GEOSUPSPLAT: GENERALIZABLE SPARSE-VIEW SURFACE RECONSTRUCTION VIA GEOMETRY-SUPERVISED GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward Gaussian splatting enables fast reconstruction from sparse images, but photorealistic rendering could not yield an accurate surface. Geometry supervision in this setting faces two coupled challenges: pseudo-depth and other pretrained cues may be ambiguous or inconsistent across views, while the opacity used for RGB compositing does not always represent geometric visibility. We present **GeoSupSplat**, a feed-forward framework for reconstructing surfaces from two sparse images using 2D Gaussian surfels. Our key insight is that appearance visibility should not fully determine geometric confidence. GeoSupSplat therefore develops a weakly decoupled Gaussian representation which assigning each surfel separate appearance and geometry opacities, allowing RGB and depth to be composited independently while sharing the same center, scale, and orientation. Furthermore, we propose a set of novel geometry supervision techniques that enable the model to learn high-quality geometry. These include scale-invariant depth, depth-to-normal, and keypoint-based 3D consistency losses, which transfer relative depth structure, local surface orientation, and cross-view agreement without imposing the teacher's unknown scale. We evaluate our model on the Re10K dataset and two zero-shot cross-dataset benchmarks: ScanNet and Replica. Our model outperforms state-of-the-art methods in mesh quality and mesh-rendering tasks.

1 INTRODUCTION

Reconstructing a surface from a sparse set of calibrated images is a fundamental problem in 3D vision, with applications in robotics, immersive telepresence, and digital content creation. Feed-forward reconstruction addresses this setting by learning geometric priors that transfer to unseen scenes in a single pass. Existing feed-forward approaches follow two main directions. Generalizable implicit methods predict occupancy or signed-distance fields from multi-view features Long et al. (2022); Ren et al. (2023); Peng et al. (2023); Xu et al. (2023); Na et al. (2024). They provide continuous surfaces but require repeated field queries during rendering and extraction. Gaussian methods instead predict explicit primitives that can be rendered efficiently by splatting Charatan et al. (2024); Chen et al. (2024). Recent extensions introduce geometry-aware Gaussians, surfels, or stronger geometric priors Yang et al. (2025); Chang et al. (2026); Dai et al. (2025); Shi et al. (2025). Yet these methods inherit a basic mismatch: Gaussian representations are optimized primarily for image formation, whereas surface reconstruction requires reliable geometric visibility and supervision. Predicting a surface-aligned primitive alone does not resolve this mismatch.

We present **GeoSupSplat**, a feed-forward 2D Gaussian-splatting framework for sparse-view surface reconstruction. Unlike prior optimization-based approaches, GeoSupSplat learns pixel-aligned Gaussian surfels across scenes and reconstructs a new scene from only two input views in a single forward pass. As illustrated in Fig. 1, the model renders color and geometry from one shared scene representation. To provide reliable geometric supervision without forcing RGB and depth to share the same compositing confidence, we combine a weakly decoupled gaussian representation with geometry supervision designed specifically for the resulting depth-rendering path.

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.

First, GeoSupSplat develops a weakly decoupled Gaussian representation with two separate prediction heads: the appearance head predicts an appearance opacity α for RGB rendering, whereas the geometry head predicts a geometry opacity β for geometry rendering. Both paths share the surfel center, scale, and orientation, while α composites color and β composites expected depth. The model therefore retains one physical surface while allowing each opacity to represent task-specific visibility.

With a dedicated geometry-rendering path in place, the remaining issue is how to supervise it without inheriting ambiguities from pretrained teachers. Pseudo-depth may contain the correct scene structure at an unknown scale. An independently predicted normal map may disagree with that depth. And per-view losses do not ensure that corresponding pixels reconstruct the same 3D point. To address these issues, we introduce three complementary objectives. A scale-invariant depth loss transfers relative depth without introducing an independent depth offset for each view. A depth-to-normal loss applies the same geometric operator to rendered and pseudo-depth, providing local orientation supervision without a separate normal teacher. Finally, a keypoint-based 3D consistency loss unprojects matched pixels and brings them to the same position in the shared scene frame. Together, these losses supervise relative structure, local orientation, and cross-view consistency.

We evaluate GeoSupSplat on surface reconstruction and NVS quality under mesh rendering. On Re10K with two context views, it reduces CD from 0.536 to 0.459 under the same TSDF protocol used by the baselines, while increasing mesh-rendering PSNR from 18.52 to 19.68 dB and reducing LPIPS from 0.242 to 0.234 relative to the best baseline for each metric. Without fine-tuning, GeoSupSplat also improves reconstruction on ScanNet and Replica. Controlled ablations further validate opacity decoupling and each geometry objective.

Our contributions are: **1)** We develop a feed-forward 2D Gaussian surfel framework with a weakly decoupled representation for sparse-view surface reconstruction. **2)** We establish a set of novel geometry supervision for sparse-view surface reconstruction via 2D Gaussian splatting. **3)** Our method achieve state-of-the-art performance across Re10K dataset and zero-shot cross-dataset benchmarks: ScanNet and Replica.

2 Related Work

2.1 Generalizable Sparse-View Surface Reconstruction

Neural implicit methods represent a surface as the level set of an occupancy or signed distance field Wang et al. (2021). Generalizable variants replace per-scene optimization with priors learned from multi-view image features. SparseNeuS constructs a geometry volume and predicts a conditional signed distance field Long et al. (2022); VolRecon models a signed ray-distance field for efficient volume rendering Ren et al. (2023); and GenS, C2F2NeuS, and ReTR improve cross-view feature fusion or rendering Peng et al. (2023); Xu et al. (2023); Liang et al. (2023). UFORecon further targets difficult input-view configurations through uncertainty-aware fusion Na et al. (2024). These methods produce continuous surfaces, but they repeatedly query an implicit field during rendering and extraction. GeoSupSplat instead predicts an explicit surfel representation that can be rendered by splatting and converted directly into a mesh.

2.2 Gaussian Splatting for Surface Reconstruction

3D Gaussian Splatting represents a radiance field with explicit anisotropic primitives and differentiable rasterization Kerbl et al. (2023). Because volumetric Gaussians optimized for appearance need not align with a surface, subsequent methods introduce surface-aware representations or regularization. SuGaR aligns flattened Gaussians with an implicit surface Guédon & Lepetit (2024); 2D Gaussian Splatting and Gaussian Surfels replace ellipsoids with oriented surfels that approximate local tangent planes Huang et al. (2024); Dai et al. (2024); Gaussian Opacity Fields extract surfaces from a learned opacity field Yu et al. (2024b); and GSDF couples Gaussian and signed distance representations Yu et al. (2024a). These approaches recover detailed geometry from dense observations, but generally optimize a separate representation for each scene. Our work adopts surface-aligned 2D Gaussians in a feed-forward setting and focuses on the supervision needed to recover reliable geometry from only a few views.

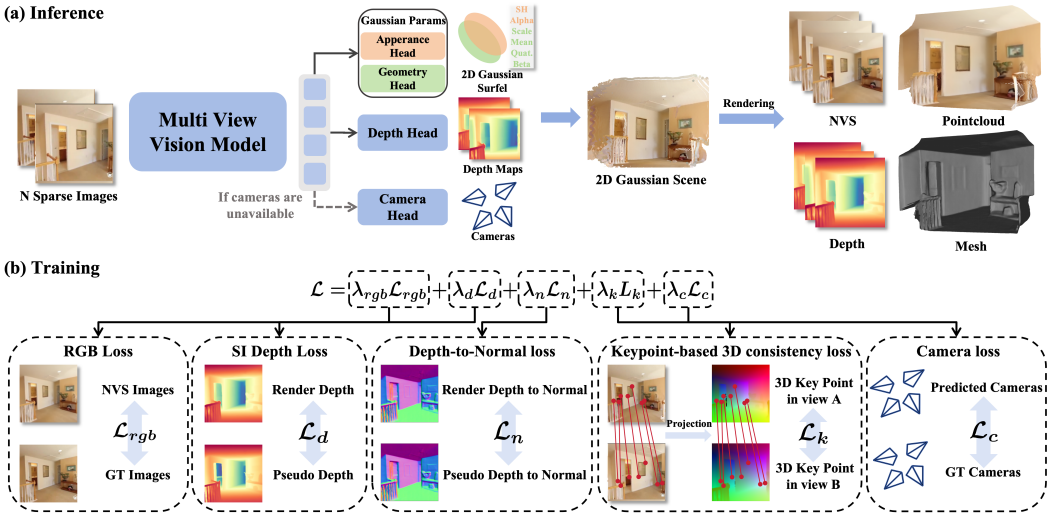

Figure 1: **Overview of GeoSupSplat.** (a) Given sparse images, a shared multi-view model feeds depth, appearance, geometry, and optional camera heads. The predicted depths and Gaussian attributes are assembled into a 2D Gaussian scene for novel-view rendering and surface reconstruction; (b) the appearance path is trained by RGB reconstruction, whereas the geometry path receives scale-invariant depth, depth-to-normal, and keypoint-based 3D consistency losses. An auxiliary camera loss anchors predicted extrinsics and intrinsics to the training cameras.

2.3 Generalizable Gaussian Reconstruction

Feed-forward Gaussian models predict a scene representation directly from one or a few images. pixelSplat regresses pixel-aligned 3D Gaussians from epipolar features, MVSplat constructs plane-sweep cost volumes for multi-view depth estimation, and FreeSplat extends this paradigm to broader indoor-view configurations Charatan et al. (2024); Chen et al. (2024); Wang et al. (2024b). These methods are designed mainly for image rendering, for which accurate appearance does not guarantee an accurate surface. Recent methods introduce stronger geometric structure and priors. GSRecon aggregates voxel- and point-level cross-view features Yang et al. (2025); MeshSplat predicts pixel-aligned 2D Gaussians and regularizes their positions and orientations with point-cloud and monocular-normal supervision Chang et al. (2026); SurfelSplat adapts surfel scale and aggregates cross-view features according to the Nyquist sampling criterion Dai et al. (2025); and PM-Loss distills a pretrained point-map prior into feed-forward Gaussian predictors Shi et al. (2025). GeoSupSplat instead first separates appearance and geometry opacity over shared surfels, then addresses scale ambiguity and cross-view inconsistency in geometry supervision.

3 Method

3.1 Overview

Given V sparse context images $\{I_v\}_{v=1}^{V}$ of a static scene, together with their camera parameters when available, our goal is to reconstruct a pixel-aligned 2D Gaussian surfel representation Huang et al. (2024) in a single forward pass. As shown in Fig. 1, a shared multi-view backbone predicts per-view features and depth, from which appearance and geometry heads construct the surfels. We first describe how separate appearance and geometry opacities define two rendering paths over one shared surface. We then introduce the geometry losses, the optional camera loss, and the surface-extraction procedures.

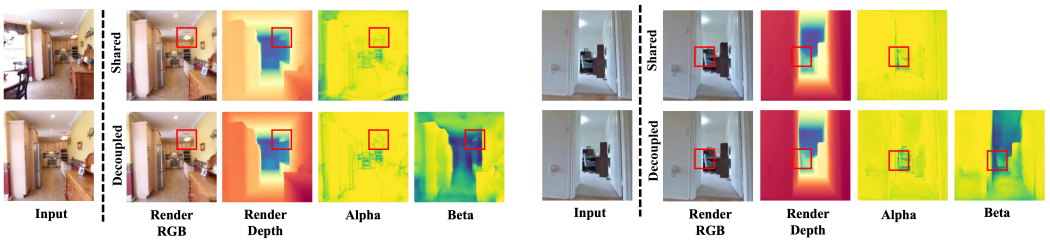

Figure 2: **Effect of a weakly decoupled gaussian representation.** The shared model uses one opacity for both RGB and depth rendering, whereas the weakly decoupled model predicts appearance opacity α and geometry opacity β over the same surfels. The highlighted regions show that a shared opacity can blur appearance details or distort depth. In contrast, the weakly decoupled model produce sharper RGB renderings and more coherent depth.

3.2 Weakly Decoupled Gaussian Representation

For each pixel u in context view v, GeoSupSplat predicts a 2D Gaussian surfel

$$\mathcal{G}_{v,u} = \big(\boldsymbol{\mu}_{v,u},\, \mathbf{s}_{v,u},\, \mathbf{R}_{v,u},\, \mathbf{c}_{v,u},\, \alpha_{v,u},\, \beta_{v,u}\big), \tag{1}$$

where $\boldsymbol{\mu}$ is its 3D center, $\mathbf{s} \in \mathbb{R}^2_+$ contains its tangential scales, $\mathbf{R}$ is its orientation, and $\mathbf{c}$ contains its spherical-harmonic color coefficients. The appearance opacity α and geometry opacity β control RGB and depth compositing, respectively. We call this representation *weakly decoupled* because the two rendering paths use separate opacity channels but share the same surfel center, scale, and orientation. Thus, the decoupling applies to compositing confidence rather than the underlying surface geometry.

