Geometry-Aware Flow Matching for Sparse-View 3D Gaussian Splatting

Abdullah Azeem[1,2] **Ruisheng Wang**[1,2,†] **Abubakar Siddique**[1,2] **Qingquan Li**[1]
[1]School of Architecture and Urban Planning, Shenzhen University
[2]Intelligent Computing Center, Shenzhen University
{abdullah.azeem, ruiswang, abubakar, liqq}@szu.edu.cn
[†]Corresponding author

Abstract

Generalizable 3D Gaussian Splatting aims to predict renderable Gaussian scenes from sparse images without per-scene optimization at test time. Existing methods learn to produce Gaussian scene parameters, using either feed-forward prediction or iterative diffusion-based generation. Despite these different mechanisms, they do not explicitly supervise how Gaussian parameters should evolve from a source state to a target under sparse-view evidence. This leaves the construction path implicit for a high-dimensional scene whose layout, geometry, visibility, opacity, orientation, and appearance interact through rendering. We propose GRiF, a geometry-aware flow matching framework that learns conditional transport directly in Gaussian parameter space. Starting from a controlled source scene, GRiF learns a view-conditioned velocity field that progressively evolves a hierarchical Gaussian state toward a renderable target. The hierarchy organizes updates from coarse layout to fine attributes, while geometry-aware paths respect the native structure of Gaussian variables, including rotations. Experiments on RealEstate10K and ACID, with zero-shot transfer evaluated on ScanNet, DL3DV, and DTU, show improved in-domain reconstruction and stronger cross-domain performance.

1 Introduction

3D Gaussian Splatting (3DGS) has emerged as a dominant representation for novel-view synthesis, combining explicit anisotropic primitives with high-quality real-time rendering [18]. Classical 3DGS optimizes a dedicated Gaussian scene per capture from calibrated cameras and SfM initialization, making it powerful but tied to scene-specific test-time optimization [18]. Recent work has relaxed these requirements. COLMAP-free and SfM-free methods remove the dependence on pose or point-cloud priors [10, 16], while feed-forward methods learn to predict Gaussian scenes directly from sparse posed or unposed inputs [37, 25, 44, 14, 17]. This progression has pushed 3DGS from a reconstruction procedure toward a learned scene-prediction problem, in which a neural network must infer the full set of Gaussian parameters, including positions, scales, rotations, opacities, and appearance coefficients, from only a handful of views in a single forward pass.

Within this learned scene-prediction paradigm, feed-forward methods have improved sparse-view 3DGS mainly by strengthening the predictor's input evidence. Multi-view aggregation and cross-view alignment improve how sparse images are fused [37, 25], depth-aware and hierarchical designs inject geometric structure [41, 24, 35], and auxiliary regularization, generated-view supervision, and surface priors reduce sparse-view artifacts [31, 4, 20, 13]. These methods improve what the network observes or how its final prediction is regularized, but feed-forward scene prediction is still supervised at the endpoint. The target is a renderable Gaussian scene, while under two-view training the direct image signal is an error for each rendered pixel. Each Gaussian has position, scale, rotation, opacity, and appearance attributes that interact through rendering, making this endpoint signal sparse relative to the structured prediction.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Diffusion-based methods provide another route for sparse-view reconstruction. Some generate or refine image observations for 3DGS reconstruction [20, 30, 48], while others sample Gaussian representations from a learned prior under test-time view guidance [26]. These approaches use iterative denoising or prior-guided sampling to produce image evidence or Gaussian configurations.

Together, feed-forward prediction and diffusion-based generation produce Gaussian scene parameters without explicitly modeling how the scene is constructed from sparse-view evidence. This motivates treating Gaussian prediction as a process in the scene representation itself. One can define intermediate Gaussian states along a source-to-target path and supervise how their parameters move under image and camera evidence, so that every state on the path remains a Gaussian scene state. The path keeps the Gaussian primitive itself unchanged and stays compatible with representation- or optimization-oriented 3DGS variants [19, 38, 43, 49, 40], while its states trace inference over scene parameters, where dynamic Gaussian methods [39, 42, 15, 36] model physical scene motion. Evolution along the path thus serves as an inference path in Gaussian scene space, allowing sparse-view reconstruction to be learned through conditioned scene updates toward the renderable target.

To instantiate this idea, we propose GRiF, a geometry-aware flow matching framework for sparse-view 3D Gaussian Splatting. During training, GRiF matches source and target scenes and supervises the conditional velocity along paths directly in Gaussian parameter space; at inference, it amortizes this view-conditioned transport to construct a renderable scene from sparse posed images. A hierarchical scene state organizes updates from coarse layout to fine detail, geometry-aware paths respect the distinct spaces of Euclidean attributes and rotations, and a cross-scale predictor conditions the updates on image and camera evidence. Experiments on RealEstate10K and ACID, together with zero-shot transfer to ScanNet, DL3DV, and DTU, show improved in-domain reconstruction and cross-domain performance. We make three contributions.

- We formulate feed-forward sparse-view 3DGS as conditional flow matching over Gaussian scene parameters, providing intermediate supervision directly on the Gaussian scene state.
- We define a geometry-aware Gaussian transport path that uses Euclidean velocities for unconstrained attributes and Lie-algebra velocities for per-Gaussian rotations on $\mathrm{SO}(3)$.
- We propose a hierarchical view-conditioned velocity predictor that guides coarse-to-fine scene-space updates using sparse image and camera evidence.

2 Related Work

3D Gaussian Splatting for reconstruction and sparse-view prediction. Reconstructing a scene from sparse views is challenging: limited input evidence must support a high-dimensional structured output. Per-scene optimization methods address this by improving the fitting process. COLMAP-Free 3DGS [10] and SfM-Free 3DGS [16] relax pose and point-cloud initialization requirements. Depth normalization [21], structural regularization [31], depth-density guidance [33], and co-adaptation control [4] stabilize per-scene fitting under sparse observations. These methods remain optimization procedures for individual scenes and do not generalize across scenes at test time.

