RECONAGENT: AN AGENTIC FRAMEWORK FOR EVIDENCE-COUPLED SPARSE-VIEW INDOOR SURFACE RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse-view indoor surface reconstruction is inherently under-constrained: generative priors can supplement missing observations, but their predictions may also introduce unsupported geometry. Accordingly, we formulate reliable generative reconstruction as two coupled decisions: where to acquire missing evidence and which generated evidence to admit. To this end, we introduce ReconAgent, a closed-loop agentic framework built around a shared scene state, designed to jointly coordinate evidence acquisition and admission. Given the shared scene state, a gain-aware planner selects complementary virtual viewpoints based on viewing-sector coverage and missing-area rate, while a reconstruction-specific vision-language program then converts each selected viewpoint into a structured generation request. The generator synthesizes candidate RGB views with corresponding depth and normal hypotheses, which are subsequently validated by the arbiter against real-view evidence and the current reconstruction before being admitted as valid geometric evidence. Even after admission, generated geometry is integrated only into regions lacking support from real observations, preventing it from overriding observation-supported structure. The admitted evidence updates a canonical gaussian scene, whose refreshed state in turn guides subsequent planning toward unresolved regions. Under the five-view protocol, ReconAgent improves the F-score over the MAtCha baseline by 6.78 points on Replica and 6.54 points on ScanNet++, achieving the best performance on all three geometric metrics while attaining the best or tied-best novel-view synthesis results among the compared methods.

1 INTRODUCTION

Sparse-view indoor surface reconstruction is a challenging problem with widespread implications in robotics, embodied AI, and AR/VR. Recent advances in neural rendering and 3D Gaussian Splatting (Kerbl et al., 2023) have substantially improved reconstruction quality, yet recovering complete geometry in unobserved regions remains difficult: occluded objects and unseen boundaries provide little or no direct geometric evidence (Guédon et al., 2025; Guédon & Lepetit, 2024). Consequently, reconstruction in these regions is inherently ambiguous: substantially different geometries can remain consistent with the same sparse observations and even yield similarly plausible renderings.

Existing approaches mitigate this ambiguity by incorporating learned and geometric priors (Yu et al., 2021; Li et al., 2024), enforcing cross-view matching constraints (Peng et al., 2024), and imposing structural regularization (Park et al., 2025). Among these methods, MAtCha (Guédon et al., 2025) represents scene surfaces with monocular-depth-initialized charts aligned to sparse SfM observations. Coupled with differentiable 2D gaussian surfel rendering, this representation enables sharp and accurate surface reconstruction from only a few RGB views while retaining high-quality novel-view synthesis. More recently, generative approaches have opened up a promising direction by synthesizing unobserved views or completing missing scene content (Kong et al., 2025; Ren et al., 2025; Wu et al., 2025b;a; Zhong et al., 2025). G4Splat (Ni et al., 2026) integrates metric-scale geometric guidance throughout the generative pipeline to improve novel-view selection and the multi-view consistency of synthesized content. Despite this progress, generated predictions remain unobserved and may introduce geometric or structural inconsistencies.

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.

This dilemma leads to two coupled decisions in sparse-view reconstruction: *which unobserved viewpoints to target* and *which generated predictions to admit as supervision*. A selected viewpoint should expose unresolved structure rather than revisit already supported regions, whereas the resulting prediction should be consistent with real observations and the current reconstruction. These decisions are coupled because each accepted update changes the scene state and, with it, the targets and predictions needed for subsequent refinement.

We build **ReconAgent** directly on the strong MAtCha baseline, retaining its chart alignment and geometric priors for gaussian reconstruction. ReconAgent extends this backbone with a shared-state closed-loop agentic framework that coordinates viewpoint selection and prediction admission. Its shared scene state records visibility, coverage, confidence, and the provenance of generated records. Given this state, the planner explores virtual viewpoints around the fixed real input cameras and prioritizes new viewing-sector coverage, followed by missing-area rate. Inspired by structured visual programs (Wüst et al., 2025), we design a reconstruction-specific Vision-Language Program (VLP) that converts each selected target into a structured generation request specifying the target region, viewpoint, and required output channels. The generator synthesizes an RGB view and then estimates the corresponding depth and normal hypotheses. The arbiter evaluates these hypotheses against real-view support and the current reconstruction before admitting them as supervision. We use evidence authority to denote a generated record's permission to supervise refinement.

ReconAgent combines accepted predictions with real observations to update a single canonical gaussian scene. Real RGB observations provide the primary photometric supervision, while geometric priors constrain surface refinement. Generated-view updates preserve the parameters of gaussians supported by real observations. After each update, the reconstructed scene and record history define the next planning state, directing subsequent requests toward unresolved regions.

We evaluate ReconAgent on Replica (Straub et al., 2019) and ScanNet++ (Yeshwanth et al., 2023) for surface reconstruction and novel-view synthesis using five input views. ReconAgent improves F-score over its MAtCha baseline by 6.78 points on Replica and 6.54 points on ScanNet++, achieving the best surface reconstruction and best or tied-best novel-view synthesis results among the compared methods. Ablation studies on Replica further support the contributions of its main components.

Our main contributions are as follows:

- We introduce ReconAgent, a shared-state closed-loop framework that couples complementary viewpoint selection with selective use of generated supervision for sparse-view indoor surface reconstruction.
- We develop a gain-aware planner that selects complementary views of unresolved regions, together with a reconstruction-specific VLP that specifies each target and the required output channels.
- We introduce an arbiter that evaluates generated hypotheses against real-view support and the current reconstruction and selectively admits predictions for subsequent refinement.
- We demonstrate the effectiveness of ReconAgent on both surface reconstruction and novel-view synthesis tasks using Replica and ScanNet++.

