVSplat | 26.39 | 0.869 | 0.128 | 2.28 | 18.12 | 0.556 | 0.370 | 58.20 |
| DepthSplat | 27.47 | 0.889 | 0.114 | 2.29 | 24.19 | 0.823 | 0.147 | **9.51** |
| AnySplat | 19.80 | 0.682 | 0.274 | 2.88 | 19.03 | 0.554 | 0.235 | 16.94 |
| YoNoSplat | 25.73 | 0.851 | 0.130 | 3.75 | 24.83 | 0.821 | **0.137** | 11.56 |
| *Generative Methods* | | | | | | | | |
| latentSplat | 23.89 | 0.813 | 0.163 | 2.22 | 17.22 | 0.514 | 0.385 | 45.66 |
| MVSplat360 | 26.41 | 0.869 | 0.126 | – | 20.07 | 0.698 | 0.257 | 18.68 |
| ReconSplat | 25.57 | 0.832 | 0.150 | 4.29 | 23.05 | 0.728 | 0.205 | 18.93 |
| **PixGR** | **28.54** | **0.905** | **0.104** | **2.19** | **25.27** | **0.840** | 0.147 | 11.26 |




Figure 3: Qualitative comparison on interpolated views. PixGR produces sharper and more geometrically consistent renderings than prior methods and handles large viewpoint changes effectively.

42 evaluation viewpoints. As shown in Tab. 1, PixGR achieves the best performance on most metrics across both datasets. The qualitative comparisons in Fig. 3 further show that PixGR preserves stable geometry, consistent appearance, and fine details across substantial viewpoint changes, while other methods exhibit blurring in challenging regions or strong generation artifacts.

### 5.2.2 View Extrapolation

We further evaluate 2-view RE10K and 4-view DL3DV on view extrapolation, where novel cameras move beyond the observed range and the model must infer geometry and appearance that are weakly or not at all constrained by the context views. For RE10K, we use its three evaluation poses as targets. For DL3DV, we sample three target poses evenly across the extrapolation range and render the resulting 3DGS at all 42 evaluation poses. As shown in Tab. 2, PixGR achieves large gains over regression-based methods across all metrics, highlighting the difficulty of deterministic regression in weakly observed regions. This is also reflected qualitatively in Fig. 4, where regression baselines exhibit blurring and severe artifacts in unseen or under-constrained areas.

Although previous generative methods improve FID over regression baselines on RE10K, they still struggle to synthesize high-quality extrapolated views. The decoupling between generation and reconstruction can introduce errors and cross-view misalignment, while compressed latent-space generation may lose fine details and cause hue shifts. As shown in Fig. 4, these methods exhibit

Table 2: Quantitative comparisons of 3D reconstruction performance on view extrapolation. Results are reported on 2-view RE10K and 4-view DL3DV. PixGR consistently outperforms prior methods.

| Method | RE10K | | | | DL3DV | | | |
|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
| *Regression-based Methods* | | | | | | | | |
| DepthSplat | 22.02 | 0.822 | 0.189 | 6.95 | 21.59 | 0.745 | 0.205 | 17.41 |
| AnySplat | 16.21 | 0.621 | 0.351 | 5.98 | 14.84 | 0.449 | 0.426 | 22.16 |
| YoNoSplat | 23.00 | 0.810 | 0.182 | 6.47 | 23.04 | 0.761 | 0.192 | 17.21 |
| *Generative Methods* | | | | | | | | |
| latentSplat | 22.62 | 0.777 | 0.196 | 2.79 | 18.35 | 0.564 | 0.345 | 37.65 |
| MVSplat360 | 23.16 | 0.810 | 0.176 | **1.79** | 17.91 | 0.609 | 0.335 | 18.23 |
| Bolt3D | 23.13 | 0.806 | 0.166 | 3.66 | – | – | – | – |
| ReconSplat | 23.98 | 0.795 | 0.184 | 4.49 | 21.23 | 0.673 | 0.249 | 23.02 |
| **PixGR** | **26.03** | **0.872** | **0.141** | 2.75 | **23.76** | **0.802** | **0.178** | **12.78** |