Feed-forward methods amortize reconstruction across scenes by learning a mapping from sparse views to a renderable Gaussian scene. Existing approaches strengthen the input signal to the predictor. Cross-view correspondence [37, 25], depth-aware cues [41, 24, 12], hierarchical organization [35], pose-free reconstruction [44, 14, 17], generated-view supervision [20], learned densification [28, 27], and surface-continuity constraints [13] all improve what the predictor observes or how the final scene is regularized. The training signal itself, however, remains a single endpoint loss against held-out views. GRiF addresses this gap by supervising the construction path of the Gaussian scene, densifying the training signal without requiring additional input views.

Flow-based reconstruction and generative 3DGS. Feed-forward sparse-view prediction relies on a single endpoint loss against held-out views, which provides one error signal per rendered pixel for a high-dimensional structured output. Flow-based formulations have begun to address different aspects of this problem. SplatFlow [11] targets the output distribution, using rectified flow over multi-view latents to generate plausible Gaussian scenes for text-conditioned synthesis and editing. Flow Distillation Sampling [5] targets the optimization signal, using pretrained matching priors as a flow regularizer for per-scene 3DGS fitting under sparse observations. FlowR [9] targets data density, mapping novel-view renderings from sparse reconstructions toward those expected

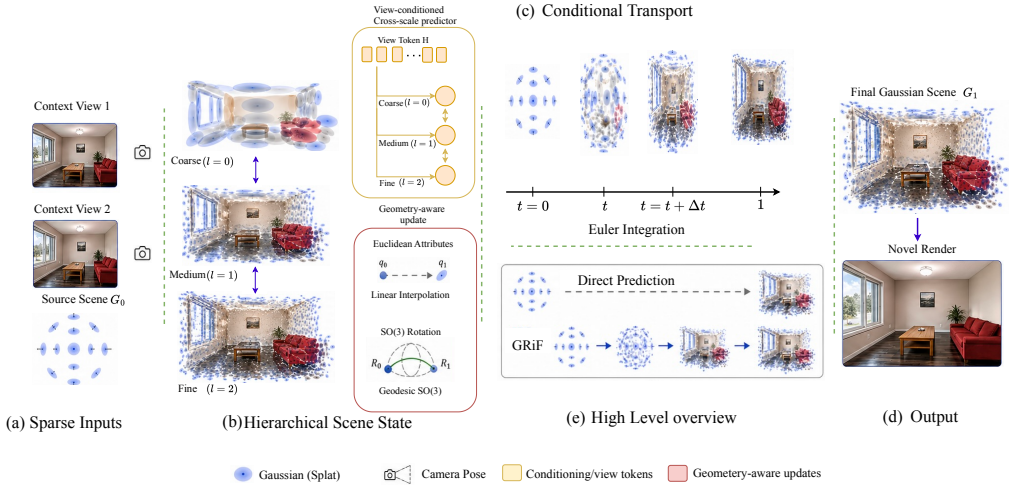


Figure 1: Overview of GRiF. Given sparse context views and a source Gaussian state, GRiF integrates a view-conditioned, geometry-aware velocity field to transport a hierarchical Gaussian scene toward the final representation used for novel-view rendering.

from dense reconstructions and using the generated views to densify the input evidence for an existing reconstruction pipeline. None of these methods supervises the construction path of the Gaussian scene itself under feed-forward prediction. GRiF addresses this gap by applying conditional transport directly over Gaussian scene parameters, providing a velocity target at every $t \in [0, 1]$ for every primitive rather than a single endpoint loss. This requires adapting flow matching on Lie groups [32, 6, 45] to a structured product-manifold state where per-primitive rotations are one component alongside Euclidean attributes, and where source-target correspondence is not given but determined by permutation-aware geometric assignment.

Dynamic and structured Gaussian scene modeling. Treating Gaussian scenes as structured evolving states rather than static parameter lists opens a richer design space for scene representation, but existing methods instantiate this idea for different goals. Dynamic methods couple Gaussian evolution to physical time: 4D-GS [39] combines 3D Gaussians with 4D neural voxels for timestamp-dependent deformations, Grid4D [42] uses decomposed hash encodings and directional attention for faithful dynamic deformation, SC-GS [15] separates sparse motion control points from dense appearance Gaussians for editable dynamic scenes, and ODE-GS [36] models Gaussian trajectories as continuous-time latent dynamics for extrapolation beyond observed timestamps. HiSplat [35] brings hierarchical organization to generalizable static 3DGS, but its hierarchy operates over pixel-aligned feature maps decoded per image scale, with inter-level interaction driven by rendering-based error maps and per-stage photometric supervision. The shared limitation is that neither line of work uses scene-space evolution as a supervisory signal for feed-forward prediction under sparse views. GRiF instantiates this principle directly, applying conditional transport from a source Gaussian scene to a target under sparse posed-view evidence, with evolution serving as an inference path rather than modeling physical time or organizing pixel-aligned features.

Learning and reformulating Gaussian representations. Raw Gaussian parameters are non-unique, heterogeneous, and couple geometry with appearance in ways that complicate direct use in neural learning systems [40]. Prior work addresses this by reformulating individual components of the representation. 3DGS-MCMC [19] reinterprets primitives as samples from an underlying scene distribution, replacing hand-designed densification heuristics with principled state transitions. Normal-GS [38] integrates surface normals into rendering to improve geometric fidelity. Spec-Gaussian [43] replaces spherical harmonics with an anisotropic appearance field for view-dependent specular effects. Perceptual-GS [49] adapts densification to perceptual sensitivity, allocating capacity to visually important regions. Each of these reformulations targets a specific component in isolation. GRiF addresses the same underlying problem at the level of the learning formulation, treating Euclidean and rotational Gaussian attributes according to their respective geometric structures and supervising their construction path rather than their final values.

3 Methodology

GRiF formalizes sparse-view Gaussian prediction as conditional flow matching over hierarchical Gaussian scene states. Given context views, a fixed source state, and a supervised target state during training, the model learns a velocity field that is integrated at inference time to produce the predicted Gaussian scene. Figure 1 summarizes the pipeline.