2 Related Work

2.1 Sparse-view Indoor Scene Reconstruction

Classical multi-view reconstruction pipelines estimate geometry from multi-view correspondences (Furukawa & Ponce, 2010), then recover surfaces through volumetric fusion (Curless & Levoy, 1996) or implicit fitting (Kazhdan et al., 2006). Neural implicit representations instead model scenes with radiance fields (Mildenhall et al., 2020) for novel-view synthesis or signed distance fields (Wang et al., 2021) for surface reconstruction. Under sparse observations, image-conditioned priors (Yu et al., 2021) and regularization over unobserved viewpoints (Niemeyer et al., 2022) help alleviate the under-constrained nature of radiance-field reconstruction. For sparse-view gaussian reconstruction, recent methods strengthen geometric reasoning through learned initialization (Fan et al., 2025), cross-view correspondence constraints (Peng et al., 2024), depth-based normalization (Li et al., 2024), and stochastic gaussian removal (Park et al., 2025). Complementary to these

optimization strategies, 2D gaussian surfels (Huang et al., 2024) explicitly represent surfaces with flattened Gaussian primitives, while surface-alignment regularization (Guédon & Lepetit, 2024) encourages gaussians to conform to the underlying scene geometry, facilitating explicit surface recovery. MAtCha (Guédon et al., 2025) further represents scene surfaces with monocular-depth-initialized charts aligned to sparse SfM observations, and couples this explicit representation with differentiable 2D gaussian surfel rendering, achieving strong performance in sparse-view surface reconstruction and novel-view synthesis. Given this strong geometric foundation, we adopt MAtCha as the reconstruction backbone for ReconAgent and retain its chart-based geometric constraints.

2.2 Generative priors for sparse-view reconstruction

Generative priors supplement missing observations in image space or directly in 3D. Image diffusion supports novel-view regularization (Wu et al., 2024), multi-view augmentation (Gao et al., 2024; Kong et al., 2025), rendered-view repair (Wu et al., 2025a), multi-view outpainting (Huang et al., 2025), and panoramic completion (Qureshi et al., 2026). Video priors promote cross-view consistency through reconstruction-conditioned generation (Wu et al., 2025b) and scene-grounding guidance (Zhong et al., 2025). Among recent generative approaches, G4Splat (Ni et al., 2026) stands out by incorporating planar geometry into depth estimation, viewpoint selection, and inpainting, providing stronger geometric guidance throughout the generation process. View selection uses uncertainty and vision-language guidance (Xu et al., 2026) or camera potential fields (An et al., 2026). Latent diffusion (Liao et al., 2025) reconstructs Gaussian scenes from a single image, while GenRecon (Schmid et al., 2026) transfers object-level priors to multi-view reconstruction through overlapping spatial chunks. OracleGS (Topaloglu et al., 2025) weights generated supervision by MVS uncertainty; ExploreGS (Kim et al., 2025) uses image- and pixel-level confidence. Meanwhile, recent agentic frameworks have explored scene-grounded editing (Liu et al., 2026), critic–actor layout refinement (Ran et al., 2026), and debate-based articulation recovery (Boudjoghra et al., 2026). ReconAgent couples this agentic reasoning with record-level admission and spatial authority constraints to govern whether, where, and to what extent generated predictions supervise reconstruction, whose updated state guides subsequent planning.

3 Method

Building on MAtCha's chart-based geometric reconstruction backbone (Guédon et al., 2025), ReconAgent introduces a shared-state closed loop that couples complementary viewpoint selection with selective admission of generated supervision, which we present in figure 1. Figure 2 shows that, at each stage, the planner first identifies unresolved regions from the current scene state. The VLP then converts the selected targets into structured generation requests, and the generator produces candidate RGB views together with their geometric hypotheses to supplement missing observations. The arbiter evaluates these candidates against real observations and the current reconstruction, applying structural and visual-quality checks to determine admission. The accepted evidence is then integrated with real observations to update the canonical gaussian scene. This update refreshes the shared scene state, which in turn guides the next round of viewpoint selection, thereby closing the reconstruction loop. We first introduce the scene representation and shared state, and then describe the planner, generator, arbiter, and reconstruction update stages. Detailed mathematical definitions and update rules are provided in **Appendix A.1**.

3.1 Scene representation and initialization

Given a sparse set of real RGB images with known camera poses, we denote the input as:

$$\mathcal{D}_{\text{real}} = \{(I_i, \mathbf{K}_i, \mathbf{T}_i)\}_{i=1}^{N}, \qquad \mathbf{T}_i = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{1}$$

where I_i is an RGB image, $\mathbf{K}_i \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix, and $\mathbf{T}_i \in \mathrm{SE}(3)$ is the world-to-camera extrinsic matrix. Its rotation $\mathbf{R}_i \in \mathrm{SO}(3)$ and translation $\mathbf{t}_i \in \mathbb{R}^3$ map a world point $\mathbf{x}$ to camera coordinates as $\mathbf{x}_{\text{cam}} = \mathbf{R}_i\mathbf{x} + \mathbf{t}_i$. Here, N is the number of input views. We refine the geometry estimated by MASt3R (Leroy et al., 2024) using MAtCha's chart alignment (Guédon et al., 2025), then initialize and optimize the gaussian scene:

$$\mathcal{G}^{(0)} = \mathrm{DenseInitAndTrain}(\mathcal{D}_{\text{real}}). \tag{2}$$

