HALLUCINATE, FILTER, AND REFINE: OCCLUDED SURFACE RECONSTRUCTION WITH GENERATIVE PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstructing reliable 3D surfaces from sparse input views remains challenging when large scene regions are unobserved due to occlusions. Generative models can expand view coverage by synthesizing missing observations, but generated views often contain hallucinated structures, pose errors, and appearance drift that can introduce unreliable supervision. We propose Hallucinate, Filter, and Refine, a reliability aware framework that exploits generative priors while controlling their uncertainty throughout reconstruction. We first establish a geometric anchor from sparse inputs and synthesize multiple candidate view sequences along augmented camera trajectories. We then progressively filter generated observations at both sequence and view levels using patch-wise visual compatibility with input views and consistency with observed geometry. Our Auxiliary Control Module (ACM) estimates patch wise reliability based on feature consistency and reconstruction difficulty. ACM regulates photometric supervision by suppressing uncertain generated regions while preserving reliable information. Since uncertain generated regions can be difficult to identify before sufficient geometric convergence, ACM adaptively activates reliability weighting based on reconstruction progress. This design preserves valid information during early optimization while reducing the influence of uncertain generated regions as reconstruction stabilizes. Experiments demonstrate improved occluded surface recovery while maintaining competitive rendering quality under sparse observations.

1 INTRODUCTION

Reconstructing complete and geometrically consistent 3D surfaces from sparse input views remains a long standing challenge in computer vision. The problem becomes particularly difficult when large portions of a scene are occluded or entirely unobserved because these regions provide no direct image evidence for reconstruction. As a result, reconstruction methods must infer missing surfaces from limited observations, which often leads to incomplete geometry or ambiguous surface estimates.

Neural radiance fields (NeRF) (Mildenhall et al., 2021) and their variants (Wang et al., 2021; Yariv et al., 2021) achieve impressive rendering quality, but typically depend on dense view coverage and struggle when input observations are limited. Explicit representations such as 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) offer efficient optimization and real time rendering, while surface oriented variants (Huang et al., 2024a; Guédon & Lepetit, 2024) further improve geometric fidelity. However, like implicit representations, Gaussian based methods degrade significantly under sparse observations because insufficient coverage leaves the underlying geometry weakly constrained. This limitation often results in incomplete surfaces and unstable reconstruction in occluded regions.

To compensate for missing observations, recent works have explored generative models for synthesizing additional views (Wu et al., 2024; Liu et al., 2024a; Topaloğlu et al., 2026). Camera conditioned diffusion models (Zhou et al., 2025; He et al., 2025a; Kuang et al., 2024; He et al., 2025b) further enable plausible view generation under specified camera trajectories, providing additional observations beyond the limited input coverage. These advances primarily focus on improving the generation process, but do not directly ensure the geometric reliability required for reconstruction.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

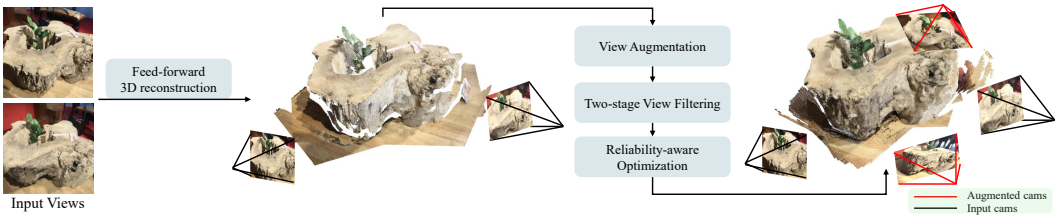


Figure 1: Proposed framework for sparse-view occluded surface reconstruction. Sparse input views are first converted into an initial geometric anchor using feed-forward 3D reconstruction. We then hallucinate additional views, filter unreliable generations, and refine the geometry with reliability-aware optimization, enabling recovery of surfaces in previously unobserved regions.

In particular, regions with limited support from the input views remain inherently ambiguous, and generated observations may still contain hallucinated structures, pose inconsistencies, or semantic drift. Consequently, improved generation alone cannot ensure that synthesized observations provide reliable supervision for surface reconstruction. This motivates a complementary approach that treats generated observations as uncertain priors and explicitly determines which observations should be retained and how strongly they should contribute during reconstruction.

Based on this observation, we propose Hallucinate, Filter, and Refine, a reliability aware framework that incorporates generative priors through progressive filtering and adaptive supervision. Starting from a sparse geometric anchor, we generate multiple candidate view sequences that expand coverage into unobserved regions. We first select a candidate sequence according to patch level visual compatibility with the input views. We then prune individual generated views by evaluating their consistency with observed geometry. The retained views provide additional supervision for joint optimization of camera poses and surfaces.

To further control residual uncertainty within the retained generated views, we introduce an Auxiliary Control Module (ACM). The module estimates patch-wise reliability from feature consistency and reconstruction difficulty to regulate photometric supervision from generated observations. Since uncertain regions can be difficult to identify before sufficient geometric convergence, reliability weighting is adaptively activated based on reconstruction progress. This strategy preserves valid generated information during early optimization while reducing the influence of uncertain regions as reconstruction stabilizes. Importantly, our framework operates without modifying or fine tuning the underlying generative model.

We extensively evaluate our approach against recent sparse view and generative reconstruction methods. Our method improves surface reconstruction, particularly in unobserved regions, while maintaining competitive rendering quality under sparse observations. We further analyze reconstruction under varying occlusion by distinguishing observed and unobserved surfaces according to input view visibility.

Our contributions are summarized as follows.

- We present a reliability aware framework for sparse view surface reconstruction that incorporates generative priors through progressive filtering and adaptive supervision.
- We develop a progressive filtering strategy together with an Auxiliary Control Module that selectively retains generated observations and adaptively controls the influence of uncertain regions during reconstruction.
- We evaluate reconstruction with explicit consideration of occlusion by distinguishing observed and unobserved surfaces and quantifying the unobserved surface fraction from the input views. Our method demonstrates improved recovery of unobserved surfaces while maintaining competitive geometric and rendering quality under sparse observations.

2 Related Work

2.1 Surface Reconstruction with Sparse Input Views

Reconstructing 3D surfaces from sparse input views is a long-standing problem in computer vision. Classical multi-view stereo (MVS) methods rely on sufficient view overlap and reliable correspondences, and therefore often degrade when the input views are sparse or affected by severe occlusions (Bernardini et al., 1999; Lorensen & Cline, 1998; Kazhdan et al., 2006; Kazhdan & Hoppe, 2013).

Neural Radiance Fields (NeRF) (Mildenhall et al., 2021) and neural implicit surface methods (Wang et al., 2021; Yariv et al., 2021) achieve high-fidelity reconstruction under dense view coverage, but their performance decreases when only a few observations are available. To address sparse-view settings, several methods introduce regularization, learned priors, or geometric constraints. RegNeRF (Niemeyer et al., 2022) and SparseNeRF (Wang et al., 2023) regularize radiance fields under limited observations, while surface-oriented methods such as UFORecon (Na et al., 2024), S-VolSDF (Wu et al., 2023), Spurfies (Raj et al., 2025), NeuSurf (Huang et al., 2024b), Sparis (Wu et al., 2025b), SparseRecon (Han et al., 2025), and SparseSurf (Gu et al., 2026) improve geometry by incorporating sparse-view surface priors, local geometric regularization, or geometric priors.

Gaussian Splatting provides an efficient explicit alternative to implicit radiance fields. 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) achieves real-time rendering with high visual quality, while surface-aware variants such as SuGaR (Guédon & Lepetit, 2024), 2DGS (Huang et al., 2024a), PGSR (Chen et al., 2024), SurfelSplat (Dai et al., 2026), and Matcha Gaussians (Guédon et al., 2025) improve geometric fidelity by encouraging Gaussian primitives to better align with surfaces or by introducing surface-aware primitive representations.