3.1 Problem Setup

Given a sparse set of context views, the goal is to predict a renderable 3D Gaussian scene without scene-specific optimization at test time (Eq. 1). During training, each sample provides a tuple $(\mathcal{C}, \mathcal{G}_0, \mathcal{G}_1)$, where $\mathcal{G}_0$ is the source scene and $\mathcal{G}_1$ is the target Gaussian scene used for supervision. The target scene is not predicted at inference time, it is used only to define supervised intermediate states during training. At test time, only $\mathcal{C}$ and $\mathcal{G}_0$ are required.

$$\mathcal{C} = \{(I_v, \pi_v)\}_{v=1}^{V}, \tag{1}$$

where I_v is the v-th context image and π_v denotes its camera parameters.

GRiF augments endpoint target supervision with intermediate scene-space supervision. It evolves a time-indexed scene state $\{\mathcal{G}_t\}_{t\in[0,1]}$ according to Eq. 2. At each time t, the model observes the current Gaussian state and predicts the local update that makes it more consistent with the context views. Integrating this field from $t = 0$ to $t = 1$ produces the end-point prediction $\widehat{\mathcal{G}}_1$, whose finest hierarchy level is rendered from novel viewpoints.

$$\frac{d\mathcal{G}_t}{dt} = \mathbf{v}_\theta(\mathcal{G}_t, \mathcal{C}, t), \qquad t \in [0, 1], \tag{2}$$

where $\mathbf{v}_\theta$ is specified in Section 3.4.

3.2 Hierarchical Gaussian Scene State

GRiF represents a transported scene using a three-level hierarchy that enables coarse-to-fine modeling. Lower levels summarize global structure using fewer primitives, while higher levels progressively increase capacity to capture finer geometric and appearance details:

$$\mathcal{G} = \left\{\mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \mathcal{G}^{(2)}\right\}, \tag{3}$$

where $\ell \in \{0, 1, 2\}$ denotes the coarse, medium, and fine levels, with $N_0 < N_1 < N_2$.

GRiF separates Euclidean attributes from rotational components at each level:

$$\mathcal{G}^{(\ell)} = \left(E^{(\ell)}, R^{(\ell)}\right), \qquad E^{(\ell)} \in \mathbb{R}^{N_\ell \times d_q}, \quad R^{(\ell)} \in \mathrm{SO}(3)^{N_\ell}, \tag{4}$$

where $E^{(\ell)}$ stores centers, log-scales, opacity logits, and appearance coefficients in unconstrained vector space, while $R^{(\ell)}$ contains one rotation matrix per primitive on the manifold $\mathrm{SO}(3)$. This defines a product-structured state space:

$$\mathcal{M}^{(\ell)} = \mathbb{R}^{N_\ell \times d_q} \times \mathrm{SO}(3)^{N_\ell}, \qquad \mathcal{M} = \prod_{\ell=0}^{2} \mathcal{M}^{(\ell)}. \tag{5}$$

Euclidean attributes admit additive updates in unconstrained coordinates before nonlinear rendering transformations are applied. Rotations follow group-consistent geodesic paths on $\mathrm{SO}(3)$ to preserve the manifold structure (Appendix A). At inference, the finest level $\mathcal{G}^{(2)}$ is rendered for novel-view synthesis, while coarser levels provide intermediate states that guide coarse-to-fine prediction.

3.3 Conditional Paths in Gaussian Scene Space

After hierarchy construction, the source and target scenes have the same number of primitives at each level. Since teacher Gaussian primitives lack a canonical ordering, GRiF first computes a level-wise assignment between source and target primitives. Let $\mathcal{P}_{N_\ell}$ denote the set of permutation matrices of size N_ℓ:

$$\Pi_\ell^\star = \arg\min_{\Pi \in \mathcal{P}_{N_\ell}} \left\langle \Pi, C^{(\ell)} \right\rangle, \tag{6}$$

where $C^{(\ell)} \in \mathbb{R}^{N_\ell \times N_\ell}$ is a geometry-based matching cost:

$$C_{ij}^{(\ell)} = \lambda_x \left\| X_{0,i}^{(\ell)} - X_{1,j}^{(\ell)} \right\|_2^2 + \lambda_s \left\| \mathbf{s}_{0,i}^{(\ell)} - \mathbf{s}_{1,j}^{(\ell)} \right\|_2^2 + \lambda_{\text{geo}} d_{\text{geo}}\left(R_{0,i}^{(\ell)}, R_{1,j}^{(\ell)}\right)^2. \tag{7}$$

Here $d_{\text{geo}}(R_a, R_b) = \| \operatorname{Log}(R_a^\top R_b)^\vee \|_2$ is the geodesic distance on $\mathrm{SO}(3)$, X denotes centers normalized by the scene bounding-box diagonal, and $\mathbf{s}$ denotes per-level normalized log-scales. We use λ_x=1, λ_s=0.1, λ_{geo}=0.01 for all levels. The assignment uses geometry only (no color), is treated as stop-gradient, and reorders the target primitives for flow supervision:

$$\widetilde{E}_1^{(\ell)} = \Pi_\ell^\star E_1^{(\ell)}, \qquad \widetilde{R}_{1,i}^{(\ell)} = R_{1,\pi_\ell^\star(i)}^{(\ell)}. \tag{8}$$

Euclidean path. The conditional path is linear in the stored coordinate space:

$$E_t^{(\ell)} = (1-t)E_0^{(\ell)} + t\widetilde{E}_1^{(\ell)}, \qquad u_{E,t}^{\star(\ell)} = \frac{\partial E_t^{(\ell)}}{\partial t} = \widetilde{E}_1^{(\ell)} - E_0^{(\ell)}. \tag{9}$$

The displacement accumulated by time t is $E_t^{(\ell)} - E_0^{(\ell)} = t u_{E,t}^{\star(\ell)}$. Linearity in stored coordinates induces geometric interpolation in rendered scale-space (since scales are exponentiated from log-scale) and sigmoid-space interpolation in rendered opacity. The path is intentionally linear in the parameter space the network operates in, keeping the Euclidean reference velocity $u_{E,t}^{\star(\ell)}$ constant. Source opacity logits and color coefficients are initialized to zero, so the source scene corresponds to a 50% opacity field of uniformly grey primitives at Halton-sampled centers. Source rotations are identity and log-scales are initialized from mean nearest-neighbor spacing. The flow field thus learns to push opacity logits and color coefficients toward target values from a fixed neutral primitives.