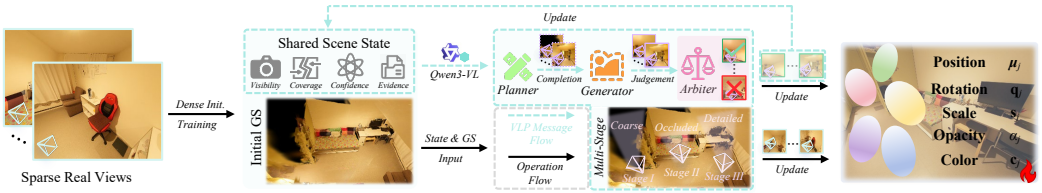

Figure 1: **Overview of ReconAgent.** MAtCha-based initialization first converts sparse real views into a canonical gaussian scene (Section 3.1). The shared state then guides the planner for target planning and VLP-based request formulation (Sections 3.2–3.3). Next, the generator produces candidate hypotheses, which the arbiter validates against real-view support and the current reconstruction (Sections 3.4–3.5). Finally, the update step integrates accepted predictions and refreshes the state, closing the loop across coarse, occluded, and detailed refinement stages (Section 3.6).

The initial gaussian field provides the geometry and rendered views for the first planning stage. At stage t, the canonical scene consists of M_t oriented gaussian surfels (Huang et al., 2024) in the Gaussian-splatting representation (Kerbl et al., 2023):

$$\mathcal{G}_t = \{g_j\}_{j=1}^{M_t}, \qquad g_j = (\boldsymbol{\mu}_j, \mathbf{q}_j, \mathbf{s}_j, \alpha_j, \mathbf{c}_j), \tag{3}$$

where $\boldsymbol{\mu}_j \in \mathbb{R}^3$ is the center, $\mathbf{q}_j \in \mathbb{S}^3$ is a unit rotation quaternion, $\mathbf{s}_j = (s_{j,1}, s_{j,2}) \in \mathbb{R}^2_+$ contains the two tangent-plane scales, $\alpha_j \in [0,1]$ is the opacity, and $\mathbf{c}_j$ contains spherical-harmonic appearance coefficients. The surfel orientation $\mathbf{R}(\mathbf{q}_j) \in \mathrm{SO}(3)$ defines the rank-two world-space covariance:

$$\boldsymbol{\Sigma}_j = \mathbf{R}(\mathbf{q}_j)\,\mathrm{diag}(s_{j,1}^2, s_{j,2}^2, 0)\mathbf{R}(\mathbf{q}_j)^\mathsf{T}. \tag{4}$$

We update this field across refinement stages. We detail projection, alpha compositing, and the rendering of depth and normals in **Appendix A.1.1**.

3.2 Shared Scene State

Planning and validation use complementary aspects of the same reconstruction state: the planner identifies unresolved regions, while the arbiter assesses hypotheses against real observations and the current geometry. ReconAgent therefore maintains a shared state $S_t \in \mathcal{S}$ over the scene domain $\Omega \subset \mathbb{R}^3$:

$$S_t = \big(\mathcal{G}_t, \mathrm{Vis}_t, \mathrm{Cov}_t, \mathrm{Conf}_t, \mathrm{Evid}_t\big). \tag{5}$$

The visibility, coverage, and confidence channels are scalar fields. The evidence ledger Evid_t stores generated records and their admission decisions, preserving their provenance:

$$(\mathrm{Vis}_t, \mathrm{Cov}_t, \mathrm{Conf}_t) : \Omega \to [0,1]^3, \qquad \mathrm{Evid}_t = \{(e_{s,k}, z_{s,k}) \mid 0 \le s < t\}. \tag{6}$$

Visibility and coverage identify unresolved regions for planning, while confidence and provenance support arbitration. Accepted predictions increase coverage but retain their generated provenance. The planner thus accounts for completed regions, while the arbiter evaluates new hypotheses using the accumulated records. The update step refreshes the state from the refined scene and recorded decisions, completing the stage transition:

$$S_t \xrightarrow{\text{PLAN}} \mathcal{P}_t \xrightarrow{\text{GENERATE}} \mathcal{E}_t \xrightarrow{\text{DECISION}} \mathcal{E}_t^+ \xrightarrow{\text{UPDATE}} S_{t+1}, \tag{7}$$

where $\mathcal{P}_t$ is the set of planned messages, $\mathcal{E}_t$ is the set of generated records, and $\mathcal{E}_t^+$ is its accepted subset. We define the shared-state channels, real-support tests, and record history in **Appendix A.1.2**.

3.3 Gain-aware Planning and VLP Requests

The planner uses the current reconstruction state to seek complementary views of unresolved regions. It samples candidate viewpoints around the fixed real input cameras. For each candidate v, it obtains the missing-area rate, conditioning support, and occupied image-grid regions from rendering, and assigns a viewing sector $\sigma(v)$. Stable-context and completion checks define the feasible candidate set $\mathcal{A}_t$. Within this set, exploration gain provides a lexicographic selection key:

$$\mathrm{Gain}_t(v) = \Big(\mathbf{1}[\sigma(v) \notin \mathcal{C}_t],\ \rho_{\mathrm{miss}}(v),\ -\mathrm{index}(v)\Big), \tag{8}$$

