BEYOND THE VISIBLE SURFACE: GEOMETRY-GUIDED JOINT OPTIMIZATION OF MULTI-LAYER GAUSSIANS FOR SINGLE-VIEW SCENE RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward Gaussian reconstruction predicts a renderable 3D scene from a single image, yet reconstructing content beyond visible surfaces remains challenging. Limited hypotheses along each source ray restrict the representation of newly exposed regions, while depth-based initialization alone provides insufficient geometric guidance for feature learning. We propose GeoJMG, a geometry-guided framework for jointly learning multi-layer Gaussian representations for single-view scene reconstruction. Ray-Stratified Gaussian Modeling (RSGM) constructs multiple depth-ordered surface hypotheses along each source ray and predicts their Gaussian parameters through independent decoder branches over shared multi-scale features. All layers are trained jointly under a common reconstruction objective without predefined visibility roles or layer-specific supervision. To guide this learning process, we derive a compact geometric descriptor from depth, camera-space coordinates, depth gradients, surface normals, and reliability. A lightweight Geometry-Conditioned Residual Adapter (GCRA) integrates these cues into multiple encoder scales through zero-initialized projections, preserving the base features at initialization. The resulting model reconstructs a layered Gaussian scene in a single forward pass without per-scene optimization. Experiments on RealEstate10K, NYUv2, and KITTI demonstrate improved novel-view synthesis quality and effective generalization across indoor and outdoor scenes.

1 INTRODUCTION

A single photograph records the surfaces visible from one viewpoint, whereas a 3D reconstruction must remain plausible as the camera moves. Single-view scene reconstruction therefore requires inferring both observed geometry and content not directly revealed by the input. It supports interactive photo exploration and immersive content creation without calibrated multi-view capture (Wiles et al., 2020; Rockwell et al., 2021; Liu et al., 2021; Sargent et al., 2024). Feed-forward models predict a scene once for rendering from many viewpoints. Explicit anisotropic primitives and efficient rasterization make 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) well suited to this setting.

Camera motion reveals surfaces occluded in the source view. A monocular metric depth estimate provides reference positions for source-visible surfaces (Piccinelli et al., 2024), but does not specify the geometry and appearance of newly exposed regions. Layered depth and multiplane representations address this ambiguity by retaining samples at multiple depths (Shade et al., 1998; Tucker & Snavely, 2020). Similarly, multiple Gaussian hypotheses along each ray provide capacity for content revealed by viewpoint changes.

Generalizable radiance fields and Gaussian models exploit cross-view evidence from sparse images (Yu et al., 2021; Wang et al., 2021; Chen et al., 2021; Johari et al., 2022; Charatan et al., 2024; Chen et al., 2024). Single-image methods instead rely on learned scene priors and monocular guidance. Flash3D (Szymanowicz et al., 2025) places depth-ordered Gaussians along source rays, while Niagara (Wu et al., 2026) integrates depth and normals through a geometric affine field and attention. CATSplat (Roh et al., 2025) uses textual and spatial context, and MLGS (Diao et al., 2025) guides visible and occlusion layers with masks and occlusion-specific supervision. However, image appearance provides incomplete evidence about surface layout, especially near occlusion boundaries

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.

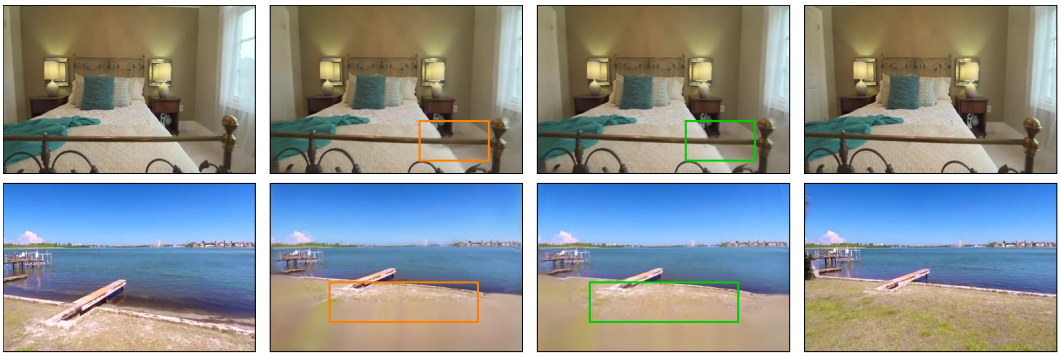

Figure 1: **Single-view scene reconstruction.** From the input images (first column), Flash3D (Szymanowicz et al., 2025) produces color artifacts around the thin bed-frame rail (top) and incompletely reconstructs the newly visible lawn (bottom, orange boxes). GeoJMG better preserves these structures and revealed regions (green boxes), producing views closer to the ground truth (last column).

and on thin or weakly textured structures. Beyond distance and orientation, spatial position, local depth variation, and reliability offer complementary cues for interpreting this evidence. We therefore condition multiple surface hypotheses on these cues to provide a common geometric reference for layer placement and Gaussian parameter prediction.

We propose GeoJMG, a geometry-guided framework for jointly learning multi-layer Gaussian scenes. Ray-Stratified Gaussian Modeling (RSGM) assigns each source ray K depth-ordered hypotheses with independently predicted Gaussian parameters. Geometry-Conditioned Residual Adapter (GCRA) encodes depth-derived geometry and injects it into multiple feature resolutions. The resulting pyramid conditions both the depth decoder and every Gaussian decoder, and all layers are jointly optimized through their combined rendering. Figure 1 illustrates the resulting improvements around thin structures and newly revealed regions.

In summary, our key contributions are:

- We propose GeoJMG. This feed-forward framework combines multi-layer Gaussians with geometric conditioning to represent observed surfaces and content revealed by camera motion in a single renderable scene.
- We develop RSGM, which combines depth-ordered surface hypotheses with independent Gaussian parameter prediction. It provides distinct geometric and appearance representations along each ray for reconstructing surfaces exposed by viewpoint changes.
- We introduce GCRA, a lightweight adapter that integrates complementary geometry across feature scales to guide layer depths and Gaussian parameters from a common spatial reference. Zero-initialized residual projections preserve the initial encoder mapping while learning geometric corrections.

2 Related Work

2.1 View Synthesis and Layered Representations

NeRF (Mildenhall et al., 2020) models continuous radiance fields. Generalizable reconstruction uses projected features (Yu et al., 2021; Wang et al., 2021), cost volumes (Chen et al., 2021; Johari et al., 2022), target-aligned volumes (Xu et al., 2024), or dense pointmap regression (Wang et al., 2024). These approaches combine observations across views, whereas single-view synthesis must infer unobserved content from one image. Layered Depth Images (Shade et al., 1998) retain multiple color-depth samples per pixel, while multiplane images use RGBA planes predicted from stereo pairs (Zhou et al., 2018) or single images (Tucker & Snavely, 2020; Tulsiani et al., 2018). Inpainting, soft layering, and adaptive planes improve their flexibility (Shih et al., 2020; Jampani et al., 2021; Han et al., 2022), although denser plane sampling increases memory and rendering costs.

Other single-view methods use projected points (Wiles et al., 2020), generated support views (Rockwell et al., 2021), or continuous radiance and density volumes (Li et al., 2021; Wimbauer et al., 2023). Our representation retains the multi-surface premise through ordered Gaussians with learned 3D parameters, rather than global planes or separate image completion.

2.2 Gaussian scene reconstruction

3DGS (Kerbl et al., 2023) represents scenes with anisotropic Gaussians that support efficient rasterization. Feed-forward methods predict these primitives from image pairs (Charatan et al., 2024; Chen et al., 2024), learn Gaussian feature representations (Wewer et al., 2024), incorporate monocular depth features (Xu et al., 2025), or reconstruct from unposed images (Ye et al., 2025). Like wide-baseline neural renderers (Du et al., 2023), these methods exploit cross-view evidence.