Rotation path. For primitive i at level ℓ, define the relative rotation and its geodesic path by

$$R_{t,i}^{(\ell)} = R_{0,i}^{(\ell)} \exp\left(t\,\Omega_i^{(\ell)}\right), \qquad \Omega_i^{(\ell)} = \operatorname{Log}\left({R_{0,i}^{(\ell)}}^\top \widetilde{R}_{1,i}^{(\ell)}\right), \tag{10}$$

The relative rotation is $\Omega_i^{(\ell)}$, and the constant reference angular velocity for primitive i is $\boldsymbol{\omega}_i^{\star(\ell)} = (\Omega_i^{(\ell)})^\vee$. Along the selected geodesic branch, $\operatorname{Log}((R_{0,i}^{(\ell)})^\top R_{t,i}^{(\ell)}) = t\Omega_i^{(\ell)}$. Branch handling near angle π and a formal validity argument are given in Appendix A. The intermediate scene state collects level-wise blocks:

$$\mathcal{G}_t = \left\{ \left(E_t^{(\ell)}, R_t^{(\ell)}\right) \right\}_{\ell=0}^{2}, \tag{11}$$

where $R_t^{(\ell)} = \{R_{t,i}^{(\ell)}\}_{i=1}^{N_\ell}$. Together, Eqs. (9) and (10) specify the reference velocities that the predictor learns to match at any sampled state.

3.4 View-Conditioned Cross-Scale Velocity Field

The velocity predictor maps the current state, context observations, and time to level-wise update directions as shown in Eq. (12).

$$\left\{ \hat{u}_{E,t}^{(\ell)}, \hat{\boldsymbol{\omega}}_t^{(\ell)} \right\}_{\ell=0}^{2} = \mathbf{v}_\theta(\mathcal{G}_t, \mathcal{C}, t), \tag{12}$$

Each context view is encoded as a joint image-camera token $H_v = W_{\text{img}}\, E_{\text{img}}(I_v) + E_{\text{cam}}(\pi_v)$, where E_{img} extracts image features and E_{cam} is a per-view MLP applied to camera intrinsics and extrinsics, broadcast spatially before addition, the full conditioning set is $H = \{H_v\}_{v=1}^{V}$. At hierarchy level ℓ, the current Gaussian state is embedded into per-primitive tokens $Z^{(\ell)} = \Phi_{\text{tok}}^{(\ell)}(E_t^{(\ell)},\, \phi(R_t^{(\ell)}),\, E_\ell,\, \gamma(t))$, where $\phi(R_t^{(\ell)})$ is a continuous rotation feature (selected columns of the rotation matrix), E_ℓ is a learned level embedding, and $\gamma(t)$ is a learned time embedding. The point tokens attend to the context tokens via Eq. 13, allowing each primitive to condition its update on the views that provide evidence for its current location. For $\ell > 0$, coarse-to-fine fusion propagates features from the previous level (Eq. 14).

$$\widetilde{Z}^{(\ell)} = Z^{(\ell)} + \operatorname{CrossAttn}\left(Z^{(\ell)}, H, H\right), \tag{13}$$

$$Z_{\mathrm{cs}}^{(\ell)} = \mathcal{U}\Big(F^{(\ell-1)},\, X_t^{(\ell-1)},\, X_t^{(\ell)}\Big), \qquad \widehat{Z}^{(\ell)} = \widetilde{Z}^{(\ell)} + \Phi_{\mathrm{fuse}}^{(\ell)}\Big(\widetilde{Z}^{(\ell)},\, Z_{\mathrm{cs}}^{(\ell)}\Big), \tag{14}$$

Here $Z_{\mathrm{cs}}^{(\ell)}$ is the transferred coarse-level feature, distinct from the Euclidean velocity notation. The hierarchy is evaluated sequentially from $\ell = 0$ to $\ell = 2$. The operator $\mathcal{U}$ transfers information from coarse tokens $F^{(\ell-1)}$ to current-level centers $X_t^{(\ell)}$ using cross-attention with pairwise spatial-distance bias; $\Phi_{\mathrm{fuse}}^{(\ell)}$ is a residual MLP block. At the coarsest level, $\widehat{Z}^{(0)} = \widetilde{Z}^{(0)}$. Time-conditioned transformer blocks then refine the fused tokens, with $\gamma(t)$ injected via adaptive layer normalization:

$$F^{(\ell)} = T^{(\ell)}\Big(\widehat{Z}^{(\ell)},\, \gamma(t)\Big), \qquad \Big(\hat{u}_{E,t}^{(\ell)},\, \hat{\boldsymbol{\omega}}_t^{(\ell)}\Big) = \Psi^{(\ell)}\Big(F^{(\ell)}\Big), \tag{15}$$

The time-conditioned transformer is evaluated at each hierarchy level in order, from coarse to fine; $\Psi^{(\ell)}$ has separate output heads for Euclidean and angular velocities.

3.5 Training and Inference

For each training tuple, we sample $t \sim \mathcal{U}[0,1]$, construct $\mathcal{G}_t$ using Section 3.3, and evaluate $\mathbf{v}_\theta(\mathcal{G}_t, \mathcal{C}, t)$. At hierarchy level ℓ, the Euclidean loss is the standard flow-matching velocity-matching objective:

$$\mathcal{L}_{\mathrm{E}}^{(\ell)} = \frac{1}{N_\ell d_q} \left\| \hat{u}_{E,t}^{(\ell)} - u_{E,t}^{\star(\ell)} \right\|_F^2. \tag{16}$$