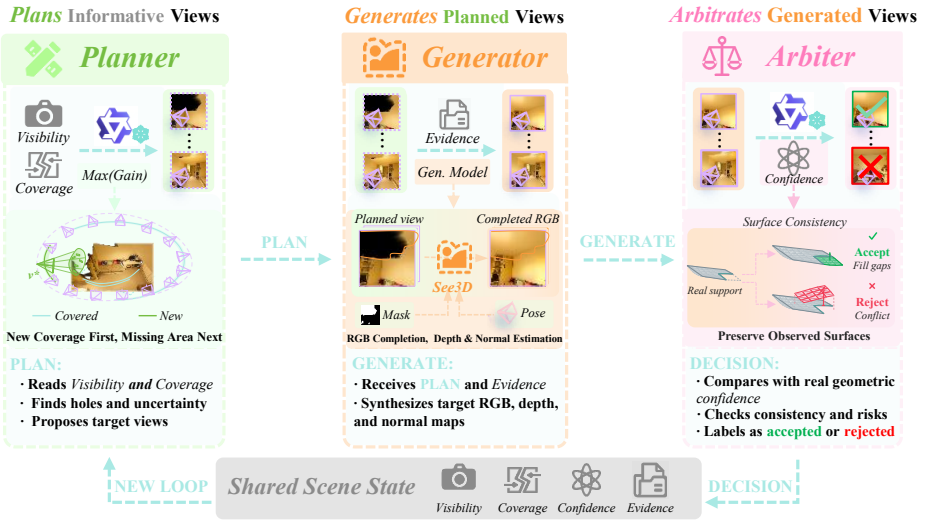

Figure 2: **Agent-level reconstruction flow.**

where $\rho_{\text{miss}}(v)$ is the rendered missing-area rate and $\mathcal{C}_t$ contains the sectors already represented during selection. The planner first prioritizes unrepresented viewing sectors, then larger missing-area rates, with smaller candidate indices breaking any remaining ties. After each choice, it adds the selected sector to $\mathcal{C}_t$ and re-ranks the remaining candidates. Sector coverage encourages requests from distinct viewing directions, while the missing-area rate directs them toward unresolved content in admissible views. Together, these cues estimate which viewpoints could add useful supervision. Let B_t denote the exploration budget at stage t and Top^{lex} the sequential selection operator. The selected queries are:

$$\mathcal{Q}_t = \text{Top}^{\text{lex}}_{B_t} \{v \in \mathcal{A}_t : \text{Gain}_t(v)\}. \tag{9}$$

Each selected query becomes a structured request specifying the region to complete and the output channels. Let $\mathcal{Q}$ denote the query space. For each $q_{t,k} \in \mathcal{Q}_t \subseteq \mathcal{Q}$, Qwen3-VL-8B-Instruct (Bai et al., 2025) instantiates a VLP adapted from structured visual programs (Wüst et al., 2025):

$$p_{t,k} = \Pi_{\text{VLP}}(S_t, q_{t,k}) = \big(r_{t,k}, v_{t,k}, \mathcal{O}_{t,k}\big), \tag{10}$$

where $r_{t,k}$ is the target image-plane region, $v_{t,k} \in \text{SE}(3)$ is the target camera's world-to-camera transform, and $\mathcal{O}_{t,k} \subseteq \mathcal{O} = \{\text{rgb}, \text{depth}, \text{normal}\}$ specifies the requested channels. ReconAgent requests all three channels. The model grounds each request in the real input images, current target-view renderings, and record history while preserving the selected viewpoint and completion mask. The messages $\mathcal{P}_t = \{p_{t,k}\}_{k=1}^{|\mathcal{Q}_t|}$ convey the planner's targets to the generator and preserve the request context for arbitration. We specify candidate filtering and sequential gain-based selection in **Appendix A.1.3** and the VLP message schema, model inputs, and validity checks in **Appendix A.1.4**.

3.4 Generator and Candidate Predictions

The generator converts the selected VLP requests into candidate records. See3D (Ma et al., 2025) synthesizes the requested RGB views jointly, and the generator then estimates the corresponding depth and normal maps. These channels form a candidate hypothesis for the arbiter to evaluate. Let $\mathcal{I}$, $\mathcal{D}_{\text{depth}}$, and $\mathcal{N}$ denote the RGB, depth, and normal-map spaces, respectively. Including each record's viewpoint and VLP program gives the record space:

$$\mathcal{Y} := \mathcal{I} \times \mathcal{D}_{\text{depth}} \times \mathcal{N}, \qquad \mathcal{V} := \text{SE}(3), \qquad \mathcal{E} := \mathcal{Y} \times \mathcal{V} \times \mathcal{P}. \tag{11}$$

The generator maps:

$$\text{Generator}_\psi : \mathcal{S} \times \mathcal{P} \longrightarrow \mathcal{E}.$$

For request $p_{t,k}$, the resulting record is:

$$e_{t,k} = \text{Generator}_\psi(S_t, p_{t,k}) = \big(\widetilde{\mathbf{Y}}_{t,k}, v_{t,k}, p_{t,k}\big), \qquad \widetilde{\mathbf{Y}}_{t,k} = (\widetilde{I}_{t,k}, \widetilde{D}_{t,k}, \widetilde{N}_{t,k}). \tag{12}$$

Here, the per-request mapping indexes one output of the jointly generated batch, with the other stage requests left implicit. The resulting records form the candidate set $\mathcal{E}_t$ for arbitration. The retained pair $(v_{t,k}, p_{t,k})$ links each hypothesis to the Planner's target and supplies the VLP context for its evaluation. We describe the map domains, joint generation inputs, and per-request provenance in **Appendix A.1.5**.