For single-image reconstruction, Splatter Image (Szymanowicz et al., 2024) predicts pixel-aligned Gaussians for objects, while Flash3D (Szymanowicz et al., 2025) uses UniDepth (Piccinelli et al., 2024) to initialize ordered Gaussian layers for scenes. Niagara (Wu et al., 2026) combines depth and normals through a geometric affine field and 3D self-attention. CATSplat (Roh et al., 2025) adds textual and point-based spatial guidance, and MLGS (Diao et al., 2025) guides visible and occlusion layers with depth-derived masks and occlusion-specific supervision.

Our method retains Flash3D's cumulative depth anchors and independent decoding structure, and conditions both depth and Gaussian parameter prediction on complementary geometry at multiple feature scales. GCRA introduces this conditioning through lightweight, zero-initialized residual projections, rather than Niagara's geometric affine field and attention. Unlike MLGS, our layers share a reconstruction objective without assigned visibility roles or layer-specific losses, allowing their contributions to emerge during learning.

3 Method

Our approach reconstructs a Gaussian scene from a single image by jointly learning multiple surface hypotheses under geometric conditioning. As shown in Figure 2(a), GCRA adds geometric residuals to RGB-D features, and the resulting feature pyramid guides both depth-anchor and Gaussian parameter prediction in RSGM. During training, the network is optimized through differentiable rendering of the combined Gaussian scene.

3.1 Problem formulation and preliminaries

Given a source image I_s and camera intrinsics $\mathbf{K}$, we predict a scene $\mathcal{S} = \bigcup_{k=1}^{K} L_k$ with one Gaussian per source pixel in each layer. A frozen UniDepth model (Piccinelli et al., 2024) estimates $D_s = \Psi(I_s)$. A ResNet encoder (He et al., 2016) processes RGB and normalized depth to produce $E = \{E_i\}_{i=0}^{4}$ at spatial resolutions from $1/2$ to $1/32$ of the input resolution. GCRA supplies $R = \{R_i\}_{i=0}^{4}$, giving the decoder input $F = \{E_i + R_i\}_{i=0}^{4}$.

For source pixel p, decoder k predicts $P_k(p) = (\mathbf{o}_k(p), \alpha_k(p), \mathbf{s}_k(p), \mathbf{q}_k(p), \mathbf{c}_k(p), \mathbf{h}_k(p))$, specifying the camera-space position offset, opacity, scale, rotation, DC color, and remaining spherical-harmonic coefficients. Given depth anchor $D_k(p)$, the Gaussian center and covariance (Kerbl et al., 2023; Szymanowicz et al., 2025) are

$$\begin{aligned} \boldsymbol{\mu}_k(p) &= D_k(p)\mathbf{K}^{-1}\tilde{\mathbf{p}} + \mathbf{o}_k(p), \\ \boldsymbol{\Sigma}_k(p) &= \mathbf{Q}(\mathbf{q}_k(p))\,\mathrm{diag}(\mathbf{s}_k(p)^2)\mathbf{Q}(\mathbf{q}_k(p))^\top, \end{aligned} \tag{1}$$

where $\tilde{\mathbf{p}}$ is the homogeneous pixel coordinate and $\mathbf{Q}$ converts a quaternion into a rotation matrix.

For a target pixel u, the rasterizer sorts the overlapping Gaussians by target-view depth and composites their colors:

$$\hat{I}_t(u) = \sum_{j=1}^{M(u)} T_j(u) a_j(u) \mathbf{c}_j^t + T_{M(u)+1}(u)\mathbf{c}_{\mathrm{bg}}, \qquad T_j(u) = \prod_{m<j} \big(1 - a_m(u)\big). \tag{2}$$

Here $M(u)$ counts the Gaussians overlapping pixel u, and $a_j(u)$ combines the opacity of Gaussian j with its projected footprint. $\mathbf{c}_j^t$ is its view-dependent color, and $\mathbf{c}_{\mathrm{bg}}$ is the background color. The index j denotes rasterization order, not the source-layer index k.

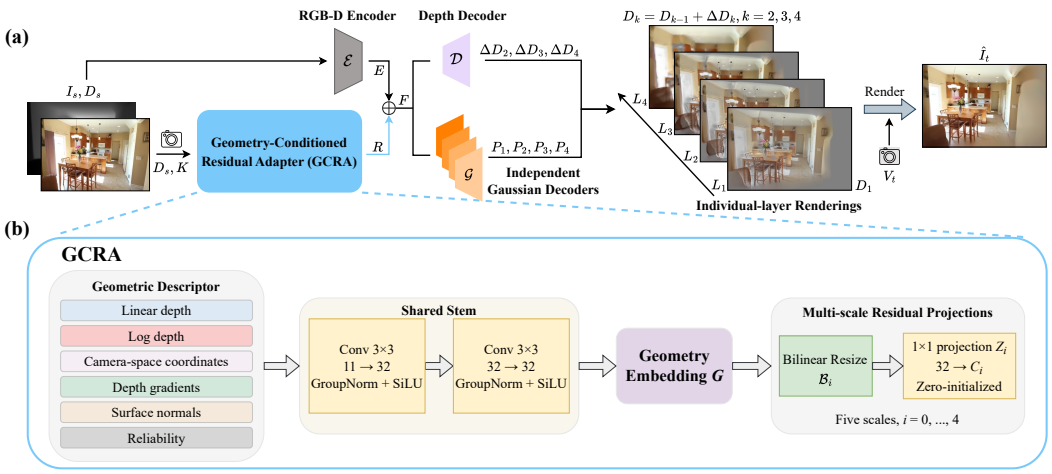

Figure 2: **Overview of GeoJMG.** (a) GCRA derives residuals R from D_s and $\mathbf{K}$ to condition RGB-D features E, yielding $F = E + R$. The depth decoder predicts $\Delta D_2, \Delta D_3, \Delta D_4$ to construct ordered depths from $D_1 = D_s$. Independent Gaussian decoders predict $P_1, \ldots, P_4$ for layers $L_1, \ldots, L_4$, illustrated by individual-layer renderings. The combined Gaussians produce $\hat{I}_t$ from viewpoint V_t. (b) Two convolutional blocks encode the 11-channel descriptor into G. Bilinear resizing $\mathcal{B}_i$ and zero-initialized 1×1 projections Z_i produce R_i at five scales. E, R, F denote feature pyramids, with $F_i = E_i + R_i$ for $i = 0, \ldots, 4$.

3.2 Ray-Stratified Gaussian Modeling (RSGM)

Ray-wise surface capacity. A source image constrains the first visible surface more directly than those behind it. RSGM retains K ordered hypotheses along each source ray to represent geometry and appearance revealed by camera motion. Independent Gaussian decoders predict distinct offsets, opacities, shapes, and colors from shared geometry-conditioned features. We use $K = 4$ to balance reconstruction quality and computational cost, as examined in Section 4.3.

Ordered anchors and independent parameters. The depth decoder $\mathcal{D}$ predicts raw log-increments r_k for $k = 2, \ldots, K$. Starting from $D_1(p) = D_s(p)$, we construct the remaining depths as

$$\begin{aligned} \Delta D_k(p) &= e^{\min\{6, \max\{-10, r_k(p)\}\}}, \\ D_k(p) &= D_{k-1}(p) + \Delta D_k(p), \qquad k = 2, \ldots, K. \end{aligned} \tag{3}$$

Following Flash3D (Szymanowicz et al., 2025), bounded positive increments preserve anchor ordering, but the offsets $\mathbf{o}_k$ can alter the order of the final centers. Rendering therefore uses actual target-view depth rather than layer index.

Each branch predicts $P_k = \mathcal{G}_k(F; \theta_k)$ for $k = 1, \ldots, K$, using the same conditioned pyramid and independent weights θ_k. Ordered anchors provide distinct reference positions, while separate decoders predict layer-specific shape, opacity, and appearance.