Recent sparse-view Gaussian methods further incorporate depth, normal, uncertainty, dense correspondence, or co-regularization priors, including FreeSplatter (Xu et al., 2025b), Sparse2DGS (Takama et al., 2025), GausSurf (Wang et al., 2024), UNG-GS (Tan et al., 2025), CoR-GS (Zhang et al., 2024), Gen-SVGS (Kong et al., 2025), DepthSplat (Xu et al., 2025a), FatesGS (Huang et al., 2025), DNgaussians (Li et al., 2024), and ExploreGS (Kim et al., 2025). These approaches improve sparse-view reconstruction by providing more reliable geometric initialization, regularizing primitive distributions, or exploiting additional priors for stable optimization.

2.2 Diffusion-Guided Surface Reconstruction

Diffusion models have recently been adopted as generative priors for sparse-view reconstruction (Blattmann et al., 2023). By synthesizing plausible intermediate or extrapolated views, camera-conditioned and video diffusion models can expand the effective view coverage when only a few input images are available (Zhou et al., 2025; He et al., 2025a; Kuang et al., 2024; He et al., 2025b). This capability has motivated reconstruction pipelines that use generated views or diffusion-based priors to compensate for missing observations.

ReconFusion (Wu et al., 2024) incorporates diffusion priors for sparse-view 3D reconstruction, while ReconX (Liu et al., 2024a) leverages video diffusion to reconstruct scenes from limited inputs. ViewExtrapolator (Liu et al., 2024b) and 3DGS-Enhancer (Liu et al., 2024c) further investigate view extrapolation or enhancement strategies for improving 3D Gaussian reconstruction under limited observations. OracleGS (Topaloğlu et al., 2026) grounds generative priors for sparse-view Gaussian Splatting and uses confidence estimates to reduce unreliable supervision. These methods demonstrate the potential of generative priors for reconstructing sparsely observed scenes.

However, diffusion-generated or extrapolated views are not guaranteed to be geometrically consistent with the underlying scene. In occluded or weakly constrained regions, generated views may contain plausible but incorrect structures, pose misalignment, or appearance drift. When such views are directly used as supervision, reconstruction models can overfit to generative artifacts, resulting in distorted surfaces or inconsistent geometry. Recent work such as OracleGS addresses this issue by grounding generative priors with confidence estimates to suppress unreliable supervision. In contrast, our method further decomposes reliability handling across the reconstruction pipeline through semantic set selection, score-based view pruning, and reliability-aware optimization.

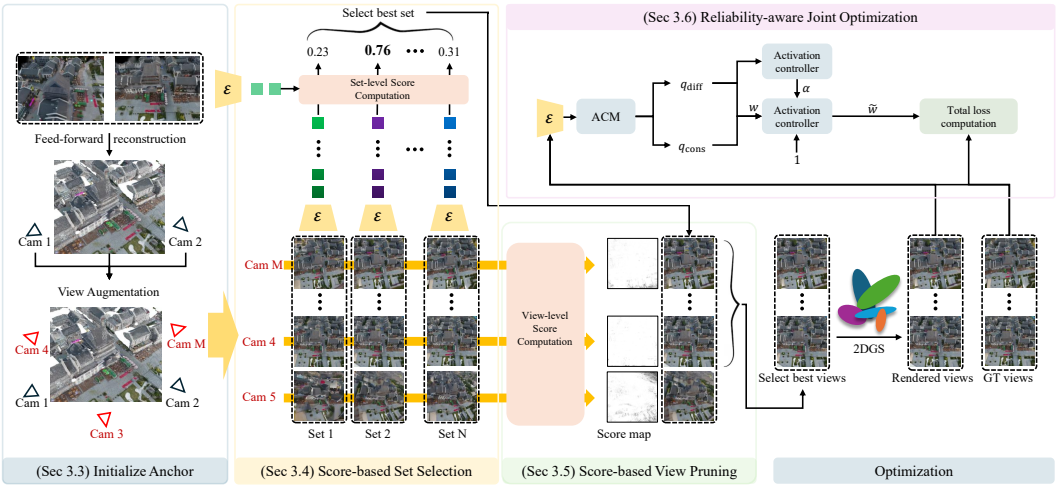


Figure 2: **Main pipeline of the proposed framework.** We initialize a sparse geometric anchor from input views, hallucinate candidate views through camera-conditioned augmentation, and filter them using score-based set selection and view pruning. During optimization, the current geometry is differentiably rendered at augmented camera poses, and ACM compares the rendered and augmented views to estimate patch wise reliability from consistency and reconstruction difficulty. The activation controller then adaptively determines when reliability weighting is applied.

3 METHOD

3.1 PRELIMINARY

2D Gaussian Splatting (2DGS) (Huang et al., 2024a) represents a scene as a set of surface aligned Gaussian primitives, each parameterized by a 3D center $\mathbf{p}_i$, normal $\mathbf{n}_i$, 2D scale $\mathbf{s}_i$, opacity α_i, and view dependent color. Novel views are rendered via differentiable alpha blended rasterization, while geometric fidelity is encouraged through normal consistency and distortion regularization losses. We adopt the original 2DGS representation without modification.

3.2 OVERVIEW

The overview of our pipeline is shown in Fig. 2. Given sparse input views $V_{\text{in}} = \{I_i^{\text{in}}\}_{i=1}^{N_{\text{in}}}$, our goal is to reconstruct a geometrically reliable surface despite large unobserved regions. We first establish a geometric anchor from V_{in} using a feed forward reconstruction model (Wang et al., 2025) and hallucinate multiple candidate view sequences $\mathcal{C}$ along an augmented camera trajectory. The candidates are progressively filtered through score based set selection, which selects a single sequence A^*, and view pruning, which retains reliable augmented views V_{aug}. Finally, we jointly optimize the 2DGS parameters Θ and camera poses using V_{in} and V_{aug}, while ACM adaptively regulates the contribution of augmented observations according to their estimated reliability.

3.3 GEOMETRIC ANCHORING AND VIEW HALLUCINATION

Given $V_{\text{in}} = \{I_i^{\text{in}}\}_{i=1}^{N_{\text{in}}}$, we first obtain initial camera poses $\hat{\Pi}_{\text{in}} = \{\hat{T}_i^{\text{in}}\}_{i=1}^{N_{\text{in}}} \subset SE(3)$ and a sparse point cloud $\mathcal{P}_{\text{in}}$ using a feed-forward 3d reconstruction model. To expand coverage in occluded regions, we sample an augmented camera trajectory $\hat{\Pi}_{\text{aug}} = \{\hat{T}_m^{\text{aug}}\}_{m=1}^{N_{\text{aug}}}$, around the two estimated input poses. The same trajectory is shared across all candidate sequences, while the synthesized content at each pose varies stochastically. We synthesize images along this trajectory using a camera conditioned video diffusion model conditioned on the two input views. Because sampling is stochastic, we generate N_{set} independent candidate sequences and collect them as $\mathcal{C} = \{A^{(s)}\}_{s=1}^{N_{\text{set}}}$, where $A^{(s)} = \{I_{s,m}^{\text{aug}}\}_{m=1}^{N_{\text{aug}}}$. The candidate sequences expand coverage but may contain semantic drift, pose errors, or hallucinated structures, motivating the filtering stages below.

3.4 SCORE-BASED SET SELECTION

We select the candidate sequence whose patch-level visual statistics most closely match those of the real input views. Concretely, we build a real-patch feature bank by extracting patch tokens from both input views by a visual representation model (Oquab et al., 2023) and pooling them into a single bank $\mathcal{B}$. For each image I in each candidate set, we compute an *unusualness score*, the mean L2 distance from each of its patch features to its nearest neighbor in $\mathcal{B}$:

$$u(I) = \frac{1}{N_p} \sum_{j=1}^{N_p} \min_{b \in \mathcal{B}} \left\| \mathcal{E}_{\text{patch}}(I)_j - b \right\|_2, \tag{1}$$

where $\mathcal{E}_{\text{patch}}(\cdot)$ denotes the patch token features and N_p is the number of patches. The set-level score is the mean unusualness over all images in the sequence, and we select the candidate with the lowest score:

$$A^* = \arg\min_s \ \frac{1}{N_{\text{aug}}} \sum_{m=1}^{N_{\text{aug}}} u\big(I_{s,m}^{\text{aug}}\big) \,. \tag{2}$$

A lower score indicates greater patch level visual compatibility between the generated sequence and the input views. The winning index is persisted before any training begins and all subsequent stages operate exclusively on A^*.