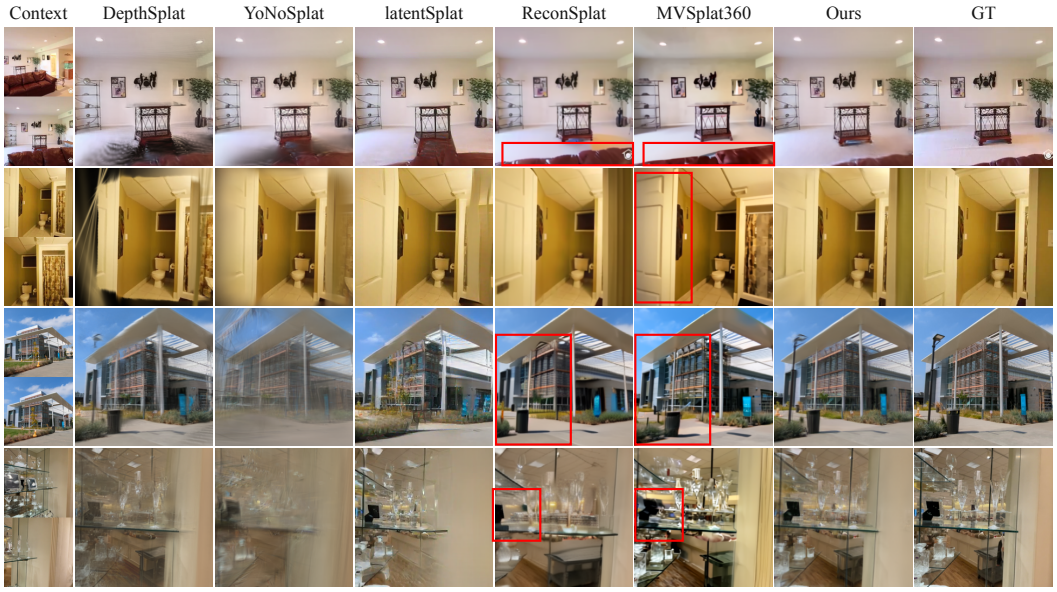



Figure 4: Qualitative comparison on extrapolated novel views. Regression-based methods fail in unseen regions, while generative baselines may produce artifacts and distortions, including geometric misalignment (e.g., the sofa in Row 1) and detail loss (e.g., missing texture on the dustbin and the missing street lamp in Row 2). In contrast, PixGR produces geometrically consistent renderings while preserving fine details.

geometric misalignment and detail loss. In contrast, PixGR performs pixel-space generation directly within the 3D reconstruction process, producing sharper, coherent novel views with finer details.

### 5.2.3 Efficiency Analysis

Fig. 5 compares inference time as the number of evaluation novel views increases on a single A100 GPU. We measure the total time to obtain a 3D Gaussian representation and render or generate the requested novel views. Regression-based methods remain fast but are less effective under ambiguous extrapolation (Sec. 5.2.2), while existing generative methods incur substantially higher cost that grows with the number of requested views. In contrast, PixGR remains below one second from 3 to 48 novel views. This favorable scaling comes from generating a single global 3D Gaussian representation using the context views and

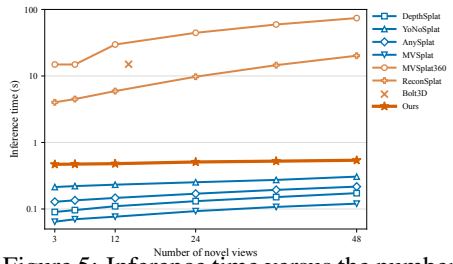



Figure 5: Inference time versus the number of novel views on a single A100 GPU.

Table 3: Ablation of encoder and decoder design on RE10K under interpolation and extrapolation.

| Variant | Interpolation | | | | Extrapolation | | | |
|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
| *Encoder ablation* | | | | | | | | |
| VGGT w/o init. | 27.32 | 0.882 | 0.134 | 2.45 | 24.98 | 0.859 | 0.167 | 3.67 |
| ViT-L encoder | 27.14 | 0.881 | 0.126 | 2.63 | 24.80 | 0.848 | 0.170 | 3.89 |
| *Decoder ablation* | | | | | | | | |
| Regressor | 27.58 | 0.883 | 0.111 | 2.38 | 24.34 | 0.821 | 0.166 | 5.42 |
| **PixGR** | **28.54** | **0.905** | **0.104** | **2.19** | **26.03** | **0.872** | **0.141** | **2.75** |

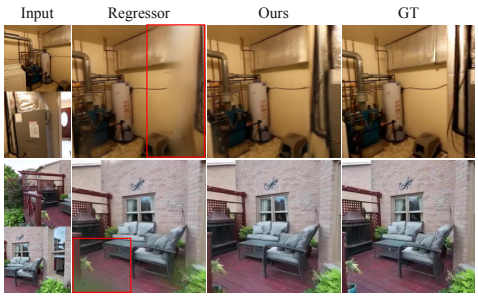



Figure 6: Qualitative comparison between a regressor and DiT as the Gaussian decoder.