Joint optimization through rendering. The branches are jointly optimized through rendering. In Eq. (2), each Gaussian's contribution depends on the transmittance of those in front of it. For fixed footprints and opacities at a target pixel,

$$\frac{\partial \hat{I}_t(u)}{\partial \mathbf{c}_j^t} = T_j(u) a_j(u) \mathbf{I}_3, \tag{4}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix. Changing a foreground opacity also changes the transmittance and contributions of the Gaussians behind it. The common rendering objective therefore couples the layers through their projected overlap and visibility, allowing different contribution patterns to develop as training views expose different scene regions.

3.3 GEOMETRY-CONDITIONED RESIDUAL ADAPTER (GCRA)

GCRA conditions the surface hypotheses in RSGM on explicit local geometry. A raw depth channel supplies distance but leaves the encoder to infer spatial position, orientation, and depth variation. GCRA derives these quantities from the depth estimate and camera intrinsics, then combines them with image features at multiple resolutions.

Complementary geometric descriptors. Let $\mathbf{x}_p = D_s(p)\mathbf{K}^{-1}\tilde{\mathbf{p}}$ be the camera-space point corresponding to pixel p. The 11-channel descriptor is

$$g_p = \left[\bar{d}_p, \tilde{d}_p, \bar{\mathbf{x}}_p^\top, \partial_x \bar{d}_p, \partial_y \bar{d}_p, \mathbf{n}_p^\top, \rho_p\right]^\top. \tag{5}$$

We use normalized depth $\bar{d}_p = \min\{\tau_d, \max\{0, D_s(p)/d_0\}\}$, log-depth $\tilde{d}_p = \log(1 + \max\{D_s(p), 0\})/\log(1 + d_0)$, and coordinates $\bar{\mathbf{x}}_p = \min\{\tau_x, \max\{-\tau_x, \mathbf{x}_p/d_0\}\}$. Here d_0 is a reference depth, τ_d, τ_x bound the normalized values, and coordinate bounds are applied componentwise. Depth gradients use forward differences. The normal $\mathbf{n}_p$ is the normalized cross product of neighboring camera-space tangents, with its sign chosen so that its camera-space z component is nonnegative.

A final channel indicates depth validity and local smoothness:

$$\rho_p = v_p e^{-\beta_g\left(|\partial_x \bar{d}_p| + |\partial_y \bar{d}_p|\right)}, \tag{6}$$

where v_p is one at valid depths and zero otherwise. Invalid descriptor entries are set to zero.

The descriptors encode complementary geometric aspects. Linear depth encodes distance directly, while log-depth compresses its dynamic range. Camera-space coordinates locate surfaces in 3D. Gradients capture local depth variation, and normals describe surface orientation, helping distinguish slanted surfaces from occlusion boundaries. The reliability channel indicates depth validity and local smoothness, with lower values near sharp depth changes. Joint encoding supplies each branch with position, orientation, and boundary information to guide geometry and appearance reconstruction under viewpoint changes.

Shared geometry, scale-specific residuals. A shared stem $\mathcal{A}$ encodes the descriptor into $G = \mathcal{A}(g)$ using two 3×3 convolution–GroupNorm–SiLU blocks, as shown in Figure 2(b). Scale-specific projections then condition the image features:

$$R_i = Z_i(\mathcal{B}_i(G)), \qquad F_i = E_i + R_i, \qquad i = 0, \ldots, 4, \tag{7}$$

Here $\mathcal{B}_i$ bilinearly resamples G to the spatial resolution of E_i, and the zero-initialized 1×1 projection Z_i maps its 32 channels to the C_i channels of E_i, producing R_i. Further architectural details are provided in Appendix A.

Scale-specific projections adapt the shared geometry embedding to fine local structure and coarser spatial context. Both the depth decoder and every Gaussian branch consume F, aligning layer placement and Gaussian parameter prediction with the same geometric cues.

Feature-preserving initialization. All projection weights and biases start at zero, giving $R_i = 0$ and $F_i = E_i$ at initialization in Eq. (7). The adapter thus preserves the initial encoder mapping while learning residual geometric corrections during training. Section 4.3 evaluates the effects of multi-scale injection and zero-initialized projections.

3.4 TRAINING OBJECTIVE

For each supervised view $t \in \mathcal{T}$, the rasterizer produces $\hat{I}_t$ from the combined scene $\mathcal{S}$. The reconstruction objective uses an ℓ_1 photometric term, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018):

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left(\lambda_1 \mathcal{L}_1^{(t)} + \lambda_{\text{SSIM}} \mathcal{L}_{\text{SSIM}}^{(t)} + \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}}^{(t)}\right), \tag{8}$$

where $\mathcal{L}_{\text{SSIM}} = (1 - \text{SSIM})/2$. Scale and camera-space offset penalties constrain excessively large values:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{scale}} \mathcal{L}_{\text{scale}} + \lambda_{\text{offset}} \mathcal{L}_{\text{offset}}. \tag{9}$$

We jointly optimize the encoder, GCRA, depth decoder, and Gaussian branches with this objective while keeping the monocular depth model frozen.

Table 1: **Novel-view synthesis on RealEstate10K.**

Method	5 frames			10 frames			Random $\mathcal{U}[-30, 30]$ frames		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
SynSin	–	–	–	–	–	–	22.30	0.740	–
SV-MPI	27.10	0.870	–	24.40	0.812	–	23.52	0.785	–
BTS	–	–	–	–	–	–	24.00	0.755	0.194
Splatter Image	28.15	0.894	0.110	25.34	0.842	0.144	24.15	0.810	0.177
MINE	28.45	0.897	0.111	25.89	0.850	0.150	24.75	0.820	0.179
Flash3D	28.46	0.899	0.100	25.94	0.857	0.133	24.93	0.833	0.160
Niagara	28.50	0.899	0.097	25.96	0.858	0.129	24.98	0.834	0.157
GeoJMG	**29.09**	**0.907**	**0.091**	**26.43**	**0.871**	**0.125**	**25.51**	**0.849**	**0.150**

4 EXPERIMENTS

We evaluate novel-view synthesis and cross-domain generalization, then conduct component ablations and analyze layer-group contributions.

4.1 EXPERIMENTAL SETUP

We train on RealEstate10K (Zhou et al., 2018) for 60,000 Adam iterations with a batch size of 16, 256×384 images, and an ImageNet-pretrained encoder. Following Flash3D (Szymanowicz et al., 2025), we evaluate 5-frame, 10-frame, and random $\mathcal{U}[-30, 30]$ target offsets, as well as interpolation and extrapolation, reporting PSNR, SSIM, and LPIPS separately for each setting. We omit MLGS from same-protocol comparisons because the official code and checkpoints were unavailable, and CAT-Splat because unresolved compatibility issues in its released implementation prevented reliable evaluation. Further experimental details are provided in the supplementary material.

Table 2: **RealEstate10K interpolation and extrapolation.** Metrics are reported as PSNR/SSIM/LPIPS.

Method	Interpolation	Extrapolation
Two-view input		
pixelNeRF	20.51/0.592/0.550	20.05/0.575/0.567
Du *et al.*	24.78/0.820/0.213	21.83/0.790/0.242
pixelSplat	26.09/0.864/0.136	21.84/0.777/0.216
latentSplat	23.93/0.812/0.164	22.62/0.777/0.196
MVSplat	**26.39/0.869/0.128**	**23.04/0.813/0.185**
Single-view input		
Flash3D	23.87/0.811/0.185	24.10/0.815/0.185
Niagara	21.73/0.792/0.191	21.95/0.796/0.189
GeoJMG	**24.56/0.819/0.174**	**24.78/0.823/0.173**

4.2 COMPARISON WITH STATE-OF-THE-ART METHODS

RealEstate10K novel-view synthesis. The fixed and random target offsets span nearby views and larger camera displacements, allowing us to assess reconstruction as visual support from the input decreases. Across all three settings in Table 1, GeoJMG achieves the best overall performance. Its PSNR margin over Niagara reaches 0.59 dB, and its advantage persists under random target sampling, where it exceeds Flash3D by 0.58 dB. These results indicate that the geometry-conditioned layered representation improves synthesis for both neighboring views and targets that reveal content beyond the input observation.