3.5 SCORE-BASED VIEW PRUNING

Although A^* is more consistent with the input observations than the other candidate sequences, individual views within it may still contain local artifacts, pose mismatch, or inconsistent structures. We therefore prune to the most reliable subset before joint optimization. After initialized geometry is re-estimated on the full input views selected set (N_{in} + N_{aug}), a co-visibility mask Ω is obtained by reprojecting the two depth maps from V_{in} into each augmented camera pose, marking which pixels of that augmented view are explained by the true input geometry. For each augmented view $I_m^{\text{aug}} \in A^*$, we render the current 2DGS model at the corresponding pose using the differentiable renderer $\mathcal{R}_\Theta$, where Θ denotes the current 2DGS parameters. We then compute a view level reliability score over the co visible region as

$$S_{\text{view}}(I_m^{\text{aug}}) = \text{SSIM}_\Omega\Big(\mathcal{R}_\Theta(\hat{T}_m^{\text{aug}}),\, I_m^{\text{aug}}\Big) \,. \tag{3}$$

We rank views by S_{view} and retain the top k_{view} views. The retained image-pose pairs form the final augmented supervision set:

$$V_{\text{aug}} = \big\{ (I_m^{\text{aug}},\, \hat{T}_m^{\text{aug}}) \big\}_{m \in \mathcal{K}}, \qquad |\mathcal{K}| = k_{\text{view}}. \tag{4}$$

3.6 RELIABILITY AWARE JOINT OPTIMIZATION

Reliability estimation. For each augmented view training iteration, let $\hat{I}^{\text{rend}}$ denote the current 2DGS render and I^{tar} denote the fixed target from V_{aug} at the same pose. We extract patch level features from both images using a frozen representation model $\mathcal{E}$ and estimate reliability from two complementary criteria. The consistency score q_{cons} measures feature space agreement between the render and the target, while the difficulty score q_{diff} measures their normalized pixel level residual. Both quantities are detached from the computational graph. We define the per patch reliability as

$$w = q_{\text{cons}} \left(1 - q_{\text{diff}}\right), \tag{5}$$

Higher values of w indicate greater reliability of the generated observation, whereas lower values correspond to regions with higher uncertainty.

Adaptive per scene activation. The reliability is not applied unconditionally throughout optimization. During early optimization, a large q_{diff} may result from insufficient geometric convergence rather than an inaccurate generated target. Directly applying w under this condition can therefore reduce the contribution of valid generated observations. To account for this ambiguity, we introduce an adaptive controller that determines when reliability weighting should influence optimization. Let $\alpha_t \in \{0, 1\}$ denote the controller state at iteration t. After an initial warm up period, the controller tracks the recent evolution of reconstruction difficulty and updates α_t every r iterations according to

$$\alpha_t = \begin{cases} 1, & \text{if the activation condition is satisfied,} \\ 0, & \text{if the deactivation condition is satisfied,} \\ \alpha_{t-r}, & \text{otherwise.} \end{cases} \tag{6}$$

The activation condition identifies a sustained decrease in reconstruction difficulty, indicating that the geometry has become sufficiently stable for the reliability estimate to be informative. Conversely, the deactivation condition identifies a sustained increase in difficulty and disables reliability weighting until the reconstruction stabilizes again. The difficulty score continues to be monitored regardless of the controller state. Detailed definitions of the activation condition and the corresponding controller parameters are provided in Appendix A.3.

We incorporate the controller state directly into the reliability weight as

$$\tilde{w}_t = (1 - \alpha_t) + \alpha_t w_t. \tag{7}$$

Thus, α_t switches between uniform weighting and reliability based weighting for augmented view optimization. The augmented photometric loss is then defined as

$$\mathcal{L}_{\text{photo}}^{\text{aug}} = (1 - \lambda_{\text{dssim}}) \frac{\sum_p \tilde{w}_{t,p} \left| \hat{I}_p^{\text{rend}} - I_p^{\text{tar}} \right|}{\sum_p \tilde{w}_{t,p} + \epsilon} + \lambda_{\text{dssim}} \left(1 - \text{SSIM}_{\tilde{w}_t} \left(\hat{I}^{\text{rend}}, I^{\text{tar}} \right) \right). \tag{8}$$

Total loss. The training objective combines photometric losses on the input and augmented views with geometric regularization terms

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{photo}}^{\text{in}} + \mathcal{L}_{\text{photo}}^{\text{aug}} + \lambda_{\text{dist}} \mathcal{L}_{\text{dist}} + \lambda_{\text{normal}} \mathcal{L}_{\text{normal}} + \lambda_{\text{dn}} \mathcal{L}_{\text{dn}}. \tag{9}$$

Here, $\mathcal{L}_{\text{dist}}$ and $\mathcal{L}_{\text{normal}}$ are the 2DGS regularizers (Huang et al., 2024a), and $\mathcal{L}_{\text{dn}}$ is the depth normal consistency loss based on monocular depth priors (Chen et al., 2025). Detailed loss definitions and hyperparameter settings are provided in Appendix A.4 and Appendix A.

4 Experiments

Implementation details. We build our method on top of 2DGS (Huang et al., 2024a) and InstantSplat (Fan et al., 2024). For initial geometric anchoring, we use Pi3 (Wang et al., 2025) to estimate camera poses and sparse geometry from the input views. Candidate views are generated using SEVA (Zhou et al., 2025), a camera-controlled video diffusion model. For set-level selection and ACM descriptor, we use DINOv2 (Oquab et al., 2023) as the pretrained visual encoder.

Evaluation protocol. We evaluate rendering quality using PSNR, SSIM, and LPIPS, and geometric quality using masked Chamfer distance and F1 score. To further assess reconstruction across different visibility conditions, we partition the ground truth surface into observed and unobserved regions according to input view visibility and report precision, recall, and F1 score for each region separately. For photometric evaluation, we apply object masks to focus on the reconstructed object regions following PixelNeRF (Yu et al., 2021). For geometry evaluation, the reconstructed surfaces are aligned with the ground truth geometry using Open3D (Zhou et al., 2018) registration before computing the geometric metrics. Additional evaluation details are provided in Appendix A.6. For fair comparison, we use the same models for geometric anchoring and candidate view generation for all experiments.

