# CoSplat: Complementary Surface Geometry Learning for Generalizable Surface Reconstruction

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

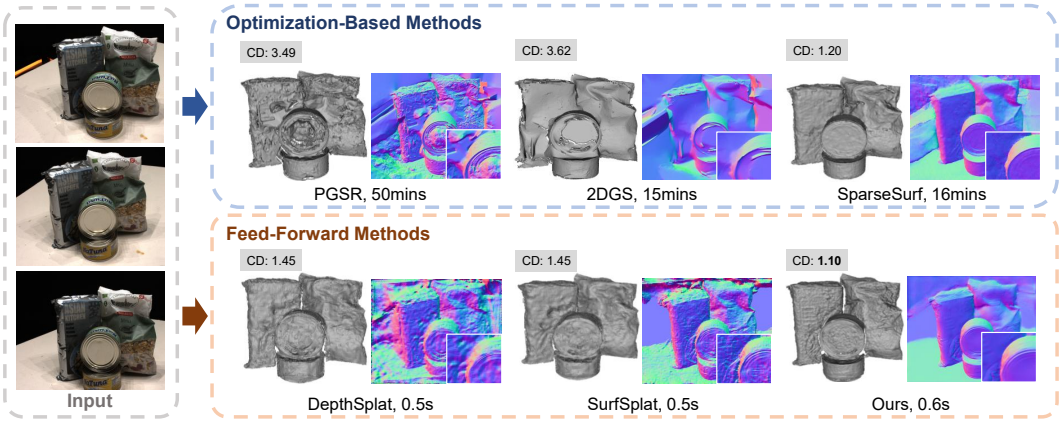



Figure 1: **Comparison of representative methods.** CoSplat reconstructs high-quality surfaces with continuous geometry and preserved details in a single feed-forward pass.

## Abstract

Feed-forward Gaussian Splatting enables efficient surface reconstruction from sparse views, yet the geometry of its predicted primitives remains ambiguous under rendering-driven supervision. Specifically, distinct primitive configurations can produce similar renderings, as geometric errors can be compensated for by adjustments to appearance parameters. To resolve this ambiguity, we propose CoSplat, a feed-forward 2DGS framework that explicitly constrains surfel geometry. We first introduce Geometry-Calibrated Normal Supervision to directly regularize primitive orientations and mitigate geometric drift caused by geometry-appearance compensation. Since local normal constraints are insufficient to recover globally coherent geometry from sparse views, we further propose Confidence-Guided Tangent-Plane Depth Propagation, which translates reliable local orientations into surface-aligned depth hypotheses and validates them using multi-view geometric evidence, thereby promoting globally coherent surface reconstruction. Extensive experiments on DTU, BlendedMVS and ETH3D demonstrate that CoSplat achieves state-of-the-art surface reconstruction accuracy among feed-forward Gaussian methods while retaining efficient inference.

## 1 Introduction

Reconstructing accurate 3D surfaces from multi-view images is a long-standing problem in computer vision and graphics. Conventional approaches, whether modeling scenes as neural implicit fields such as NeRF (Mildenhall et al., 2020), VolSDF (Yariv et al., 2021), and NeuS (Wang et al., 2021), or utilizing explicit 3DGS (Kerbl et al., 2023) and its surface-oriented variants such as 2DGS (Huang et al., 2024a) and PGSR (Chen et al., 2025), fundamentally rely on per-scene



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.

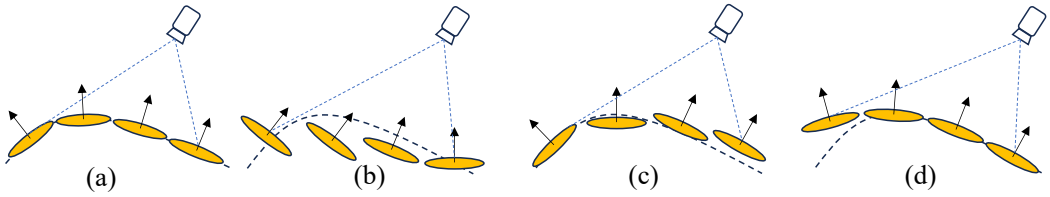



Figure 2: **Geometric ambiguity in rendering-oriented feed-forward reconstruction.** (a) The ground-truth surface; (b) underconstrained orientations biased toward the camera; (c) supervised by monocular normals but misaligned with the reconstructed geometry; (d) locally self-consistent distortions caused by inaccurate depth estimates. Figure 8 provides additional details.

optimization. Despite achieving high-fidelity geometry, these per-scene methods require time-consuming optimization for every new scene, severely limiting their scalability. To overcome the per-scene bottleneck, feed-forward frameworks have emerged. Unlike early implicit methods such as VolRecon (Ren et al., 2023) and ReTR (Liang et al., 2023) that retain costly volumetric inference, recent explicit Gaussian methods such as pixelSplat, MVSplat, DepthSplat, and HiSplat (Charatan et al., 2024; Chen et al., 2024; Xu et al., 2025; Tang et al., 2025) bypass per-scene optimization entirely by directly predicting primitives. However, their rendering-driven supervision leaves primitive geometry ambiguous, particularly under sparse input views.

An accurate surface intrinsically requires surfels to possess geometrically correct orientations and spatially coherent centers, while preserving genuine discontinuities and fine structures. However, without explicit geometric constraints, rendering-driven supervision allows attributes such as appearance, opacity, and scale to freely compensate for geometric errors. As illustrated in Fig. 2(b), surfels can simply face the input camera while still reproducing similar images. Thus, photometrically equivalent renderings do not necessarily imply geometrically equivalent surfaces.

To constrain this under-determined geometry, existing methods introduce additional geometric cues. MeshSplat (Chang et al., 2026) supervises predicted surfel normals using a pretrained monocular normal model, providing independent local shape cues when sparse-view image evidence is insufficient. However, a pixel-wise normal constrains orientation along a camera ray but leaves the surface position on that ray undetermined, so correctly oriented surfels may still form a fragmented surface, as in Fig. 2(c). In contrast, SurfSplat (He et al., 2026) derives local surface constraints from predicted depth to determine surfel rotations. Because these constraints are coupled with the predicted centers, they encourage positional continuity and remain aligned with the current reconstruction. Nevertheless, inaccurate or over-smoothed depth can yield a distorted yet locally self-consistent surface, as in Fig. 2(d). The desired reconstruction must instead combine accurate orientations with consistent positions to recover the surface in Fig. 2(a). For Gaussian surfels, the center and orientation of each surfel jointly define a local tangent plane. Reliable normals therefore provide the bridge between accurate orientation and positional coherence.

Although both monocular and depth-derived cues constrain surface normals, they play fundamentally distinct roles in feed-forward reconstruction. Depth-derived local geometry is intrinsically coupled with the predicted 3D positions, making it ideal for establishing a robust local reference frame. In contrast, monocular normal priors are estimated independently of the recovered multi-view geometry. Rather than overriding this geometric base, they excel at capturing challenging details that the reconstruction itself fails to resolve. Treating these two cues as interchangeable full-field targets ignores their asymmetric contributions. This forces the network to compromise between incompatible absolute orientations. This raises a fundamental question: **Can a feed-forward model incorporate monocular details while preserving consistency with the reconstructed geometry?**

To answer this, we present CoSplat, a feed-forward 2DGS surface reconstruction framework. We introduce Geometry-Calibrated Normal Supervision (GCNS), which decomposes normal supervision according to the asymmetric roles of the two cues: the depth-derived frame anchors the absolute orientation, while the monocular prior contributes solely local angular detail. This asymmetric supervision yields detail-preserving normals compatible with the predicted surfel positions. Using these normals, we further design Confidence-Guided Tangent-Plane Depth Propagation (TDP) to constrain surfel positions by filtering propagated depth hypotheses with geometric confidence and

validating them using multi-view evidence, thereby promoting complete and accurate surface reconstruction. Our contributions are summarized as follows:

- We propose **CoSplat**, a feed-forward 2DGS framework that couples complementary constraints on surfel orientations and positions to improve sparse-view surface reconstruction.
- We introduce **GCNS**, **an asymmetric normal supervision scheme** that calibrates the monocular normal prior to the geometry-derived reference while preserving both the geometric base and monocular detail. The learned normals further constrain surfel positions through **TDP**.
- We conduct experiments on DTU, BlendedMVS and ETH3D datasets. CoSplat achieves state-of-the-art among feed-forward Gaussian methods while retaining high efficiency.