Interpolation and extrapolation. Table 2 compares methods under interpolation and extrapolation protocols. Among single-view methods, GeoJMG performs best in both, improving PSNR over Flash3D by 0.69 dB for interpolation and 0.68 dB for extrapolation. MVSplat achieves higher interpolation accuracy with a second source view. In extrapolation, GeoJMG exceeds every listed two-view method on all three metrics. Although the input conditions differ, the extrapolation results indicate that the layered representation can reconstruct content weakly constrained by the source observation.

Cross-domain generalization. We assess transfer from RealEstate10K to NYUv2 and KITTI without fine-tuning. As reported in Table 3, GeoJMG outperforms both baselines across all three metrics on NYUv2, extending its reconstruction gains to a different indoor dataset. On KITTI, it matches Flash3D's PSNR at the reported precision and achieves higher SSIM and lower LPIPS, indicating

comparable pixel-level fidelity with improved structural and perceptual similarity. It also surpasses Niagara on all three metrics. These results demonstrate generalization to both indoor and outdoor scenes.

Qualitative comparison. Figure 3 shows differences in local image structures. Competing renderings blur fine patterns, propagate foreground colors across occlusion boundaries, or distort repeated thin structures. GeoJMG better preserves these regions and their surrounding appearance under viewpoint changes.

Table 3: **Cross-domain novel-view synthesis.**

Method	NYUv2	KITTI
Flash3D	25.09/0.775/0.182	**21.96**/0.826/0.132
Niagara	25.07/0.781/0.172	21.33/0.802/0.152
GeoJMG	**25.46/0.793/0.161**	**21.96/0.827/0.128**

4.3 Ablation studies

We first isolate the effects of layered capacity and geometric conditioning, then examine the structural choices within each component.

Core components. Table 4 compares the baseline with RSGM alone, GCRA alone, and their combination. Each component improves rendering quality across all target settings, and the complete model consistently outperforms both individual variants. These results show that expanded surface capacity and geometric conditioning provide complementary gains. Under random target sampling, the full model improves PSNR by 0.58 dB and reduces LPIPS by 0.010 relative to the baseline.

Table 4: **Core component ablation.**

Method	5 frames	10 frames	Random
Baseline	28.46/0.899/0.100	25.94/0.857/0.133	24.93/0.833/0.160
w/ RSGM	28.91/0.903/0.092	26.29/0.862/0.126	25.34/0.836/0.154
w/ GCRA	28.85/0.903/0.096	26.17/0.861/0.128	25.18/0.835/0.156
GeoJMG	**29.09/0.907/0.091**	**26.43/0.871/0.125**	**25.51/0.849/0.150**

RSGM design. The parameter-matched comparisons in Table 5 show that increased parameter count alone does not explain the improvement. Widening the two-branch decoder reduces reconstruction quality despite bringing its parameter count close to the four-layer model. At four layers, independent branches outperform the parameter-matched joint decoder on every metric while requiring 20.1% fewer FLOPs. These comparisons support the structured K-layer design and independent Gaussian parameter prediction over increasing network width alone.

Increasing K from 4 to 6 improves PSNR by only 0.04–0.07 dB, with similarly small changes in SSIM and LPIPS. Parameter count and FLOPs increase by 25.49% and 31.23%, respectively, while training time increases by 39.2%, as detailed in Appendix B. Given these diminishing returns, we use $K = 4$ to balance reconstruction quality and computational efficiency.

Table 5: **RSGM ablation.** Independent and joint denote separate and shared Gaussian decoders. Wide denotes widened branches. Metrics follow the order 5-frame / 10-frame / random $\mathcal{U}[-30, 30]$.

Model	Gaussians/ pixel	Decoder	Params (M)	FLOPs (G)	PSNR↑	SSIM↑	LPIPS↓
a	2	Independent	52.762	26.819	28.85 / 26.17 / 25.18	0.903 / 0.861 / 0.835	0.096 / 0.128 / 0.156
b	2	Independent (wide)	70.683	42.308	28.64 / 25.98 / 24.99	0.887 / 0.847 / 0.821	0.101 / 0.136 / 0.163
c	4	Joint	70.675	48.790	28.93 / 26.25 / 25.21	0.904 / 0.867 / 0.835	0.095 / 0.128 / 0.153
d	4	Independent	70.811	38.996	29.09 / 26.43 / 25.51	**0.907** / 0.871 / 0.849	**0.091** / **0.125** / 0.150
e	6	Independent	88.860	51.173	**29.13** / **26.47** / **25.58**	**0.907** / **0.872** / **0.851**	**0.091** / **0.125** / **0.149**

GCRA design. Table 6 evaluates geometric inputs under a fixed adapter architecture. Using depth together with log-depth improves reconstruction over the zero-input control, and the full descriptor further improves all three metrics across the target settings. Compared with depth and log-depth conditioning, the full descriptor increases PSNR by 0.09 dB and reduces LPIPS from 0.153 to 0.150 on random targets. Removing any descriptor group reduces PSNR in every setting, with decreases of 0.14–0.16 dB on random targets. These consistent reductions indicate that reconstruction benefits from combining distance, spatial position, surface orientation, local variation, and reliability.

With the full descriptor fixed, injecting geometry at multiple encoder scales yields higher PSNR than injection at the deepest scale alone, with a 0.17 dB gain on random targets. Zero initialization is also

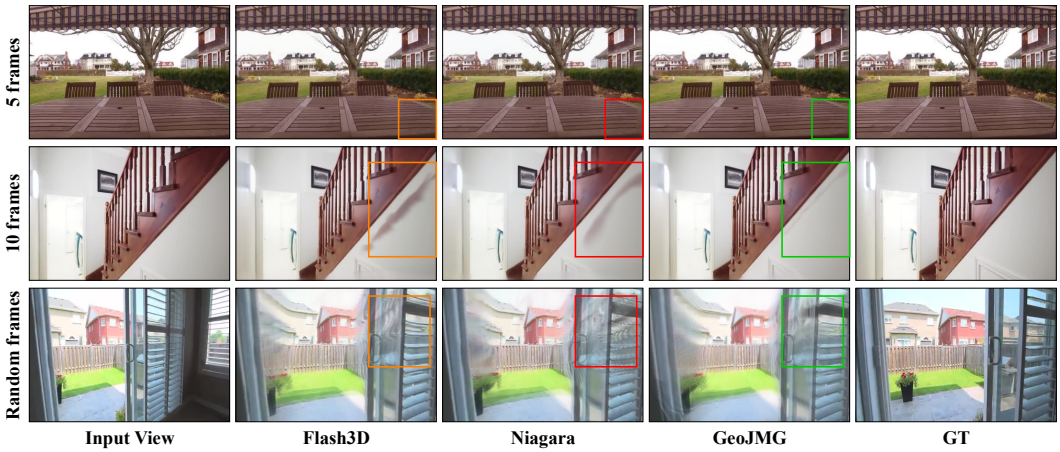

Figure 3: **Qualitative results on RealEstate10K.** GeoJMG achieves more faithful reconstruction across varying ranges of camera motion than Flash3D (Szymanowicz et al., 2025) and Niagara (Wu et al., 2026).

Table 6: **GCRA ablation.** XYZ denotes camera-space coordinates, and depth includes log-depth. Deepest scale only uses $1/32$ resolution. Random init. replaces zero initialization of the 1×1 projection weights and biases with PyTorch Conv2d defaults.