For rotations, directly comparing coordinates ignores the group structure. Instead, we apply both the reference and predicted angular velocities for a bounded step $\delta_t = \min(\delta,\, 1-t)$ and compare the resulting rotations on SO(3):

$$\mathcal{L}_{\mathrm{R}}^{(\ell)} = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \frac{1}{\bar{\delta}_t^2}\, d_{\mathrm{geo}}\Big(R_{t,i}^{(\ell)} \exp\Big(\delta_t\, \hat{\boldsymbol{\omega}}_{t,i}^{(\ell)\wedge}\Big),\, R_{t,i}^{(\ell)} \exp\Big(\delta_t\, \boldsymbol{\omega}_i^{\star(\ell)\wedge}\Big)\Big)^2, \tag{17}$$

where d_{geo} is the geodesic distance and $\bar{\delta}_t = \max(\delta_t,\, \epsilon_\delta)$ avoids division by zero near $t{=}1$. The $\bar{\delta}_t^{-2}$ normalization keeps the loss on the scale of angular-velocity matching regardless of step size. We use $\delta{=}1/50$, $\epsilon_\delta{=}10^{-4}$, and $\lambda_{\mathrm{R}}{=}1$ in all experiments. The full objective averages over hierarchy levels and adds a photometric rendering loss on the terminal state:

$$\mathcal{L} = \frac{1}{3} \sum_{\ell=0}^{2} \Big(\mathcal{L}_{\mathrm{E}}^{(\ell)} + \lambda_{\mathrm{R}}\, \mathcal{L}_{\mathrm{R}}^{(\ell)}\Big) + \lambda_{\mathrm{img}}\, \mathcal{L}_{\mathrm{img}}\Big(\widehat{\mathcal{G}}_1^{(2)}\Big), \qquad \min_\theta\, E_{(\mathcal{C},\mathcal{G}_0,\mathcal{G}_1),\,t}[\mathcal{L}], \tag{18}$$

where $\lambda_{\mathrm{img}} = 0.2$ and $\mathcal{L}_{\mathrm{img}}$ is an MSE + LPIPS loss between the rendered finest-level output and the held-out target view. The renderer follows the standard 3DGS rasterizer [18].

At inference, the source state is integrated for K Euler steps with $\Delta t{=}1/K$:

$$E_{k+1}^{(\ell)} = E_k^{(\ell)} + \Delta t\, \hat{u}_{E,k}^{(\ell)}, \qquad R_{k+1,i}^{(\ell)} = R_{k,i}^{(\ell)} \exp\Big(\Delta t\, \hat{\boldsymbol{\omega}}_{k,i}^{(\ell)\wedge}\Big). \tag{19}$$

After K steps, GRiF renders the finest level $\widehat{\mathcal{G}}_1^{(2)}$ for novel-view synthesis.

4 Experiments

4.1 Experimental Setup

We evaluate GRiF under the sparse-view protocol used by SurfSplat [13]. The datasets are chosen to cover different scene distributions and surface types. RealEstate10K [50] contains real indoor and outdoor property videos dominated by architectural layouts, furniture, walls, floors, and other man-made surfaces, while ACID [23] contains aerial outdoor scenes with coastlines, vegetation, water, sky, and large depth variation. We use these datasets for in-domain evaluation with two context views and held-out target views. To test cross-domain transfer, we train on RealEstate10K and evaluate zero-shot on ScanNet [8], DL3DV [22], and DTU [1] without finetuning. Following the SurfSplat protocol [13], all input and rendered images use 256×256 resolution, with two context views, held-out target-view sampling, and the same LPIPS configuration for posed sparse-view comparisons. Metrics are averaged over target images and then over scenes.

We report PSNR, SSIM, and LPIPS on all datasets. Unless otherwise stated, experiments use a three-level hierarchy with $N_0 = 512$, $N_1 = 1024$, and $N_2 = 4096$ Gaussian primitives, spherical harmonics degree 0, canonical source-scene initialization, and the default target Gaussian supervision. At inference, we integrate the learned velocity field with $K = 16$ Euler steps and render the finest hierarchy level. Full architecture, training, and scene-state construction details are provided in Appendix B. Main ablations use the same protocol, metrics, renderer, source initialization, target scenes, and inference procedure unless the ablated factor is explicitly named.

4.2 Ablation Studies

We conduct three ablation studies on GRiF: (i) the cumulative contribution of the transport formulation, hierarchy, rotation treatment, and cross-scale fusion (Table 1), (ii) the role of hierarchy versus Gaussian count and cross-scale communication (Table 2), and (iii) the view-conditioning pathway used by the velocity field (Table 3).

Table 1 ablates the contributions of conditional flow matching (FM), hierarchical scene state (H), and cross-scale fusion (CSF). Our baseline, a single-level endpoint regressor trained against identical teacher targets ($-$FM, $-$H, $-$CSF), achieves 25.50 PSNR. FM yields the most significant performance boost, delivering consistent gains of $+1.22$ to $+1.36$ PSNR across all configurations. Notably, FM provides a $+1.22$ PSNR improvement over the raw baseline, which scales up to a $+1.36$ PSNR improvement when integrated with both H and CSF. This indicates that FM operates most effectively when transporting a structured hierarchical representation. H alone contributes a smaller direct effect ($+0.44$ PSNR over the baseline), consistent with the observation that a parameter-space hierarchy without inter-level communication provides limited additional structure. H becomes substantially more valuable once CSF is introduced: CSF adds $+1.08$ PSNR given H and FM, making it the second largest contributor. The full configuration improves over the baseline by $+2.76$ PSNR, with no single factor accounting for the gain in isolation.

Table 2 separates the contribution of the hierarchical scene state from Gaussian count and inter-level communication. The fine-only variant predicts a single-level scene with N_2=4096 primitives and reaches only 25.82 PSNR on RealEstate10K, 2.44 dB below full GRiF. Increasing the budget to $5{,}632$ primitives in fine-only matched-budget recovers $+0.68$ PSNR but still falls 1.76 dB short, showing that primitive count alone does not explain the gain. Reintroducing the three-level hierarchy without learned inter-level communication (hierarchy, no fusion) adds $+0.47$ PSNR over the matched-budget flat model, and learned cross-scale fusion in full GRiF adds a further $+1.29$ PSNR. Across both datasets, the three sources of improvement, primitive count, parameter-space hierarchy, and cross-scale fusion, are each necessary, with cross-scale fusion contributing the largest hierarchy-related effect.

Table 3 evaluates how context evidence reaches the velocity field while keeping the scene representation, transport paths, and training objective fixed. Global image pooling, which provides only scene-level appearance, reaches just 22.30 PSNR on RealEstate10K. Adding camera embeddings (global + camera) raises this by $+3.58$ PSNR by giving the predictor access to viewing geometry. Replacing pooled features with attention from Gaussian tokens to spatial image tokens (image attention) adds another $+0.68$ PSNR, confirming that primitives benefit from spatially-localized evidence. Combining attention with camera embeddings, the configuration used by full GRiF, adds a final $+1.70$ PSNR on RE10K and $+1.87$ on ACID. The pattern is consistent across both datasets: each component contributes additively, with the largest single jump coming from the introduction of camera embeddings.