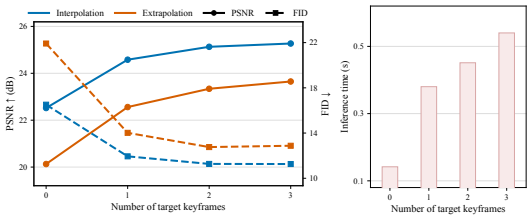



Figure 7: Effect of target keyframes on DL3DV. The left panel reports PSNR and FID, while the right shows inference time as the number of target keyframes varies.

three key target frames, then rendering all evaluation cameras from the shared representation rather than repeating generation for each view. These results show that our paradigm can model uncertainty while maintaining practical feed-forward efficiency.

### 5.2.4 Ablation Study

**Encoder Design.** As shown in Tab. 3, removing VGGT initialization or replacing the VGGT-based encoder with ViT-L degrades all metrics, indicating that the generative decoder benefits from a 3D-aware scene latent. Nevertheless, both variants still outperform most baselines in Tabs. 1 and 2, demonstrating that the decoder remains effective across different encoder choices and that our paradigm does not rely entirely on pretrained 3D priors.

**Generative vs. Deterministic Decoding.** In Tab. 3, replacing the DiT decoder with a 12-block deterministic regressor of comparable model size leads to a consistent drop under interpolation and a substantially larger drop under extrapolation. The widening gap as ambiguity increases highlights the importance of explicitly modeling multiple plausible 3D explanations. Fig. 6 further shows that the generative decoder handles uncertainty better when context views have little or no overlap.

**Keyframe Selection.** We also ablate the number of target keyframes provided to the model. As shown in Fig. 7, using more keyframes consistently improves performance, with larger gains under extrapolation where scene ambiguity is higher. Most gains are achieved with two keyframes, while a third further improves quality with only 0.09 s additional latency. We therefore use three by default.

## 6 Conclusion

We present PixGR, a novel framework for generative 3D reconstruction that combines a 3D-aware scene encoder with a lightweight generative decoder. Despite using a lightweight decoder trained without pretrained generative priors, PixGR achieves strong performance in both interpolation and extrapolation, outperforming regression-based methods and remaining competitive with heavier generative approaches. These results highlight the potential of integrating efficient pixel-space generative modeling directly into 3D reconstruction.

### AI USE STATEMENT

In this work, we have not used generative AI tools for helping develop theoretical models or conceptual frameworks, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, supporting qualitative and thematic data analysis, interpreting results, and generating synthetic data sets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, cleaning and reformatting dataset are not applicable to this work. Additionally, we used generative AI tools for editing a research paper to improve readability. We have reviewed all AI-assisted work. LLM-generated texts were verified for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REFERENCES

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023.

Titas Anciukevicius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12608–12618, 2023.

Titas Anciukevicius, Fabian Manhardt, Federico Tombari, and Paul Henderson. Denoising diffusion via image-based rendering. In *The Twelfth International Conference on Learning Representations*, 2024.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Yuanhao Cai, He Zhang, Kai Zhang, Yixun Liang, Mengwei Ren, Fujun Luan, Qing Liu, Soo Ye Kim, Jianming Zhang, Zhifei Zhang, et al. Baking gaussian splatting into diffusion denoiser for fast and scalable single-stage image-to-3d generation and reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25062–25072, 2025.

Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4217–4229, 2023.

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.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025a.

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. In *European Conference on Computer Vision*, pp. 370–386. Springer, 2024a.

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. *Advances in Neural Information Processing Systems*, 37:107064–107086, 2024b.

Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space. *arXiv preprint arXiv:2511.18822*, 2025b.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for video diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Lukas Hollein, Aljaz Bozic, Norman Muller, David Novotny, Hung-Yu Tseng, Christian Richardt, Michael Zollhofer, and Matthias Niessner. Viewdiff: 3d-consistent image generation with text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5043–5052, 2024.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. In *The Twelfth International Conference on Learning Representations*, 2024.