## 2 Related Work

**Per-scene 3D Reconstruction.** Classical systems such as COLMAP (Schönberger et al., 2016) use multi-view stereo geometry to estimate and fuse depth maps. NeRF (Mildenhall et al., 2020) represents scenes with implicit neural fields. Subsequent methods adapt implicit fields to surface reconstruction (Yariv et al., 2021; Wang et al., 2021). 3DGS (Kerbl et al., 2023) uses explicit Gaussian primitives for efficient optimization and rendering, and SuGaR, 2DGS, PGSR, Gaussian Surfels, and GOF (Guédon & Lepetit, 2024; Huang et al., 2024a; Chen et al., 2025; Dai et al., 2024; Yu et al., 2024) adapt Gaussian representations to surface extraction. Although their representations differ, these methods require sufficient observations and per-scene computation to reconstruct each scene independently.

**Sparse-view Surface Reconstruction.** With sparse inputs, incomplete correspondence leaves per-scene reconstruction underconstrained. Additional priors regularize implicit reconstruction from few views (Jain et al., 2021; Niemeyer et al., 2022; Yang et al., 2023; Wang et al., 2023; Huang et al., 2024b). DNGaussian, CoR-GS, FSGS, Sparse2DGS, FatesGS, and SparseSurf (Li et al., 2024; Zhang et al., 2024; Zhu et al., 2024; Wu et al., 2025; Huang et al., 2025; Gu et al., 2026) regularize sparse-view Gaussian optimization with depth, consistency, or geometric priors. These priors reduce sparse-view degeneracy, but optimization can still overfit the observed cameras. The resulting geometry may explain the input images without matching the underlying surface.

**Generalizable 3D Reconstruction.** Feed-forward methods learn cross-scene priors from large multi-scene datasets. These priors support efficient inference and multi-view-consistent geometry under sparse observations. MVSNet, TransMVSNet, and MonoMVSNet (Yao et al., 2018; Ding et al., 2022; Jiang et al., 2025) predict multi-view depth maps for surface fusion. Other feed-forward methods infer implicit geometry from image features (Ren et al., 2023; Liang et al., 2023; Na et al., 2024). Splatter Image, pixelSplat, MVSplat, DepthSplat, HiSplat, and MonoSplat (Szymanowicz et al., 2024; Charatan et al., 2024; Chen et al., 2024; Xu et al., 2025; Tang et al., 2025; Liu et al., 2025) directly predict Gaussian representations. IDESplat refines depth estimates, while MeshSplat and SurfSplat target feed-forward surface reconstruction (Long et al., 2026; Chang et al., 2026; He et al., 2026). Rendering-oriented Gaussian predictors can compensate for inaccurate centers or orientations through appearance, opacity, and covariance, producing high-fidelity images but blurred or fragmented surfaces. CoSplat instead learns surface orientations. It uses them to propagate and refine depth with multi-view evidence.

## 3 Method

### 3.1 Overview

Given a set of $V$ calibrated sparse-view images $\mathcal{I} = \{\mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}\}_{i=1}^{V}$, let their camera projection matrices be $\mathcal{P} = \{\mathbf{P}_i = \mathbf{K}_i[\mathbf{R}_i \mid \mathbf{t}_i]\}_{i=1}^{V}$, where $\mathbf{K}_i$ is the intrinsic matrix and $\mathbf{R}_i, \mathbf{t}_i$ specify the camera pose. We learn a feed-forward reconstruction network $f_\theta$ that maps the calibrated observations to a set of 2D Gaussian surfels,

$$f_\theta(\mathcal{I}, \mathcal{P}) \longrightarrow \mathcal{G} = \{(\boldsymbol{\mu}_k, \mathbf{R}_k, \mathbf{s}_k, \alpha_k, \mathbf{c}_k)\}_{k=1}^{M}. \tag{1}$$

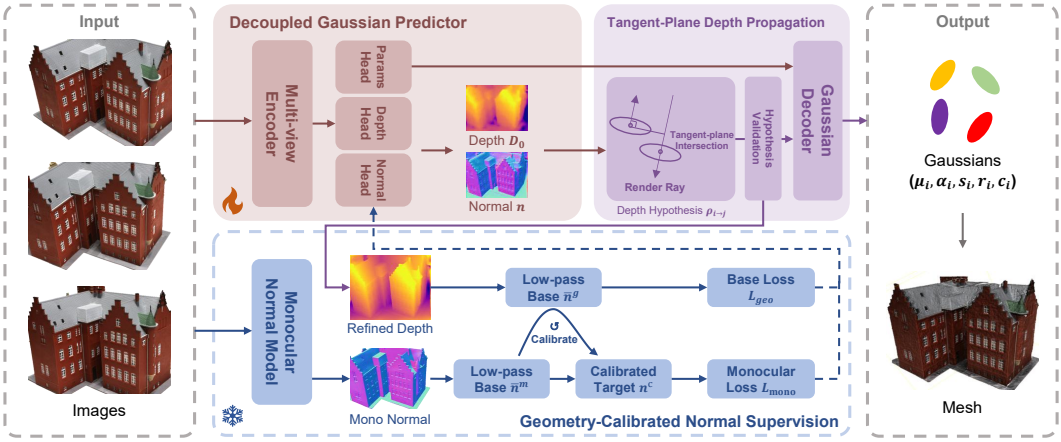



Figure 3: **Overview of CoSplat.** CoSplat uses a multi-view encoder with GCNS and TDP to predict Gaussian attributes in a single forward pass. (a) **Geometry-Calibrated Normal Supervision** uses a geometry-derived local orientation base and transfers the monocular angular residual to predict surfel normals. (b) **Confidence-Guided Tangent-Plane Depth Propagation** combines the learned normals with coarse depth to construct surface-aligned hypotheses. It validates and aggregates these hypotheses to produce refined depth.

For the $k$-th surfel, $\boldsymbol{\mu}_k \in \mathbb{R}^3$ denotes its center, $\mathbf{R}_k \in \mathrm{SO}(3)$ its orientation, $\mathbf{s}_k \in \mathbb{R}^2_+$ its in-plane scale, $\alpha_k$ its opacity, and $\mathbf{c}_k$ its spherical harmonics. Fig. 3 summarizes the pipeline. Following DepthSplat Xu et al. (2025), we use a CNN–Transformer multi-view encoder. The encoder predicts per-view depth, normals, and the remaining pixel-aligned Gaussian attributes. Camera parameters convert the depth and normals into Gaussian centers and rotations. GCNS improves their geometric consistency using DSINE Bae & Davison (2024) as a frozen monocular normal prior to learn surfel normals compatible with the reconstructed positions. TDP then combines the normals with coarse depth and constructs surface-aligned depth hypotheses for validation. The refined depth determines the surfel centers used for the final reconstruction.

### 3.2 Geometry-Calibrated Normal Supervision

Rendering supervision alone weakly constrains Gaussian surfel orientations, motivating explicit orientation supervision. We therefore introduce a dedicated normal predictor to estimate normals from the input views. Its supervision must preserve local detail while remaining compatible with the reconstructed geometry. Monocular normal priors provide fine-grained local shape cues in photometrically challenging regions. In contrast, depth-derived local geometry provides a local orientation base tied to the predicted surface positions. We treat these cues as complementary rather than interchangeable supervision targets. GCNS resolves disagreement between their local orientation bases: it retains the geometry-derived base and transfers only the angular residual of the monocular prior.