Configuration	5 frames	10 frames	Random frames
Geometry input			
Zero	29.01 / 0.897 / 0.095	26.34 / 0.863 / 0.128	25.26 / 0.838 / 0.153
Depth + log-depth	29.03 / 0.901 / 0.093	26.37 / 0.867 / 0.126	25.42 / 0.841 / 0.153
Full geometry	**29.09 / 0.907 / 0.091**	**26.43 / 0.871 / 0.125**	**25.51 / 0.849 / 0.150**
Descriptor-group removal			
w/o depth	29.07 / 0.904 / 0.094	26.36 / 0.863 / 0.127	25.35 / 0.835 / 0.157
w/o XYZ	29.08 / 0.905 / 0.093	26.37 / 0.864 / 0.126	25.37 / 0.837 / 0.154
w/o gradients	29.07 / 0.906 / 0.094	26.36 / 0.862 / 0.127	25.36 / 0.835 / 0.156
w/o normals	29.06 / 0.905 / 0.093	26.36 / 0.863 / 0.126	25.35 / 0.837 / 0.154
w/o reliability	29.08 / 0.905 / 0.095	26.37 / 0.863 / 0.126	25.37 / 0.836 / 0.155
Adapter design (full geometry input)			
Deepest scale only	29.04 / 0.904 / 0.094	26.35 / 0.863 / 0.126	25.34 / 0.837 / 0.155
Random init.	28.67 / 0.902 / 0.095	26.11 / 0.862 / 0.128	25.11 / 0.833 / 0.157

beneficial, since random initialization of the projection layers reduces PSNR by 0.32–0.42 dB across all three settings. Initializing the residuals to zero preserves the encoder's feature mapping at the start of training, after which the adapter learns geometric corrections through the residual pathway.

4.4 Layer-group contribution analysis

We analyze the learned layer-group contributions using removal statistics in Table 7 and spatial error maps in Figure 4. These removals quantify each group's overall influence and how that influence varies across visibility regions.

Global removal sensitivity. Jointly removing L2 and L3 lowers full-image PSNR by 3.25–5.27 dB, more than removing L1 or L4 individually. All three removals reduce PSNR in every setting, indicating that the model uses each group, with the middle pair contributing most in aggregate.

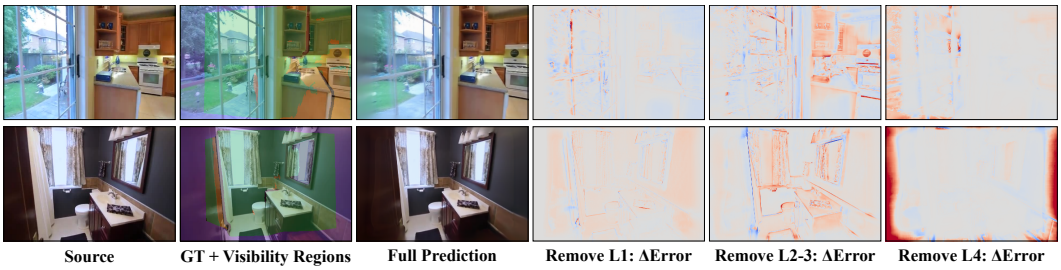

Figure 4: **Spatial effects of layer-group removal.** Rows show 10-frame and random targets. Green, orange, and purple mark source-covisible, disoccluded, and out-of-frustum regions, respectively. L1, L2–3, and L4 denote the first, middle two, and fourth layers. The last three columns show $\Delta\text{Error} = |I_{\text{remove}} - I_{\text{GT}}| - |I_{\text{full}} - I_{\text{GT}}|$, using removed-layer and full-model renderings. All maps use $[-0.25, 0.25]$, with warm/cool colors indicating increased/decreased error.

Visibility-dependent contributions. The regional contrasts show different tendencies for the outer groups. Removing L1 degrades source-covisible regions more than disoccluded regions, whereas L4 contributes more strongly to disoccluded content. Removing L4 causes a larger PSNR drop in out-of-frustum than covisible regions for 10-frame and random targets, but the opposite holds for 5-frame targets (-0.219 dB). These results indicate visibility-dependent contributions, consistent with the rendering weights in Eq. (4), rather than a fixed partition of scene content among layers.

The spatial error maps in Figure 4 clarify these statistics. Removing L2–3 produces widespread changes across scene structures, consistent with its dominant global sensitivity. Removing L1 causes more concentrated changes in source-covisible content, whereas removing L4 affects newly exposed regions more strongly, particularly near out-of-frustum areas in the random-view example. The regional statistics and spatial maps therefore show differentiated but overlapping layer-group contributions.

Table 7: **Layer-group removal sensitivity.** (a) Full-image PSNR drops. (b) Differences in pooled regional PSNR drops. All values are in dB. Cov., Dis., and OoF denote covisible, disoccluded, and out-of-frustum regions, respectively. Disoccluded regions include out-of-frustum content. Positive values in (b) indicate a larger drop in the first listed region.

(a) Global PSNR drop

Target	L1	L2–3	L4
5 frames	1.386	**5.266**	1.225
10 frames	0.858	**3.589**	1.039
Random	0.977	**3.246**	1.046

(b) Regional contrast

Target	L1 Cov.−Dis.	L4 Dis.−Cov.	L4 OoF−Cov.
5 frames	+0.155	+0.106	-0.219
10 frames	+0.216	+0.332	+0.610
Random	+0.359	+0.533	+0.663

5 Conclusion

We presented GeoJMG, a feed-forward framework that combines layered Gaussian representations with geometric conditioning for single-view scene reconstruction. RSGM represents each source ray with ordered surface hypotheses and independent Gaussian parameter prediction, providing additional capacity for content exposed by camera motion. GCRA guides both layer depths and Gaussian parameters with complementary geometric cues through multi-scale, zero-initialized residual projections. Visibility-dependent rendering couples the layers under a common reconstruction objective. Experiments demonstrate improved novel-view synthesis on RealEstate10K and generalization to NYUv2 and KITTI without fine-tuning. Controlled ablations support both the structured layer organization and the complementary geometric descriptor, while layer-group removal reveals differentiated but overlapping contributions. Future work will investigate content-adaptive Gaussian allocation to concentrate capacity in weakly observed regions while reducing computational cost.

AI USE STATEMENT

We used generative AI tools to assist with drafting, restructuring, and polishing the manuscript, including translation and LaTeX formatting. We also used them for literature retrieval and discovery, summarizing related work, and organizing references. In the research workflow, these tools assisted with developing and refining ideas, designing ablations, and implementing and debugging experimental code. The authors are responsible for the final text, methodological claims, code, and reported experimental results.

REFERENCES

D. Charatan, S. L. Li, A. Tagliasacchi, and V. Sitzmann. pixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 19457–19467, 2024.

A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 14124–14133, 2021.

Y. Chen, H. Xu, C. Zheng, B. Zhuang, M. Pollefeys, A. Geiger, T.-J. Cham, and J. Cai. MVSplat: Efficient 3D Gaussian splatting from sparse multi-view images. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 370–386, 2024.

S. Diao, Y. Zhao, Y. Chen, Z. Zhang, W. Jia, and R. Wang. Multi-layer Gaussian splatting for single-image feed-forward spatial scene reconstruction. In *Proc. ACM Int. Conf. Multimedia (MM)*, pp. 3751–3759, 2025.

Y. Du, C. Smith, A. Tewari, and V. Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4970–4980, 2023.

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.*, 32(11):1231–1237, 2013.

Y. Han, R. Wang, and J. Yang. Single-view view synthesis in the wild with learned adaptive multi-plane images. In *Proc. ACM SIGGRAPH Conf.*, pp. 1–8, 2022.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, 2016.

V. Jampani, H. Chang, K. Sargent, A. Kar, R. Tucker, M. Krainin, D. Kaeser, W. T. Freeman, D. Salesin, B. Curless, and C. Liu. SLIDE: Single image 3D photography with soft layering and depth-aware inpainting. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 12518–12527, 2021.