Consistency-based acceleration. We additionally apply post-training flow-map consistency distillation (FCD) [34, 2]. The pretrained K=16 GRiF model is frozen and used to generate intermediate Gaussian states. A student initialized from the same backbone learns direct transitions between these states, using Euclidean consistency for unconstrained attributes and geodesic consistency for rotations on $\mathrm{SO}(3)$. One-step FCD achieves 27.65 dB in 0.09 s, while two-step FCD achieves 28.05 dB in 0.14 s, compared with 28.26 dB in 0.70 s for the original integration. Thus, FCD retains most of GRiF's reconstruction quality without retraining the original model from scratch. Table 4 summarizes this quality–runtime trade-off.

4.3 Comparison with Existing Methods

We compare GRiF with feed-forward Gaussian-splatting methods that represent different approaches to sparse-view prediction: image-space Gaussian prediction with pixelSplat [3], multi-view aggre-

Table 1: Component ablation on RealEstate10K. All variants use the same encoder, Gaussian budget, teacher targets, and training schedule. FM: conditional flow matching. H: hierarchy. CSF: cross-scale fusion. The highlighted row is the setting used in GRiF.

FM	H	CSF	PSNR↑	SSIM↑	LPIPS↓
✗	✗	✗	25.50	0.80	0.22
✓	✗	✗	26.72	0.82	0.20
✗	✓	✗	25.94	0.82	0.19
✓	✓	✗	27.18	0.85	0.15
✗	✓	✓	26.90	0.88	0.13
✓	✓	✓	28.26	0.91	0.10

Table 2: Hierarchy and cross-scale fusion ablation on RealEstate10K and ACID.

Variant	RealEstate10K			ACID		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Fine only	25.82	0.78	0.21	25.17	0.81	0.19
Fine only, matched budget	26.50	0.81	0.18	26.25	0.82	0.17
Hierarchy, no fusion	26.97	0.85	0.15	27.50	0.84	0.15
GRiF	28.26	0.91	0.10	29.37	0.87	0.13

Table 3: View-conditioning ablation on RealEstate10K and ACID.

Variant	RealEstate10K			ACID		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Global image	22.30	0.80	0.20	24.37	0.80	0.19
Global + camera	25.88	0.83	0.16	26.73	0.82	0.16
Image attention	26.56	0.88	0.13	27.50	0.84	0.14
Image + camera attn.	28.26	0.91	0.10	29.37	0.87	0.13

gation with MVSplat [7], transformer-based cross-view fusion with TranSplat [46], hierarchical reconstruction with HiSplat [35], depth-aware Gaussian prediction with DepthSplat [41], large reconstruction model pretraining with GS-LRM [47], and surface-continuity regularization with SurfSplat [13]. Table 5 evaluates in-domain reconstruction on RealEstate10K [50] and ACID [23], while Table 6 evaluates zero-shot transfer to ScanNet [8], DL3DV [22], and DTU [1] after training on RealEstate10K.

On RealEstate10K, GRiF improves the strongest prior result from 27.54 to 28.26 PSNR ($+0.72$), from 0.89 to 0.91 SSIM, and from 0.11 to 0.10 LPIPS, advancing all three metrics simultaneously over SurfSplat [13] and DepthSplat [41] which had previously traded margins on individual metrics. On ACID, GRiF improves the best prior PSNR by $+1.03$ dB ($28.34 \rightarrow 29.37$), SSIM from 0.85 to 0.87, and LPIPS from 0.14 to 0.13. The ACID gain is the larger of the two and is informative because ACID contains aerial outdoor scenes with depth ranges and surface statistics distinct from the indoor architectural distribution that dominates RealEstate10K, indicating that the scene-space transport formulation is not specialized to a single visual domain. Across both datasets, the consistency of the improvement across distortion (PSNR), structural (SSIM), and perceptual (LPIPS) metrics is consistent with predicting a better Gaussian scene rather than overfitting to a single objective, which Figure 2 corroborates qualitatively.

Generalization The zero-shot evaluation in Table 6 tests whether the learned scene-space transport generalizes beyond the training distribution. Models are trained on RealEstate10K and evaluated without finetuning on three datasets that differ from training in distinct ways: ScanNet [8] contains cluttered indoor RGB-D scans with different camera statistics, DL3DV [22] contains diverse real-world scenes with broader appearance and motion, and DTU [1] provides controlled object-centric captures with a camera geometry far removed from the indoor video distribution. GRiF obtains the best PSNR on all three transfer datasets, with gains that grow with the distribution gap: $+2.43$ dB on ScanNet ($20.31 \rightarrow 22.74$), $+3.19$ dB on DL3DV ($28.14 \rightarrow 31.33$), and $+3.14$ dB on DTU ($15.54 \rightarrow 18.68$), where all baselines cluster between 12.75 and 15.54 PSNR. The improvements are consistent across PSNR, SSIM, and LPIPS, indicating genuinely better reconstructions rather than gains on a single metric. The widening margin under increasing domain shift suggests that GRiF's

Table 4: Quality and inference time for flow-consistency distillation on RealEstate10K under the two-view, 256×256 setting.

Model	Steps	Time/scene ↓	PSNR ↑	SSIM ↑
GRiF	16	0.70s	28.26	0.91
GRiF-FCD	1	0.09s	27.65	0.90
GRiF-FCD	2	0.14s	28.05	0.91



Figure 2: Qualitative comparison of pixelSplat, MVSplat, SurfSplat, GRIF, and ground truth.

Table 5: Comparison on RealEstate10K and ACID.

Method	RealEstate10K			ACID		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
pixelSplat	26.05	0.86	0.14	28.28	0.84	0.15
MVSplat	26.36	0.87	0.13	28.20	0.84	0.15
TranSplat	26.69	0.88	0.13	28.34	0.85	0.14
HiSplat	27.21	0.88	0.12	28.75	0.85	0.13
DepthSplat	27.50	0.89	0.11	28.86	0.86	0.14
GS-LRM	27.68	0.81	0.21	28.31	0.83	0.18
SurfSplat	27.54	0.89	0.11	28.34	0.85	0.14
GRiF	28.26	0.91	0.10	29.37	0.87	0.13

Table 6: Zero-shot transfer for models trained on RealEstate10K.