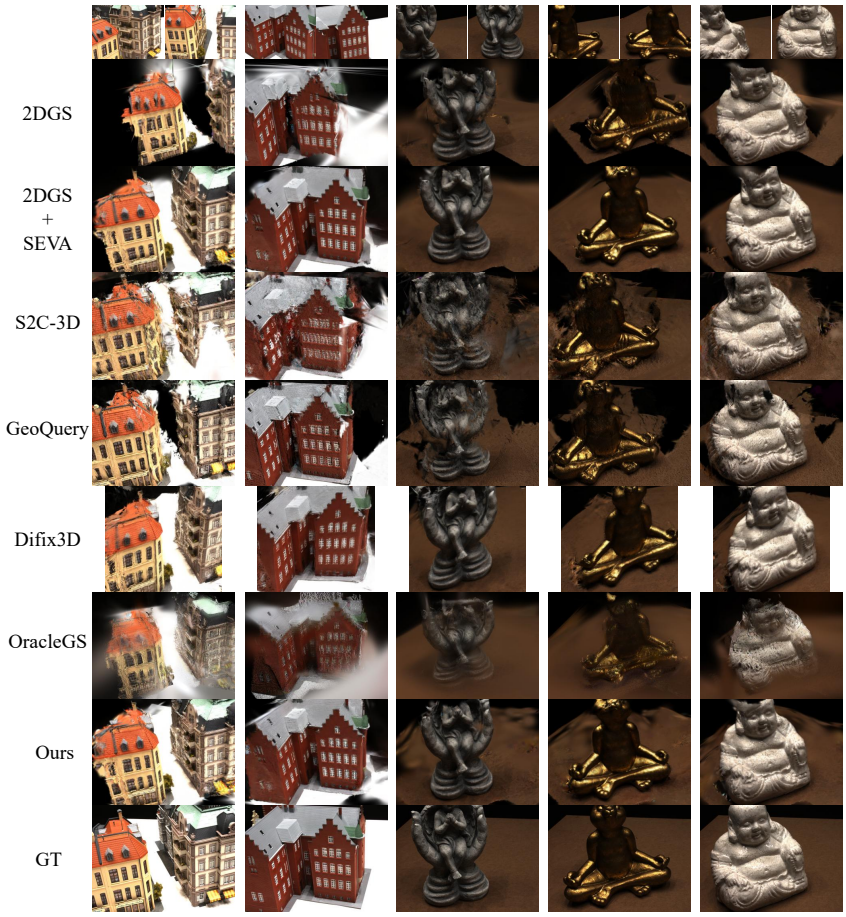


Figure 3: **Qualitative comparison on the DTU dataset.** From top to bottom, two sparse input views, 2DGS (Huang et al., 2024a), 2DGS + SEVA (Zhou et al., 2025), S2C-3D (Shen et al., 2026), GeoQuery (Cao et al., 2026), Difix3D (Wu et al., 2025a), OracleGS (Topaloğlu et al., 2026), Ours, and ground truth.

Datasets and baselines. We perform the main evaluation on five scans from S VolSDF (Wu et al., 2023) based on the DTU dataset (Jensen et al., 2014). We organize the compared methods into three groups according to how generative priors are incorporated into reconstruction. The first group includes 2DGS (Huang et al., 2024a) as the reconstruction baseline and its direct augmentation with SEVA (Zhou et al., 2025). The second group consists of generative reconstruction methods that improve reconstruction through dedicated generative modeling or adaptation, including S2C-3D (Shen et al., 2026), GeoQuery (Cao et al., 2026), and Difix3D (Wu et al., 2025a). The final group compares reconstruction frameworks that focus on how generated observations are incorporated into the reconstruction process, including OracleGS (Topaloğlu et al., 2026) and our method.

4.1 Qualitative Results

Figure 3 presents qualitative comparisons on the DTU dataset. The sparse input setting leads to substantial reconstruction errors in the 2DGS baseline, particularly in regions with limited input view coverage. While the compared methods recover additional scene content to varying degrees, several results exhibit incomplete surfaces, distorted structures, or noticeable appearance inconsistencies. Our method produces more complete and coherent reconstructions across the evaluated scenes. In particular, the reconstructed surfaces preserve the overall object structure while recovering regions that are poorly represented by the sparse input views. The resulting appearance also remains consistent with the observed scene, without introducing substantial structural deviations in the recovered regions.

4.2 Quantitative Results

Table 1: **Quantitative comparison on the DTU dataset.** The best results for each scenario are highlighted in **bold** and second best results are underlined.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Chamfer ↓	Obs. F1 ↑	Obs. Prec. ↑	Obs. Rec. ↑	Unobs. F1 ↑	Unobs. Prec. ↑	Unobs. Rec. ↑
2DGS (Huang et al., 2024a)	11.5544	0.3285	0.6643	34.3015	0.6066	0.7519	0.5230	0.7201	**0.9951**	0.5804
2DGS + SEVA (Zhou et al., 2025)	12.4986	0.3770	0.6074	16.8117	0.8255	0.8623	0.7946	0.9017	0.9845	0.8367
S2C-3D (Shen et al., 2026)	**13.8245**	0.3665	0.5975	28.8329	0.5671	0.6514	0.5881	0.7020	0.8200	0.6709
GeoQuery (Cao et al., 2026)	13.5279	0.3460	**0.5776**	43.4244	0.4173	0.6766	0.3562	0.4885	0.8602	0.3749
Difix3D (Wu et al., 2025a)	12.6265	0.3784	0.6134	32.1026	0.5371	0.7134	0.4879	0.6659	0.8977	0.5551
OracleGS (Topaloğlu et al., 2026)	13.2353	**0.3901**	0.6284	13.6807	0.8673	0.9344	0.8148	0.9360	0.9950	0.8852
Ours	13.0398	0.3773	0.5946	**11.6398**	**0.9179**	**0.9819**	**0.8649**	**0.9405**	0.9949	**0.8936**

Table 1 presents the quantitative comparison on the DTU dataset. Our method achieves the strongest overall geometric reconstruction while maintaining competitive rendering quality. Several compared methods achieve favorable rendering metrics but show substantially weaker geometric reconstruction, indicating that improved rendering quality does not necessarily translate into accurate surface recovery under sparse observations. The advantage of our method is particularly evident when reconstruction quality is evaluated according to input view visibility. Our method achieves the strongest F1 score, precision, and recall on observed surfaces, indicating that the use of generated observations does not compromise regions supported by the input views. On unobserved surfaces, our method achieves the strongest F1 score and recall while maintaining high precision. These results indicate that the proposed framework effectively extends surface reconstruction beyond the input coverage while preserving geometric accuracy in observed regions. The consistent improvements across both visibility regions support the use of explicit filtering and reliability aware optimization for incorporating generated observations into surface reconstruction.

4.3 Occlusion Severity Analysis

Table 2: **Reconstruction performance under different occlusion levels.** Low and High correspond to input configurations with different unobserved surface fractions, where ρ denotes the fraction of the ground truth surface that is unobserved from the input views. F1 scores are reported over the complete surface and separately over observed and unobserved regions. Best results are shown in **bold** and second best results are underlined.

Method	**Low** ($\rho = 0.264$)			**High** ($\rho = 0.282$)		
	All	**Obs.**	**Unobs.**	**All**	**Obs.**	**Unobs.**
2DGS	0.435	0.400	0.694	0.087	0.073	0.332
2DGS + SEVA	0.366	0.345	0.498	0.071	0.064	0.104
GeoQuery	0.478	0.431	0.643	0.363	0.322	0.488
Difix3D	0.047	0.051	0.023	0.622	0.570	0.619
OracleGS	0.376	0.337	**0.697**	0.126	0.112	0.449
Ours	**0.569**	**0.525**	0.660	**0.651**	**0.621**	**0.681**

We further evaluate reconstruction under different levels of input visibility on DTU scan24. We consider *Low* and *High* occlusion configurations with different fractions ρ of the ground truth surface that are unobserved from the input views. For each configuration, the ground truth surface is partitioned into observed and unobserved regions according to visibility from the input views. Further details on the definition of ρ, visibility computation, evaluation split, and input camera configurations are provided in Appendix A.7.

Table 2 reports F1 scores over the complete surface and separately over the observed and unobserved regions. Under the *Low* configuration, our method achieves the strongest reconstruction over the complete and observed surfaces, while several compared methods perform better on the unobserved region. As the unobserved surface fraction increases, the differences between the methods become more pronounced. Under the *High* configuration, our method achieves the strongest F1 score across the complete, observed, and unobserved surfaces. This behavior indicates that the proposed framework remains effective when direct input coverage becomes more limited.

Table 3: **Pipeline ablation study.** We progressively add generative view augmentation (Gen.), reliability aware filtering (Filter), and the proposed auxiliary control module (ACM). The best results for each metric are highlighted in **bold** and second best results are underlined.