Standard Gaussian splatting learns opacity mainly through photometric reconstruction, while geometry-aware extensions often reuse the same opacity to composite depth or normals. This design assumes that a primitive's contribution to image formation agrees with its geometric visibility. The two signals can favor different primitives along a ray, however, especially around transparent surfaces and view-dependent appearance. A shared opacity is then driven toward a compromise that can blur rendered appearance and distort geometry, as illustrated in Fig. 2. We therefore decouple appearance and geometry opacity.

Given the shared feature map F_v and depth prediction Z_v, separate heads predict task-specific attributes:

$$(\mathbf{c}_{v,u}, a_{v,u}) = \mathcal{H}_{\text{app}}(F_v)_u, \tag{2}$$

$$(\Delta u_{v,u}, \mathbf{s}_{v,u}, \mathbf{q}_{v,u}, b_{v,u}) = \mathcal{H}_{\text{geo}}(F_v)_u. \tag{3}$$

Here, Δu refines the pixel location and the quaternion $\mathbf{q}$ determines the surfel orientation $\mathbf{R}$. We obtain $\boldsymbol{\mu}_{v,u}$ by unprojecting the refined pixel with Z_v. The two rendering paths share the resulting physical geometry $\mathbf{g}_{v,u} = (\boldsymbol{\mu}_{v,u}, \mathbf{s}_{v,u}, \mathbf{R}_{v,u})$; only their color and opacity parameters are task specific.

We map the predicted logits to opacity as

$$\alpha = \text{sigmoid}(a), \qquad \beta = \beta_{\min} + (1 - \beta_{\min})\,\text{sigmoid}(b), \tag{4}$$

where $\beta_{\min} = 0.05$ prevents the trivial all-zero geometry-opacity solution. The geometry opacity changes only how the shared surfels are composited; it does not define a second surface.

The two opacity channels define separate compositing weights over the same depth-sorted surfels. Let $\rho_i(x)$ denote the projected footprint of surfel i at image location x. For $\gamma \in \{\alpha, \beta\}$, its front-to-back compositing weight is

$$w_i^{\gamma}(x) = \gamma_i \rho_i(x) \prod_{j<i} \big(1 - \gamma_j \rho_j(x)\big). \tag{5}$$

The appearance path uses w^{α} to render color and appearance coverage,

$$\hat{I}(x) = \sum_i w_i^{\alpha}(x)\mathbf{c}_i(x), \qquad A(x) = \sum_i w_i^{\alpha}(x), \tag{6}$$

whereas the geometry path uses w^β to render geometry coverage and expected depth,

$$B(x) = \sum_i w_i^\beta(x), \qquad \hat{D}(x) = \frac{\sum_i w_i^\beta(x) z_i(x)}{B(x) + \epsilon}, \tag{7}$$

where $z_i(x)$ is the intersection depth between the ray and surfel i. Photometric supervision updates $\mathbf{g}$, $\mathbf{c}$, and α, whereas geometry supervision updates $\mathbf{g}$ and β. The shared geometry therefore receives both signals while each opacity remains task specific.

3.3 Geometry Supervision

The geometry path in Eq. 7 allows geometric priors to supervise the predicted surfels through rendered depth. We use three complementary objectives for global depth structure, local surface orientation, and cross-view 3D consistency.

Reliable mask for geometry supervision. Teacher supervision is reliable only where the pseudo-label is valid and the rendered scene has sufficient coverage. In particular, the coverage-normalized depth in Eq. 7 becomes sensitive to noise as B_u approaches zero. We therefore define the valid pixel set

$$\mathcal{M} = \{u \mid v_u = 1,\ \mathrm{sg}(A_u) \geq \tau_\alpha,\ \mathrm{sg}(B_u) \geq \tau_\beta\}, \tag{8}$$

where v_u is the teacher-validity mask obtained by thresholding the depth confidence predicted by the teacher, and A_u and B_u are the accumulated appearance and geometry opacities along the ray. The fixed thresholds τ_α and τ_β remove insufficiently covered pixels, while $\mathrm{sg}(\cdot)$ prevents the losses from changing their own valid regions. The depth-to-normal mask $\mathcal{M}'$ also requires valid stencil neighbors, and both endpoints of a keypoint match must pass the same test.

Scale-invariant depth loss. A frozen multi-view geometry teacher provides a pseudo-depth map $\tilde{D}$, while $\hat{D}$ denotes the depth rendered by Eq. 7. The pseudo-depth contains useful scene structure, but its numerical scale is not aligned with our camera-normalized scene frame. A direct absolute ℓ_1 loss assumes that $\hat{D}$ and $\tilde{D}$ share the same scale and would penalize a geometrically correct prediction that differs from the teacher only by a global scale factor. We instead fit one scale through the origin and penalize the aligned residual:

$$\begin{aligned} s^\star &= \frac{\sum_{u\in\mathcal{M}} \hat{D}_u \tilde{D}_u}{\sum_{u\in\mathcal{M}} \hat{D}_u^2 + \epsilon}, \\ \mathcal{L}_d &= \frac{1}{|\mathcal{M}| + \epsilon} \sum_{u\in\mathcal{M}} \left(s^\star \hat{D}_u - \tilde{D}_u\right)^2. \end{aligned} \tag{9}$$

The scale is fitted independently for each rendered view. We do not fit an independent depth shift. While the teacher may differ from the rendered depth by a multiplicative scale, a per-view additive offset moves points along different camera rays and generally cannot be represented by a single shared scene transformation. Allowing such offsets would therefore absorb view-dependent geometric errors. Accordingly, $\mathcal{L}_d$ preserves the camera-relative depth offsets, while the keypoint-based loss further enforces cross-view consistency.

Depth-to-normal loss. Depth supervision constrains surface position but is less sensitive to local orientation. Rather than introducing an independent normal teacher, we derive normals from both depth maps with the same operator $\mathcal{N}[\cdot]$:

$$\hat{\mathbf{n}}_u = \mathcal{N}[\hat{D}]_u, \qquad \tilde{\mathbf{n}}_u = \mathcal{N}[\tilde{D}]_u. \tag{10}$$

The operator unprojects depth using the camera intrinsics and estimates local orientation with a fixed four-neighbor finite-difference stencil. Thus, $\hat{\mathbf{n}}_u$ is derived from the rendered depth, whereas $\tilde{\mathbf{n}}_u$ is the pseudo-normal derived from the teacher depth. We minimize their angular difference:

$$\mathcal{L}_n = \frac{1}{|\mathcal{M}'| + \epsilon} \sum_{u\in\mathcal{M}'} \left[1 - \langle \hat{\mathbf{n}}_u, \mathrm{sg}(\tilde{\mathbf{n}}_u)\rangle\right]. \tag{11}$$

Using the same operator on both sides prevents systematic differences between separate depth and normal predictors from entering the loss. Gradients flow through $\hat{\mathbf{n}}_u$ to the rendered depth, while the pseudo-normal target is detached.

Keypoint-based 3D consistency loss. The preceding losses act on each rendered view independently and do not ensure that different views reconstruct one common surface. We therefore use 2D matches from a frozen correspondence model. For a valid match $c = (u_a^c, u_b^c)$ between views v_a and v_b, we read the two rendered depths and unproject the endpoints into the shared scene frame:

$$\mathbf{X}_a^c = \mathcal{U}_{v_a}\left(u_a^c, \hat{D}_{v_a}(u_a^c)\right), \tag{12}$$

$$\mathbf{X}_b^c = \mathcal{U}_{v_b}\left(u_b^c, \hat{D}_{v_b}(u_b^c)\right), \tag{13}$$

where $\mathcal{U}_v(u, d)$ denotes unprojection using the intrinsics and pose of view v. Matched pixels should describe the same scene point, so we minimize

$$\mathcal{L}_k = \frac{1}{|\mathcal{C}| + \epsilon} \sum_{c \in \mathcal{C}} \left\| \mathbf{X}_a^c - \mathbf{X}_b^c \right\|_1, \tag{14}$$

where $\mathcal{C}$ contains matches whose two endpoints satisfy the validity and coverage conditions. This objective directly enforces cross-view agreement without requiring additional 3D labels or constraining the absolute scene scale.

Camera loss. When input cameras are unavailable, an optional camera head predicts extrinsics and focal length. During training, we supervise relative rotation, full relative translation, and normalized focal length. For ordered view pairs $\mathcal{P} = \{(i, j) \mid i \neq j\}$, let $P_i^{-1} P_j = [R_{ij} \mid \mathbf{t}_{ij}]$ and $\hat{P}_i^{-1} \hat{P}_j = [\hat{R}_{ij} \mid \hat{\mathbf{t}}_{ij}]$. The camera objective is

$$\begin{aligned} \mathcal{L}_c = \omega_R \operatorname*{mean}_{(i,j)\in\mathcal{P}} \|\hat{R}_{ij} - R_{ij}\|_F + \omega_t \operatorname*{mean}_{(i,j)\in\mathcal{P}} \rho_{0.1}\left(\hat{\mathbf{t}}_{ij} - \mathbf{t}_{ij}\right) \\ + \omega_f \operatorname*{mean}_i \left| \log \hat{f}_i^{\text{norm}} - \log f_i^{\text{norm}} \right|. \end{aligned} \tag{15}$$

Here, $\rho_{0.1}$ is the Huber loss and $f_i^{\text{norm}} = f_i^{\text{px}}/W$. We enforce square-pixel intrinsics by sharing the focal length predicted from the horizontal field of view. Relative poses remove the arbitrary global camera frame, while the translation term preserves the normalized baseline rather than metric scale. At inference, we use provided cameras when available and predicted cameras otherwise.

Full objective. The photometric objective combines ℓ_1, SSIM Wang et al. (2004), and LPIPS Zhang et al. (2018):

$$\begin{aligned} \mathcal{L}_{\text{rgb}} = \omega_1 \|\hat{I} - I\|_1 + \omega_{\text{ssim}}\big(1 - \text{SSIM}(\hat{I}, I)\big) \\ + \omega_{\text{lpips}}\, \text{LPIPS}(\hat{I}, I), \end{aligned} \tag{16}$$

where I and $\hat{I}$ are the target and rendered images, respectively. The complete training objective is

$$\mathcal{L} = \lambda_{\text{rgb}} \mathcal{L}_{\text{rgb}} + \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n + \lambda_k \mathcal{L}_k + \lambda_c \mathcal{L}_c. \tag{17}$$

The camera term is included only when the optional camera head is trained.

3.4 Surface Extraction

At inference time, we render an expected depth map $\hat{D}_v$ and geometry coverage B_v from each context and query camera. We then obtain a mesh using either TSDF fusion Curless & Levoy (1996) or Pixel-Grid Meshing (PGM), a parameter-free alternative adapted from MoGe Wang et al. (2025a). TSDF integrates the filtered depth maps into one volumetric field, whereas PGM unprojects valid pixels and connects neighboring samples using the original image-grid topology. The two extraction procedures are detailed and compared in Appendix A.4.

4 Experiments

4.1 Datasets and Preprocessing

We train GeoSupSplat on 190,000 samples from 20,000 RealEstate10K (Re10K) scenes Flynn et al. (2018). Every experiment uses $N = 2$ context views. Before training, a separate frozen Depth

Table 1: **Surface reconstruction and mesh rendering quality on Re10K with $N = 2$ context views.** Surface quality is evaluated against dense-view COLMAP reference point clouds, while mesh rendering quality is measured at held-out target views. All baselines use TSDF-fused meshes; GeoSupSplat (TSDF) is therefore our primary comparison, while PGM is reported as an alternative extraction method. The best and second-best results are marked in bold and underlined, respectively.

	Surface Reconstruction				Mesh Rendering Quality		
Method	CD ↓	Prec. ↑	Rec. ↑	F1 ↑	PSNR ↑	SSIM ↑	LPIPS ↓
pixelSplat Charatan et al. (2024)	0.778	0.188	0.207	0.233	17.41	0.720	0.349
MVSplat Chen et al. (2024)	0.547	0.352	0.291	0.312	18.52	0.783	0.256
DepthSplat Xu et al. (2025)	0.548	0.398	<u>0.297</u>	0.336	18.49	0.788	0.251
SurfelSplat Dai et al. (2025)	0.884	0.396	0.067	0.117	11.81	0.499	0.597
MeshSplat Chang et al. (2026)	0.536	0.369	0.290	0.315	17.78	0.784	0.242
GeoSupSplat (TSDF)	<u>0.459</u>	<u>0.446</u>	**0.306**	<u>0.346</u>	<u>19.68</u>	<u>0.791</u>	**0.234**
GeoSupSplat (PGM)	**0.432**	**0.471**	0.290	**0.351**	**22.15**	**0.795**	<u>0.241</u>

Anything 3 (DA3) model Lin et al. (2025) generates pseudo-depth for the depth and depth-to-normal losses, while RoMa Edstedt et al. (2024) provides cross-view matches for the keypoint-based 3D consistency loss. This offline teacher is distinct from the DA3-LARGE backbone that is fine-tuned as part of GeoSupSplat. The same Re10K-trained model is evaluated on ScanNet and Replica without fine-tuning.