**Plane-consistent geometric reference construction and supervision.** For a pixel $p$, we obtain its Gaussian center $\mathbf{P}_p$ by back-projecting the predicted depth along the corresponding camera ray. We estimate two local tangent directions from valid horizontal and vertical neighbors and compute the geometry-derived normal as

$$\mathbf{n}_p^g = \mathrm{normalize}\Big(\big(\mathbf{P}_{p_x^+} - \mathbf{P}_{p_x^-}\big) \times \big(\mathbf{P}_{p_y^+} - \mathbf{P}_{p_y^-}\big)\Big). \tag{2}$$

Neighbors crossing depth discontinuities are excluded to prevent local planes from spanning different surfaces. For a normal field $\mathbf{n}$ and reliability $w$, we define the image-guided low-pass operator

$$\mathcal{B}(\mathbf{n}; w)_p = \mathrm{normalize}\left(\frac{\sum_{q\in\mathcal{N}(p)} a_{pq} w_q \mathbf{n}_q}{\sum_{q\in\mathcal{N}(p)} a_{pq} w_q}\right), \quad a_{pq} = k^s_{pq} \exp\left(-\frac{\|\mathbf{I}_p - \mathbf{I}_q\|_1}{3\tau_c}\right), \tag{3}$$

where $k^s_{pq}$ is a spatial kernel. The associated support is $\mathcal{S}(w)_p = \sum_q a_{pq} w_q / \sum_q a_{pq}$. We construct the depth reliability $w^d$ by masking invalid neighbors and depth discontinuities and down-weighting

uncertain depth estimates. Applying the operator gives the local orientation base $\overline{\mathbf{n}} = \mathcal{B}(\mathbf{n}; w^d)$ and the geometry-derived base $\overline{\mathbf{n}}^g = \mathcal{B}(\mathbf{n}^g; w^d)$, which serves as the geometric reference for both anchor supervision and monocular calibration. We set $w^g = \mathcal{S}(w^d)$ and constrain the coarse normal orientation with the anchor loss

$$\mathcal{L}_{\text{geo}} = \frac{\sum_p w_p^g \left(1 - \overline{\mathbf{n}}_p \cdot \overline{\mathbf{n}}_p^g\right)}{\sum_p w_p^g}. \tag{4}$$

**Reference-calibrated monocular normal supervision.** Let $\mathbf{n}^m$ be the prediction of a frozen monocular normal prior and $m^m$ its validity mask. We define the monocular base as $\overline{\mathbf{n}}^m = \mathcal{B}(\mathbf{n}^m; m^m)$. Using $\mathbf{n}^m$ directly as another absolute target can conflict with the geometry-derived base. We instead compute the minimal rotation $\mathbf{A}_p$ that aligns the monocular base with the geometry-derived base and apply the same rotation to the full monocular normal:

$$\mathbf{A}_p = \text{Rot}_{\min}\left(\overline{\mathbf{n}}_p^m \to \overline{\mathbf{n}}_p^g\right), \qquad \mathbf{n}_p^c = \text{normalize}(\mathbf{A}_p \mathbf{n}_p^m). \tag{5}$$

Because rotations preserve inner products and $\mathbf{A}_p \overline{\mathbf{n}}_p^m = \overline{\mathbf{n}}_p^g$, the angular residual relative to the local base is preserved exactly:

$$d_{\mathbb{S}^2}(\mathbf{n}_p^c, \overline{\mathbf{n}}_p^g) = d_{\mathbb{S}^2}(\mathbf{n}_p^m, \overline{\mathbf{n}}_p^m), \qquad d_{\mathbb{S}^2}(\mathbf{a}, \mathbf{b}) = \arccos(\mathbf{a}^\top \mathbf{b}). \tag{6}$$

Thus, $\mathbf{n}^c$ is obtained by calibrating the monocular normal to the geometry-derived base while retaining its angular residual. Its weight is $w_p^c = m_p^{\text{fg}} \min\{w_p^g, \mathcal{S}(m^m)_p\}$. We supervise the full-resolution normal using a robust angular loss,

$$\mathcal{L}_{\text{mono}} = \frac{\sum_p w_p^c \left(\sqrt{\left(1 - \mathbf{n}_p \cdot \mathbf{n}_p^c\right)^2 + \epsilon^2} - \epsilon\right)}{\sum_p w_p^c}, \tag{7}$$

where $\mathbf{n}_p^c$ is the calibrated target, $w_p^c$ masks and weights pixels supported by both cues, and $\epsilon$ is a small constant that makes the angular penalty robust near zero. The complete normal supervision combines both objectives as $\mathcal{L}_{\text{normal}} = \mathcal{L}_{\text{geo}} + \mathcal{L}_{\text{mono}}$. The resulting field preserves the monocular detail and remains compatible with the reconstructed geometry. It orients the predicted 2D Gaussian surfels.

### 3.3 Confidence-Guided Tangent-Plane Depth Propagation

GCNS provides orientations compatible with the reconstructed geometry, but orientations alone do not ensure a coherent surface. Independently predicted Gaussian centers may remain inconsistent with the recovered normal field. We propagate local tangent-plane hypotheses to neighboring rendering rays and weight them using chained normal and cross-view depth reprojection confidence. Multi-view matching then refines the surface-aligned depth candidates.

**Normal and reprojection confidence.** Fig. 4 illustrates the construction. For each target pixel $p$, we collect supports $q_r = p + r\boldsymbol{\delta}$ along the eight horizontal, vertical, and diagonal directions at radii $r \in \{1, 2\}$. With $q_t = p + t\boldsymbol{\delta}$, the chained normal confidence along the resulting straight path is

$$c^n_{q_r \to p} = \prod_{t=1}^{r} \left[(\mathbf{n}_{q_{t-1}})^\top \mathbf{n}_{q_t}\right]_+ \prod_{t=1}^{r} m_{q_t}, \tag{8}$$

where $[\cdot]_+$ clamps adjacent-normal agreement to $[0, 1]$, and $m$ masks invalid path elements. Links crossing a depth discontinuity or producing an invalid plane–ray intersection are rejected, and a rejected near link blocks farther supports along the same direction.

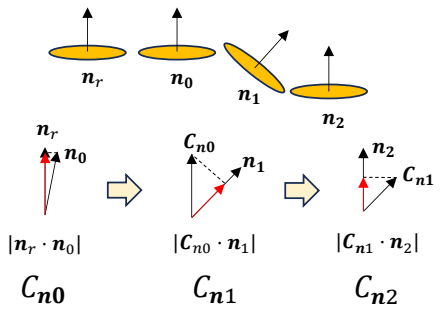



Figure 4: **Illustration of Chained Normal Confidence.**

Chained normal confidence measures path-wise propagation validity, but it does not assess the cross-view consistency of the supporting depth. We project each propagated 3D point from reference view

$r$ into a neighboring view $t$. We then sample and back-project its coarse depth to compute the forward–backward pixel error $e_{r\leftrightarrow t}$. The corresponding reprojection confidence is

$$c^{\text{repr}}_{r\leftrightarrow t}(\mathbf{p}_r) = \begin{cases} \exp(-e_{r\leftrightarrow t}(\mathbf{p}_r)/\tau_{\text{repr}}), & e_{r\leftrightarrow t}(\mathbf{p}_r) \le \epsilon_{\text{repr}}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Invalid projections receive zero confidence. We take the maximum over the selected neighboring views. Because the backward projection uses an independently predicted source-view depth, the error measures cross-view surface agreement. The voting weight for each propagated hypothesis is the product of its chained normal and depth reprojection confidences.

**Tangent-Plane Depth Propagation.** For each valid support $q$, its coarse depth $d_q$ and predicted normal $\mathbf{n}_q$ define a local tangent plane through $d_q\mathbf{r}_q$, where $\mathbf{r}_x = \mathbf{K}^{-1}\widetilde{\mathbf{x}}$ is the camera ray at pixel $x$. Intersecting this plane with the target ray $\mathbf{r}_p$ propagates the local surface geometry from $q$ to $p$, yielding a surface-aligned inverse-depth hypothesis

$$\rho_{q\to p} = \rho_q \frac{(\mathbf{n}_q)^\top \mathbf{r}_p}{(\mathbf{n}_q)^\top \mathbf{r}_q}, \qquad \rho = \frac{1}{d}. \tag{10}$$

Together with the coarse inverse depth, the valid propagated hypotheses bound a local range. We uniformly sample $K$ candidates from this range. Each hypothesis casts a linearly interpolated vote weighted by its combined confidence, while the coarse depth contributes an additional vote. The accumulated votes define the propagation prior and support over the candidate domain.

**Depth hypothesis validation and refinement.** Following the cost-volume construction in our multi-view encoder, we warp source-view features at each candidate depth and aggregate their group-wise correlations into a local matching volume. A lightweight 3D CNN combines this multi-view evidence with the propagation prior to predict a residual depth and update the coarse depth.

### 3.4 Loss Functions

The entire training is divided into two stages. We first pretrain the model on the large-scale RE10K (Zhou et al., 2018) dataset to learn generalizable appearance and geometry priors. The pretraining objective is

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{mse}} + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}} + \lambda_{\text{normal}}\mathcal{L}_{\text{normal}}, \tag{11}$$

where $\mathcal{L}_{\text{mse}}$ is the mean squared error loss, $\mathcal{L}_{\text{lpips}}$ is the perceptual similarity loss (Zhang et al., 2018), and $\mathcal{L}_{\text{normal}}$ is defined in Sec. 3.2. The weight $\lambda_{\text{lpips}}$ is set to 0.05, and $\lambda_{\text{normal}}$ is set to 0.01.