M. M. Johari, Y. Lepoittevin, and F. Fleuret. GeoNeRF: Generalizing NeRF with geometry priors. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 18365–18375, 2022.

B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), 2023. Article 139.

J. Li, Z. Feng, Q. She, H. Ding, C. Wang, and G. H. Lee. MINE: Towards continuous depth MPI with NeRF for novel view synthesis. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 12578–12588, 2021.

A. Liu, R. Tucker, V. Jampani, A. Makadia, N. Snavely, and A. Kanazawa. Infinite Nature: Perpetual view generation of natural scenes from a single image. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 14458–14467, 2021.

B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 405–421, 2020.

L. Piccinelli, Y.-H. Yang, C. Sakaridis, M. Segu, S. Li, L. Van Gool, and F. Yu. UniDepth: Universal monocular metric depth estimation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 10106–10116, 2024.

C. Rockwell, D. F. Fouhey, and J. Johnson. PixelSynth: Generating a 3D-consistent experience from a single image. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 14104–14113, 2021.

W. Roh, H. Jung, J. W. Kim, S. Lee, I. Yoo, A. Lugmayr, S. Chi, K. Ramani, and S. Kim. CATSplat: Context-aware transformer with spatial guidance for generalizable 3D Gaussian splatting from a single-view image. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 28228–28238, 2025.

K. Sargent, Z. Li, T. Shah, C. Herrmann, H.-X. Yu, Y. Zhang, E. R. Chan, D. Lagun, L. Fei-Fei, D. Sun, and J. Wu. ZeroNVS: Zero-shot 360-degree view synthesis from a single image. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 9420–9429, 2024.

J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. In *Proc. 25th Annu. Conf. Comput. Graph. Interactive Techn. (SIGGRAPH)*, pp. 231–242, 1998.

M.-L. Shih, S.-Y. Su, J. Kopf, and J.-B. Huang. 3D photography using context-aware layered depth inpainting. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 8028–8038, 2020.

N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 746–760, 2012.

S. Szymanowicz, C. Rupprecht, and A. Vedaldi. Splatter Image: Ultra-fast single-view 3D reconstruction. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 10208–10217, 2024.

S. Szymanowicz, E. Insafutdinov, C. Zheng, D. Campbell, J. F. Henriques, C. Rupprecht, and A. Vedaldi. Flash3D: Feed-forward generalisable 3D scene reconstruction from a single image. In *Proc. Int. Conf. 3D Vis. (3DV)*, pp. 670–681, 2025.

R. Tucker and N. Snavely. Single-view view synthesis with multiplane images. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 551–560, 2020.

S. Tulsiani, R. Tucker, and N. Snavely. Layer-structured 3D scene inference via view synthesis. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 311–327, 2018.

Q. Wang, Z. Wang, K. Genova, P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser. IBRNet: Learning multi-view image-based rendering. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4690–4699, 2021.

S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. DUSt3R: Geometric 3D vision made easy. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 20697–20709, 2024.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

C. Wewer, K. Raj, E. Ilg, B. Schiele, and J. E. Lenssen. latentSplat: Autoencoding variational Gaussians for fast generalizable 3D reconstruction. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 456–473, 2024.

O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson. SynSin: End-to-end view synthesis from a single image. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 7467–7477, 2020.

F. Wimbauer, N. Yang, C. Rupprecht, and D. Cremers. Behind the scenes: Density fields for single view reconstruction. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 9076–9086, 2023.

X. Wu, Z. Ai, H. Yang, S. Lim, J. Liu, and H. Wang. Niagara: Normal-integrated geometric affine field for scene reconstruction from a single view. *IEEE Trans. Circuits Syst. Video Technol.*, 36 (5):6866–6876, 2026.

Y. Wu and K. He. Group normalization. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 3–19, 2018.

H. Xu, A. Chen, Y. Chen, C. Sakaridis, Y. Zhang, M. Pollefeys, A. Geiger, and F. Yu. MuRF: Multi-baseline radiance fields. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 20041–20050, 2024.

H. Xu, S. Peng, F. Wang, H. Blum, D. Barath, A. Geiger, and M. Pollefeys. DepthSplat: Connecting Gaussian splatting and depth. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 16453–16463, 2025.

B. Ye, S. Liu, H. Xu, X. Li, M. Pollefeys, M.-H. Yang, and S. Peng. No pose, no problem: Surprisingly simple 3D Gaussian splats from sparse unposed images. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.

A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4578–4587, 2021.

R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 586–595, 2018.

T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4), 2018. Article 65.

SUPPLEMENTARY MATERIAL

This supplement provides additional experimental details in Appendix A and computational efficiency measurements in Appendix B. Appendix C compares reconstruction accuracy across visibility regions using shared masks. Appendix D presents further qualitative results, and Appendix E discusses limitations and future directions.

A ADDITIONAL EXPERIMENTAL DETAILS

A.1 DATASETS AND EVALUATION PROTOCOLS

RealEstate10K. We use the RealEstate10K splits (Zhou et al., 2018) and evaluation protocols adopted by Flash3D (Szymanowicz et al., 2025). Within each of the 5-frame, 10-frame, and random $\mathcal{U}[-30, 30]$ settings, all methods use the same 2,960 available source-target pairs. Images are evaluated at 256×384, with the outer 5% removed on each side before computing PSNR, SSIM, and LPIPS. Interpolation and extrapolation use the prescribed source-target pairs at 256×256 without border cropping, with 18,162 and 18,015 valid examples, respectively. All comparisons use a common valid subset within each protocol.

Cross-domain evaluation. Models trained on RealEstate10K are evaluated on NYUv2 (Silberman et al., 2012) and KITTI (Geiger et al., 2013) without fine-tuning. NYUv2 uses targets sampled within $[-30, 30]$ frames of the source and contains 231 valid examples after excluding missing images or unavailable poses. KITTI follows the split of Tulsiani et al. (2018), with 1,079 source images and synchronized stereo targets. Both datasets use 256×384 images and the Flash3D evaluation protocol.

A.2 TRAINING AND MODEL SETTINGS

We train with Adam for 60,000 iterations at a learning rate of 10^{-4} and a batch size of 16, using FP32 on one RTX 4090 GPU. The ResNet-50 encoder is initialized from ImageNet, the GCRA output projections are initialized to zero, and the remaining trainable modules are initialized randomly. Experiments use seeds 42, 45, and 47, and reconstruction metrics are reported as means over the three runs. Evaluation uses the exponential moving average of the model parameters.

For each source image, we sample three distinct nonzero temporal offsets from $[-30, 30]$ and supervise the corresponding target views together with the source view. Offsets beyond a sequence boundary are clipped to the nearest valid frame. The weights of the ℓ_1, SSIM, and LPIPS terms in Eq. (8) are 1.0, 0.85, and 0.01, respectively, with LPIPS enabled after 50,000 steps. The scale and offset regularizers in Eq. (9) have weights of 0.001 and 0.01. The scale penalty averages components exceeding 2 m, while the offset penalty averages squared components whose absolute values exceed 1 m; either term is zero when no component meets its threshold.

The GCRA geometry stem contains two 3×3 convolutions with 32 hidden channels, each followed by GroupNorm with eight groups (Wu & He, 2018) and SiLU. The five residual projections match encoder features with downsampling factors and channel counts $(s_i, C_i) \in \{(2, 64), (4, 256), (8, 512), (16, 1024), (32, 2048)\}$, where the spatial resolution at scale i is $1/s_i$ of the input resolution. For GCRA, the reference depth is $d_0 = 20$ m, the normalized depth and coordinate bounds are $\tau_d = \tau_x = 5$, and the reliability coefficient is $\beta_g = 10$. Invalid or non-finite descriptor values are set to zero. Projection initialization follows Section 3.3.