4.2 Evaluation Protocol

We compare against the feed-forward Gaussian methods pixelSplat Charatan et al. (2024), MVSplat Chen et al. (2024), DepthSplat Xu et al. (2025), SurfelSplat Dai et al. (2025), and MeshSplat Chang et al. (2026). Surface reconstruction is evaluated with Chamfer distance (CD), precision, recall, and F1. We additionally render each reconstructed mesh at held-out target views and report PSNR, SSIM, and LPIPS as novel-view quality under mesh rendering. Unless stated otherwise, all methods use the same TSDF protocol with depth percentile $p = 95$, and all surface metrics use the same global scale-and-translation alignment. PGM results are identified explicitly. Appendix A.4 and Appendix A.2 provide the complete extraction and evaluation settings.

4.3 Quantitative Evaluation

Surface reconstruction and NVS quality under mesh rendering. Tab. 1 compares feed-forward Gaussian methods on Re10K under the two-view setting. Under the common TSDF protocol, GeoSupSplat outperforms every baseline on all seven metrics: it reduces CD from 0.536 to 0.459, raises F1 from 0.336 to 0.346, increases mesh-rendering PSNR from 18.52 to 19.68 dB, and lowers LPIPS from 0.242 to 0.234. PGM further improves CD, precision, F1, PSNR, and SSIM to 0.432, 0.471, 0.351, 22.15 dB, and 0.795, respectively, while TSDF retains higher recall and slightly lower LPIPS. These results show that GeoSupSplat improves both surface accuracy and the visual quality supported by the reconstructed mesh.

Fig. 3 compares meshes reconstructed from the same two input views. The baseline surfaces exhibit strong local roughness, distorted planar regions, and incomplete object boundaries. GeoSupSplat produces cleaner walls and openings while preserving thin structures such as the hanging fixture, consistent with its improvements in CD and mesh-rendering quality. We provided more qualitative visualizations in the Appendix A.11

Cross-dataset generalization. Tab. 2 evaluates the Re10K-trained model on ScanNet and Replica without fine-tuning. Under TSDF extraction, GeoSupSplat reduces the best baseline CD from 0.261 to 0.207 on ScanNet and from 0.238 to 0.163 on Replica, while improving F1 from 0.455 to 0.594 and from 0.448 to 0.470, respectively. PGM further lowers CD to 0.178 and 0.142 and raises F1 to 0.630 and 0.478. The consistent gains under both extraction schemes show that the learned geometric prior transfers beyond the Re10K training distribution.

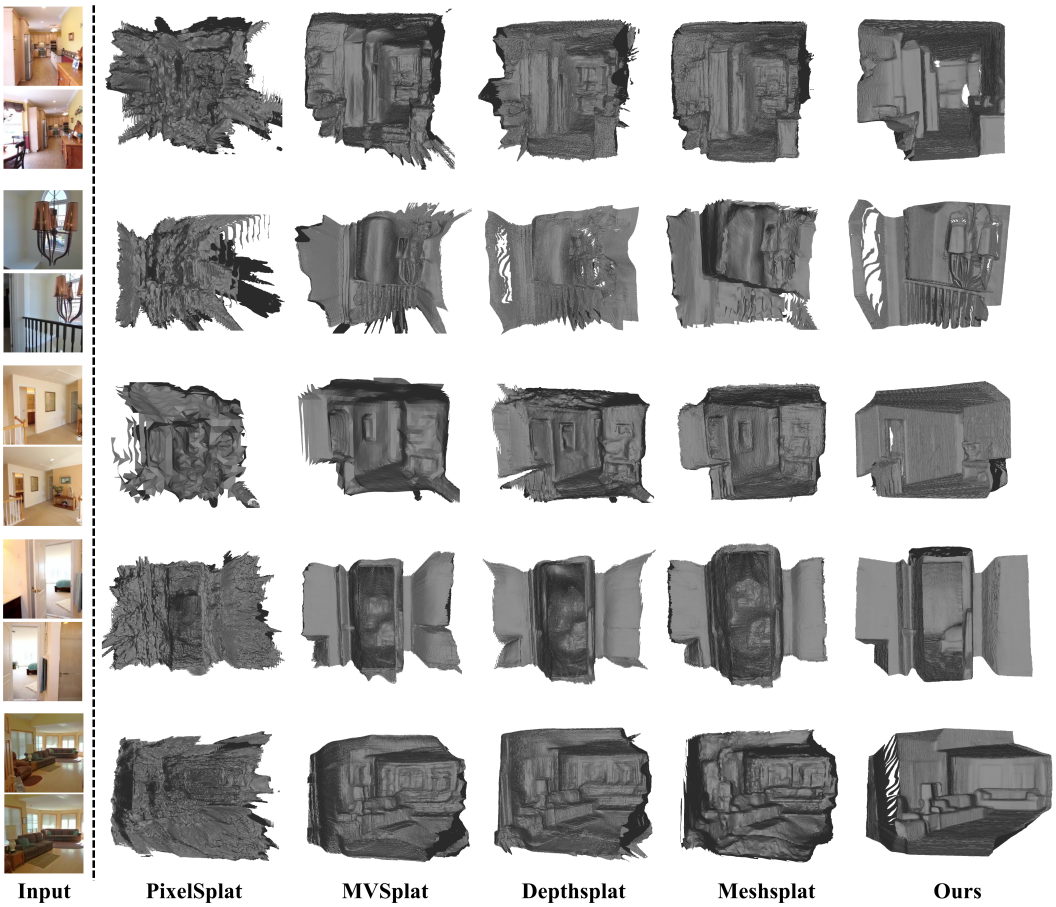

Figure 3: **Qualitative surface reconstruction on Re10K from two context views.** Under the same TSDF extraction protocol, existing methods often produce noisy surfaces, holes, or distorted scene structure. GeoSupSplat recovers cleaner planar regions and more coherent room geometry while preserving object and boundary details.

Table 2: **Zero-shot cross-dataset surface reconstruction.** All methods are trained on Re10K and evaluated without dataset-specific fine-tuning. The best and second-best results are marked in bold and underlined, respectively.

	ScanNet				Replica			
Method	CD ↓	Prec. ↑	Rec. ↑	F1 ↑	CD ↓	Prec. ↑	Rec. ↑	F1 ↑
pixelSplat Charatan et al. (2024)	0.503	0.232	0.351	0.260	0.492	0.271	0.228	0.211
MVSplat Chen et al. (2024)	0.366	0.463	0.356	0.394	0.247	0.662	0.291	0.419
SurfelSplat Dai et al. (2025)	0.721	0.252	0.110	0.105	0.706	0.294	0.071	0.085
DepthSplat Xu et al. (2025)	0.411	0.482	0.474	0.455	0.238	0.764	0.310	0.448
MeshSplat Chang et al. (2026)	0.261	0.531	0.402	0.450	0.256	0.621	0.261	0.365
GeoSupSplat (TSDF)	<u>0.207</u>	<u>0.755</u>	<u>0.501</u>	<u>0.594</u>	<u>0.163</u>	<u>0.830</u>	**0.334**	<u>0.470</u>
GeoSupSplat (PGM)	**0.178**	**0.784**	**0.540**	**0.630**	**0.142**	**0.873**	<u>0.331</u>	**0.478**

4.4 Ablation Study

Main components. Tab. 3 isolates opacity decoupling and the three geometry objectives. Adding geometry supervision with a shared opacity reduces CD from 0.771 to 0.494 and raises mesh-rendering PSNR from 10.36 to 18.28 dB. The full weakly decoupled model further improves CD

Table 3: **Main component ablation on Re10K.** "Shared" uses one opacity for appearance and geometry rendering, whereas "Decoupled" uses α and β. All variants use TSDF for surface extraction, and all unlisted settings are kept identical. The best and second-best results are shown in bold and underlined, respectively.

Variant	Opacity	$\mathcal{L}_d$	$\mathcal{L}_n$	$\mathcal{L}_k$	CD ↓	Prec. ↑	PSNR ↑	LPIPS ↓
RGB-only baseline	Shared	–	–	–	0.771	0.342	10.36	0.620
Geometry supervision	Shared	✓	✓	✓	0.494	0.388	18.28	0.268
w/o scale-invariant depth	Decoupled	–	✓	✓	0.576	0.368	19.48	0.246
w/o depth-to-normal	Decoupled	✓	–	✓	0.473	0.421	<u>19.53</u>	<u>0.241</u>
w/o keypoint 3D consistency	Decoupled	✓	✓	–	<u>0.470</u>	<u>0.423</u>	19.44	0.244
Full model	Decoupled	✓	✓	✓	**0.459**	**0.446**	**19.68**	**0.234**

Table 4: **Ablation of geometry-supervision design choices.** Each group changes one component of the full model. All variants use TSDF for surface extraction. The best and second-best results within each group are shown in bold and underlined, respectively.

Component	Variant	CD ↓	Prec. ↑	PSNR ↑	LPIPS ↓
Depth alignment	Absolute ℓ_1 depth	0.524	0.340	17.67	0.255
	Scale-and-shift-invariant depth	<u>0.469</u>	<u>0.428</u>	<u>19.44</u>	<u>0.239</u>
	Scale-invariant depth (ours)	**0.459**	**0.446**	**19.68**	**0.234**
Normal target	Omnidata Eftekhar et al. (2021)	0.486	0.384	18.87	0.256
	MoGe-2 Wang et al. (2025b)	<u>0.475</u>	<u>0.406</u>	<u>19.40</u>	<u>0.244</u>
	Normal derived from depth (ours)	**0.459**	**0.446**	**19.68**	**0.234**
Pseudo-depth source	MoGe-2 Wang et al. (2025b)	<u>0.484</u>	<u>0.418</u>	<u>19.39</u>	<u>0.240</u>
	DUSt3R Wang et al. (2024a)	0.547	0.392	19.13	0.251
	DA3 Lin et al. (2025) (ours)	**0.459**	**0.446**	**19.68**	**0.234**
Geometry backbone	MVSplat Chen et al. (2024)	0.508	0.381	19.20	0.244
	DUSt3R Wang et al. (2024a)	<u>0.464</u>	<u>0.426</u>	<u>19.32</u>	<u>0.239</u>
	DA3 Lin et al. (2025) (ours)	**0.459**	**0.446**	**19.68**	**0.234**

to 0.459, precision to 0.446, PSNR to 19.68 dB, and LPIPS to 0.234. Removing scale-invariant depth causes the largest geometric degradation, increasing CD to 0.576, while removing the normal or keypoint loss yields smaller but consistent drops. This confirms that opacity decoupling and the three geometry objectives provide complementary benefits.

Geometry-supervision design. Tab. 4 compares alternative depth alignment, normal targets, pseudo-depth teachers, and geometry backbones. Scale-invariant depth gives the best overall result, reducing CD from 0.524 with absolute ℓ_1 supervision and 0.469 with scale-and-shift alignment to 0.459. Normals derived from pseudo-depth likewise outperform independent Omnidata and MoGe-2 targets, while the DA3 pseudo-depth source and backbone achieve the best CD, precision, and PSNR in their groups. MoGe-2 yields the lowest LPIPS among pseudo-depth sources, indicating a small appearance–geometry trade-off, but the full DA3 design remains strongest on geometric accuracy.

5 Conclusion

We presented GeoSupSplat, a feed-forward 2D Gaussian framework for surface reconstruction from sparse views. It combines a weakly decoupled gaussian representation with scale-invariant depth, depth-to-normal, and keypoint-based 3D consistency supervision. GeoSupSplat improves both surface reconstruction and NVS quality under mesh rendering on Re10K and generalizes to ScanNet and Replica without fine-tuning. The current study focuses on static scenes and relies on pretrained geometric priors during training. Extending the framework to dynamic scenes, wider camera configurations, and geometry supervision learned jointly from raw video is a promising direction for future work.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools solely for language polishing and grammar checking. We did not use generative AI tools for research ideation, method or experimental design, code writing, experiment execution, or the analysis and interpretation of experimental results. All AI-assisted edits were manually reviewed by the authors to ensure technical accuracy and consistency with the intended meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

See the ICLR 2027 AI Policy for Authors for more details. This statement should not be more than 1 page.

REPRODUCIBILITY STATEMENT

We support reproducibility by documenting the proposed representation, rendering procedures, and training objectives in Section 3, while Section 4 describes the experimental setup and evaluation methodology. Appendix A.1 provides implementation and optimization details, Appendix A.3 describes the construction of the training and evaluation benchmarks, and Appendices A.2 and A.4 specify the evaluation and surface-extraction protocols used for all reported results.

REFERENCES

Hanzhi Chang, Ruijie Zhu, Wenjie Chang, Mulin Yu, Yanzhe Liang, Jiahao Lu, Zhuoyuan Li, and Tianzhu Zhang. MeshSplat: Generalizable sparse-view surface reconstruction via gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2716–2724, 2026. doi: 10.1609/aaai.v40i4.37260.

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

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pp. 370–386, 2024.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 303–312, 1996.

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*, pp. 5828–5839, 2017.

Chensheng Dai, Shengjun Zhang, Min Chen, and Yueqi Duan. SurfelSplat: Learning efficient and generalizable gaussian surfel representations for sparse-view surface reconstruction. In *Advances in Neural Information Processing Systems*, 2025.

Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19790–19800, 2024.

Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10786–10796, 2021.

John Flynn, Tinghui Zhou, Richard Tucker, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM SIGGRAPH 2018 Papers*, pp. 1–12, 2018.

Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5354–5363, 2024.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

Yixun Liang, Hao He, and Yingcong Chen. ReTR: Modeling rendering via transformer for generalizable neural surface reconstruction. In *Advances in Neural Information Processing Systems*, volume 36, pp. 62332–62351, 2023.

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

Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. SparseNeuS: Fast generalizable neural surface reconstruction from sparse views. In *European Conference on Computer Vision*, pp. 210–227, 2022.

Youngju Na, Woo Jae Kim, Kyu Beom Han, Suhyeon Ha, and Sung-Eui Yoon. UFORecon: Generalizable sparse-view surface reconstruction from arbitrary and unfavorable sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5094–5104, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Rui Peng, Xiaodong Gu, Luyang Tang, Shihe Shen, Fanqi Yu, and Ronggang Wang. GenS: Generalizable neural surface reconstruction from multi-view images. In *Advances in Neural Information Processing Systems*, volume 36, pp. 56932–56945, 2023.

Yufan Ren, Fangjinhua Wang, Tong Zhang, Marc Pollefeys, and Sabine Süsstrunk. VolRecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16685–16695, 2023.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pp. 501–518, 2016.

Duochao Shi, Weijie Wang, Donny Y. Chen, Zeyu Zhang, Jia-Wang Bian, Bohan Zhuang, and Chunhua Shen. Revisiting depth representations for feed-forward 3d gaussian splatting. *arXiv preprint arXiv:2506.05327*, 2025.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27171–27183, 2021.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5261–5271, 2025a.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. In *Advances in Neural Information Processing Systems*, volume 38, pp. 40377–40408, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024a.

Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. FreeSplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. In *Advances in Neural Information Processing Systems*, volume 37, pp. 107326–107349, 2024b.

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

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16453–16463, 2025.

Luoyuan Xu, Tao Guan, Yuesong Wang, Wenkai Liu, Zhaojie Zeng, Junle Wang, and Wei Yang. C2F2NeuS: Cascade cost frustum fusion for high fidelity and generalizable neural surface reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18291–18301, 2023.

Hang Yang, Le Hui, Jianjun Qian, Jin Xie, and Jian Yang. GSRecon: Efficient generalizable gaussian splatting for surface reconstruction from sparse views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25346–25356, 2025.

Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. GSDF: 3dgs meets sdf for improved rendering and reconstruction. In *Advances in Neural Information Processing Systems*, volume 37, pp. 129507–129530, 2024a.

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 43(6):1–13, 2024b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

A Additional Details

A.1 Implementation Details

Architecture and inputs. GeoSupSplat uses a fine-tuned DA3-LARGE backbone Lin et al. (2025) with a DINOv2 ViT-L trunk Oquab et al. (2024). We extract features from layers 11, 15, 19, and 23 and decode them with separate 128-channel appearance and geometry heads. The model predicts degree-3 spherical harmonics, appearance and geometry opacity, and an image-plane center offset. We do not predict an additional depth residual or separate rotations for the two rendering paths. Camera translations are normalized by the maximum pairwise distance between the context cameras. During training, the probability of using provided cameras decreases linearly from 1.0 to 0.85 over the first 20,000 steps and remains fixed thereafter.

Optimization. We train for 100,000 optimizer steps with a per-process batch size of 4 and gradient accumulation over 8 iterations, corresponding to 32 scenes per optimizer step per process. We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.05, and global gradient-norm clipping at 1.0. The base learning rate is 2×10^{-4} for the prediction heads and parameter adapter and 2×10^{-5} for the DA3 trunk. A 1,000-step linear warm-up is followed by cosine decay to 5% of the initial learning rate. Training uses bfloat16 mixed precision and random seed 42, while rendering and camera calculations use FP32. We save checkpoints every 1,000 steps and retain the three most recent checkpoints. All training and evaluation runs are conducted on a single NVIDIA RTX 5090 GPU.

Supervision and loss aggregation. Each training sample contains two context and four target views. We average $\mathcal{L}_{\text{rgb}}$ over the target renders and apply $\mathcal{L}_d$ and $\mathcal{L}_n$ to both target views and context views re-rendered from their own cameras. The keypoint term is averaged over valid RoMa correspondences between context and target training views. For sub-pixel matches, rendered depth is sampled bilinearly before unprojection. A view or view pair with no valid supervision contributes zero to the corresponding loss. The depth-to-normal and keypoint terms are enabled after the initial photometric and scale-invariant depth warm-up, whereas RGB and camera supervision are active from the start. All loss weights and schedules are fixed across datasets.

The reliable mask uses $\tau_\alpha = 0.1$ and $\tau_\beta = 0.1$, and a depth-based term is evaluated only when at least 256 pixels remain valid. Teacher validity requires finite positive pseudo-depth and the validity supplied with the pseudo-label. To compute depth-derived normals, we first form the point map $\mathbf{X}_D(u) = \mathcal{U}_v(u, D(u))$ and apply the same central-difference operator to rendered and teacher depth:

$$\mathcal{N}[D]_u = \text{normalize}\big(\big[\mathbf{X}_D(u + e_x) - \mathbf{X}_D(u - e_x)\big] \times \big[\mathbf{X}_D(u + e_y) - \mathbf{X}_D(u - e_y)\big]\big) . \quad (18)$$

We use the same cross-product order and camera convention on both sides, and require the stencil center and all four neighbors to be valid.

The RGB terms use $(\omega_1, \omega_{\text{ssim}}, \omega_{\text{lpips}}) = (1.0, 0.05, 0.05)$. Camera supervision uses $(\omega_R, \omega_t, \omega_f) = (0.1, 0.1, 0.1)$, with camera translations normalized by the maximum pairwise context-camera distance before relative-pose supervision. The Huber penalty in Eq. 15 is applied element-wise and averaged over the three translation coordinates. The final loss weights are $(\lambda_{\text{rgb}}, \lambda_d, \lambda_n, \lambda_k) = (1.0, 0.1, 0.1, 0.3)$.

A.2 Evaluation Protocol

All methods use the same two context views, three target cameras, reference point clouds, and evaluation masks. Predicted meshes are mapped from network-normalized coordinates to the original dataset frame with the stored `world_from_normalized` similarity transform. The complete transform is applied to camera centers, whereas only its rotation is applied to camera orientations. Predicted vertices and reference point clouds are then voxel-downsampled to $2\,\text{cm}$.

Context-visible reference region. Because reconstruction is conditioned on only two images, we evaluate against reference geometry visible from at least one context camera. A reference point is retained when it has positive depth and projects into either normalized image rectangle with a border

allowance of 0.05. To remove distant ray-like artifacts in the dense reference cloud, its nearest valid context depth must also be no greater than 2.5 times the median context depth. Only the reference cloud is cropped. The complete prediction is retained so that unsupported or floating geometry remains penalized. We use the full reference cloud only if numerical projection errors leave the visible set empty.

Global alignment. The scale-invariant depth loss supervises relative scene structure without requiring the model to reproduce the teacher's arbitrary scale. In addition, feed-forward methods use different conventions for scene scale and origin. A direct raw comparison can therefore mix local reconstruction error with a single global coordinate-frame offset. **We report controlled alignment settings to separate these effects, applying each setting identically to GeoSupSplat and every baseline. Unless stated otherwise, all surface results use scale-and-translation (ST) alignment.**

We obtain alignment correspondences by projecting the predicted and reference point clouds into the shared context and target cameras. A z-buffer retains the nearest point on a 224×224 grid, producing paired points $(\widehat{\mathbf{x}}_k, \mathbf{x}_k)$ at jointly valid pixels. With a fixed seed, we select at most 2,048 pairs and require at least 50. Otherwise, the identity transform is used. Following MoGe Wang et al. (2025a), alignment minimizes a robust coordinate-wise ℓ_1 objective without filtering pairs by their final surface distance.

Raw evaluation. The raw setting compares prediction and reference directly in the original dataset coordinates and therefore retains errors in global scale, translation, and local shape.

Scale-only alignment. To remove only the global scale ambiguity while preserving the predicted origin and orientation, we solve

$$s^\star = \arg\min_s \sum_k \|s\widehat{\mathbf{x}}_k - \mathbf{x}_k\|_1 , \tag{19}$$

and apply $s^\star$ to the complete prediction.

Scale-and-translation alignment. ST alignment additionally removes one global translation:

$$(s^\star, \mathbf{t}^\star) = \arg\min_{s,\mathbf{t}} \sum_k \|s\widehat{\mathbf{x}}_k + \mathbf{t} - \mathbf{x}_k\|_1 . \tag{20}$$

We do not optimize rotation because the camera frame already fixes the global orientation. Alignment cannot correct local shape errors or cross-view inconsistency, and the complete transformed prediction is retained for evaluation. Tab. 5 reports all three settings under otherwise identical conditions.

GeoSupSplat obtains the lowest CD and highest precision under all three protocols. Its CD decreases from 0.575 in the raw frame to 0.475 with scale-only alignment and 0.459 with ST alignment. Under ST alignment, it also achieves the highest recall and F1. Because every method receives the same alignment, the remaining advantage reflects local reconstruction quality rather than a favorable global coordinate convention.

ST alignment cannot correct per-view depth offsets, local surface deformation, or cross-view inconsistency. Because it removes global placement error, the default surface metrics should be interpreted as aligned shape quality. GeoSupSplat remains the strongest CD and precision metric under the raw and scale-only protocols, showing that its gains are not produced solely by ST alignment.

Surface metrics. Let $\mathcal{P}$ and $\mathcal{G}$ denote the predicted and context-visible reference point sets, and let $d(\mathbf{x}, \mathcal{S}) = \min_{\mathbf{y}\in\mathcal{S}} \|\mathbf{x} - \mathbf{y}\|_2$. We report bidirectional Chamfer distance

$$\mathrm{CD}(\mathcal{P}, \mathcal{G}) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p}\in\mathcal{P}} d(\mathbf{p}, \mathcal{G}) + \frac{1}{|\mathcal{G}|} \sum_{\mathbf{g}\in\mathcal{G}} d(\mathbf{g}, \mathcal{P}), \tag{21}$$

and precision, recall, and F1 at $\tau = 0.05$ in the original dataset coordinates. Define

$$\begin{aligned} \mathcal{P}_\tau &= \{\mathbf{p} \in \mathcal{P} \mid d(\mathbf{p}, \mathcal{G}) < \tau\} , \\ \mathcal{G}_\tau &= \{\mathbf{g} \in \mathcal{G} \mid d(\mathbf{g}, \mathcal{P}) < \tau\} . \end{aligned} \tag{22}$$

Table 5: **Surface reconstruction under different global-alignment protocols on Re10K.** Raw results preserve the complete prediction error; scale-only alignment removes a single global scale factor; and scale-and-translation alignment additionally removes one global shift. ST alignment is the default used in our other surface-reconstruction tables. No protocol optimizes rotation or crops the prediction. The best and second-best results within each block are marked in bold and underlined, respectively.

Method	CD ↓	Prec. ↑	Rec. ↑	F1 ↑
Raw				
pixelSplat Charatan et al. (2024)	1.937	0.094	0.244	0.125
MVSplat Chen et al. (2024)	0.637	0.338	0.275	0.300
DepthSplat Xu et al. (2025)	<u>0.619</u>	<u>0.374</u>	<u>0.283</u>	**0.307**
MeshSplat Chang et al. (2026)	0.637	0.353	**0.292**	<u>0.306</u>
GeoSupSplat (ours)	**0.575**	**0.393**	0.269	0.302
Scale-only aligned				
pixelSplat Charatan et al. (2024)	0.924	0.153	0.203	0.205
MVSplat Chen et al. (2024)	0.578	0.346	0.280	0.307
DepthSplat Xu et al. (2025)	0.580	<u>0.387</u>	**0.295**	<u>0.321</u>
MeshSplat Chang et al. (2026)	<u>0.575</u>	0.359	<u>0.294</u>	0.311
GeoSupSplat (ours)	**0.475**	**0.429**	0.285	**0.330**
Scale-and-translation aligned				
pixelSplat Charatan et al. (2024)	0.778	0.188	0.207	0.233
MVSplat Chen et al. (2024)	0.547	0.352	0.291	0.312
DepthSplat Xu et al. (2025)	0.548	<u>0.398</u>	<u>0.297</u>	<u>0.336</u>
MeshSplat Chang et al. (2026)	<u>0.536</u>	0.369	0.290	0.315
GeoSupSplat (ours)	**0.459**	**0.446**	**0.306**	**0.346**

The remaining metrics are

$$\mathrm{Prec} = \frac{|\mathcal{P}_\tau|}{|\mathcal{P}|}, \tag{23}$$

$$\mathrm{Rec} = \frac{|\mathcal{G}_\tau|}{|\mathcal{G}|}, \tag{24}$$

$$\mathrm{F1} = \frac{2\ \mathrm{Prec}\ \mathrm{Rec}}{\mathrm{Prec} + \mathrm{Rec}}. \tag{25}$$

Mesh rendering. We render each mesh from the three target cameras at 224×224 resolution with $2\times$ supersampling. NVS quality under mesh rendering is measured by PSNR, SSIM, and VGG-based LPIPS against the target images. We also record the fraction of target pixels covered by the mesh.