We then perform a short fine-tuning stage on the DTU (Jensen et al., 2014) training set to adapt the model to the scene scale of DTU. We also use the ground-truth depth to compute an $L_1$ depth loss. The fine-tuning loss is $\mathcal{L}_{\text{fine}} = \mathcal{L}_{\text{pre}} + \mathcal{L}_{\text{depth}}$.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** RealEstate10K (RE10K) (Zhou et al., 2018) is a large-scale real-world video dataset collected from YouTube. It contains diverse indoor and outdoor scenes, with 67,477 training scenes and 7,289 testing scenes. We use the RE10K training set for large-scale pretraining at a resolution of $256\times256$. DTU (Jensen et al., 2014) is a calibrated, object-centric multi-view stereo benchmark with ground-truth geometry. We use its training scenes for fine-tuning and evaluate surface reconstruction on its testing scenes. We also provide cross-dataset generalization on ETH3D (Schöps et al., 2017) and BlendedMVS (Yao et al., 2020).

**Evaluation Metrics.** We report the Chamfer distance (CD, in millimeters) between points sampled from the reconstructed mesh and the ground-truth point cloud. We also report per-scene optimization time for optimization-based methods and network inference time for feed-forward methods.

Table 1: Quantitative results on DTU with 3 views setting (large overlap). The subscript "ft" denotes fine-tuning under the same setting as ours. Background colors indicate the best, second-best, and third-best results.

| Method | 24 | 37 | 40 | 55 | 63 | 65 | 69 | 83 | 97 | 105 | 106 | 110 | 114 | 118 | 122 | Mean | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COLMAP (Schönberger et al., 2016) | 0.90 | 2.89 | 1.63 | 1.08 | 2.18 | 1.94 | 1.61 | 1.30 | 2.34 | 1.28 | 1.10 | 1.42 | 0.76 | 1.17 | 1.14 | 1.52 | 1 min |
| TransMVSNet (Ding et al., 2022) | 1.07 | 3.14 | 2.39 | 1.30 | 1.35 | 1.61 | 0.73 | 1.60 | 1.15 | 0.94 | 1.34 | 0.46 | 0.60 | 1.20 | 1.46 | 1.36 | 0.4 s |
| MonoMVSNet (Jiang et al., 2025) | 0.66 | 1.95 | 1.33 | 1.01 | 1.69 | 1.95 | 1.45 | 1.48 | 1.28 | 1.65 | 1.04 | 0.51 | 0.55 | 1.34 | 1.21 | 1.27 | 0.3 s |
| 2DGS (Huang et al., 2024a) | 3.14 | 3.79 | 2.31 | 1.51 | 5.34 | 2.22 | 1.85 | 2.79 | 3.62 | 1.63 | 3.05 | 4.11 | 1.05 | 3.35 | 2.44 | 2.81 | 15 min |
| GOF (Yu et al., 2024) | 3.27 | 5.05 | 2.77 | 2.04 | 3.84 | 2.61 | 2.23 | 2.65 | 3.30 | 1.65 | 2.58 | 4.44 | 1.16 | 2.29 | 2.39 | 2.82 | 25 min |
| PGSR (Chen et al., 2025) | 3.36 | 3.28 | 2.75 | 1.27 | 5.15 | 1.84 | 0.88 | 1.79 | 3.49 | 1.19 | 1.86 | 1.11 | 0.61 | 1.09 | 1.52 | 2.08 | 50 min |
| Gaussian Surfel (Dai et al., 2024) | 1.71 | 3.21 | 2.50 | 1.59 | 3.25 | 2.72 | 1.85 | 3.36 | 2.57 | 2.00 | 2.45 | 2.50 | 1.22 | 1.71 | 2.08 | 2.31 | 8 min |
| Sparse2DGS (Wu et al., 2025) | 1.05 | 2.35 | 1.38 | 0.83 | 1.37 | 1.45 | 0.84 | 1.16 | 1.43 | 0.74 | 0.85 | 0.84 | 0.57 | 0.95 | 1.01 | 1.12 | 10 min |
| SparseSurf (Gu et al., 2026) | 0.72 | 1.61 | 1.17 | 0.72 | 1.11 | 1.13 | 0.57 | 1.30 | 1.20 | 0.67 | 0.70 | 0.45 | 0.36 | 0.70 | 0.87 | 0.89 | 16 min |
| VolRecon (Ren et al., 2023) | 1.20 | 2.59 | 1.56 | 1.08 | 1.43 | 1.92 | 1.11 | 1.48 | 1.42 | 1.05 | 1.19 | 1.38 | 0.74 | 1.23 | 1.27 | 1.38 | 70 s |
| ReTR (Liang et al., 2023) | 1.04 | 2.32 | 1.47 | 0.96 | 1.21 | 1.52 | 0.88 | 1.35 | 1.28 | 0.86 | 1.07 | 0.77 | 0.58 | 1.05 | 1.11 | 1.16 | 90 s |
| UFORecon (Na et al., 2024) | 0.77 | 2.06 | 1.44 | 0.86 | 1.09 | 1.15 | 0.74 | 1.22 | 1.12 | 0.69 | 0.87 | 0.59 | 0.56 | 0.90 | 1.01 | 1.00 | 120 s |
| DepthSplat$_{ft}$ (Xu et al., 2025) | 1.36 | 2.25 | 1.43 | 0.88 | 1.74 | 1.41 | 0.96 | 1.64 | 1.45 | 0.91 | 1.06 | 1.07 | 0.72 | 1.09 | 1.18 | 1.28 | 0.5 s |
| IDESplat$_{ft}$ (Long et al., 2026) | 1.32 | 3.27 | 1.60 | 0.89 | 2.31 | 1.57 | 0.94 | 1.81 | 1.33 | 1.15 | 1.05 | 0.99 | 0.70 | 1.00 | 1.17 | 1.41 | 0.9 s |
| SurfSplat$_{ft}$ (He et al., 2026) | 1.60 | 3.45 | 1.66 | 1.01 | 1.60 | 1.98 | 1.09 | 1.40 | 1.45 | 0.85 | 1.06 | 0.93 | 0.83 | 1.23 | 1.26 | 1.43 | 0.5 s |
| Ours | 0.74 | 1.96 | 1.21 | 0.83 | 1.03 | 1.16 | 0.63 | 0.97 | 1.10 | 0.71 | 0.81 | 0.47 | 0.44 | 0.85 | 0.96 | 0.92 | 0.6 s |

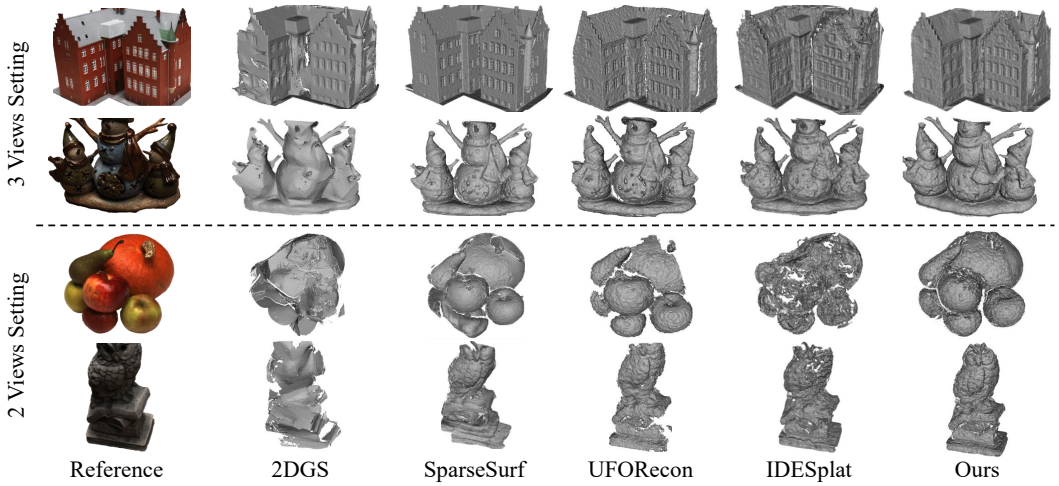



Figure 5: **Qualitative surface reconstruction results on DTU with 3 views (large overlap) and 2 views (little overlap) sparse setting.**

**Implementation Details.** We pretrain CoSplat on RE10K for 300k iterations with a batch size of 4 on a single NVIDIA A100 (40GB) GPU, then fine-tune it for 10k iterations on the DTU training set. All feed-forward GS methods use the same $512 \times 640$ input resolution, mesh extraction procedure, and evaluation pipeline. For the remaining methods, we retain their official evaluation pipelines. Since all feed-forward GS baselines are pretrained on RE10K, for a fair comparison, we report their results after fine-tuning under the same setting as our method and, where applicable, their zero-shot results with range-rescaled depth residuals. Appendix A.3 provides additional details.