3.5 Arbiter and Authority-Aware Update

A viewpoint that exposes missing structure can still yield inconsistent predictions. The arbiter evaluates each generated hypothesis against real-view support and the current scene. It combines a multimodal structural assessment with a separate RGB quality check; explicit structural conflicts or severe visual degradation trigger rejection. Its admission decision is:

$$z_{t,k} := \mathrm{Arbiter}(e_{t,k}, S_t) \in \{\texttt{accepted}, \texttt{rejected}\}. \tag{13}$$

The update step uses the accepted set:

$$\mathcal{E}_t^+ = \{e_{t,k} \in \mathcal{E}_t : z_{t,k} = \texttt{accepted}\}. \tag{14}$$

The ledger stores admission decisions and assessment validity separately, including unresolved assessments retained by the fallback; only accepted records enter the Update step. The admissible domain of generated geometry depends on real-view support, free-space admissibility, and local confidence. Let m_t^{hole} mark regions lacking sufficient real support, m_t^{free} encode free-space admissibility, and $\gamma_{t,k}$ measure local confidence in the generated geometry. These quantities define the geometric mask:

$$\begin{aligned} m_{t,k}^{\mathrm{geo}}(\mathbf{x}) = \mathbf{1}[z_{t,k} = \texttt{accepted}]\, m_t^{\mathrm{hole}}(\mathbf{x}) m_t^{\mathrm{free}}(\mathbf{x}) \\ \cdot \mathbf{1}[\gamma_{t,k}(\mathbf{x}) \geq \tau_{\mathrm{geo}}]\,. \end{aligned} \tag{15}$$

This mask restricts the admissible domain of generated geometry to regions without sufficient real-view support. The threshold τ_{geo} sets the required local confidence. The requested-region mask $m_{t,k}^{\mathrm{app}}$, defined in the target image plane, specifies appearance support. The arbiter's binary decision selects the records used for refinement, and the masks delimit their channel-specific supervision regions. We apply these masks to the corresponding image losses and restrict parameter updates from generated supervision to geometrically admissible gaussians.

The update step combines accepted records with real-view supervision:

$$(\mathcal{G}_{t+1}, S_{t+1}) = \mathcal{U}\big(\mathcal{G}_t, S_t, \mathcal{D}_{\mathrm{real}}, \mathcal{E}_t^+\big). \tag{16}$$

The backend refines 2DGS (Huang et al., 2024) with MAtCha-derived geometric supervision (Guédon et al., 2025). Before each stage's gaussian reinitialization, we align the depths of accumulated accepted views to frozen geometry inferred from the real inputs using a bounded affine fit in inverse depth. RGB targets and geometric priors then supervise refinement, with greater photometric weight on real RGB. Generated-view steps preserve parameters and optimizer states outside the admissible gaussian set, while real-view steps retain full-scene supervision. The objective combines depth, normal, and curvature priors with normal consistency, depth ordering, and anisotropy regularization. The refined field refreshes the shared state, closing the planning loop. We detail arbitration and supervision masks in **Appendix A.1.6**, depth alignment in **Appendix A.1.7**, masked losses and parameter updates in **Appendix A.1.8**, and loss schedules in **Appendix A.2.3**.

3.6 Multi-Stage Refinement

ReconAgent applies this loop across three refinement stages with distinct spatial priorities. The coarse stage targets large missing structures, the occluded stage focuses on surfaces hidden from the initial views, and the detailed stage addresses residual local structure. Each stage plans from the current field and accumulated records. The update step optimizes gaussian position, rotation, scale, opacity, and appearance, while the ledger carries both acceptance and rejection decisions forward. Refinement with accepted predictions changes the geometry and unresolved targets in the next state, coupling admission to subsequent planning. The detailed stage returns the final scene. We formalize stage transitions and record-history propagation in **Appendix A.1.9** and give the complete reconstruction procedure in **Appendix A.1.10**.

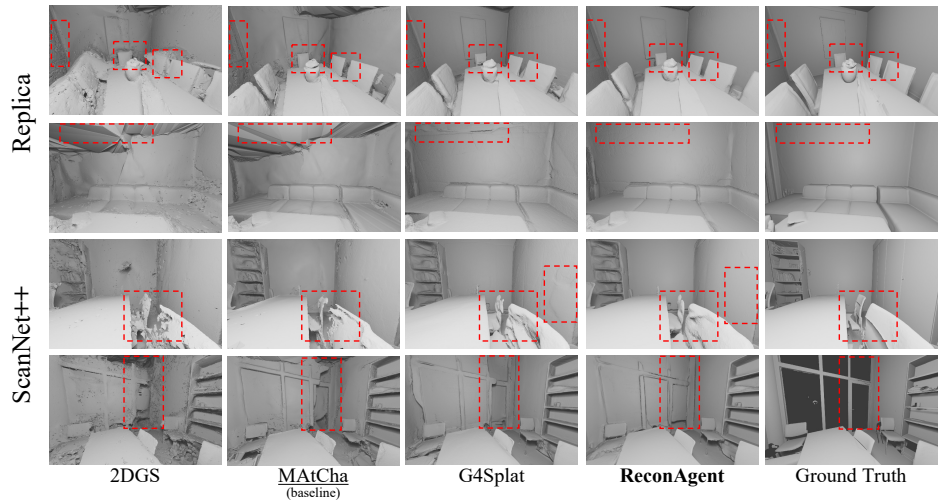

Figure 3: **Surface reconstruction on Replica and ScanNet++.**

4 EXPERIMENTS

We compare ReconAgent against MAtCha and other competitive methods on surface reconstruction and novel-view synthesis. Qualitative comparisons further examine surface continuity and appearance fidelity, while ablation studies evaluate the contributions of shared-state coordination, gain-aware planning, structured requests, generation, and prediction admission. We give data and training details in **Appendix A.2.1** and **Appendix A.2.2**, respectively.