Method	ScanNet			DL3DV			DTU		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
pixelSplat	19.61	0.71	0.32	27.20	0.88	0.10	12.75	0.33	0.64
MVSplat	18.73	0.69	0.33	23.84	0.77	0.16	13.91	0.47	0.39
HiSplat	18.27	0.66	0.31	23.37	0.74	0.16	14.24	0.51	0.35
TranSplat	18.94	0.71	0.33	23.91	0.77	0.16	14.96	0.53	0.33
GS-LRM	20.23	0.71	0.32	27.57	0.84	0.12	15.22	0.48	0.35
DepthSplat	20.20	0.74	0.31	28.14	0.91	0.08	14.59	0.43	0.44
SurfSplat	20.31	0.73	0.31	27.38	0.89	0.11	15.54	0.49	0.33
GRiF	22.74	0.78	0.25	31.33	0.91	0.07	18.68	0.53	0.30

improvements are not tied to the RealEstate10K distribution, we attribute this to two properties of the formulation: conditional transport densifies the training signal beyond endpoint photometric supervision, and the hierarchical scene state with cross-scale fusion separates coarse layout from fine

appearance, allowing in-distribution layout priors to be reused when out-of-distribution appearance differs.

5 Conclusion

We introduced GRiF, a flow-matching framework that supervises sparse-view 3D Gaussian prediction through conditional transport in scene space. GRiF transports a hierarchical Gaussian state from a fixed source to a renderable target. It uses a view-conditioned velocity field, with Euclidean paths for unconstrained attributes and Lie-algebra updates on $\mathrm{SO}(3)$ for rotations. Experiments show consistent gains on RealEstate10K and ACID. GRiF also shows stronger zero-shot transfer to ScanNet, DL3DV, and DTU, with margins that widen under larger domain shift. The results indicate that conditional transport, hierarchical scene representation, and cross-scale fusion jointly improve sparse-view Gaussian scene prediction.

Limitations. GRiF assumes posed input views and static scenes, and its scene-space supervision depends on optimized teacher Gaussian scenes. As a result, teacher quality, hierarchy construction, and the hand-designed assignment cost can affect the learned transport. GRiF also uses multi-step Euler integration at inference, adding computation relative to single-pass endpoint predictors. Future work should reduce dependence on teacher targets, accelerate integration, and extend scene-space flow to pose-free, dynamic, reflective, and transparent-scene settings.

Acknowledgments

This work is supported by the Key Technological Innovation Program of Ningbo City under Grant No. 2024Z297 and the Intelligent Computing Center of Shenzhen University.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016.

[2] Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Trans. Mach. Learn. Res.*, 2025, 2024. URL `https://api.semanticscholar.org/CorpusID:270380005`.

[3] 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 (CVPR)*, 2024.

[4] Kangjie Chen, Yingji Zhong, Zhihao Li, Jiaqi Lin, Youyu Chen, Minghan Qin, and Haoqian Wang. Quantifying and alleviating co-adaptation in sparse-view 3d gaussian splatting. 2025.

[5] Lin-Zhuo Chen, Kangjie Liu, Youtian Lin, Zhihao Li, Siyu Zhu, Xun Cao, and Yao Yao. Flow distillation sampling: Regularizing 3d gaussians with pre-trained matching priors. In *International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=BzsjHiBfLk`.

[6] Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, 2024.

[7] 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 (ECCV)*, 2024. URL `https://arxiv.org/abs/2403.14627`.

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] Tobias Fischer, Samuel Rota Bulò, Yung-Hsu Yang, Nikhil Varma Keetha, Lorenzo Porzi, Norman Müller, Katja Schwarz, Jonathon Luiten, Marc Pollefeys, and Peter Kontschieder. Flowr: Flowing from sparse to dense 3d reconstructions. In *ICCV 2025 Workshop on Wild 3D*, 2025. URL https://openreview.net/forum?id=cUREnXF0hn.

[10] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20796–20805, 2024.

[11] Hyojun Go, Byeongjun Park, Jiho Jang, Jin-Young Kim, Soonwoo Kwon, and Changick Kim. Splatflow: Multi-view rectified flow model for 3d gaussian splatting synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21524–21536, June 2025.

[12] Liang Han, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Binocular-guided 3d gaussian splatting with view consistency for sparse view synthesis. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2191. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7eca17ef54789b0663cab421f2e9dbf5-Abstract-Conference.html.

[13] Bing He, Jingnan Gao, Yunuo Chen, Ning Cao, Gang Chen, Zhengxue Cheng, Li Song, and Wenjun Zhang. Surfsplat: Conquering feedforward 2d gaussian splatting with surface continuity priors. In *ICLR 2026*, 2026.

[14] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. Pf3plat: Pose-free feed-forward 3d gaussian splatting. 2024.

[15] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4220–4230, 2024.

[16] Bo Ji and Angela Yao. Sfm-free 3d gaussian splatting via hierarchical training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21654–21663, 2025.

[17] Gyeongjin Kang, Jisang Yoo, Jihyeon Park, Seungtae Nam, Hyeonsoo Im, Sangheon Shin, Sangpil Kim, and Eunbyung Park. Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22012–22022, 2025.

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

[19] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. volume 37, pages 80965–80986, 2024.

[20] Hanyang Kong, Xingyi Yang, and Xinchao Wang. Generative sparse-view gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26745–26755, 2025.

[21] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20775–20785, 2024.

[22] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanhao Li, Xingyu Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, Aniket Bera, and Haitao Yang. 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 (CVPR)*, 2024.

[23] 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 (ICCV)*, 2021. URL https://infinite-nature.github.io/.

[24] Yifan Liu, Keyu Fan, Weihao Yu, Chenxin Li, Hao Lu, and Yixuan Yuan. Monosplat: Generalizable 3d gaussian splatting from monocular depth foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21570–21579, 2025.

[25] Zhiyuan Min, Yawei Luo, Jianwen Sun, and Yi Yang. Epipolar-free 3d gaussian splatting for generalizable novel view synthesis. volume 37, pages 39573–39596, 2024.