### 4.2 Main Results

**Surface Reconstruction.** Table 1 compares sparse-view surface reconstruction on DTU across MVS-based, optimization-based GS, implicit, and feed-forward GS methods. CoSplat obtains a mean CD of 0.92. Compared with the per-view optimization method SparseSurf, CoSplat remains within 0.03 while avoiding its 16-minute per-scene optimization. CoSplat attains a lower mean CD than the implicit baselines while reducing inference time from 120 s to 0.6 s. Table 2 evaluates reconstruction using two views with little overlap. Optimization-based methods such as SparseSurf rely on COLMAP dense point cloud initialization. CoSplat avoids this dependency and reduces mean CD by 0.43.

**Cross-Dataset Generalization.** We evaluate the RE10K-pretrained model directly on DTU, BlendedMVS, and ETH3D without target-dataset fine-tuning. On DTU, our method achieves the lowest

Table 2: Quantitative results on DTU with 2 views setting (little overlap). The subscript "ft" denotes fine-tuning under the same setting as ours. Background colors indicate the best, second-best, and third-best results.

| Method | 24 | 37 | 40 | 55 | 63 | 65 | 69 | 83 | 97 | 105 | 106 | 110 | 114 | 118 | 122 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2DGS (Huang et al., 2024a) | 7.11 | 8.65 | 6.53 | 3.29 | 12.86 | 5.64 | 6.08 | 13.46 | 5.48 | 7.57 | 4.31 | 6.63 | 3.68 | 6.53 | 6.20 | 6.93 |
| Sparse2DGS (Wu et al., 2025) | 2.29 | 3.67 | 2.59 | 1.51 | 2.31 | 2.30 | 1.56 | 1.49 | 2.82 | 1.49 | 1.81 | 2.10 | 0.72 | 2.01 | 1.34 | 2.00 |
| SparseSurf (Gu et al., 2026) | 2.16 | 4.74 | 1.70 | 1.13 | 2.31 | 1.62 | 1.07 | 3.25 | 1.98 | 1.47 | 1.75 | 0.99 | 0.64 | 1.41 | 1.84 | 1.87 |
| UFORecon (Na et al., 2024) | 2.11 | 2.99 | 1.62 | 1.79 | 1.88 | 1.94 | 1.57 | 1.69 | 1.66 | 1.47 | 2.09 | 1.35 | 0.92 | 2.20 | 1.63 | 1.79 |
| DepthSplat$_{ft}$ (Xu et al., 2025) | 1.93 | 3.04 | 2.19 | 1.89 | 2.09 | 2.28 | 1.74 | 1.91 | 2.18 | 1.56 | 2.14 | 1.25 | 1.37 | 2.00 | 1.73 | 1.95 |
| SurfSplat$_{ft}$ (He et al., 2026) | 2.08 | 4.79 | 1.97 | 1.82 | 2.10 | 1.68 | 1.33 | 1.59 | 2.20 | 1.27 | 1.80 | 1.48 | 0.88 | 1.81 | 1.23 | 1.87 |
| IDESplat$_{ft}$ (Long et al., 2026) | 1.87 | 3.83 | 1.85 | 1.37 | 2.36 | 1.69 | 1.14 | 2.33 | 1.76 | 1.74 | 1.57 | 1.36 | 0.70 | 1.60 | 1.14 | 1.75 |
| Ours | 1.57 | 2.30 | 1.82 | 1.31 | 1.68 | 1.76 | 1.17 | 1.39 | 1.71 | 1.21 | 1.48 | 0.80 | 0.79 | 1.46 | 1.22 | 1.44 |

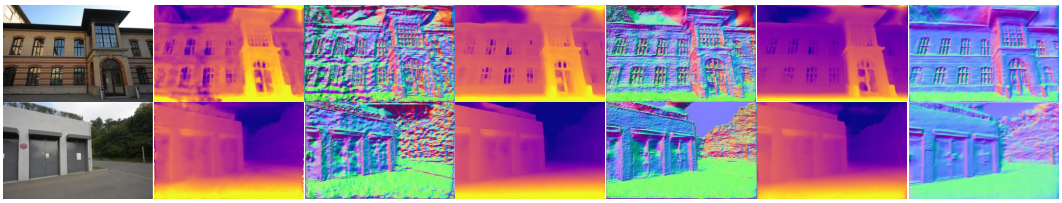



Figure 6: **Qualitative comparison of zero-shot depth and normal predictions on ETH3D.**

CD among the compared feed-forward Gaussian methods (Table 3). Figure 7 shows qualitative surface reconstruction results on BlendedMVS, while Figure 6 compares depth and normal predictions on ETH3D. These results demonstrate consistent transfer across datasets with different scene distributions. Appendix A.3 reports both the original and range-rescaled zero-shot results.

Table 3: Quantitative comparison of zero-shot results on DTU with 3 views setting. * denotes results with range-rescaled depth residuals.

| Method | Accuracy | Completion | CD |
|---|---|---|---|
| MVSplat (Chen et al., 2024) | 2.73 | 4.20 | 3.47 |
| MeshSplat (Chang et al., 2026) | 3.01 | 4.20 | 3.61 |
| DepthSplat* (Xu et al., 2025) | 1.36 | 2.08 | 1.72 |
| SurfSplat* (He et al., 2026) | 1.34 | 2.10 | 1.72 |
| IDESplat (Long et al., 2026) | 1.99 | 2.97 | 2.33 |
| Ours | **0.87** | **1.92** | **1.39** |

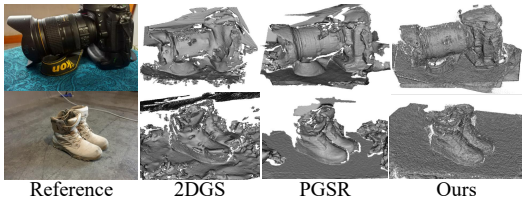



Figure 7: **Qualitative comparison of zero-shot surface reconstruction on BlendedMVS.**

### 4.3 Ablation Study

**Effects of Main Strategies.** We ablate the two main strategies of CoSplat: GCNS and TDP. Base denotes the original DepthSplat architecture trained from scratch using our training strategy, rather than its official configuration of 600K steps with a batch size of 8. Table 4 shows that replacing Gaussian primitives with surfels increases the CD, a degradation consistent with surfel orientations remaining underconstrained. GCNS indirectly improves depth prediction via rendering signals, while TDP cannot effectively refine depth without accurate normals. Combining the two therefore yields the best performance.

**Normal Supervision Strategy.** Under the complete CoSplat setting, Table 5 compares the direct use of monocular normals with our learned normal predictors trained under different supervision schemes. Directly using DSINE outputs as normals improves orientation, but they remain insufficiently adapted to the recovered multi-view geometry. Monocular and local depth supervision each outperform the setting without normal supervision, yet their naive combination reaches a normal MAE of $26.15^\circ$ while degrading CD to 1.42, worse than either cue alone. And GCNS achieves the best normal MAE of $22.94^\circ$ and CD of 0.92. This result is consistent with a conflict between full-field targets with different local orientation bases. Figure 8 shows the qualitative effect.