Gen.	Filter	ACM	PSNR ↑	SSIM ↑	LPIPS ↓	Chamfer ↓	F1 ↑	Observed			Unobserved		
								F1 ↑	Prec. ↑	Rec. ↑	F1 ↑	Prec. ↑	Rec. ↑
			13.369	0.3686	**0.5763**	23.727	0.7436	0.7217	0.7492	0.7092	0.7552	0.9653	0.6225
✓			12.850	0.3727	0.6010	21.049	0.8142	0.7817	0.9022	0.7005	0.8406	0.9919	0.7322
✓	✓		12.910	0.3717	0.5978	13.453	0.9093	0.8943	0.9449	0.8493	0.9244	0.9871	0.8727
✓	✓	✓	13.040	**0.3773**	0.5946	**11.640**	**0.9305**	**0.9179**	**0.9819**	**0.8649**	**0.9405**	**0.9949**	**0.8936**

4.4 Ablation Study

Ablation on Pipeline Components. Table 3 quantitatively analyzes the contribution of each component in our reconstruction pipeline. Adding generated views to the baseline improves geometric reconstruction, with particularly clear gains in unobserved regions, while rendering quality decreases. This result indicates that additional generated observations expand geometric coverage but may introduce visual inconsistencies. Introducing reliability aware filtering substantially improves geometric reconstruction in both observed and unobserved regions, particularly in recall and F1 score. Finally, incorporating ACM further improves the geometric metrics, achieving the strongest overall F1 score and the strongest reconstruction in both observed and unobserved regions. These results show that progressive filtering and reliability aware optimization enable generated observations to contribute more effectively to geometric reconstruction.

Table 4: **Ablation of the reliability formulation in ACM.** Cons. and Diff. denote q_{cons} and q_{diff}, respectively, and checkmarks indicate the terms used to construct the reliability weight. All variants use the same adaptive activation controller. Best results are shown in **bold** and second best results are underlined.

Cons.	Diff.	PSNR ↑	SSIM ↑	LPIPS ↓	Chamfer ↓	F1 ↑	Observed			Unobserved		
							F1 ↑	Prec. ↑	Rec. ↑	F1 ↑	Prec. ↑	Rec. ↑
✓		12.907	0.3750	0.5978	12.700	0.9142	0.8990	0.9526	0.8519	0.9352	0.9900	0.8879
	✓	13.005	0.3750	0.5963	**11.640**	**0.9321**	**0.9198**	0.9695	**0.8758**	**0.9456**	0.9934	**0.9039**
✓	✓	**13.040**	**0.3773**	**0.5946**	**11.640**	0.9305	0.9179	**0.9819**	0.8649	0.9405	**0.9949**	0.8936

Auxiliary Control Module. We analyze the contribution of the consistency and difficulty terms in the ACM reliability formulation while keeping the adaptive activation controller fixed across all variants in Table 4. Using the consistency term alone provides limited geometric improvement, whereas the difficulty term substantially improves reconstruction across both observed and unobserved regions. Combining both terms further improves precision in both regions and achieves better rendering metrics while maintaining competitive geometric reconstruction. These results indicate that reconstruction difficulty provides the primary criterion for identifying uncertain regions, while feature consistency enables more selective reliability weighting when combined with the difficulty term.

5 Conclusion

We presented a reliability aware framework for sparse view surface reconstruction that leverages generative priors to recover geometry beyond direct input coverage. Our approach first generates candidate observations from sparse input views and progressively filters them through sequence selection and view pruning. The retained observations are then incorporated into joint optimization, where ACM estimates patch wise reliability from feature consistency and reconstruction difficulty. An adaptive activation controller further determines when reliability weighting should influence optimization based on reconstruction progress. Experiments on challenging sparse view settings demonstrate that our approach improves geometric reconstruction while maintaining competitive rendering quality. In particular, the evaluation of observed and unobserved surfaces shows that the proposed framework effectively recovers geometry beyond input visibility while preserving reconstruction quality in observed regions. These results demonstrate that generative priors can provide useful information for occluded surface reconstruction when generated observations are carefully filtered and their contribution is adaptively controlled during optimization.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, assist with translation. We did not use generative AI tools to generate synthetic data sets, support qualitative and thematic data analysis, interpret results, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, or design or provide feedback on research methodology or experiments. Cleaning and reformatting datasets is not applicable to this work. We have reviewed all AI-assisted work. LLM-assisted code for implementing baseline methods was reviewed and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed framework and its individual components in Sec. 3. The appendix provides implementation details, optimization settings, loss definitions, and hyperparameters used in our experiments. We also describe the datasets, input view configurations, compared methods, and evaluation settings used for quantitative evaluation. The evaluation protocol and the visibility based partition of observed and unobserved surfaces are detailed in Appendix A.6 and Appendix A.7, respectively.

REFERENCES

Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359, 1999.

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

Xiao Cao, Yuze Li, Youmin Zhang, Jiayu Song, Cheng Yan, Wen Li, and Lixin Duan. Geoquery: Geometry-query diffusion for sparse-view reconstruction. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2026.

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, 2024.

Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22831–22840, 2025.

Chensheng Dai, Shengjun Zhang, Min Chen, and Yueqi Duan. Surfelsplat: Learning efficient and generalizable gaussian surfel representations for sparse-view surface reconstruction. *arXiv preprint arXiv:2604.08370*, 2026.

Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Sparse-view gaussian splatting in seconds. *arXiv preprint arXiv:2403.20309*, 2024.

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.

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.

Antoine Guédon, Tomoki Ichikawa, Kohei Yamashita, and Ko Nishino. Matcha gaussians: Atlas of charts for high-quality geometry and photorealism from sparse views. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6001–6011, 2025.

Liang Han, Xu Zhang, Haichuan Song, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Sparserecon: Neural implicit surface reconstruction from sparse views with feature and depth consistencies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28514–28524, 2025.

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

Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13416–13426, 2025b.

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, 2024a.

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*, volume 39, pp. 3644–3652, 2025.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 406–413, 2014.

Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

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/`.

Minsu Kim, Subin Jeon, In Cho, Mijin Yoo, and Seon Joo Kim. Exploregs: Explorable 3d scene reconstruction with virtual camera samplings and diffusion priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27042–27051, 2025.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

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

Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas J Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with camera control. *Advances in Neural Information Processing Systems*, 37:16240–16271, 2024.

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*, pp. 20775–20785, 2024.

Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model. *arXiv preprint arXiv:2408.16767*, 2024a.

Kunhao Liu, Ling Shao, and Shijian Lu. Novel view extrapolation with video diffusion priors. *arXiv preprint arXiv:2411.14208*, 2024b.

Xi Liu, Chaoyi Zhou, and Siyu Huang. 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. *Advances in Neural Information Processing Systems*, 37:133305–133327, 2024c.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

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 SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5480–5490, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Kevin Raj, Christopher Wewer, Raza Yunus, Eddy Ilg, and Jan Eric Lenssen. Spurfies: Sparse surface reconstruction using local geometry priors. *3DV*, 2025.

Yiyang Shen, Yin Yang, Kun Zhou, and Tianjia Shao. Sparse-to-complete: From sparse image captures to complete 3d scenes. *arXiv preprint arXiv:2605.05664*, 2026.

Natsuki Takama, Shintaro Ito, Koichi Ito, Hwann-Tzong Chen, and Takafumi Aoki. Sparse2dgs: Sparse-view surface reconstruction using 2d gaussian splatting with dense point cloud. *arXiv preprint arXiv:2505.19854*, 2025.

Zhen Tan, Xieyuanli Chen, Jinpu Zhang, Lei Feng, and Dewen Hu. Uncertainty-aware normal-guided gaussian splatting for surface reconstruction from sparse image sequences. *arXiv preprint arXiv:2503.11172*, 2025.

Atakan Topaloğlu, Kunyi Li, Michael Niemeyer, Nassir Navab, A Murat Tekalp, and Federico Tombari. Oraclegs: Grounding generative priors for sparse-view gaussian splatting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 77–87, 2026.

Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9065–9076, 2023.