[26] Yuxuan Mu, Xinxin Zuo, Chuan Guo, Yilin Wang, Juwei Lu, Xiaofeng Wu, Songcen Xu, Peng Dai, Youliang Yan, and Li Cheng. Gsd: View-guided gaussian splatting diffusion for 3d reconstruction. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXIX*, page 55–72, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72985-0. doi: 10.1007/978-3-031-72986-7_4.

[27] 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 (CVPR)*, pages 26683–26693, June 2025.

[28] Junfeng Ni, Yixin Chen, Zhifei Yang, Yu Liu, Ruijie Lu, Song-Chun Zhu, and Siyuan Huang. G4Splat: Geometry-guided gaussian splatting with generative prior. In *ICLR 2026*, 2026. URL https://openreview.net/forum?id=kdPmsMVhZf.

[29] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

[30] Avinash Paliwal, Xi-Long Zhou, Wei Ye, Jinhui Xiong, Rakesh Ranjan, and Nima Khademi Kalantari. Ri3d: Few-shot gaussian splatting with repair and inpainting diffusion priors. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 25094–25103, 2025. URL https://api.semanticscholar.org/CorpusID:277043155.

[31] Hyunwoo Park, Gun Ryu, and Wonjun Kim. Dropgaussian: Structural regularization for sparse-view gaussian splatting. In *Proceedings of the computer vision and pattern recognition conference*, pages 21600–21609, 2025.

[32] Finn M. Sherry and Bart M.N. Smets. Flow matching on Lie groups. *arXiv preprint arXiv:2504.00494*, 2025.

[33] Meixi Song, Xin Lin, Dizhe Zhang, Haodong Li, Xiangtai Li, Bo Du, and Lu Qi. D^2GS: Depth-and-density guided gaussian splatting for stable and accurate sparse-view reconstruction. In *ICLR 2026*, 2026. URL https://openreview.net/forum?id=7yvz93kBw9.

[34] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *ArXiv*, abs/2303.01469, 2023. URL https://api.semanticscholar.org/CorpusID:257280191.

[35] Shengji Tang, Weicai Ye, Peng Ye, Weihao Lin, Yang Zhou, Tao Chen, and Wanli Ouyang. Hisplat: Hierarchical 3d gaussian splatting for generalizable sparse-view reconstruction. In *ICLR 2025*, 2025. URL https://openreview.net/forum?id=SBzIbJojs8.

[36] Daniel Wang, Patrick Rim, Tian Tian, Dong Lao, Alex Wong, and Ganesh Sundaramoorthi. Ode-gs: Latent odes for dynamic scene extrapolation with 3d gaussian splatting. In *ICLR 2026*, 2026.

[37] Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. Freesplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. volume 37, pages 107326–107349, 2024.

[38] Meng Wei, Qianyi Wu, Jianmin Zheng, Hamid Rezatofighi, and Jianfei Cai. Normal-GS: 3d gaussian splatting with normal-involved rendering. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[39] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20310–20320, 2024.

[40] Yuelin Xin, Yuheng Liu, Xiaohui Xie, and Xinke Li. Learning unified representation of 3d gaussian splatting. In *ICLR 2026*, 2026.

[41] 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*, pages 16453–16463, 2025.

[42] Jiawei Xu, Zexin Fan, Jian Yang, and Jin Xie. Grid4d: 4d decomposed hash encoding for high-fidelity dynamic gaussian splatting. volume 37, pages 123787–123811, 2024.

[43] Ziyi Yang, Xinyu Gao, Yang-Tian Sun, Yi-Hua Huang, Xiaoyang Lyu, Wen Zhou, Shaohui Jiao, Xiaojuan Qi, and Xiaogang Jin. Spec-Gaussian: Anisotropic view-dependent appearance for 3d gaussian splatting. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[44] 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. 2024.

[45] Jason Yim, Andrew Campbell, Andrew Y. K. Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Regina Barzilay, Tommi Jaakkola, and Frank Noé. Fast protein backbone generation with SE(3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

[46] Chuanrui Zhang, Yingshuang Zou, Zhuoling Li, Minmin Yi, and Haoqian Wang. TranSplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9869–9877, 2025. doi: 10.1609/aaai.v39i9.33070. URL `https://arxiv.org/abs/2408.13770`.

[47] Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Kalyan Sunkavalli, Noah Snavely, and Zexiang Xu. GS-LRM: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024. URL `https://arxiv.org/abs/2404.19702`.

[48] Ying-Ji Zhong, Zhi-Hao Li, Dave Zhenyu Chen, Lanqing Hong, and Dan Xu. Taming video diffusion prior with scene-grounding guidance for 3d gaussian splatting from sparse inputs. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6133–6143, 2025. URL `https://api.semanticscholar.org/CorpusID:276884590`.

[49] Hongbi Zhou and Zhangkai Ni. Perceptual-GS: Scene-adaptive perceptual densification for Gaussian splatting. In *Proceedings of the 42nd International Conference on Machine Learning*.

[50] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37 (4), 2018. URL `https://google.github.io/realestate10k/`.

[51] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5738–5746, 2018. URL `https://api.semanticscholar.org/CorpusID:56178817`.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes],
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction clearly align with the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The limitations are explicitly discussed in the conclusion section.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The assumptions for each theoretical result are clearly stated, and the correctness is analyzed in detail in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification:The paper presents comprehensive details on the experimental setup, model architecture, training procedures, and evaluation metrics, enabling reproduction of the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The source code will be released.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have specified all training and testing details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: We have not conducted experiments with error bars yet.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research we conducted fully conforms to the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper focuses mainly on the method and evaluation of geometry-aware flow matching for sparse-view 3DGS. While GRiF may benefit broader applications in generalizable 3D reconstruction and novel-view synthesis, we do not explicitly discuss these larger impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: Our work focuses on geometry-aware flow matching for sparse-view 3D Gaussian Splatting. It does not involve releasing models, data, or artifacts with high risk of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification:All the datasets used are publically available.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: No dataset has been proposed in this paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No human subjects were involved.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects were involved.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: GPT5 was used for writing purposes and and did not affect the core methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.