Ranran Huang and Krystian Mikolajczyk. No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025a.

Ranran Huang and Krystian Mikolajczyk. Spfsplatv2: Efficient self-supervised pose-free 3d gaussian splatting from sparse views. *arXiv preprint arXiv:2509.17246*, 2025b.

Ranran Huang, Weixun Luo, Ye Mao, and Krystian Mikolajczyk. From none to all: Self-supervised 3d reconstruction via novel view synthesis. *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2026.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *arXiv preprint arXiv:2505.23716*, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pp. 71–91. Springer, 2024.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Chenguo Lin, Panwang Pan, Bangbang Yang, Zeming Li, and Yadong Mu. Diffsplat: Repurposing image diffusion models for scalable gaussian splat generation. *arXiv preprint arXiv:2501.16764*, 2025a.

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*, 2025b.

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.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14458–14467, 2021.

Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model. *IEEE Transactions on Image Processing*, 2026.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9298–9309, 2023a.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023b.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9970–9980, 2024.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158*, 2026.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

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.

Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4328–4338, 2023.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022a.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022b.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9420–9429, 2024.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. In *The Twelfth International Conference on Learning Representations*, 2024.

J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20875–20886, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Giuseppe Stracquadanio, Kevin Raj, Julia Grabinski, and Stefan Roth. Reconsplat: Generalizable 3d scene reconstruction beyond observed views. *arXiv preprint arXiv:2608.28895*, 2026.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhu, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with decoupled video diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13695–13706, 2025.

Stanislaw Szymanowicz, Jason Y Zhang, Pratul Srinivasan, Ruiqi Gao, Arthur Brussee, Aleksander Holynski, Ricardo Martin-Brualla, Jonathan T Barron, and Philipp Henzler. Bolt3d: Generating 3d scenes in seconds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24846–24857, 2025.

Stanislaw Szymanowicz, Minghao Chen, Jianyuan Wang, Christian Rupprecht, and Andrea Vedaldi. Lagernvs: Latent geometry for fully neural real-time novel view synthesis. *arXiv preprint arXiv:2603.20176*, 2026.

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.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. $\pi^3$: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *European conference on computer vision*, pp. 456–473. Springer, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26024–26035. IEEE, 2025.

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21551–21561, 2024.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21469–21480, 2025.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 16453–16463, 2025.

Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023.

Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *European Conference on Computer Vision*, pp. 1–20. Springer, 2024.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024.

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*, 2025a.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5916–5926, 2025a.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. *arXiv preprint arXiv:2511.20645*, 2025c.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: A structured and explicit radiance representation for 3d generative modeling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pp. 1–19. Springer, 2024b.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Qitao Zhao, Hao Tan, Qianqian Wang, Sai Bi, Kai Zhang, Kalyan Sunkavalli, Shubham Tulsiani, and Hanwen Jiang. E-rayzer: Self-supervised 3d reconstruction as spatial visual pre-training. *arXiv preprint arXiv:2512.10950*, 2025.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4): 1–12, 2018.

Xiaojin Jerry Zhu. Semi-supervised learning literature survey. *Computer Sciences TR*, 2005.

# A METHOD

Algorithms 1 and 2 summarize the training and inference procedures, respectively.

**Algorithm 1** Training.

**Require:** Training set $\mathcal{D}$; Scene encoder $\mathcal{E}$; Generative Gaussian decoder $\mathcal{D}$; Renderer $\mathcal{R}$; number of training iterations $K$.