Jiepeng Wang, Yuan Liu, Peng Wang, Cheng Lin, Junhui Hou, Xin Li, Taku Komura, and Wenping Wang. Gaussurf: Geometry-guided 3d gaussian splatting for surface reconstruction. *arXiv preprint arXiv:2411.19454*, 2024.

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. *Advances in neural information processing systems*, 2021.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.

Haoyu Wu, Alexandros Graikos, and Dimitris Samaras. S-volsdf: Sparse multi-view stereo regularization of neural implicit surfaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3556–3568, 2023.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26024–26035, 2025a.

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

Yulun Wu, Han Huang, Wenyuan Zhang, Chao Deng, Ge Gao, Ming Gu, and Yu-Shen Liu. Sparis: Neural implicit surface reconstruction of indoor scenes from sparse views. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8514–8522, 2025b.

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

Jiale Xu, Shenghua Gao, and Ying Shan. Freesplatter: Pose-free gaussian splatting for sparse-view 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25442–25452, 2025b.

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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1790–1799, 2020.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4578–4587, 2021.

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.

Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12405–12414, 2025.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.

A Implementation Details

We provide implementation details for the notation, hyperparameters, and training configuration used in the proposed framework. Unless otherwise specified, the underlying 2DGS optimization follows the default InstantSplat (Fan et al., 2024) configuration. We report only the hyperparameters that are newly introduced or modified by our reliability-aware pipeline, including candidate-view generation and pruning, camera-pose optimization, ACM training, and additional loss weights.

To improve optimization stability, we first optimize the 2DGS representation using the sparse input views and then enable camera-pose refinement after the initial geometry has been established. In our implementation, pose optimization starts after 1,000 training iterations. This delayed update schedule prevents early pose changes from destabilizing the coarse geometric anchor.

Table 5: Notation and hyperparameters used in the proposed framework. The upper part summarizes the main notation, while the lower part reports the default hyperparameter values used in our experiments.

Notation or Hyperparameter	**Description or Value**
Notation	
V_{in}	Input view set
$\hat{T}_m^{\text{aug}}$	Estimated pose of augmented view m
$\mathcal{C}$	Set of generated candidate sequences
$A^{(s)}$	Generated candidate sequence s
A^*	Selected candidate sequence after set selection
V_{aug}	Retained augmented image and pose pairs after view pruning
S_{set}	Candidate sequence score for set selection
S_{view}	View reliability score for view pruning
Θ	2DGS parameters
q_{cons}	Patch level consistency score
q_{diff}	Patch level reconstruction difficulty score
w	Patch level reliability weight
α_t	Reliability activation state at iteration t
$\tilde{w}_t$	Effective reliability weight at iteration t
Ω	Co visibility mask used for augmented view pruning
Ω_{rend}	Valid rendering region used for geometric regularization
Hyperparameters	
N_{in}	2 input views
N_{set}	32 candidate view sequences
k_{view}	14 augmented views retained after view pruning
Pruning iteration	1,000
Pose optimization start	Iteration 1,000
Total training iterations	10,000

A.1 Resources

All experiments are conducted on a workstation with an Intel Xeon Silver 4210 CPU at 2.20GHz, 128GB RAM, and an NVIDIA RTX 3090 GPU with 24GB VRAM. For each scene, generating 32 augmented candidate sets takes approximately 30 minutes. Training and evaluation each take approximately 30 minutes.

A.2 Camera trajectory construction.

We construct the augmented camera trajectory using spherical grid sampling rather than direct interpolation between the input camera poses. We first estimate a common look at center from the optical axes of the two input cameras and represent both camera poses in spherical coordinates relative to this center. We then sample a 4×4 grid of camera poses over the azimuth and elevation ranges de-

Table 6: Optimization settings and loss weights used in the final configuration. Camera pose learning rates are scaled relative to the default pose learning rates of the backbone implementation.

Category	Setting
Learning rates and optimization	
Camera rotation LR	10^{-4}
Camera translation LR	10^{-4}
Controller warm up	3,000 iterations
Loss weights	
λ_{normal}	0.05
λ_{dist}	200

fined by the input cameras. To increase coverage of occluded regions, we expand the elevation range by 10° in both directions and increase the average camera radius by 0.15. This procedure produces $N_{\text{aug}} = 16$ augmented camera poses distributed across the resulting spherical region. Finally, each augmented camera is oriented toward the common look at center, yielding the trajectory $\hat{\Pi}_{\text{aug}}$ used for candidate view generation.

A.3 Details of the Auxiliary Control Module

ACM takes a rendered target image pair $(\hat{I}^{\text{rend}}, I^{\text{tar}})$ as input and extracts patch level feature maps from both images using a frozen representation model $\mathcal{E}$, obtaining $f_r, f_t \in \mathbb{R}^{C\times h\times w}$. Both images are resized to a fixed resolution before encoding, and the encoder weights remain frozen throughout optimization. The extracted features are detached from the computational graph so that the resulting reliability estimate does not receive gradients from the reconstruction objective.

Reliability score computation. The consistency score q_{cons} is defined as the per patch cosine similarity between the render and target features, clamped to a nonnegative range

$$q_{\text{cons}}[i,j] = \max\Big(0,\ \frac{f_r[:,i,j]\cdot f_t[:,i,j]}{\|f_r[:,i,j]\|\,\|f_t[:,i,j]\|}\Big). \tag{10}$$

A higher q_{cons} indicates greater feature consistency between the rendered and target patches, while a lower value indicates greater disagreement that may arise from inaccurate generated content, appearance variation, or pose error.

The difficulty score q_{diff} is derived from the pixel level absolute residual between the render and target. The residual is spatially pooled to the patch resolution and normalized independently for each image

$$q_{\text{diff}}[i,j] = \frac{r[i,j]-\min r}{\max r-\min r+\epsilon}, \qquad r = \text{AvgPool}_{h\times w}\big(|\hat{I}^{\text{rend}} - I^{\text{tar}}|\big), \tag{11}$$

where ϵ is a small constant for numerical stability. A higher q_{diff} indicates stronger disagreement between the rendered and target patches. Such disagreement may arise from inaccurate generated content or from regions that have not yet been sufficiently reconstructed by the current geometry.

Both scores are detached from the computational graph and are used only to construct the reliability weight. We combine them as

$$w = q_{\text{cons}}\,(1 - q_{\text{diff}})\,. \tag{12}$$

The resulting patch-wise reliability weight is bilinearly upsampled to full image resolution before being incorporated into the augmented view photometric loss.

Weighted photometric loss. The effective reliability map $\tilde{w}_t$ is applied to both components of the augmented view photometric loss. For the L1 term, pixel residuals are weighted and normalized by the sum of the reliability weights. For the SSIM term, $\tilde{w}_t$ is applied through element wise masking of the rendered and target images before the SSIM window computation.

Adaptive activation controller. We implement the adaptive activation controller using the temporal evolution of the spatially averaged difficulty score $\bar{q}_{\text{diff}}(t)$. After a warm up period t_0, the controller is evaluated every r iterations. At each evaluation, we compute the mean difficulty over two consecutive windows of size W

$$\mu_{\text{recent}}(t) = \frac{1}{W} \sum_{\tau=t-W}^{t-1} \bar{q}_{\text{diff}}(\tau), \qquad \mu_{\text{older}}(t) = \frac{1}{W} \sum_{\tau=t-2W}^{t-W-1} \bar{q}_{\text{diff}}(\tau). \tag{13}$$

The controller state $\alpha_t \in \{0, 1\}$ is updated using a relative threshold δ

$$\alpha_t = \begin{cases} 1, & \mu_{\text{recent}}(t) < (1-\delta)\mu_{\text{older}}(t), \\ 0, & \mu_{\text{recent}}(t) > (1+\delta)\mu_{\text{older}}(t), \\ \alpha_{t-r}, & \text{otherwise.} \end{cases} \tag{14}$$

The difficulty score is recorded regardless of the controller state.