B COMPUTATIONAL EFFICIENCY

We measure Flash3D, Niagara, and the four- and six-layer variants of GeoJMG locally under the same input resolution, precision, and timing scope. Inference uses FP32, batch size one, and 256×384 inputs on an RTX 4090, with 20 warm-up iterations followed by 100 timed iterations. Timing starts from cached geometric priors and prepared camera poses and includes the core reconstruction network and rasterization of one target view. External depth and normal estimation and pose preprocessing are excluded. Parameter and operation counts cover the core reconstruction net-

Table 8: **Computational cost.** Measurements on one RTX 4090 for 60K-step training and batch-one inference. Inference includes core reconstruction and one target-view rendering, excluding external prior estimation. One MAC counts as one operation.

Method	Training (h)	Inference (s)	Params (M)	FLOPs (G)
Flash3D	14.786	0.0058	52.621	25.485
Niagara	12.833	0.0082	75.048	46.945
GeoJMG ($K = 4$)	20.376	0.0086	70.811	38.996
GeoJMG ($K = 6$)	28.362	0.0106	88.860	51.173

work; the operation count includes convolutional and linear layers, with one multiply-accumulate counted as one operation.

The measurements in Table 8 reveal different quality–cost trade-offs. **The four-layer GeoJMG model uses a smaller core network and fewer operations than Niagara while maintaining similar inference latency.** Its training time is longer, however, and Flash3D remains less expensive at inference. Relative to Flash3D, the reconstruction gains thus come with additional computation rather than a reduction in cost.

Increasing the number of independent layers from four to six raises the parameter count and operation count by 25.49% and 31.23%, respectively. Training time increases by 39.2%, from 20.376 to 28.362 h, and inference latency also increases, as reported in Table 8. These additional costs yield only a small improvement in reconstruction quality in Table 5. We therefore retain four layers as the main configuration to balance reconstruction quality and computational cost.

C Reconstruction in visibility regions

This evaluation directly compares GeoJMG, Flash3D, and Niagara on target pixels with different visibility relative to the source image. It complements the full-image results by measuring reconstruction accuracy on surfaces revealed by occlusion changes and on content outside the source field of view.

C.1 Shared masks and regional metrics

Each target setting uses 2,960 source-target pairs from 592 scenes. All methods share the camera poses and visibility masks. The masks are constructed from source and target metric depths predicted by a frozen UniDepth model, independently of the reconstruction being evaluated. For a target pixel p, we back-project its estimated depth and transform the point into the source camera:

$$\mathbf{X}_t(p) = \hat{D}_t(p)\mathbf{K}_t^{-1}\tilde{\mathbf{p}}, \qquad \mathbf{X}_s(p) = \mathbf{R}_{s\leftarrow t}\mathbf{X}_t(p) + \mathbf{t}_{s\leftarrow t}. \tag{10}$$

Projection into the source image gives $\mathbf{u}_s(p) = \pi(\mathbf{K}_s\mathbf{X}_s(p))$, where π denotes perspective division. Let $d_s(p)$ be the source depth sampled at $\mathbf{u}_s(p)$, and let $\epsilon(p) = 0.05\ \mathrm{m} + 0.05|d_s(p)|$ be the depth-consistency tolerance.

For valid, in-frustum projections away from depth edges, a pixel is *covisible* if $|X_s^z(p) - d_s(p)| \leq \epsilon(p)$ and belongs to *in-frustum disocclusion* if $X_s^z(p) > d_s(p) + \epsilon(p)$. The latter identifies target-visible surfaces behind the surface observed in the source image. The *out-of-frustum* mask contains valid target pixels projected outside the source image or behind the source camera. The full *disoccluded* mask is the union of in-frustum disocclusion and out-of-frustum pixels and therefore overlaps with both constituent regions.

We use nearest-neighbor depth sampling and exclude invalid or unresolved depth comparisons. Depth edges are identified by adjacent valid depths differing by more than 5% of the smaller value and are excluded from covisible and in-frustum disocclusion masks. We omit a two-pixel image boundary during mask construction and apply the central 90% evaluation crop before aggregation. The resulting regional coverage is summarized in Table 11.

Table 9: **Reconstruction accuracy by visibility region.** PSNR (dB) under identical masks, test pairs, and camera poses. Bold marks the best result for each setting and region.

Target setting	Method	Covisible	Disoccluded	In-frustum disocclusion	Out-of-frustum
5 frames	Flash3D	27.408	27.083	27.749	13.155
	Niagara	27.319	26.981	27.614	**13.248**
	GeoJMG	**27.625**	**27.276**	**27.974**	13.170
10 frames	Flash3D	25.193	24.296	24.921	16.059
	Niagara	25.161	24.241	24.850	**16.094**
	GeoJMG	**25.349**	**24.411**	**25.064**	16.030
Random $\mathcal{U}[-30, 30]$ frames	Flash3D	24.446	21.094	23.581	15.516
	Niagara	24.413	21.098	23.523	**15.579**
	GeoJMG	**24.597**	**21.210**	**23.773**	15.562

For region r, we pool squared RGB errors over valid regional pixels before computing PSNR:

$$\mathrm{PSNR}_r = -10 \log_{10} \left(\frac{\sum_i \sum_{p \in r_i} \sum_{c=1}^{3} (\hat{I}_{ipc} - I_{ipc})^2}{3 \sum_i |r_i|} \right). \tag{11}$$

This weights images by their regional pixel counts; empty masks contribute neither errors nor pixels. The same masks and aggregation are used for every method.

C.2 Accuracy on visible and newly revealed surfaces

The comparison in Table 9 uses identical visibility masks and shows that **GeoJMG achieves the highest PSNR in covisible, disoccluded, and in-frustum disocclusion regions across all three target settings**. The gains extend beyond source-visible content to surfaces revealed by a change in viewpoint.

Separating in-frustum disocclusion from out-of-frustum content isolates surfaces hidden by foreground occluders within the source view. The consistent advantage over both baselines in this region directly demonstrates more accurate reconstruction of newly revealed surfaces, complementing the full-image results in the main paper.

The improvement is also evident at the image level: GeoJMG reduces disoccluded-region mean squared error on more than 70% of valid images against either baseline in every target setting. These comparisons evaluate the complete method and complement the component ablations in Table 4.

C.3 Statistical support and out-of-frustum results

We estimate paired 95% confidence intervals by resampling the 592 scenes with replacement 10,000 times, retaining all samples from each selected scene and using identical draws for the compared methods. The 2.5th and 97.5th percentiles give the interval bounds. These intervals quantify test-scene sampling uncertainty. All intervals for disoccluded and in-frustum disocclusion gains reported in Table 10 lie above zero, indicating that the improvements over both baselines are robust to test-scene resampling.

Out-of-frustum differences are smaller, and their confidence intervals include zero in all three settings. Although GeoJMG lowers the regional error on a majority of valid images against either baseline, the pooled results do not establish a consistent advantage. As the coverage statistics in Table 11 indicate, these regions occupy only a small fraction of test pixels, particularly for fixed offsets. The strongest evidence of improved reconstruction therefore concerns newly revealed surfaces within the source frustum.

Table 10: **Paired regional PSNR gains.** Differences (dB) are GeoJMG minus the baseline, computed before rounding. Brackets indicate 95% scene-bootstrap confidence intervals. Each pair uses identical non-empty masks.