Reproducibility. Evaluation indices are stored explicitly and shared across methods. Every run uses the same target cameras, reference clouds, context-visible regions, point-cloud downsampling, thresholds, and alignment procedure. We save per-scene metrics, aggregate summaries, resolved configurations, coordinate transforms, and mesh-rendering statistics.

A.3 Dataset and Benchmark Construction

Training data. We use the Re10K Flynn et al. (2018) training subset described in the main paper, containing 190,000 samples from 20,000 RealEstate10K (Re10K). Each sample contains two context and four target views, with context-frame separations sampled between 45 and 135 frames. Native 360×640 images are resized and cropped to 224×224. We remove samples in which a detected person occupies more than 5% of an image, using a segmentation confidence threshold of 0.35. The training index is generated with seed 0, and model training uses seed 42.

Re10K test benchmark. The benchmark contains 50 scenes, each with two context and three target views. Because Re10K does not provide ground-truth surfaces, we follow MeshSplat Chang et al. (2026) and construct dense COLMAP reference point clouds Schönberger & Frahm (2016); Schönberger et al. (2016). We produce 50 additional clouds reconstructed from at most 80 evenly spaced frames with fixed dataset intrinsics and poses. Candidate scenes provide sufficient camera motion, parallax, and texture for dense reconstruction. The released benchmark index defines the evaluated view groups exactly.

ScanNet and Replica. For cross-dataset evaluation, we use all scenes in our ScanNet Dai et al. (2017) and Replica Straub et al. (2019) test sets, comprising 8 and 16 scenes, respectively. Because our task reconstructs a scene from sparse observations, we deterministically sample five view groups from each scene and select them to cover different scene regions as broadly as possible. This procedure yields 40 ScanNet samples and 80 Replica samples, each containing two context and three target views. Context pairs have symmetric geometric overlap between 30% and 80%, a translation baseline between $0.15\,\mathrm{m}$ and $2.5\,\mathrm{m}$, and relative rotation between 5° and 90°. Directed overlap is estimated from stride-16 depth samples and requires reprojection agreement within $\max(0.05\,\mathrm{m}, 0.05z)$; the smaller directional overlap defines the symmetric score. Each target view must be supported by both context views, and context frames are not reused across groups within the same scene. Reference clouds are built by fusing ground-truth RGB-D frames and poses with a $1\,\mathrm{cm}$ voxel hash, pixel stride 4, and a valid depth range of 0.1–$10\,\mathrm{m}$, followed by the common $2\,\mathrm{cm}$ downsampling.

A.4 Surface Extraction

We evaluate TSDF fusion and Pixel-Grid Meshing (PGM) from the same rendered depth and geometry-coverage maps. This controlled comparison isolates the effect of extraction from the learned geometry.

TSDF fusion. We discard depth pixels below the coverage threshold and fuse the remaining observations into a truncated signed distance field Curless & Levoy (1996). The zero level set gives one fused surface. TSDF can combine overlapping observations and suppress view-specific depth noise, but it resamples the geometry on a voxel grid and requires scene-dependent voxel and truncation scales.

Unless stated otherwise, TSDF uses expected depths rendered from both context and all three target cameras. Target images and ground-truth target depths are never used. The target poses only define additional views of the predicted representation. Let $\mathcal{D}_{\mathrm{valid}}$ be the positive rendered depths retained by the coverage mask. For percentile p, the scene depth extent is

$$d_p = Q_p(\mathcal{D}_{\mathrm{valid}})\,, \tag{26}$$

where Q_p is the p-th percentile. We use no additional multiplicative margin and set $d_p = 5.0$ when no valid depth is available. The voxel size v and truncation distance μ are

$$v = \frac{d_p}{1024}, \qquad \mu = 80v. \tag{27}$$

Depth-edge rejection is disabled for every method, and we retain at most the 50 largest connected components containing at least 50 triangles. All standard results use $p = 95$ and the high-percentile setting appears only in the study below.

Pixel-Grid Meshing. As a parameter-free alternative, we adapt the depth-to-mesh conversion used by MoGe Wang et al. (2025a), which we call *Pixel-Grid Meshing* (PGM). Every coverage-valid depth pixel is unprojected into a 3D vertex. Each valid 2×2 cell in the original image grid forms two triangles, after which the per-view meshes are transformed into the shared scene frame and concatenated. PGM performs no cross-view fusion or vertex merging and requires neither a voxel size nor a truncation distance.

Controlled extraction comparison. Tab. 6 compares TSDF and PGM using identical GeoSupSplat predictions. Alongside CD and F1, extraction time and face count characterize computational cost and mesh size.

Table 6: **Comparison of surface-extraction methods.** Both methods receive identical rendered depth and geometry-coverage maps at 224×224 resolution. Runtime and mesh size are averaged over 50 Re10K scenes on a single RTX 5090. The better result for each metric is shown in bold.

Method	CD ↓	F1 ↑	Time (s) ↓	#Faces ↓
TSDF fusion	0.459	0.346	31.57	5.14M
Pixel-Grid Meshing	**0.432**	**0.351**	**0.78**	**0.496M**

Table 7: **Baseline comparison using Pixel-Grid Meshing.** PGM is applied to the rendered depth maps of every method. We report surface reconstruction and mesh-rendering quality on Re10K and under zero-shot transfer to ScanNet and Replica. Re10K contains 50 test scenes, while ScanNet and Replica contain 40 and 80 sparse-view groups, respectively. The best and second-best results within each dataset are marked in bold and underlined.

	Surface Reconstruction				Mesh Rendering Quality		
Method	CD ↓	Prec. ↑	Rec. ↑	F1 ↑	PSNR ↑	SSIM ↑	LPIPS ↓
Re10K							
pixelSplat Charatan et al. (2024)	0.588	0.264	0.239	0.277	21.09	0.748	0.293
MVSplat Chen et al. (2024)	0.543	0.430	0.207	0.352	22.00	0.789	0.242
DepthSplat Xu et al. (2025)	<u>0.487</u>	<u>0.445</u>	**0.310**	<u>0.348</u>	<u>21.75</u>	<u>0.794</u>	**0.224**
MeshSplat Chang et al. (2026)	0.539	0.431	0.281	0.336	21.64	0.785	0.259
GeoSupSplat (ours)	**0.432**	**0.471**	<u>0.290</u>	**0.351**	**22.15**	**0.795**	<u>0.241</u>
ScanNet							
pixelSplat Charatan et al. (2024)	0.733	0.424	0.277	0.247	19.32	0.655	0.413
MVSplat Chen et al. (2024)	0.657	0.550	0.323	0.371	19.40	0.657	0.348
DepthSplat Xu et al. (2025)	<u>0.540</u>	0.537	<u>0.462</u>	<u>0.468</u>	<u>20.01</u>	<u>0.678</u>	<u>0.346</u>
MeshSplat Chang et al. (2026)	0.753	<u>0.572</u>	0.409	0.466	19.35	0.674	0.358
GeoSupSplat (ours)	**0.178**	**0.784**	**0.540**	**0.630**	**20.22**	**0.687**	**0.344**
Replica							
pixelSplat Charatan et al. (2024)	0.495	0.439	0.194	<u>0.408</u>	20.20	0.738	0.191
MVSplat Chen et al. (2024)	0.300	0.656	0.254	0.364	<u>21.73</u>	0.765	0.216
DepthSplat Xu et al. (2025)	<u>0.230</u>	<u>0.681</u>	<u>0.285</u>	0.399	21.15	<u>0.776</u>	<u>0.206</u>
MeshSplat Chang et al. (2026)	0.365	0.672	0.271	0.384	21.43	0.766	0.225
GeoSupSplat (ours)	**0.142**	**0.873**	**0.331**	**0.478**	**22.34**	**0.778**	**0.247**

Using identical rendered geometry, PGM reduces CD from 0.459 to 0.432 and raises F1 from 0.346 to 0.351. It also lowers extraction time from 31.57 to 0.78 seconds and mesh size from 5.14M to 0.496M faces, corresponding to approximately $40\times$ faster extraction and $10\times$ fewer faces. Thus, PGM preserves the predicted surface more directly while avoiding the computational and discretization costs of volumetric fusion.

We also apply PGM to every method in Tab. 7, removing extraction-method differences from the comparison. The main-paper protocol instead uses the default TSDF meshes released or reproduced for the baselines and reports both TSDF and PGM for GeoSupSplat.

Under the common PGM extractor, GeoSupSplat achieves the best CD, precision, F1, PSNR, and SSIM on Re10K, with CD improving from the second-best 0.487 to 0.432. It further achieves the best four surface metrics on both ScanNet and Replica, reducing CD from 0.540 to 0.178 and from 0.230 to 0.142, respectively.

Qualitative behavior. Fig. 4 shows that TSDF may introduce fine staircase patterns when continuous depth is resampled on a finite voxel grid. A finer grid reduces this effect at the cost of memory and extraction time. PGM avoids voxelization and more directly preserves local depth detail, but it can retain view-specific roughness and produce cracks or disconnected pieces near invalid regions. Thus, TSDF favors a compact fused surface, whereas PGM preserves per-view geometry without scene-scale extraction parameters.

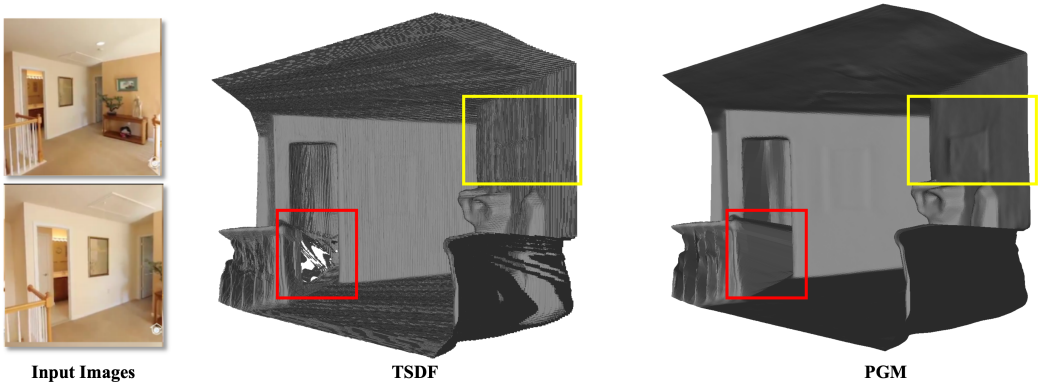

Figure 4: **Qualitative comparison of surface extraction.** Both meshes are extracted from the same rendered depth and geometry-coverage maps. TSDF produces fine staircase and ripple patterns on planar regions (yellow box) and fragmented geometry near depth discontinuities (red box). PGM avoids voxel-grid artifacts and preserves smoother local surfaces, but its direct pixel-grid construction may connect samples across a depth boundary, producing stretched triangles near the doorway (red box).

Table 8: **Effect of the TSDF depth percentile.** Each setting reuses the same saved RGB-D predictions and changes only the percentile used to determine the fusion extent and reject far depths. Metrics use scale-and-translation alignment, and $p = 95$ is the default elsewhere. The best and second-best results within each dataset and percentile block are marked in bold and underlined, respectively.

		$p = 100$				$p = 95$			
Dataset	Method	CD ↓	Prec. ↑	Rec. ↑	F1 ↑	CD ↓	Prec. ↑	Rec. ↑	F1 ↑
Re10K	pixelSplat Charatan et al. (2024)	3.349	0.141	0.190	0.121	0.778	0.188	0.207	0.233
	MVSplat Chen et al. (2024)	3.205	0.275	0.260	0.238	0.547	0.352	0.291	0.312
	DepthSplat Xu et al. (2025)	<u>1.279</u>	<u>0.337</u>	<u>0.290</u>	<u>0.328</u>	0.548	<u>0.398</u>	<u>0.297</u>	<u>0.336</u>
	MeshSplat Chang et al. (2026)	2.764	0.286	0.237	0.247	<u>0.536</u>	0.369	0.290	0.315
	GeoSupSplat (ours)	**0.448**	**0.435**	**0.300**	**0.337**	**0.459**	**0.446**	**0.306**	**0.346**
ScanNet	pixelSplat Charatan et al. (2024)	1.414	0.433	0.034	0.028	0.503	0.232	0.351	0.260
	MVSplat Chen et al. (2024)	3.204	0.365	0.205	0.199	0.366	0.463	0.356	0.394
	DepthSplat Xu et al. (2025)	<u>1.371</u>	0.326	<u>0.401</u>	<u>0.309</u>	0.411	0.482	<u>0.474</u>	<u>0.455</u>
	MeshSplat Chang et al. (2026)	3.718	<u>0.468</u>	0.236	0.244	<u>0.261</u>	<u>0.531</u>	0.402	0.450
	GeoSupSplat (ours)	**0.187**	**0.752**	**0.536**	**0.617**	**0.207**	**0.755**	**0.501**	**0.594**
Replica	pixelSplat Charatan et al. (2024)	5.621	0.232	0.173	0.161	0.492	0.271	0.228	0.211
	MVSplat Chen et al. (2024)	3.294	0.527	0.235	0.312	0.247	0.662	0.291	0.419
	DepthSplat Xu et al. (2025)	<u>0.326</u>	<u>0.585</u>	<u>0.279</u>	<u>0.372</u>	<u>0.238</u>	<u>0.764</u>	<u>0.310</u>	<u>0.448</u>
	MeshSplat Chang et al. (2026)	5.531	0.552	0.215	0.292	0.256	0.621	0.261	0.365
	GeoSupSplat (ours)	**0.152**	**0.826**	**0.334**	**0.473**	**0.163**	**0.830**	**0.334**	**0.470**

Depth-percentile study. We compare the default $p = 95$ setting with a high-percentile $p = 100$ setting on Re10K, ScanNet, and Replica in Tab. 8. A high percentile preserves nearly the complete depth range but allows a few distant predictions to enlarge the fusion volume and voxel size. The $p = 95$ setting rejects depths above d_{95} only during TSDF integration. Within each method, we keep the RGB-D predictions, cameras, fusion views, truncation multiplier, target resolution, component filtering, and evaluation protocol fixed.