Hyperparameters. Table 7 summarizes the hyperparameters used for the adaptive activation controller.

Table 7: Hyperparameters of the adaptive activation controller.

Parameter	Symbol	Value
Warm up iteration	t_0	3000
Averaging window	W	500
Relative threshold	δ	0.05
Evaluation interval	r	200

A.4 Loss Functions

We use the standard reconstruction losses of 2DGS (Huang et al., 2024a) together with the reliability aware photometric loss for augmented views and a depth normal consistency term. Unless otherwise specified, the original 2DGS loss formulation is retained.

Photometric loss. For the input views, we use the standard photometric objective consisting of an L1 term and SSIM

$$\mathcal{L}_{\text{photo}}^{\text{in}} = (1 - \lambda_{\text{dssim}}) \left\| \hat{I}^{\text{rend}} - I^{\text{tar}} \right\|_1 + \lambda_{\text{dssim}} \left(1 - \text{SSIM} \left(\hat{I}^{\text{rend}}, I^{\text{tar}} \right) \right). \tag{15}$$

We set $\lambda_{\text{dssim}} = 0.2$ in all experiments.

For augmented views, we use the effective reliability weight $\tilde{w}_t$ defined in Appendix A.3. The corresponding photometric objective is

$$\mathcal{L}_{\text{photo}}^{\text{aug}} = (1 - \lambda_{\text{dssim}}) \frac{\sum_p \tilde{w}_{t,p} \left| \hat{I}_p^{\text{rend}} - I_p^{\text{tar}} \right|}{\sum_p \tilde{w}_{t,p} + \epsilon} + \lambda_{\text{dssim}} \left(1 - \text{SSIM}_{\tilde{w}_t} \left(\hat{I}^{\text{rend}}, I^{\text{tar}} \right) \right). \tag{16}$$

The implementation of the reliability weighted SSIM term is described in Appendix A.3.

Geometric regularization. Following 2DGS (Huang et al., 2024a), we use distortion and normal consistency regularization. The distortion loss is computed from the rendered distortion map as

$$\mathcal{L}_{\text{dist}} = \frac{1}{|\Omega_{\text{rend}}|} \sum_{\mathbf{u} \in \Omega_{\text{rend}}} D_{\text{dist}}(\mathbf{u}), \tag{17}$$

where D_{dist} denotes the rendered distortion map and Ω_{rend} denotes the valid rendering region.

The normal consistency loss encourages agreement between the rendered normal $\hat{\mathbf{n}}(\mathbf{u})$ and the normal $\mathbf{N}(\mathbf{u})$ estimated from the rendered depth

$$\mathcal{L}_{\text{normal}} = \frac{1}{|\Omega_{\text{rend}}|} \sum_{\mathbf{u} \in \Omega_{\text{rend}}} \left(1 - \hat{\mathbf{n}}(\mathbf{u})^\top \mathbf{N}(\mathbf{u})\right). \tag{18}$$

Depth normal consistency. We additionally use monocular depth estimates to provide geometric guidance during optimization. Let $\mathbf{N}^{\text{mono}}$ denote the surface normal derived from the monocular depth estimate. The depth normal consistency loss encourages agreement between the reconstructed surface orientation and this prior

$$\mathcal{L}_{\text{dn}} = \frac{1}{|\Omega_{\text{rend}}|} \sum_{\mathbf{u} \in \Omega_{\text{rend}}} \left(1 - \hat{\mathbf{n}}(\mathbf{u})^\top \mathbf{N}^{\text{mono}}(\mathbf{u})\right). \tag{19}$$

Total objective. The complete optimization objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}^{\text{in}}_{\text{photo}} + \mathcal{L}^{\text{aug}}_{\text{photo}} + \lambda_{\text{dist}}\mathcal{L}_{\text{dist}} + \lambda_{\text{normal}}\mathcal{L}_{\text{normal}} + \lambda_{\text{dn}}\mathcal{L}_{\text{dn}}. \tag{20}$$

The loss weights used in our experiments are reported in Table 6.

A.5 Dataset selection

For surface evaluation on DTU, we use the evaluation data provided by S-VolSDF (Wu et al., 2023), which are derived from the DTU dataset. Among the 11 scans included in this evaluation protocol, we select five scans for our quantitative evaluation, namely scan21, scan24, scan110, scan114, and scan118. For all methods, we use the same sparse input views, geometric anchoring model, and candidate view generation model to ensure a consistent comparison. We rerun all compared methods under this common experimental setting rather than relying on previously reported results.

We also present additional examples on Tanks and Temples (Knapitsch et al., 2017) and BlendedMVS (Yao et al., 2020) to examine the behavior of the proposed framework beyond the main DTU evaluation.

A.6 Evaluation Protocol Details

We evaluate rendering quality using PSNR, SSIM, and LPIPS, and geometric quality using masked Chamfer distance, precision, recall, and F1 score. The same evaluation procedure is applied to all compared methods.

Rendering evaluation. Following the PixelNeRF evaluation protocol (Yu et al., 2021), we evaluate rendering quality on 16 views selected from the 49 DTU images. The evaluation view indices are $\{1, 2, 9, 10, 11, 12, 14, 15, 22, 26, 27, 31, 32, 35, 42, 46\}$. Object masks are applied to restrict the computation of PSNR, SSIM, and LPIPS to the foreground object regions.

Geometry evaluation. Before computing geometry metrics, each reconstructed surface is aligned with the ground truth geometry using the Open3D (Zhou et al., 2018) registration procedure described in Algorithm 1. After registration, we compute masked Chamfer distance, precision, recall, and F1 score using the processed DTU surface evaluation data described in the preceding section.

Algorithm 1: Point Cloud Registration and Alignment

Input : Source vertices $\mathbf{V}_s \in \mathbb{R}^{N\times 3}$, pre-alignment rotation $\mathbf{R} \in \mathbb{R}^{3\times 3}$, translation $\mathbf{t} \in \mathbb{R}^3$, scale factor s, ground-truth point cloud $\mathcal{P}_{gt}$, and ICP threshold thr_{icp}.

Function: In Open3D,
`PointCloud`: converts vertices into a point cloud object.
`PointCloud.transform`: applies a rigid transformation.
`execute_global_registration`: performs RANSAC-based global alignment.
`registration_icp`: refines the alignment using ICP.

Output : Aligned point cloud $\mathbf{V}_{aligned}$.

$\mathbf{V}_{scaled} \leftarrow s(\mathbf{R}\mathbf{V}_s + \mathbf{t})$

$\widehat{\mathcal{P}} \leftarrow \texttt{PointCloud}(\mathbf{V}_{scaled})$

$\text{result}_{ransac} \leftarrow \texttt{execute_global_registration}(\widehat{\mathcal{P}}, \mathcal{P}_{gt})$

$\text{reg}_{icp} \leftarrow \texttt{registration_icp}(\widehat{\mathcal{P}}, \mathcal{P}_{gt}, \text{thr}_{icp}, \text{result}_{ransac}.\text{transformation})$

$\widehat{\mathcal{P}}_{aligned} \leftarrow \widehat{\mathcal{P}}.\texttt{transform}(\text{reg}_{icp}.\text{transformation})$

$\mathbf{V}_{aligned} \leftarrow \texttt{asarray}(\widehat{\mathcal{P}}_{aligned}.\text{points})$

return $\mathbf{V}_{aligned}$

For precision, recall, and F1 score, we use a distance threshold of $15\,\text{mm}$ for all methods and experiments.

Observed and unobserved evaluation. We additionally evaluate geometric reconstruction according to input view visibility. Each ground truth surface point is classified using the z buffer visibility procedure described in Appendix A.7. A point is assigned to the observed subset if it is visible from at least one input view and to the unobserved subset otherwise. Precision, recall, and F1 score are computed separately for the observed and unobserved subsets after applying the same registration procedure. We use the same $15\,\text{mm}$ distance threshold for both subsets.