1: **for** $i = 1, \dots, K$ **do**
2: Draw a batch from the training set: $(\mathcal{I}_{\text{cxt}}, \mathcal{I}_{\text{tgt}}, \mathcal{I}_{\text{nov}}, \mathcal{P}_{\text{cxt}}, \mathcal{P}_{\text{tgt}}, \mathcal{P}_{\text{nov}}) \sim \mathcal{D}$.
3: Sample a timestep $t = \sigma(\xi)$, $\xi \sim \mathcal{N}(\mu_t,\ \sigma_t^2)$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$.
4: Corrupt clean target images: $\mathcal{I}_{\text{tgt}}^{(t)} = \{\boldsymbol{I}_{\text{tgt}}^{(t)} = t\boldsymbol{I}_{\text{tgt}} + (1-t)\boldsymbol{\epsilon}_v \mid \boldsymbol{I}_{\text{tgt}} \in \mathcal{I}_{\text{tgt}}\}$.
5: Predict pixel-aligned 3D Gaussians: $\boldsymbol{S} \leftarrow \mathcal{E}(\mathcal{I}_{\text{cxt}}, \mathcal{P}_{\text{cxt}}),\ \boldsymbol{G} \leftarrow \mathcal{D}\left(\boldsymbol{S},\ \mathcal{I}_{\text{tgt}}^{(t)},\ \mathcal{P}_{\text{tgt}},\ t\right)$.
6: Render images at all viewpoints: $\widehat{\mathcal{I}} = \{\hat{\boldsymbol{I}} = \mathcal{R}(\boldsymbol{G}, \boldsymbol{P}) \mid \boldsymbol{P} \in \mathcal{P}_{\text{cxt}} \cup \mathcal{P}_{\text{tgt}} \cup \mathcal{P}_{\text{nov}}\}$.
7: Compute main objective $\mathcal{L} = \mathbb{E}_{(\boldsymbol{I}, \hat{\boldsymbol{I}})}\left[\omega(\boldsymbol{I}, t)\left\|\hat{\boldsymbol{I}} - \boldsymbol{I}\right\|^2 + \lambda_{\text{lpips}}\,\text{LPIPS}\left(\hat{\boldsymbol{I}}, \boldsymbol{I}\right)\right]$, where $\omega(\boldsymbol{I}, t) = (1-t)^{-2}$ for $\boldsymbol{I} \in \mathcal{I}_{\text{tgt}}$, and $\omega(\boldsymbol{I}, t) = 1$ for $\boldsymbol{I} \in \mathcal{I}_{\text{cxt}} \cup \mathcal{I}_{\text{nov}}$, and update $\mathcal{E}$, $\mathcal{D}$ via backpropagation.
8: **end for**
9: **return** $e, h$.

**Algorithm 2** Inference with Euler Solver.

**Require:** Context images $\mathcal{I}_{\text{cxt}}$, Context camera parameters $\mathcal{P}_{\text{cxt}}$, Target camera parameters $\mathcal{P}_{\text{tgt}}$, Scene encoder $\mathcal{E}$; Generative Gaussian decoder $\mathcal{D}$; Renderer $\mathcal{R}$; number of denoising steps $N$.

1: Initialize target images: $\mathcal{I}_{\text{tgt}}^{(0)} = \{\boldsymbol{I}_{\text{tgt}}^{j,(0)} \mid \boldsymbol{I}_{\text{tgt}}^{j,(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\}_{j=1}^{V_{\text{tgt}}}$.
2: Encode scene representation: $\boldsymbol{S} \leftarrow \mathcal{E}(\mathcal{I}_{\text{cxt}},\ \mathcal{P}_{\text{cxt}})$.
3: **for** $i = 0, \dots, N-1$ **do**
4: Set timestep: $t_i = i/N,\ t_{i+1} = (i+1)/N$.
5: Decode 3D Gaussians at the current step: $\boldsymbol{G} \leftarrow \mathcal{D}\big(\boldsymbol{S},\ \mathcal{I}_{\text{tgt}}^{(t_i)},\ \mathcal{P}_{\text{tgt}},\ t_i\big)$.
6: Compute velocity using the rendered images: $\{\boldsymbol{v}^j \mid \boldsymbol{v}^j = \big(\mathcal{R}(\boldsymbol{G}, \boldsymbol{P}_{\text{tgt}}^j) - \boldsymbol{I}_{\text{tgt}}^{j,(t_i)}\big)/(1-t_i)\}_{j=1}^{V_{\text{tgt}}}$.
7: Advance the target images: $\mathcal{I}_{\text{tgt}}^{(t_{i+1})} = \{\boldsymbol{I}_{\text{tgt}}^{j,(t_{i+1})} \mid \boldsymbol{I}_{\text{tgt}}^{j,(t_{i+1})} = \boldsymbol{I}_{\text{tgt}}^{j,(t_i)} + \boldsymbol{v}^j(t_{i+1} - t_i)\}$.
8: **end for**
9: **return** $\boldsymbol{G}$.