Target setting	Compared with	Disoccluded	In-frustum disocclusion	Out-of-frustum
5 frames	Flash3D	$+0.193$ $[+0.145, +0.254]$	$+0.225$ $[+0.182, +0.272]$	$+0.015$ $[-0.121, +0.216]$
	Niagara	$+0.295$ $[+0.213, +0.394]$	$+0.360$ $[+0.299, +0.428]$	-0.079 $[-0.246, +0.153]$
10 frames	Flash3D	$+0.116$ $[+0.085, +0.149]$	$+0.142$ $[+0.112, +0.175]$	-0.029 $[-0.135, +0.060]$
	Niagara	$+0.171$ $[+0.133, +0.213]$	$+0.214$ $[+0.174, +0.256]$	-0.064 $[-0.149, +0.003]$
Random $\mathcal{U}[-30, 30]$ frames	Flash3D	$+0.116$ $[+0.087, +0.149]$	$+0.192$ $[+0.164, +0.221]$	$+0.046$ $[-0.005, +0.106]$
	Niagara	$+0.112$ $[+0.075, +0.150]$	$+0.250$ $[+0.213, +0.288]$	-0.017 $[-0.076, +0.047]$

Table 11: **Visibility-region coverage.** Statistics for 2,960 test pairs from 592 scenes. Pixel fractions use all cropped test pixels as the denominator. Valid images and scenes contain at least one regional pixel. Disoccluded includes both in-frustum and out-of-frustum disocclusion.

Target setting	Region	Mean pixel fraction (%)	Valid images	Valid scenes
5 frames	Covisible	49.015	2,806	588
	Disoccluded	34.155	2,721	584
	In-frustum disocclusion	33.951	2,694	580
	Out-of-frustum	0.204	205	89
10 frames	Covisible	46.682	2,828	588
	Disoccluded	35.939	2,819	590
	In-frustum disocclusion	35.108	2,767	586
	Out-of-frustum	0.831	604	206
Random $\mathcal{U}[-30, 30]$ frames	Covisible	43.957	2,852	589
	Disoccluded	38.795	2,830	591
	In-frustum disocclusion	33.247	2,753	589
	Out-of-frustum	5.548	1,339	515

D Additional qualitative results

Layer-group removal. As illustrated in Figure 5, removing different layer groups changes both local structure and newly revealed content across the three target settings. These examples extend Figure 4 using the same opacity-zeroing procedure and include a 5-frame boundary case. Removing L2–3 measures the aggregate effect of two layers, not their individual importance relative to L1 or L4.

Comparison with Flash3D and Niagara. The six scenes in Figure 6, two per target setting, cover object boundaries, depth discontinuities, reflective or weakly textured surfaces, and newly exposed content near image borders. **GeoJMG better preserves local structure and reduces boundary and color-bleeding artifacts in these examples.** These visual differences complement the regional comparisons in Appendix C.

E Limitations and future work

A single input image cannot directly constrain every surface visible from a novel viewpoint. Recovering fine detail remains difficult under large viewpoint changes, particularly when content lies outside the source field of view. Gaussian placement and GCRA conditioning also depend on the accuracy of the monocular depth prior near ambiguous boundaries. Finally, assigning four Gaussians to every source pixel may allocate more capacity than necessary in simple regions. Future work will investigate content-adaptive allocation and pruning, together with improved geometric priors, to better balance reconstruction quality and computational cost.

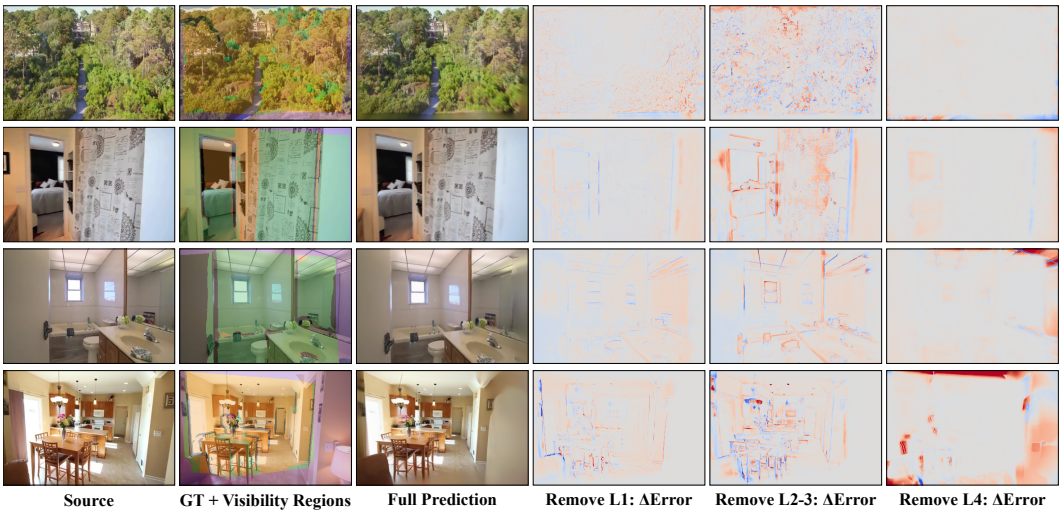

Figure 5: **Additional layer-group removal examples.** The first row is a 5-frame boundary case. Region colors and error maps follow Figure 4, with warm/cool colors indicating increased/decreased error on the same $[-0.25, 0.25]$ scale.

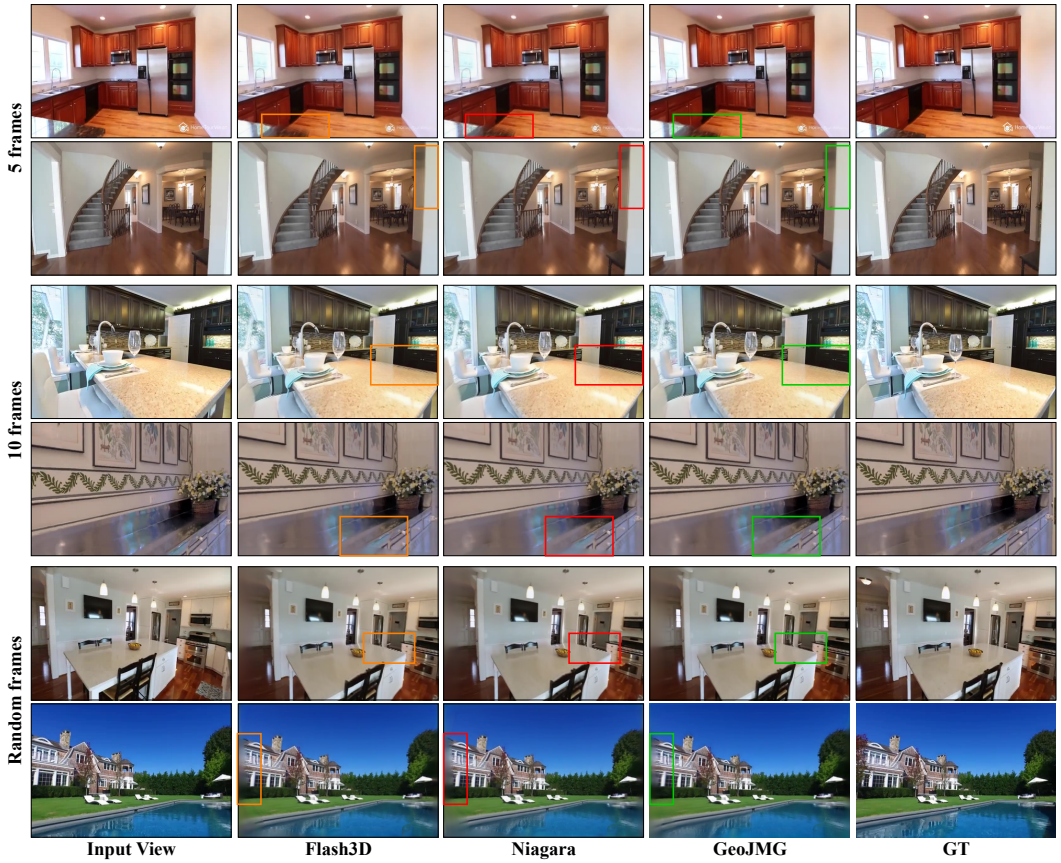

Figure 6: **Additional qualitative comparisons on RealEstate10K.** Two examples per target setting. Boxes highlight local reconstruction differences.