4.1 MAIN RESULTS

Table 1: **Surface reconstruction and novel-view synthesis on Replica and ScanNet++ with five input views.** Red/orange/yellow backgrounds denote the top three ranks. Red parentheses show signed differences from MAtCha.

Dataset	Method	Surface reconstruction			Novel-view synthesis		
		CD ↓	F-score ↑	NC ↑	PSNR ↑	SSIM ↑	LPIPS ↓
Replica	3DGS	16.61	27.72	64.34	18.29	0.744	0.254
	2DGS	14.64	48.01	74.14	18.43	0.735	0.306
	FSGS	18.17	26.87	64.16	19.19	0.766	0.259
	InstantSplat	21.00	19.67	62.01	19.39	0.762	0.255
	G4Splat	6.61	65.14	83.98	23.90	0.836	0.199
	MAtCha	10.12	60.90	79.33	17.81	0.752	0.228
	ReconAgent	6.35 (−3.77)	67.68 (+6.78)	84.11 (+4.78)	24.28 (+6.47)	0.849 (+0.097)	0.192 (−0.036)
ScanNet++	3DGS	16.60	31.92	65.35	14.28	0.696	0.372
	2DGS	14.34	51.97	70.01	13.91	0.661	0.429
	FSGS	23.80	27.86	64.53	14.80	0.731	0.362
	InstantSplat	21.32	25.44	60.67	15.02	0.742	0.355
	G4Splat	6.34	67.12	77.45	18.69	0.792	0.314
	MAtCha	11.55	62.98	73.61	13.58	0.677	0.351
	ReconAgent	6.11 (−5.44)	69.52 (+6.54)	77.90 (+4.29)	18.79 (+5.21)	0.793 (+0.116)	0.314 (−0.037)

Surface reconstruction. Table 1 shows that ReconAgent achieves the lowest CD and highest F-score and NC on both datasets. Relative to its MAtCha backbone, it reduces CD by 3.77 cm on Replica and 5.44 cm on ScanNet++ and increases F-score by 6.78 and 6.54 points, respectively. The margins over G4Splat are smaller, with CD reductions of 0.26 and 0.23 cm and F-score gains of 2.54 and 2.40 points. Figure 3 localizes the geometric improvements: ReconAgent reduces MAtCha's folded ceiling artifacts and produces cleaner wall and furniture contours. Compared with

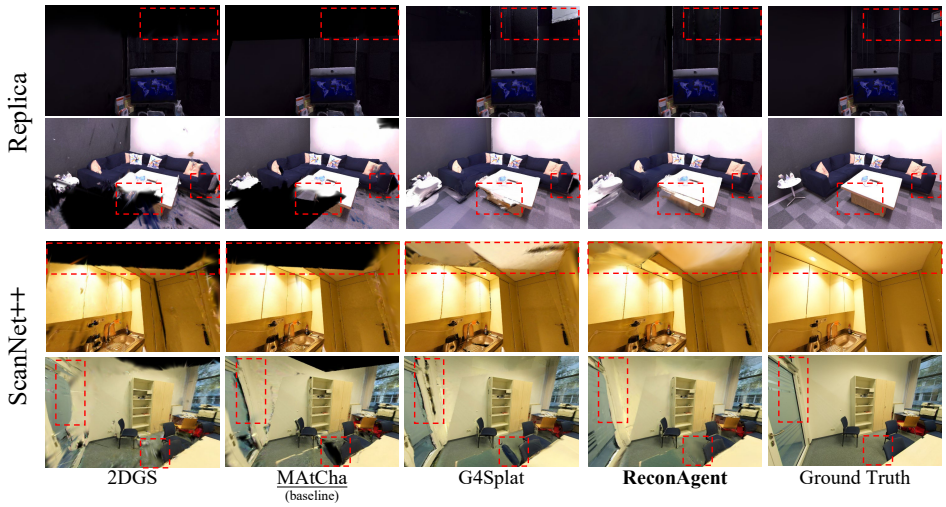

Figure 4: **Novel-view synthesis on Replica and ScanNet++.**

G4Splat, it reconstructs a more continuous ceiling strip above the sofa and fewer ridges on the highlighted ScanNet++ wall, although the chair and window retain geometric discrepancies. These local improvements can be visually prominent within similar room layouts because CD and NC average surface distances and normal agreement, while F-score measures thresholded proximity. These results support complementary planning and selective admission that complete unresolved regions while preserving observation-supported geometry.

Novel-view synthesis. Table 1 shows that ReconAgent improves PSNR over MAtCha by 6.47 dB on Replica and 5.21 dB on ScanNet++, alongside higher SSIM and lower LPIPS. It achieves the highest PSNR and SSIM on both datasets, with the lowest LPIPS on Replica and a tie with G4Splat on ScanNet++. The PSNR margins over G4Splat are 0.38 and 0.10 dB, respectively, and the ScanNet++ SSIM difference is 0.001. Figure 4 shows more pronounced improvements in local appearance: ReconAgent avoids G4Splat's bright wall patch and mottled coffee-table front on Replica, while reducing ceiling streaks and frame tearing on ScanNet++. These regions reveal more coherent surface appearance and boundaries even when image-level scores remain close, since localized artifacts contribute only part of the aggregate error. These localized gains support combining structured generation requests with joint appearance and geometry validation before refinement. We define metrics in **Appendix A.2.4**, extend surface reconstruction and novel-view synthesis comparisons in **Appendix A.3.4** and **Appendix A.3.5**, and analyze view counts and costs in **Appendix A.3.1** and **Appendix A.3.2**.