Table 4: Main-strategy ablation on DTU.

| Base | 2DGS | GCNS | TDP | CD ↓ |
|---|---|---|---|---|
| ✓ | | | | 1.55 |
| ✓ | ✓ | | | 1.82 |
| ✓ | ✓ | ✓ | | 1.23 |
| ✓ | ✓ | | ✓ | 1.49 |
| ✓ | ✓ | ✓ | ✓ | **0.92** |

Table 5: Normal-supervision ablation on DTU.

| Normal strategy | Normal MAE (°) ↓ | CD ↓ |
|---|---|---|
| None | 52.94 | 1.49 |
| DSINE direct | 25.00 | 1.22 |
| Monocular supervision | 27.56 | 1.24 |
| Local depth supervision | 26.67 | 1.17 |
| Naive joint supervision | 26.15 | 1.42 |
| GCNS (Ours) | **22.94** | **0.92** |

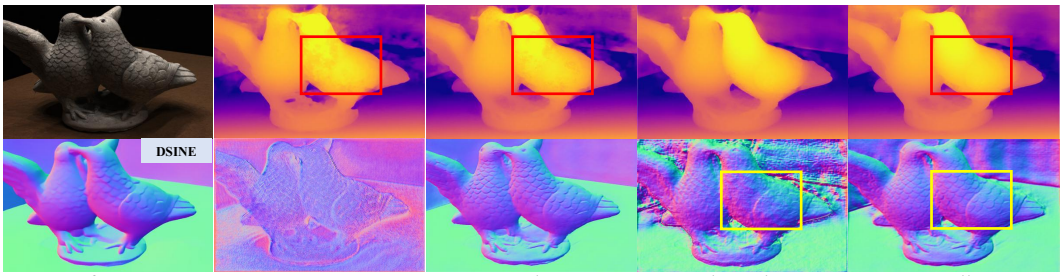



Figure 8: **Qualitative comparison of depth and normal predictions under different normal supervision strategies.** Sup. denotes supervision.

**Tangent-Plane Depth Propagation.** Table 6 summarizes the TDP ablations. Similar to cascaded MVS and IDESplat (Gu et al., 2020; Long et al., 2026), we use an FPN to construct fine-stage features as multi-view evidence. Unlike their fixed local search, TDP uses propagated tangent-plane hypotheses to define a surface-adaptive candidate range that promotes local surface continuity. Chained normal and reprojection confidences reject intra-view and cross-view propagation errors, respectively, while multi-view matching further validates the propagated depths. Combining them reduces CD from 1.09 to 0.92.

Table 6: Ablation of Tangent-Plane Depth Propagation on DTU.

| Candidate domain | MV matching | Normal cons. | Reproj. cons. | CD ↓ |
|---|---|---|---|---|
| No refinement | | | | 1.23 |
| Fixed local | ✓ | | | 1.18 |
| Propagated candidates | ✓ | | | 1.09 |
| Propagated candidates | | ✓ | ✓ | 1.21 |
| Propagated candidates | ✓ | ✓ | | 1.03 |
| Propagated candidates | ✓ | | ✓ | 1.04 |
| Propagated candidates | ✓ | ✓ | ✓ | **0.92** |

## 5 Conclusion

We presented CoSplat, a generalizable feed-forward 2DGS framework for accurate and efficient surface reconstruction. GCNS calibrates the monocular normal prior to the geometry-derived base and preserves its angular residual to learn reliable surfel orientations, which guide TDP to refine surfel positions. Experiments on DTU show that CoSplat achieves state-of-the-art surface reconstruction accuracy among feed-forward Gaussian methods while retaining efficient inference, highlighting the importance of jointly constraining surfel orientation and position. Zero-shot evaluations on DTU, BlendedMVS, and ETH3D further demonstrate strong cross-dataset generalization.

### AI USE STATEMENT

We used generative AI tools to assist with manuscript preparation and software development. For the manuscript, these tools were used for text translation and language refinement to improve readability and maintain logical clarity across sections. The overall conception, organization, and substantive writing of the paper were carried out by the authors, who reviewed and approved every AI-assisted revision. We also used generative AI to improve coding efficiency by assisting with the implementation and analysis of specific software functions. Generative AI did not contribute to the research conception, methodology, or experimental design. All ideas, experiments, and scientific analyses were developed and conducted by the authors. All AI-assisted text and code were reviewed and verified by the authors. The authors take full responsibility for the manuscript, including any text generated or edited with AI assistance. We reviewed all AI-assisted content for accuracy and ensured that it complies with applicable ethical standards and neither constitutes nor facilitates plagiarism or other scientific misconduct.

## REFERENCES

Gwangbin Bae and Andrew J. Davison. Rethinking inductive biases for surface normal estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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.

David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024.

Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 31 (9):6100–6111, 2025. doi: 10.1109/TVCG.2024.3494046.

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

R.T. Collins. A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 358–363, June 1996. doi: 10.1109/CVPR.1996.517097.

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*. Association for Computing Machinery, 2024.

Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvsnet: Global context-aware multi-view stereo network with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8585–8594, 2022.

Meiying Gu, Jiawei Zhang, Jiahe Li, Xiaohan Yu, Haonan Luo, Jin Zheng, and Xiao Bai. Sparsesurf: Sparse-view 3d gaussian splatting for surface reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4311–4319, 2026.

Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2495–2504, 2020.

Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024.

Bing He, Jingnan Gao, Yunuo Chen, Ning Cao, Gang Chen, Zhengxue Cheng, Li Song, and Wenjun Zhang. Surfsplat: Conquering feedforward 2d gaussian splatting with surface continuity priors. In *The Fourteenth International Conference on Learning Representations*, 2026.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024a. doi: 10.1145/3641519.3657428.

Han Huang, Yulun Wu, Junsheng Zhou, Ge Gao, Ming Gu, and Yu-Shen Liu. Neusurf: On-surface priors for neural surface reconstruction from sparse input views. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 2312–2320, 2024b.

Han Huang, Yulun Wu, Chao Deng, Ge Gao, Ming Gu, and Yu-Shen Liu. Fatesgs: Fast and accurate sparse-view surface reconstruction using gaussian splatting with depth-feature consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5885–5894, October 2021.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 406–413. IEEE, 2014.

Jianfei Jiang, Qiankun Liu, Haochen Yu, Hongyuan Liu, Liyong Wang, Jiansheng Chen, and Huimin Ma. Monomvsnet: Monocular priors guided multi-view stereo network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27806–27816, October 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL `https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/`.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20775–20785, June 2024.

Yixun Liang, Hao He, and Yingcong Chen. Retr: Modeling rendering via transformer for generalizable neural surface reconstruction. *Advances in neural information processing systems*, 36: 62332–62351, 2023.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017. doi: 10.1109/CVPR.2017.106.

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

Wei Long, Haifeng Wu, Shiyin Jiang, Jinhua Zhang, Xinchun Ji, and Shuhang Gu. Idesplat: Iterative depth probability estimation for generalizable 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33248–33258, June 2026.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

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.

Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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 (CVPR)*, pp. 16685–16695, June 2023.

Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 501–518, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46487-9.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Shengji Tang, Weicai Ye, Peng Ye, Weihao Lin, Yang Zhou, Tao Chen, and Wanli Ouyang. Hisplat: Hierarchical 3d gaussian splatting for generalizable sparse-view reconstruction. In *International Conference on Learning Representations*, volume 2025, pp. 48436–48449, 2025.

Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

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. *arXiv preprint arXiv:2106.10689*, 2021.

Jiang Wu, Rui Li, Yu Zhu, Rong Guo, Jinqiu Sun, and Yanning Zhang. Sparse2dgs: Geometry-prioritized gaussian splatting for surface reconstruction from sparse views. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11307–11316, 2025. doi: 10.1109/CVPR52734.2025.01056.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *CVPR*, 2025.

Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *European Conference on Computer Vision (ECCV)*, 2018.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024.

Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. Cor-gs: sparse-view 3d gaussian splatting via co-regularization. In *European conference on computer vision*, pp. 335–352. Springer, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

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

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European conference on computer vision*, pp. 145–163. Springer, 2024.

# A Appendix

## A.1 Datasets

**DTU.** DTU (Jensen et al., 2014) is an object-centric multi-view dataset containing 124 scenes captured under varying lighting conditions. Following Sparse2DGS (Wu et al., 2025), we use scans 24, 37, 40, 55, 63, 65, 69, 83, 97, 105, 106, 110, 114, 118, and 122 as the test set and the remaining 109 scenes for fine-tuning. For three-view reconstruction, we follow the previous setting and use views 23, 24, and 33. For the more challenging two-view setting, we use views 16 and 36, which have little overlap. For all feed-forward GS methods, the input views are resized to $512 \times 640$. We normalize the DTU camera coordinate scale by a factor of $1/200$ to match the model and set the initial depth search range to $[2.125, 4.525]$. Other baseline methods retain their original experimental settings.