# B MORE IMPLEMENTATION DETAILS

**Implementation Details.** PixGR is trained for $100{,}000$ steps and an initial learning rate of $1\text{e}{-4}$. During training, target views are sampled from an extrapolated range of the context views, so that the target views may lie outside or inside the context range. All cameras in one scene are expressed relative to the first context view, and the camera poses are normalized to a fixed scale across all scenes. For interpolation, we adopt the evaluation splits from Charatan et al. (2024); Xu et al. (2025) for 2-view RE10K and ACID (Tab. 5), and the 6-view DL3DV benchmark. For extrapolation, we adopt the extrapolated evaluation splits from latentSplat for RE10K, and remove the first and last two views from the 6-view DL3DV benchmark to form a 4-view extrapolation setting. We use three target keyframes for both interpolation and extrapolation unless otherwise specified.

**Architecture.** VGGT operates on images with patch size 14, while our decoder tokenizer processes 256×256 images with patch size 8. To align the token counts between encoder and decoder, we resize the conditioning images to 448× 448, so that VGGT outputs 1024 (32× 32) tokens per view, matching the number of target tokens. The per-view tokens are then concatenated and passed to the decoder to produce per-view pixel-aligned Gaussians. We use DiT with channel dimension 768, 12 heads and 12 transformer blocks.

**Implementation Details of Baselines.** For YoNoSplat, we use its GT-pose- and GT-intrinsics-required setting and evaluate at $256 \times 256$. As Bolt3D is not open-sourced, we compare to it only quantitatively, reporting the numbers from their paper under the same experimental settings.

Table 4: Geometric and generative priors and parameter counts of the compared methods. A dash indicates that no corresponding prior is used or that a separate parameter count is unavailable.

| Method | Prior | | Params (B) | | |
|---|---|---|---|---|---|
| | Geometric | Generative | Encoder | Decoder | Total |
| *Regression-based Methods* | | | | | |
| DepthSplat | DA2 (Yang et al., 2024) | – | 0.347 | – | 0.347 |
| AnySplat | VGGT (Wang et al., 2025) | – | 1.191 | – | 1.191 |
| YoNoSplat | Pi3 (Wang et al., 2026) | – | 0.965 | – | 0.965 |
| DA3 | DA3 (Lin et al., 2025b) | – | 1.356 | – | 1.356 |
| NoPoSplat | MASt3R (Leroy et al., 2024) | – | 0.612 | – | 0.612 |
| NAS3R | VGGT (Wang et al., 2025) | – | 1.223 | – | 1.223 |
| SPFSplat | MASt3R (Leroy et al., 2024) | – | 0.613 | – | 0.613 |
| *Generative Methods* | | | | | |
| Bolt3D | MASt3R (Leroy et al., 2024) | CAT3D (Gao et al., 2024) | – | – | – |
| latentSplat | – | VAE-GAN (Rombach et al., 2022a) | 0.101 | 0.084 | 0.185 |
| MVSplat360 | MVSplat (Chen et al., 2024a) | SVD (Blattmann et al., 2023) | 0.012 | 2.389 | 2.401 |
| ReconSplat | MVSplat (Chen et al., 2024a) | SD (Rombach et al., 2022a) | 0.012 | 1.298 | 1.310 |
| **PixGR(ViT-L)** | – | – | **0.304** | **0.131** | **0.434** |
| **PixGR** | VGGT (Wang et al., 2025) | – | **0.912** | **0.131** | **1.042** |

**Implementation Details of Efficiency Analysis** For a fair comparison, all methods use two context views at a resolution of $256 \times 256$, except AnySplat, for which we use $252 \times 252$ inputs to accommodate its patch size; this minor difference has a negligible effect on computational cost. Since Bolt3D is not publicly available, we report its published runtime for a fixed input of 16 views as a reference.

**Priors and Parameter Counts.** Tab. 4 summarizes the geometric and pretrained generative priors used by the compared methods together with their encoder, decoder, and total parameter counts when available. Here we also include some additional methods such as epth Anything V3 (DA3) (Lin et al., 2025b), NoPoSplat (Ye et al., 2025b), SPFSplat (Huang & Mikolajczyk, 2025a) and NAS3R (Huang et al., 2026) which will be evaluated in Tab. 6.

Regression-based approaches leverage geometric foundation models but do not incorporate generative priors for resolving ambiguity in extrapolated regions. Existing generative approaches instead build on large pretrained image or video generators, whereas PixGR combines a VGGT geometric prior with a lightweight pixel-space generative decoder trained from scratch. We also include the PixGR variant introduced in Sec. 5.2.4, which replaces the VGGT encoder with ViT-L.