The effect of the percentile differs notably between GeoSupSplat and the baselines. Reducing the percentile from $p = 100$ to $p = 95$ lowers MVSplat's CD from 3.205 to 0.547 on Re10K, from 3.204 to 0.366 on ScanNet, and from 3.294 to 0.247 on Replica, indicating that a few far-depth predictions can strongly enlarge its fusion volume. GeoSupSplat is less sensitive: its CD changes from 0.448 to 0.459, from 0.187 to 0.207, and from 0.152 to 0.163 on the three datasets. This smaller variation

Table 9: **Ablation of opacity decoupling and the reliable supervision mask.** Within each group, all components not named in the variant are fixed to the full model. The best and second-best valid results within each group are shown in bold and underlined, respectively.

Component	Variant	CD ↓	F1 ↑	PSNR ↑	LPIPS ↓
Opacity	Shared	0.494	0.308	18.28	0.260
	Decoupled, shared prediction head	0.471	<u>0.335</u>	<u>19.52</u>	<u>0.235</u>
	Decoupled, separate heads, $\beta_{\min} = 0$	<u>0.469</u>	0.327	19.20	0.249
	Decoupled, separate heads, $\beta_{\min} = 0.05$ (ours)	**0.459**	**0.346**	**19.68**	**0.234**
Supervision mask	Teacher validity v only	0.489	0.314	18.62	0.251
	v with appearance-coverage filtering	<u>0.474</u>	<u>0.329</u>	18.98	<u>0.239</u>
	v with geometry-coverage filtering	0.479	0.327	<u>19.06</u>	0.248
	Full mask $\mathcal{M}$ (ours)	**0.459**	**0.346**	**19.68**	**0.234**

Table 10: **Ablation of camera prediction on Re10K.** RRE, RTE, and TDE denote relative rotation, translation, and translation-direction errors, respectively. Focal error is computed on normalized focal length.

Camera input	$\mathcal{L}_c$	RRE (°) ↓	RTE ↓	TDE (°) ↓	Focal error ↓	CD ↓	F1 ↑	PSNR ↑
GT cameras	–	0	0	0	0	0.459	0.346	19.68
Predicted cameras	✓	1.15	0.04	2.15	0.027	0.467	0.340	18.13

suggests fewer harmful far-depth outliers, although $p = 95$ can remove some valid distant geometry. Nevertheless, we use $p = 95$ as a common robust default for all methods rather than tuning the percentile separately for each model.

A.5 Opacity Decoupling and Reliable Masking

Tab. 9 examines whether separate opacity channels and prediction heads are necessary and whether $\beta_{\min}$ prevents geometry coverage from collapsing. It also removes teacher validity v, appearance coverage A, and geometry coverage B from the supervision mask in turn.

Decoupling the opacity channels improves CD from 0.494 with shared opacity to 0.459 and raises F1 from 0.308 to 0.346. Separate prediction heads and the lower bound $\beta_{\min} = 0.05$ provide the strongest variant, while removing the lower bound reduces F1 to 0.327 and increases LPIPS to 0.249. For supervision masking, combining teacher validity with both appearance and geometry coverage outperforms either coverage filter alone.

A.6 Camera Prediction

The main experiments use calibrated input cameras. We separately evaluate the optional camera head when these cameras are unavailable and test whether $\mathcal{L}_c$ limits camera drift. Tab. 10 reports relative rotation error (RRE), relative translation error (RTE), translation-direction error (TDE), normalized focal-length error, and reconstruction quality. RRE, RTE, and focal error correspond to Eq. 15, while TDE provides an additional diagnostic.

Replacing ground-truth cameras with predicted cameras increases CD only from 0.459 to 0.467 and reduces F1 from 0.346 to 0.340, while PSNR decreases from 19.68 to 18.13 dB. The camera head attains an RRE of $1.15°$, an RTE of 0.04, and a focal error of 0.027. This moderate degradation shows that GeoSupSplat remains usable when calibration is unavailable, while provided cameras remain preferable when present.

A.7 Model Parameters and Inference Time

We compare the computational cost of GeoSupSplat with pixelSplat, MVSplat, DepthSplat, and MeshSplat in Tab. 11. To isolate network efficiency from implementation-dependent post-processing, parameter count and FLOPs cover the complete prediction network, while runtime mea-

Table 11: **Model complexity and inference efficiency.** Parameters and FLOPs are reported for the full prediction network. Runtime measures one representation-prediction forward pass with $N = 2$ context views at 224×224 resolution; rendering and surface extraction are excluded. All methods are evaluated on a single RTX 5090. The best and second-best results are shown in bold and underlined, respectively.

Method	Params (M) ↓	FLOPs (G) ↓	Inference Time (s) ↓
pixelSplat Charatan et al. (2024)	119.0	1165	0.069
MVSplat Chen et al. (2024)	**12.0**	**250**	0.079
DepthSplat Xu et al. (2025)	354.3	834	0.087
MeshSplat Chang et al. (2026)	<u>13.3</u>	<u>365</u>	<u>0.050</u>
GeoSupSplat	456.5	712	**0.045**

Table 12: **Direct RGB-rendering NVS on Re10K with $N = 2$ context views.** Following pixelSplat Charatan et al. (2024), held-out target views are rendered directly from the predicted Gaussian representation, without surface extraction or mesh rendering. The best and second-best results are shown in bold and underlined, respectively.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
pixelSplat Charatan et al. (2024)	24.99	0.841	0.156
MVSplat Chen et al. (2024)	<u>26.37</u>	<u>0.869</u>	<u>0.123</u>
DepthSplat Xu et al. (2025)	**27.17**	**0.888**	**0.110**
MeshSplat Chang et al. (2026)	25.05	0.847	0.141
GeoSupSplat	25.58	0.848	0.158

sures only the forward pass that constructs the scene representation. All methods use two 224×224 context views, batch size one, identical numerical precision, and the same hardware. Gaussian rendering and surface extraction are excluded from this timing.

GeoSupSplat gives the fastest measured forward pass at 0.045 seconds, compared with 0.050 seconds for the second-fastest MeshSplat. It nevertheless has the largest parameter count (456.5M), while its 712G FLOPs remain below DepthSplat and pixelSplat. MVSplat is the most compact model at 12.0M parameters and 250G FLOPs. Thus, GeoSupSplat provides low inference latency at the cost of a larger model footprint.

A.8 NVS Comparison

The main paper evaluates NVS quality under mesh rendering, which jointly reflects reconstruction and extraction quality. We additionally follow the pixelSplat protocol Charatan et al. (2024) and render held-out Re10K target views directly from each predicted Gaussian representation. Tab. 12 therefore measures appearance prediction without introducing TSDF, PGM, or a mesh renderer. Every method uses the same two context views and target-view split.

For direct Gaussian RGB rendering, DepthSplat performs best with 27.17 dB PSNR, 0.888 SSIM, and 0.110 LPIPS, followed by MVSplat. GeoSupSplat obtains 25.58 dB PSNR and 0.848 SSIM, outperforming pixelSplat and MeshSplat on these two metrics but trailing the strongest appearance-oriented baselines. Its LPIPS of 0.158 is also slightly worse than pixelSplat's 0.156. This comparison indicates that GeoSupSplat's main advantage lies in surface reconstruction and NVS through the recovered mesh rather than direct RGB rendering alone.

A.9 Depth and Depth-to-Normal Benchmark

We evaluate the predicted geometry before surface extraction on 1,000 test samples each from ScanNet Dai et al. (2017) and Replica Straub et al. (2019). All methods receive the same two calibrated context views, and their rendered depth maps are evaluated without mesh extraction or test-time refinement. Because these feed-forward models can predict geometry under different global depth

scales and offsets, we omit metric-depth results and report both scale-invariant (SI) and scale-and-shift-invariant (SSI) depth metrics. For each alignment protocol, we measure AbsRel and δ_1.

For normal evaluation, we derive reference normals from ground-truth depth and predicted normals from rendered depth using the same camera-aware depth-to-normal operator. We report the mean angular error and the fraction of pixels whose angular error is below 11.25°. The results are summarized in Tab. 13.

Table 13: **Scale-ambiguous depth and depth-to-normal estimation on ScanNet and Replica.** We sample 1,000 test examples from each dataset and report depth after scale-only (SI) and scale-and-shift (SSI) alignment; metric-depth results are omitted. Reference normals are derived from ground-truth depth, while predicted normals are derived from rendered depth using the same depth-to-normal operator. The best and second-best results within each dataset are shown in bold and underlined, respectively.

	SI Depth		SSI Depth		Depth-to-Normal	
Method	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	Mean (°) ↓	11.25° ↑
ScanNet ($n = 1000$)						
pixelSplat Charatan et al. (2024)	0.255	0.660	0.194	0.689	84.46	0.005
MVSplat Chen et al. (2024)	0.175	0.724	0.176	0.725	61.39	0.027
DepthSplat Xu et al. (2025)	<u>0.125</u>	<u>0.848</u>	<u>0.100</u>	<u>0.908</u>	<u>47.57</u>	<u>0.064</u>
MeshSplat Chang et al. (2026)	0.151	0.781	0.137	0.824	51.98	0.042
GeoSupSplat (ours)	**0.053**	**0.963**	**0.051**	**0.971**	**30.83**	**0.159**
Replica ($n = 1000$)						
pixelSplat Charatan et al. (2024)	0.373	0.529	0.393	0.517	80.56	0.012
MVSplat Chen et al. (2024)	0.357	0.550	0.353	0.548	57.68	0.037
DepthSplat Xu et al. (2025)	<u>0.267</u>	<u>0.702</u>	<u>0.167</u>	<u>0.843</u>	<u>42.45</u>	<u>0.116</u>
MeshSplat Chang et al. (2026)	0.318	0.599	0.269	0.648	49.33	0.053
GeoSupSplat (ours)	**0.077**	**0.954**	**0.093**	**0.951**	**12.95**	**0.638**

GeoSupSplat performs best across all depth and normal metrics on both datasets. Compared with the second-best DepthSplat, it reduces SI-AbsRel from 0.125 to 0.053 on ScanNet and from 0.267 to 0.077 on Replica, while improving SSI-δ_1 from 0.908 to 0.971 and from 0.843 to 0.951, respectively. The improvement is particularly clear for local orientation: the mean normal error decreases from 47.57° to 30.83° on ScanNet and from 42.45° to 12.95° on Replica. These results show that the learned geometry retains both accurate relative depth and local surface orientation before mesh extraction. We provided the qualitative visualizations in the Fig. 12 and Fig. 13.

A.10 Limitations and Future Work

Our current study focuses on the sparse two-view setting ($N = 2$). Although the framework is not restricted to two inputs and can be extended to denser view configurations, systematically studying this setting is left for future work. In addition, the optional camera head does not yet match the accuracy of provided camera poses, leading to a moderate drop in mesh-rendering quality when calibration is unavailable as shown in the Tab. 10. Future work may improve camera prediction by increasing the supervision weight or adopting a stronger pose objective. Finally, GeoSupSplat does not achieve the strongest direct NVS performance among the compared methods as shown in the Tab. 12. Improving appearance prediction while retaining its geometric accuracy is therefore another important direction.