**BlendedMVS.** BlendedMVS (Yao et al., 2020) is a large-scale MVS dataset designed for generalized multi-view stereo networks. It contains 17k MVS training samples from 113 scenes spanning architectures, sculptures, and small objects. We randomly select three images as input, resize them to $576 \times 768$, and extract the mesh to assess cross-dataset generalization.

**ETH3D.** ETH3D (Schöps et al., 2017) is a real-world MVS benchmark covering indoor and outdoor scenes captured with high-resolution images and multi-camera videos. We randomly select three images as input and resize them to $512 \times 768$. We use these samples to assess zero-shot depth estimation and normal prediction on unseen scenes.

## A.2 Implementation Details

**Model architecture.** We follow the CNN–Transformer multi-view encoder architecture of DepthSplat (Xu et al., 2025) and augment it with GCNS and TDP. In the multi-view matching branch, we uniformly sample 128 depth hypotheses between the near and far bounds in relative inverse-depth space and construct a cost volume using plane-sweep stereo (Collins, 1996). In the monocular branch, we initialize the ViT-S backbone with pretrained Depth Anything V2-Small weights (Yang et al., 2024) to extract monocular features.

**Edge-aware low-pass operator.** The operator $\mathcal{B}(\cdot)$ is implemented as two full-resolution RGB-guided $3 \times 3$ atrous filtering passes with dilation rates 1 and 2. Each pass uses the spatial kernel $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ and the color weight $\exp(-\|\mathbf{I}_p - \mathbf{I}_q\|_1/(3\tau_c))$, where $\tau_c = 0.1$. We use replication padding for images, normals, and reliability maps, so the output retains the input resolution without introducing invalid image boundaries.

**Depth reliability and valid masks.** We back-project the predicted depth into camera space and estimate horizontal and vertical tangents from its immediate neighbors. A neighbor is valid only when its depth is finite and positive and its relative difference from the center depth is below 0.05. Central differences are used when both neighbors are valid and one-sided differences otherwise. The geometry normal is valid only when both tangent directions are available and their cross product is non-degenerate. Thus, invalid neighborhoods and depth discontinuities form a hard validity mask $m^d$. We additionally compute the normalized entropy $u_p$ from the depth matching distribution at the coarse depth prediction stage and define the soft depth reliability as $r_p^d = 0.1 + 0.9(1 - u_p)^2$. The weight used by $\mathcal{B}$ is $w_p^d = m_p^d r_p^d m_p^{\mathrm{fg}}$, where $m^{\mathrm{fg}}$ is the foreground mask when available.

**Depth hypothesis validation and refinement.** We sample 9 candidates from each pixel's local relative-inverse-depth domain. We concatenate the coarse-scale monocular and multi-view features and pass them through an FPN (Lin et al., 2017) to construct full-resolution 32-channel matching descriptors. These source-view descriptors are warped for each candidate, divided into 8 channel groups, and their valid group-wise cosine correlations are averaged across two selected source views to form a local matching volume. We concatenate it with the propagation prior and normalized candidate coordinates. A lightweight 3D CNN with 16 hidden channels and three separable residual blocks aggregates along the candidate and spatial axes and predicts residual candidate logits. The

softmax expectation of the resulting distribution gives a relative inverse-depth residual, which is weighted by its entropy-derived confidence to update the coarse depth.

**Mesh extraction.** To ensure a fair comparison on DTU, all feed-forward GS methods use the same mesh extraction and evaluation pipeline. We extract meshes using TSDF fusion with a voxel size of 0.004, an SDF truncation distance of 0.02, and a depth truncation distance of 5.0. For the other baselines, we follow the evaluation pipelines provided in their official implementations.

**Normal evaluation.** For the normal ablation, we rasterize the official DTU ground-truth point cloud into each evaluation camera to obtain a z-buffered depth map, and derive ground-truth normals in camera coordinates. Pixels are discarded when the center or any required neighbor has invalid depth, when a relative depth jump exceeds 0.05, or when the resulting cross product is degenerate. We compute the oriented angular error $\arccos(\mathbf{n}_{\mathrm{pred}}^{\top}\mathbf{n}_{\mathrm{gt}})$ in degrees, over the intersection of the DTU foreground mask, valid ground-truth normals. The reported MAE is aggregated over all valid pixels rather than averaging per-view means.

### A.3 Baselines

We compare with representative methods from four families: the MVS-based methods COLMAP (Schönberger et al., 2016), TransMVSNet (Ding et al., 2022), and MonoMVSNet (Jiang et al., 2025); the optimization-based methods 2DGS (Huang et al., 2024a), GOF (Yu et al., 2024), PGSR (Chen et al., 2025), Gaussian Surfel (Dai et al., 2024), Sparse2DGS (Wu et al., 2025), and SparseSurf (Gu et al., 2026); the implicit reconstruction methods VolRecon (Ren et al., 2023), ReTR (Liang et al., 2023), and UFORecon (Na et al., 2024); and the feed-forward Gaussian methods MVSplat (Chen et al., 2024), MeshSplat (Chang et al., 2026), DepthSplat (Xu et al., 2025), IDESplat (Long et al., 2026), and SurfSplat (He et al., 2026).

**Fine-tuning feed-forward Gaussian baselines.** For a fair comparison on the DTU test set, we fine-tune the recent feed-forward GS baselines using the same setting as CoSplat. Each method is initialized from its officially released pretrained weights and fine-tuned 10k iterations with a batch size of 4 and a learning rate of $2 \times 10^{-4}$. We retain each method's original loss, denoted by $\mathcal{L}_{\mathrm{pre}}$, and add the $L_1$ depth loss commonly used in feed-forward reconstruction. The final objective is $\mathcal{L} = \mathcal{L}_{\mathrm{pre}} + \mathcal{L}_{\mathrm{depth}}$.

**Depth-Range Rescaling.** The feed-forward Gaussian baselines are pretrained on scenes whose depth ranges differ substantially from DTU, which contributes to their weaker zero-shot reconstruction. In particular, the learned DPT upsampling heads of DepthSplat and SurfSplat add an inverse-depth residual whose magnitude is tied to the training range. To ensure a fair comparison of zero-shot performance, we apply Depth-Range Rescaling to these methods. For their range-rescaled variants, we rescale this residual by $\Delta\rho_{\mathrm{test}}/\Delta\rho_{\mathrm{train}}$, where $\Delta\rho = 1/d_{\mathrm{near}} - 1/d_{\mathrm{far}}$, before adding it to the bilinearly upsampled coarse prediction and clamping the result to the test range.

**Zero-shot on DTU.** Matched fine-tuning and depth-range rescaling address different sources of mismatch for a fair comparison. The former aligns target-domain optimization, whereas the latter retains zero-shot evaluation and corrects the depth-range mismatch without updating model weights. Table 7 reports the zero-shot results. Range rescaling reduces the mean CD of DepthSplat from 9.58 to 1.72 and that of SurfSplat from 9.25 to 1.72. Ours obtains the lowest mean CD of 1.39.

Table 7: Comparison of zero-shot results on DTU with 3 views setting. * denotes results with range-rescaled depth residuals.

| Method | 24 | 37 | 40 | 55 | 63 | 65 | 69 | 83 | 97 | 105 | 106 | 110 | 114 | 118 | 122 | Mean | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MVSplat | 3.56 | 4.81 | 4.10 | 2.60 | 4.81 | 3.53 | 3.18 | 4.44 | 2.96 | 2.65 | 3.04 | 2.66 | 3.41 | 2.98 | 3.22 | 3.46 | 0.4 s |
| MeshSplat | 3.88 | 4.50 | 4.49 | 2.92 | 3.36 | 4.10 | 3.32 | 3.76 | 4.10 | 2.09 | 4.13 | 1.92 | 5.71 | 2.70 | 3.15 | 3.61 | 1 s |
| DepthSplat | 7.17 | 9.41 | 7.76 | 9.76 | 9.87 | 8.00 | 12.70 | 9.68 | 10.09 | 10.18 | 9.09 | 6.31 | 8.46 | 14.15 | 11.01 | 9.58 | 0.5 s |
| DepthSplat* | 1.79 | 2.80 | 2.75 | 1.70 | 2.04 | 2.17 | 1.57 | 1.44 | 1.42 | 0.88 | 1.57 | 1.87 | 0.78 | 1.33 | 1.71 | 1.72 | 0.5 s |
| IDESplat | 2.10 | 3.81 | 3.90 | 1.25 | 3.44 | 2.24 | 1.48 | 1.27 | 2.34 | 0.75 | 3.32 | 1.50 | 4.72 | 1.27 | 1.50 | 2.33 | 0.9 s |
| SurfSplat | 11.35 | 8.14 | 7.46 | 8.08 | 7.79 | 7.35 | 9.67 | 10.88 | 9.04 | 11.53 | 8.89 | 13.59 | 9.17 | 7.96 | 7.88 | 9.25 | 0.5 s |
| SurfSplat* | 1.71 | 2.87 | 2.52 | 1.57 | 1.90 | 2.31 | 1.62 | 1.46 | 1.35 | 0.90 | 1.67 | 1.71 | 0.92 | 1.55 | 1.77 | 1.72 | 0.5 s |
| Ours | 1.21 | 2.35 | 1.64 | 1.19 | 1.63 | 2.15 | 1.24 | 1.23 | 1.39 | 0.99 | 1.26 | 1.29 | 0.75 | 1.25 | 1.31 | 1.39 | 0.6 s |