A.7 Occlusion Severity Evaluation Protocol

Occlusion severity metric. Given a set of N_{in} input cameras and the ground truth surface point cloud $\mathcal{P}_{\text{gt}}$, we compute a per point visibility label using a z buffer visibility test. Each point $\mathbf{p} \in \mathcal{P}_{\text{gt}}$ is projected into each input camera and compared with the rendered depth map using a fixed tolerance. A point is labeled *observed* if it is visible from at least one input camera and *unobserved* otherwise. We define the occlusion severity of an input configuration as

$$\rho = 1 - \frac{1}{|\mathcal{P}_{\text{gt}}|} \sum_{\mathbf{p}\in\mathcal{P}_{\text{gt}}} \mathbf{1}[\mathbf{p} \text{ is observed}], \tag{21}$$

where $\rho \in [0, 1]$ represents the fraction of the ground truth surface that is unobserved from the input views. A larger ρ indicates that a greater fraction of the surface lacks direct observations. The metric is independent of the reconstruction method and depends only on the input camera configuration and the ground truth surface.

Evaluation split. The same visibility labels are used to partition $\mathcal{P}_{\text{gt}}$ into observed and unobserved subsets for region specific evaluation. Precision, recall, and F1 score are computed on these subsets using the evaluation procedure described in Appendix A.6.

Study setup. We conduct the occlusion severity analysis on DTU scan24 using two input camera configurations with different unobserved surface fractions. The Low configuration uses cameras 22 and 25, while the High configuration uses cameras 28 and 40. Table 8 reports the measured ρ for each configuration.

Table 8: Input configurations used for the occlusion severity analysis on DTU scan24. ρ denotes the fraction of the ground truth surface that is unobserved from the two input views.

Severity level	Input cameras	ρ
Low	22, 25	0.264
High	28, 40	0.282



Figure 4: Qualitative comparisons between the 2DGS baseline (Huang et al., 2024a) and our method. From top to bottom, we show results on Tanks and Temples (Knapitsch et al., 2017), BlendedMVS (Yao et al., 2020), and other challenging scenes.

The measured unobserved surface fraction increases from the Low configuration to the High configuration.

B Additional Experimental Results

In this section, we provide additional qualitative results on challenging sparse view scenes from other datasets and comparisons between the best and worst augmented view sets.

B.1 Additional Qualitative Results

We provide additional qualitative examples on challenging sparse view scenes. The results further illustrate the ability of the proposed framework to recover surfaces in regions that are poorly supported by the initial reconstruction. Figure 4 presents additional examples from Tanks and Temples Knapitsch et al. (2017), BlendedMVS Yao et al. (2020), and other challenging scenes, comparing the 2DGS baseline with our method.

B.2 Comparison of Best and Worst Augmented View Sets

We further examine how the quality of generated augmented views affects geometric reconstruction. Figure 5 compares representative augmented view sets from DTU scan21. The top row shows the sparse input views, while the following rows show the best and worst generated view sets among the candidates. The score maps illustrate differences in agreement between the generated views and the observed scene.

The best augmented view set preserves the global appearance and structure of the scene more consistently with the sparse input views. In contrast, the worst augmented view set exhibits stronger

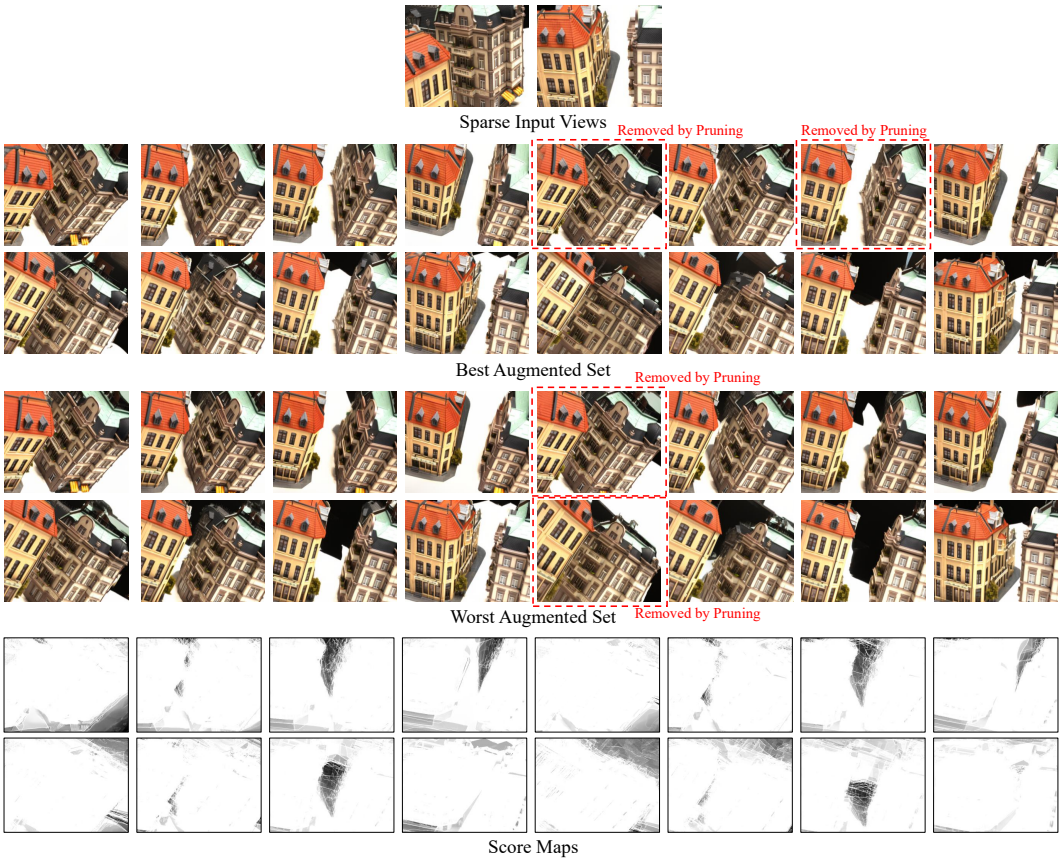


Figure 5: Comparison between the best and worst augmented view sets on DTU scan21. We visualize the sparse input views, the best generated view set, the worst generated view set, and their corresponding score maps. The worst set exhibits stronger appearance drift, pose mismatch, and local hallucinated structures, which can degrade reconstruction when used as additional observations.

appearance drift, pose mismatch, and local hallucinated structures. Such inconsistencies can affect the geometry obtained during subsequent joint optimization.

Table 9 compares the geometric reconstruction obtained using the best and worst augmented view sets. Using the worst set increases the geometric error over the complete surface, with a more pronounced degradation in the observed region. The observed surface also shows lower recall and F1 score when the worst set is used. In contrast, the differences in the unobserved region are relatively small, and the worst set does not consistently underperform the best set across the corresponding metrics. These results provide illustrative evidence that candidate set quality affects subsequent geometric reconstruction, particularly over the complete and observed surfaces, supporting the use of score based set selection before joint optimization.

Table 9: **Effect of candidate set selection on geometric reconstruction.** We compare the highest and lowest scoring augmented view sets among the candidate sequences on DTU scan21 while keeping all other reconstruction settings fixed. Best results are shown in **bold**.

Candidate Set	All				Observed				Unobserved			
	Chamfer ↓	Prec. ↑	Rec. ↑	F1 ↑	Chamfer ↓	Prec. ↑	Rec. ↑	F1 ↑	Chamfer ↓	Prec. ↑	Rec. ↑	F1 ↑
Best set	**9.833**	**0.9809**	**0.9214**	**0.9502**	**10.442**	**0.9822**	**0.8951**	**0.9366**	**9.573**	0.9784	**0.9609**	0.9695
Worst set	11.639	0.9798	0.9072	0.9421	13.127	0.9766	0.8697	0.9200	9.911	**0.9849**	0.9578	**0.9712**