A.11 More visualization results



Figure 5: **Qualitative TSDF-based surface reconstruction on Re10K, Scannet, Replica from two context views.**



Figure 6: **Qualitative TSDF-based surface reconstruction on Scannet, Replica from two context views.**

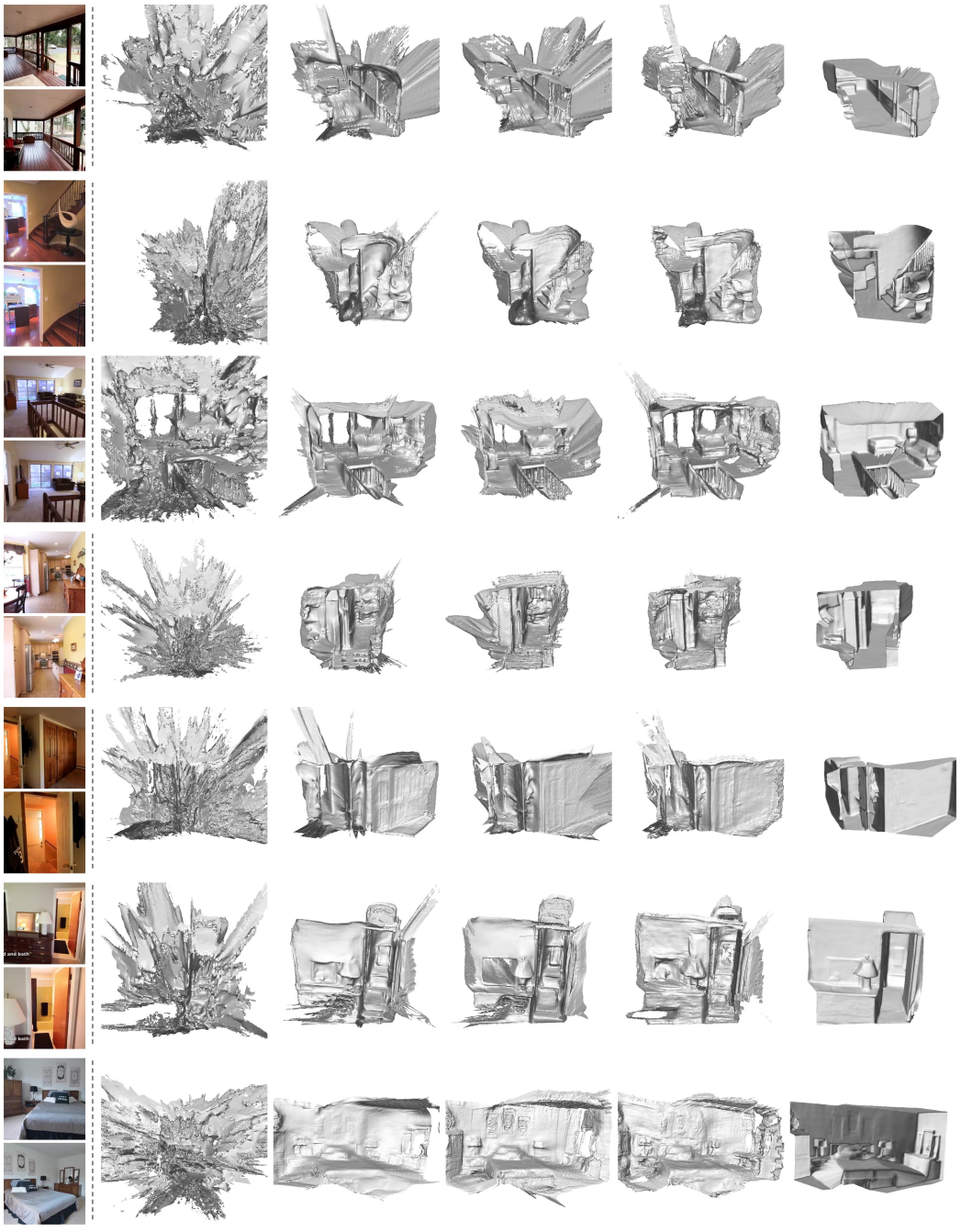

Figure 7: **Qualitative PGM-based surface reconstruction on Re10K from two context views.**

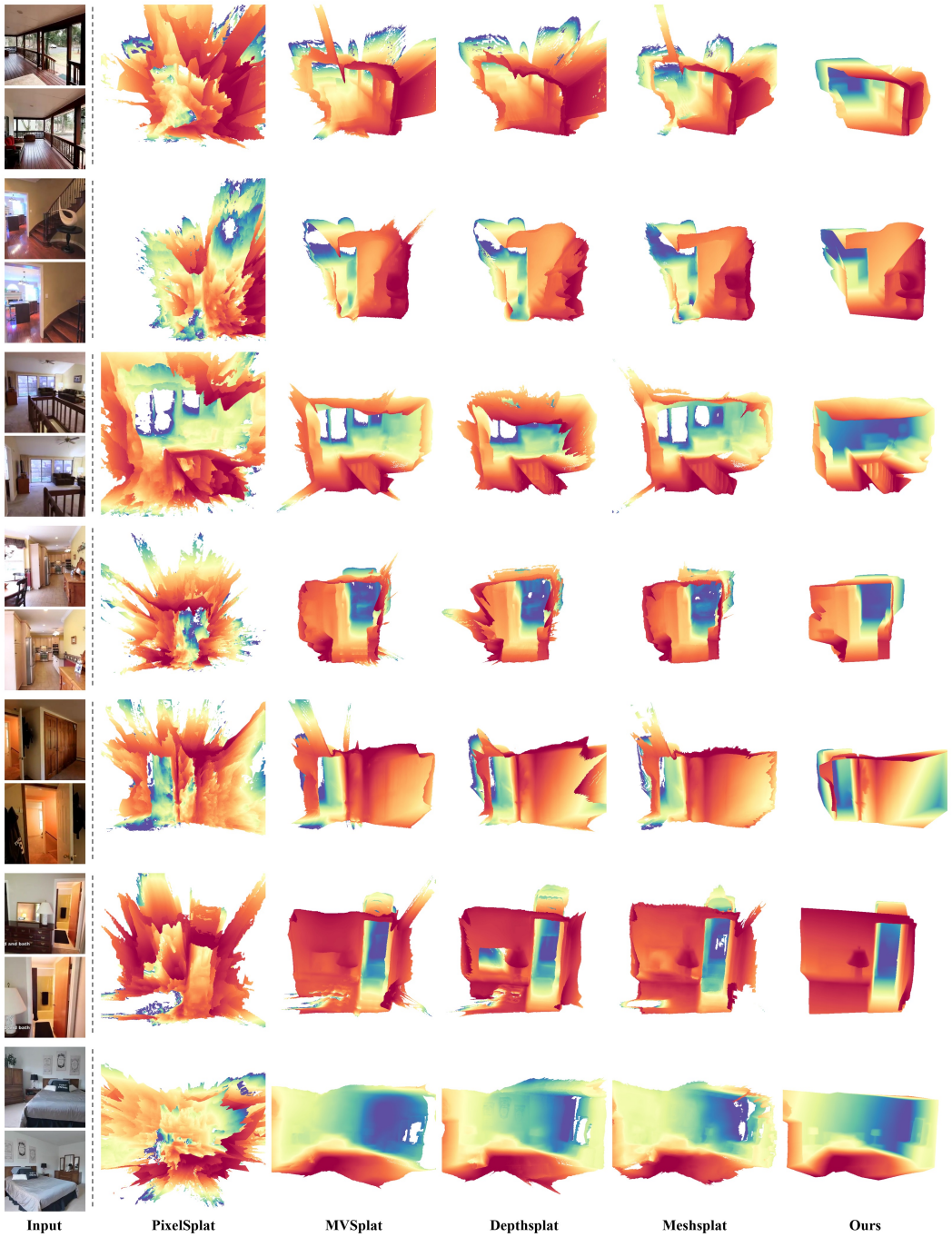

Figure 8: **Qualitative depth from mesh on Re10K, Scannet, Replica from two context views.**

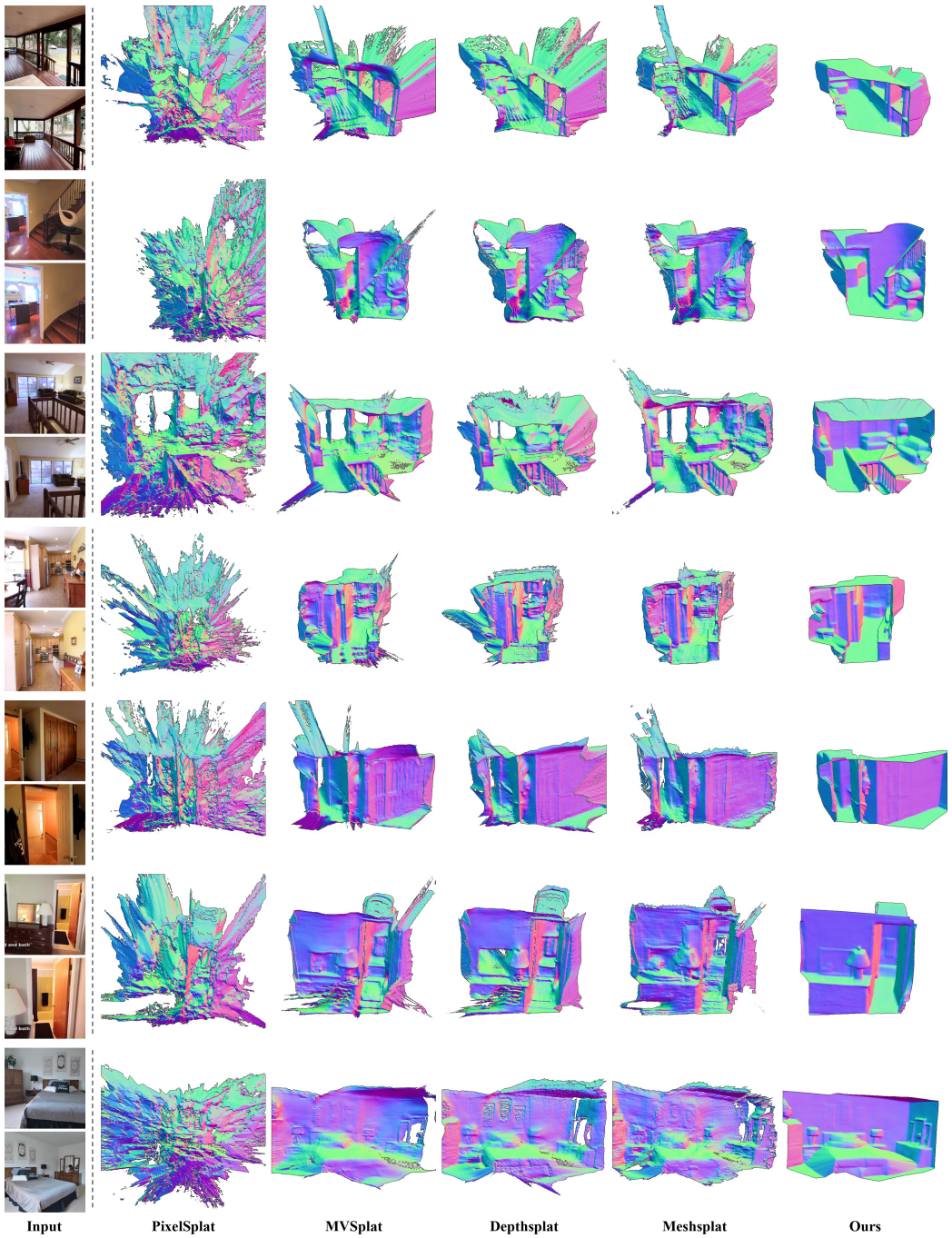

Figure 9: **Qualitative depth-to-normal from mesh on Re10K, Scannet, Replica from two context views.**

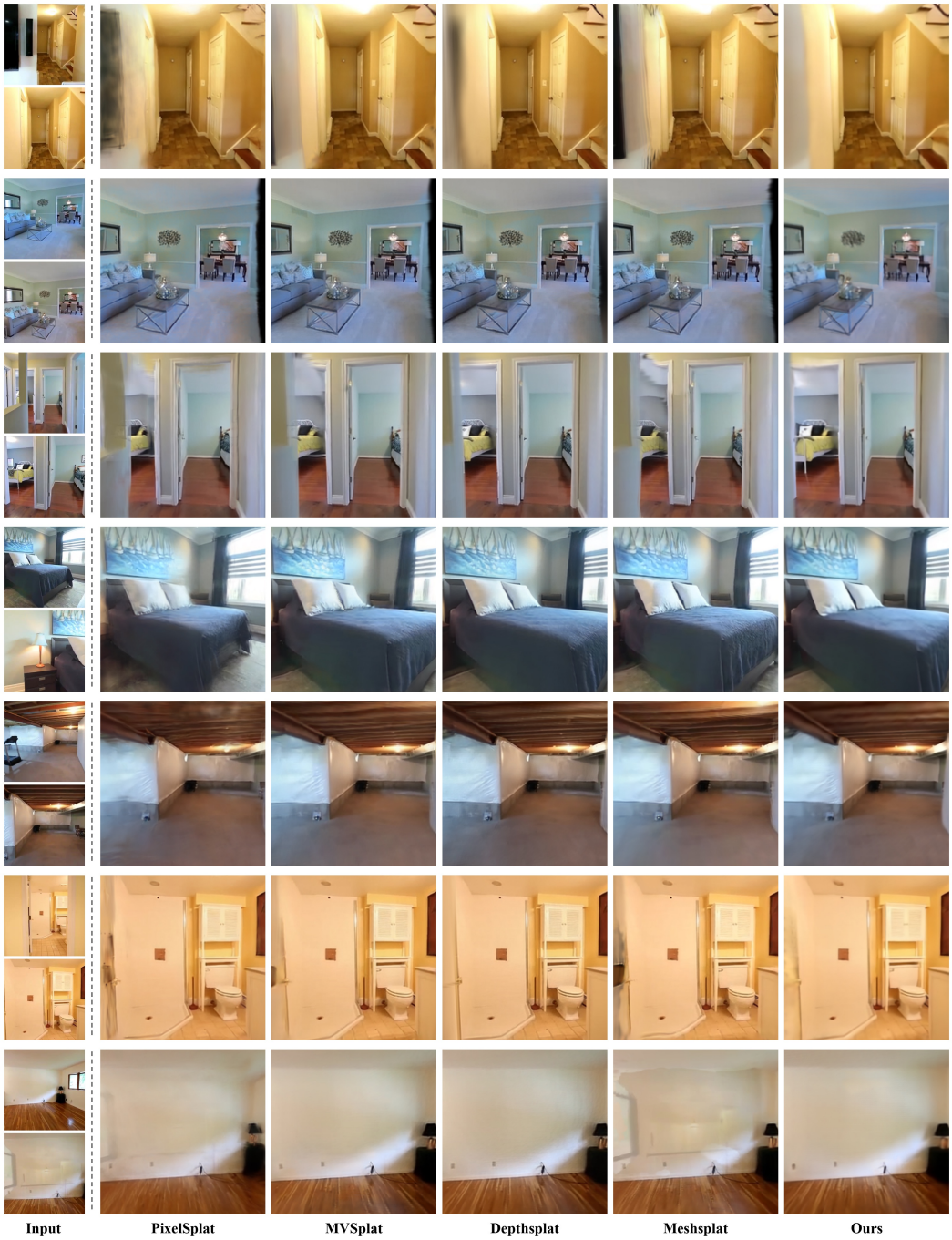

Figure 10: **Qualitative New Views Synthesis on Re10K, Scannet, Replica from two context views.**