**Novel view synthesis.** We also report novel-view synthesis performance on RE10K in Table 8. Compared with the 3DGS-based DepthSplat-S baseline, our method has lower PSNR and SSIM and higher LPIPS. Although we do not isolate the source of this gap, it may arise because the surface-constrained 2D Gaussian representation provides less flexibility for appearance fitting, while our model does not explicitly regularize opacity. As discussed in SurfSplat (He et al., 2026), highly opaque Gaussians can dominate alpha blending and suppress deeper contributions, allowing the network to reduce the rendering loss without learning fully view-consistent geometry. We regard this rendering degradation as a limitation of the current model.

Table 8: Novel-view synthesis comparison on RE10K.

| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
|---|---|---|---|
| DepthSplat-S | 26.77 | 0.877 | 0.123 |
| SurfSplat-S | 27.00 | 0.883 | 0.118 |
| IDESplat | **27.80** | **0.893** | **0.108** |
| Ours | 26.14 | 0.863 | 0.146 |

### A.4 Additional Visualization Results

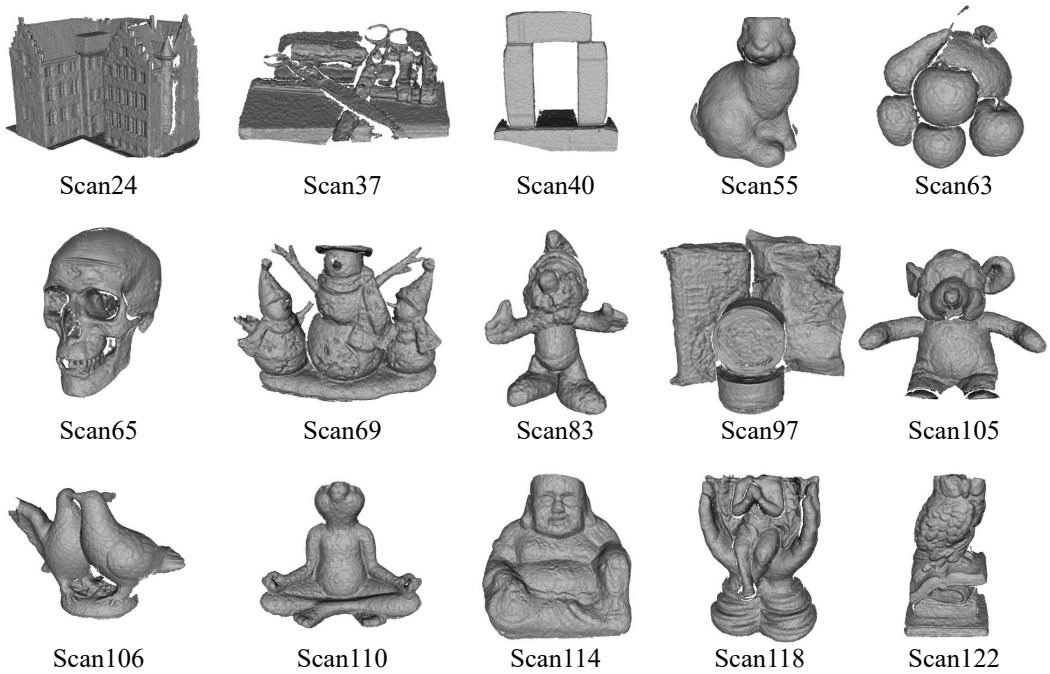



Figure 9: **Additional qualitative reconstruction results on DTU using three input views (large-overlap setting).**

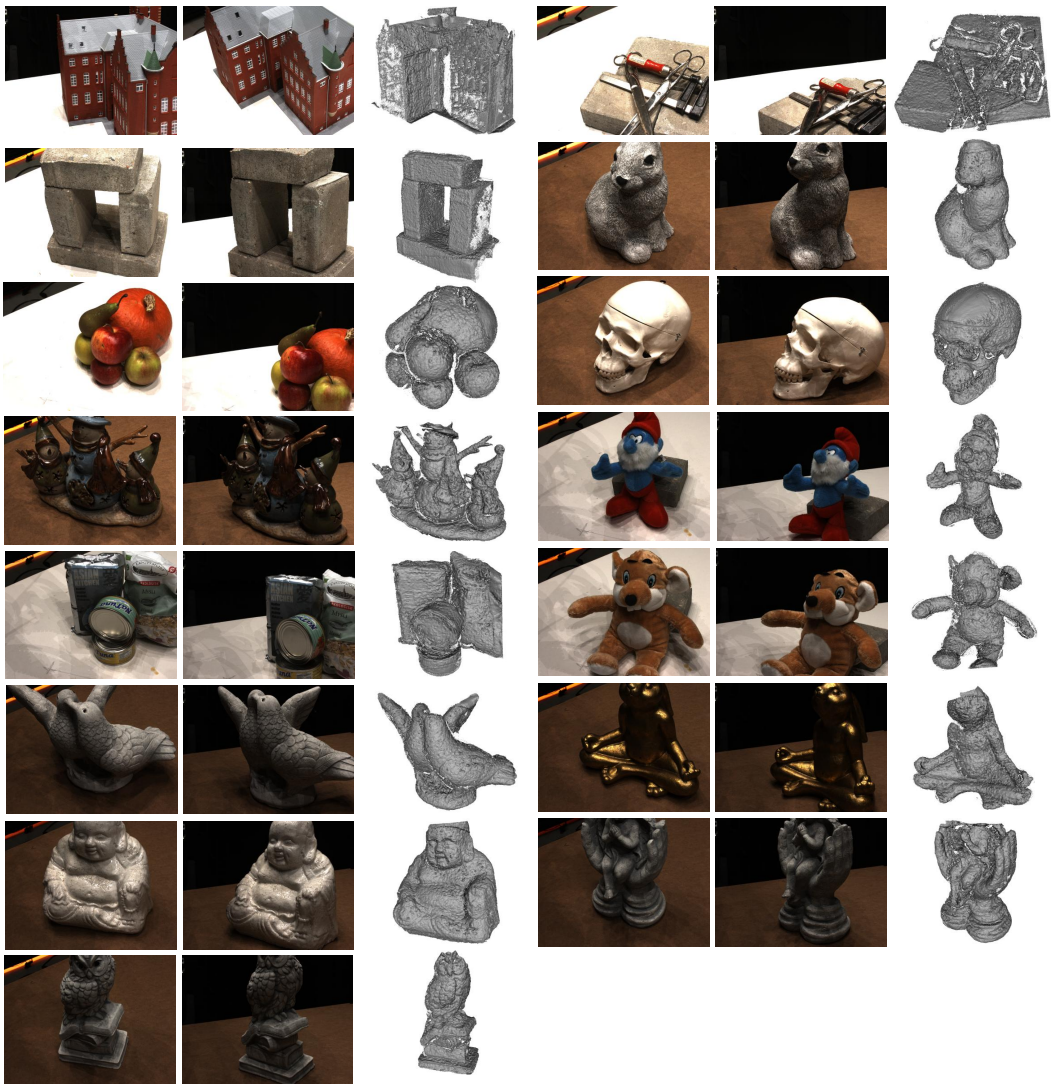


Figure 10: **Additional qualitative reconstruction results on DTU using two input views (little-overlap setting).**