4.2 Ablation Studies

Figure 5 compares the full ReconAgent system (A) with variants that remove shared state (B), replace VLP with natural-language communication (C), replace gain-aware planning with random view selection (D), or remove the generator (E) or the arbiter (F). G denotes the underlying MAtCha reconstruction baseline.

Shared-state coordination. We notice that removing the shared state (B) increases CD from 6.35 to 6.78 and reduces F-score from 67.68 to 66.02. The visible gap in the ceiling further reflects degraded surface completeness when acquisition and admission operate without a shared scene state.

Viewpoint selection. We sample viewpoints uniformly without replacement from the same admissible pool as A, while preserving the same view count at each stage. F-score decreases by 0.44 points to 67.24, PSNR drops to 23.82, and LPIPS increases to 0.222, while SSIM remains at 0.849. These results suggest that allocating the view budget according to complementary coverage and missing-area cues leads to more effective viewpoint selection. We analyze stage-wise metrics in **Appendix A.3.3** and planner trajectories in **Appendix A.3.6**.

Metric	**A**	B	C	D	E	F	G
CD↓	**6.35**	6.78	7.62	6.49	9.35	6.68	10.12
F-score↑	**67.68**	66.02	64.55	67.24	62.85	66.54	60.90
NC↑	**84.11**	82.95	82.15	83.82	79.74	83.36	79.33
PSNR↑	**24.28**	23.55	23.18	23.82	18.35	23.58	17.81
SSIM↑	**0.849**	0.837	0.817	0.849	0.751	0.832	0.752
LPIPS↓	**0.192**	0.217	0.225	0.222	0.275	0.211	0.228

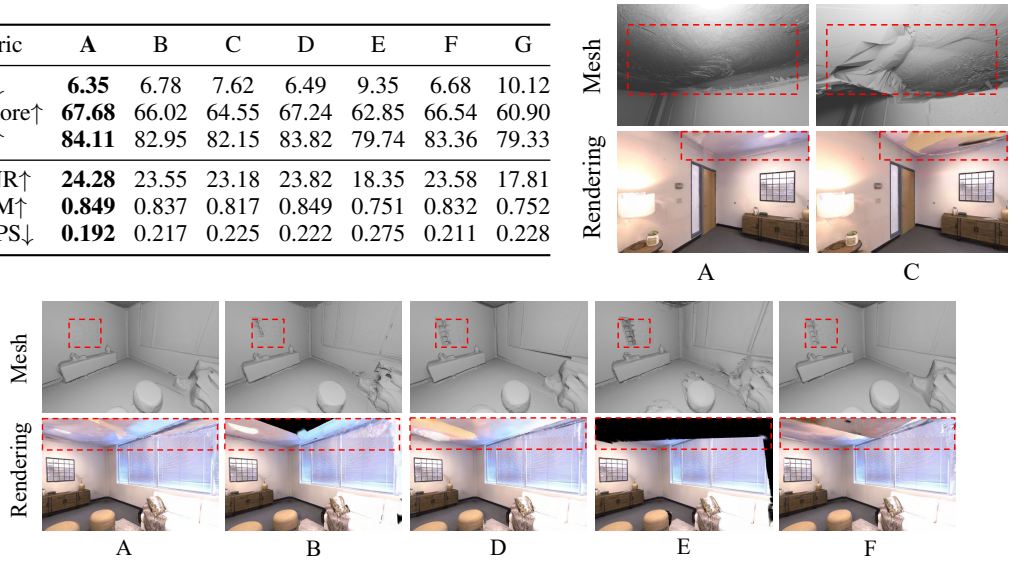

Figure 5: **Component ablations on Replica.**

Generation. Removing the generator (E) increases CD from 6.35 to 9.35, reduces F-score from 67.68 to 62.85, and lowers PSNR from 24.28 to 18.35 dB. It incurs the largest CD and PSNR degradation among the component ablations. Its pronounced ceiling gap further shows the loss of completeness without generated supervision. Together, these results support the generator's role in supplying complementary supervision for regions poorly constrained by the sparse input views, improving both surface reconstruction and novel-view synthesis.

Structured requests. Replacing VLP with natural-language messages for inter-agent communication (C), while retaining all other components and settings, raises CD to 7.62 and lowers PSNR to 23.18 dB. The visual comparison shows folded ceiling geometry and angular appearance artifacts, supporting structured information exchange across planning, generation, and arbitration.

Prediction admission. Removing prediction admission (F) reduces F-score, NC, and PSNR to 66.54, 83.36, and 23.58, respectively, compared with 67.68, 84.11, and 24.28 for the full model (A). While retaining the same planning and generation configurations, it evolves candidate predictions from its own reconstruction state. The resulting degradation supports validating generated predictions against real-view evidence and the current reconstruction before refinement. Supplementary examples further show that a rejected candidate can partially fill a ceiling gap yet retain fragmented appearance, indicating that apparent completeness alone is insufficient for admission. We present qualitative rejection examples with corresponding novel-view synthesis results in **Appendix A.3.7**.

5 Conclusion and Future Work

We presented ReconAgent, a closed-loop framework that rethinks how generative priors participate in sparse-view indoor surface reconstruction. Rather than treating generated views as additional pseudo-observations, ReconAgent formulates reconstruction as an evidence-coupled process of acquisition, admission, and authority-aware refinement, all conditioned on an evolving shared scene state. Generated hypotheses are first validated against real-view evidence and the current reconstruction, and even accepted evidence is spatially restricted to regions lacking real-view support, preventing generated supervision from overriding observation-supported geometry. Built on MAtCha, ReconAgent coordinates complementary viewpoint planning, structured generation, and selective prediction admission within this closed loop. On Replica and ScanNet++, ReconAgent improves over MAtCha, achieving the best surface reconstruction and the best or tied-best novel-view synthesis results among the compared methods. In future work, we plan to extend ReconAgent toward PBR-aware mesh reconstruction, jointly recovering geometry and physically based materials to produce richer, simulation-ready 3D assets for embodied AI applications.