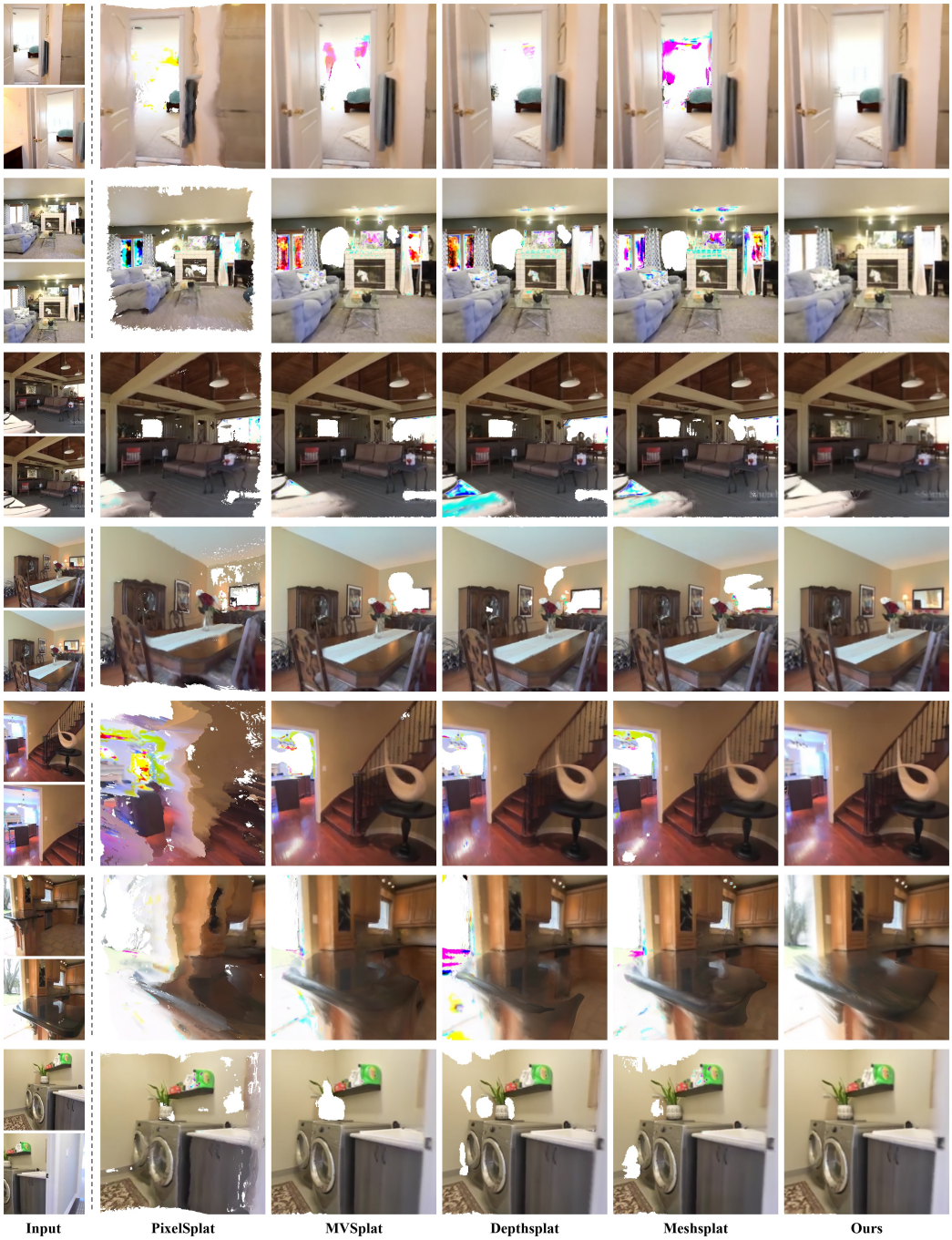

Figure 11: **Qualitative Mesh New Views Synthesis on Re10K, Scannet, Replica from two context views.**

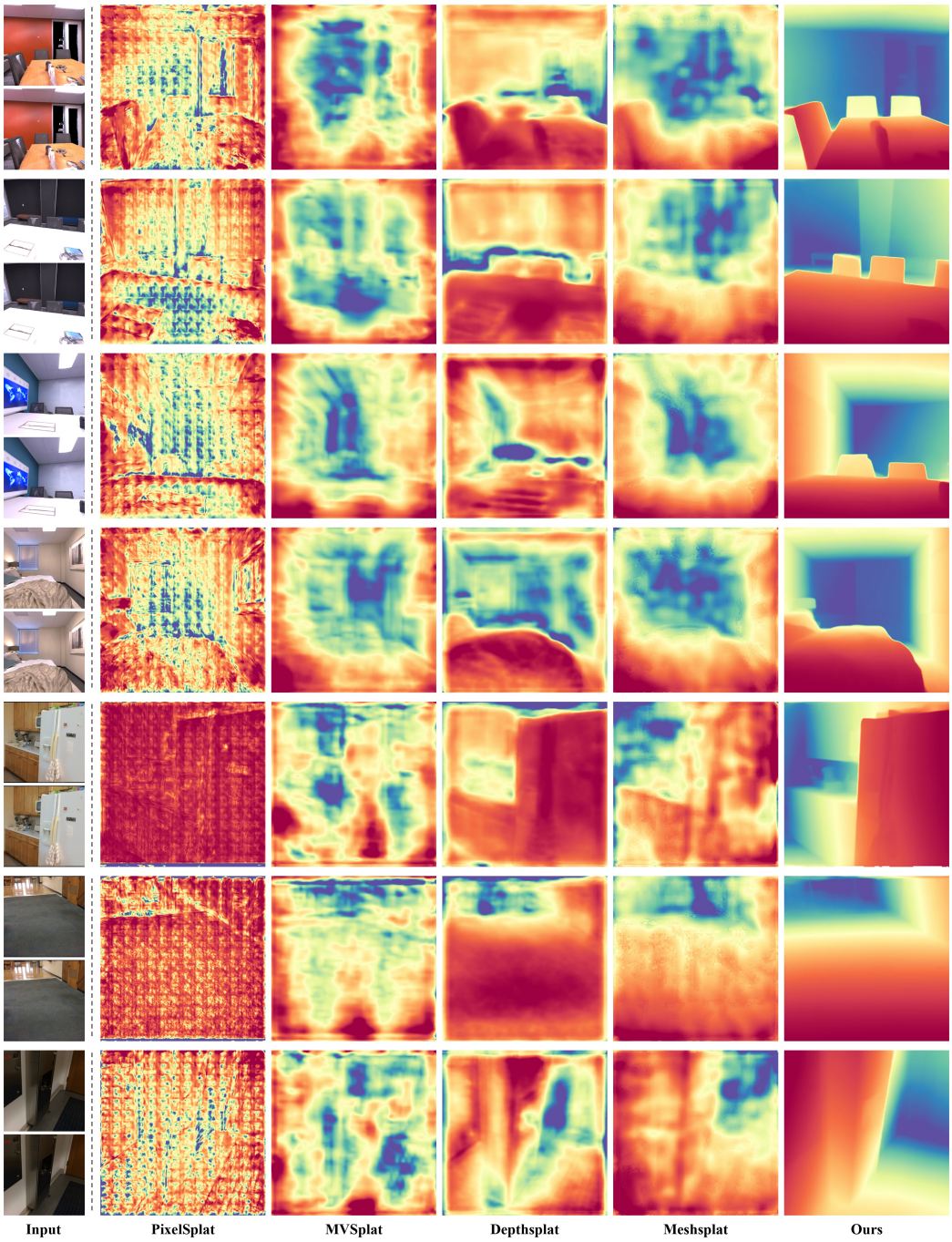

Figure 12: **Qualitative depth render on Scannet, Replica from two context views.**

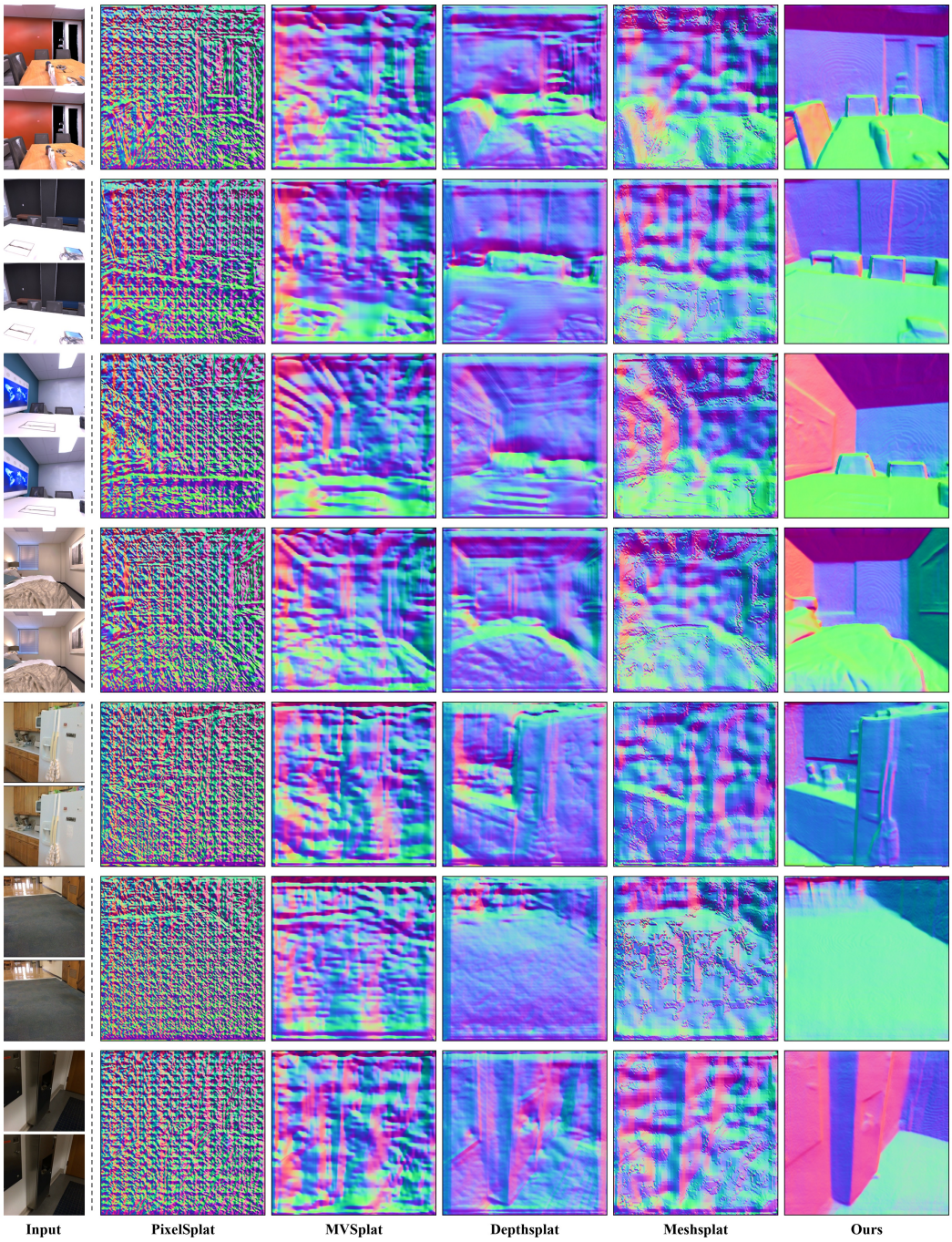

Figure 13: **Qualitative depth-to-normal render on Scannet, Replica from two context views.**