## C Additional Quantitative Results

**Out-of-Domain Evaluation on ACID.** As shown in Tab. 5, PixGR consistently outperforms both regression-based and generative baselines on the out-of-domain ACID benchmark, achieving the best results across all four metrics. In particular, it surpasses the strongest competing method, MVSplat, by 1.13 dB in PSNR while also improving SSIM, LPIPS, and FID. These results demonstrate the robustness of PixGR and highlight the effectiveness of pixel-space generative decoding for producing high-fidelity and structurally consistent novel views under previously unseen visual distributions.

Table 5: Out-of-domain novel-view synthesis results on ACID.

| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| *Regression-based Methods* | | | | |
| MVSplat | <u>28.25</u> | <u>0.843</u> | <u>0.144</u> | <u>6.38</u> |
| DepthSplat | 19.65 | 0.628 | 0.298 | 28.36 |
| AnySplat | 21.49 | 0.631 | 0.300 | 8.47 |
| YoNoSplat | 25.72 | 0.768 | 0.185 | 10.14 |
| *Generative Methods* | | | | |
| latentSplat | 25.78 | 0.789 | 0.185 | 6.55 |
| MVSplat360 | 21.91 | 0.686 | 0.296 | 25.24 |
| ReconSplat | 22.53 | 0.656 | 0.286 | 17.28 |
| **PixGR** | **29.38** | **0.866** | **0.138** | **6.29** |

**Comparison with Additional Regression-Based Methods.** We mainly compare with pose-require methods in our main experiments, as they are the most relevant to our setting. We addition-

Table 6: Comparison with additional regression-based methods on 2-view RE10K extrapolation.

| **Method** | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| Depth Anything V3 | 17.91 | 0.574 | 0.339 | 13.28 |
| NoPoSplat | 21.41 | 0.757 | 0.207 | 6.09 |
| NAS3R | 19.92 | 0.782 | 0.205 | 7.78 |
| SPFSplat | 20.22 | 0.775 | 0.210 | 7.49 |
| **PixGR** | **26.03** | **0.872** | **0.141** | **2.75** |

Table 7: One-view reconstruction results on RE10K compared with DiffusionGS (Cai et al., 2025). **Best** results are highlighted. Our method consistently outperforms DiffusionGS across all metrics, demonstrating better performance in reconstructing scenes from a single view.

| **Method** | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| DiffusionGS | 17.09 | 0.560 | 0.420 | 7.88 |
| **PixGR** | **17.65** | **0.600** | **0.384** | **6.15** |




Figure 8: Visualization of one-view reconstruction on RE10K.

ally compare with recent pose-free regression-based reconstruction methods, Depth Anything V3 (DA3) (Lin et al., 2025b), NoPoSplat (Ye et al., 2025b), SPFSplat (Huang & Mikolajczyk, 2025a) and NAS3R (Huang et al., 2026) on the 2-view RE10K extrapolation setting. We adopt NAS3R's variant that initializes from VGGT. As shown in Tab. 6, PixGR substantially outperforms all additional baselines across all metrics, further demonstrating the benefit of generative decoding when novel views lie outside the observed camera range.

**One-View Reconstruction on RE10K.** We further evaluate reconstruction from a single context view, where scene geometry is highly under-constrained. As shown in Tab. 7, PixGR consistently outperforms DiffusionGS (Cai et al., 2025), improving PSNR by $0.56$ dB and reducing FID by $1.73$. The qualitative comparison in Fig. 8 further shows sharper details and more accurate scene structure, while DiffusionGS often produces renderings that deviate from the target viewpoint. These results demonstrate that conditioning pixel-space generative decoding on a 3D-aware scene representation remains effective even in this extremely sparse setting, providing consistent geometric guidance across sampling steps.

## D Limitations and Future Work

While PixGR achieves strong performance in both interpolation and extrapolation, it has several limitations. First, PixGR employs a lightweight generative decoder trained from scratch, which is less expressive than large pretrained generative models. This limits its ability to synthesize high-fidelity novel views in highly ambiguous scenarios, such as long-sequence generation from a single view. Future work could explore incorporating pretrained generative priors while preserving efficient inference. Second, the current 3D Gaussian representation is limited to static scenes and cannot model dynamic changes such as motion or deformation. Extending this paradigm to dynamic scenes with 4D representations is a promising direction for future work.