AI USE STATEMENT

We used generative AI tools to assist with language polishing, LaTeX formatting, and figure layout. These uses are separate from the vision-language and generative models that form part of ReconAgent and are described in the method and experimental settings. Quantitative results come from experimental evaluation. The authors take responsibility for the final text, figures, results, and claims.

ETHICS STATEMENT

This work studies indoor reconstruction using the existing Replica and ScanNet++ benchmarks, without collecting new human-subject data. Indoor reconstructions can reveal room layouts and personal belongings; deployment therefore requires appropriate authorization and privacy safeguards. Use and redistribution of datasets and pretrained models must respect their respective licenses. Generated regions remain hypotheses rather than verified measurements: arbitration and spatially restricted updates reduce the influence of inconsistent predictions but do not guarantee geometric correctness. Safety-critical applications require independent validation of reconstructed geometry.

REPRODUCIBILITY STATEMENT

Section 3 specifies the reconstruction framework, and **Appendix A.1** details rendering, candidate selection, structured requests, arbitration, depth alignment, masked optimization, and the complete stage-wise algorithm. **Appendix A.2** reports data preparation and view selection, pretrained checkpoints, GPU hardware, optimization settings, loss weights and schedules, and evaluation metrics. The controlled ablations share the initial field, stage schedule, view budget, backend, and evaluator. **Appendix A.3** provides additional input-view comparisons, computational costs, stage-wise results, and qualitative analyses. We will publicly release our code upon acceptance of the paper.

REFERENCES

Zeyuan An, Yanghang Xiao, Zhiying Leng, Yijun Feng, and Xiaohui Liang. GSPotential: Camera potential field for sparse-view 3D gaussian splatting. *arXiv preprint arXiv:2608.29346*, 2026. URL https://arxiv.org/abs/2608.29346.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Mohamed El Amine Boudjoghra, Ivan Laptev, and Angela Dai. UnfoldArt: Zero-shot recovery of full articulated 3D objects from text or image. *arXiv preprint arXiv:2606.30608*, 2026. URL https://arxiv.org/abs/2606.30608.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 303–312, 1996. doi: 10.1145/237170.237269. URL https://graphics.stanford.edu/papers/volrange/.

Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. MASt3R-SfM: A fully-integrated solution for unconstrained structure-from-motion. In *2025 International Conference on 3D Vision (3DV)*, pp. 1–10. IEEE, 2025. doi: 10.1109/3DV66043.2025.00008.

Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, Zhangyang Wang, and Yue Wang. InstantSplat: Sparse-view gaussian splatting in seconds. *arXiv preprint arXiv:2403.20309v6*, 2025. URL https://arxiv.org/abs/2403.20309v6.

Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010. doi: 10.1109/TPAMI.2009.161.

Ruiqi Gao, Aleksander Hołyński, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: Create anything in 3D with multi-view diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 75468–75494, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/89e4433fec4b99f1d859db57af1e0a0f-Abstract-Conference.html.

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.

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

Yi-Chuan Huang, Hao-Jen Chien, Chin-Yang Lin, Ying-Huan Chen, and Yu-Lun Liu. GaMO: Geometry-aware multi-view diffusion outpainting for sparse-view 3D reconstruction. *arXiv preprint arXiv:2512.25073*, 2025. URL https://arxiv.org/abs/2512.25073.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177703732.

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, pp. 61–70, 2006. URL https://hhoppe.com/proj/poissonrecon/.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):1–14, 2023. doi: 10.1145/3592433.

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*, 2025. URL https://arxiv.org/abs/2508.06014.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

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.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision*, pp. 71–91. Springer, 2024. doi: 10.1007/978-3-031-73220-1_5.

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.

Ziwei Liao, Mohamed Sayed, Steven L. Waslander, Sara Vicente, Daniyar Turmukhambetov, and Michael Firman. Complete gaussian splats from a single image with denoising diffusion models. *arXiv preprint arXiv:2508.21542*, 2025. URL https://arxiv.org/abs/2508.21542.

Xiujin Liu, Tianyu Yang, Yilun Zhao, and Xiangliang Zhang. DesignAgent3D: Interactive 3D scene editing via designer-like multimodal reasoning. *arXiv preprint arXiv:2608.21438*, 2026. URL https://arxiv.org/abs/2608.21438.

Baorui Ma, Huachen Gao, Haoge Deng, Zhengxiong Luo, Tiejun Huang, Lulu Tang, and Xinlong Wang. You see it, you got it: Learning 3D creation on pose-free videos at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2016–2029, 2025.

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 *Computer Vision – ECCV 2020*, pp. 405–421. Springer, 2020. doi: 10.1007/978-3-030-58452-8_24.

Junfeng Ni, Yixin Chen, Zhifei Yang, Yu Liu, Ruijie Lu, Song-Chun Zhu, and Siyuan Huang. G4Splat: Geometry-guided gaussian splatting with generative prior. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kdPmsMVhZf.

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.

Hyunwoo Park, Gun Ryu, and Wonjun Kim. DropGaussian: Structural regularization for sparse-view gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21600–21609, 2025.

Rui Peng, Wangze Xu, Luyang Tang, Liwei Liao, Jianbo Jiao, and Ronggang Wang. Structure consistent gaussian splatting with matching prior for few-shot novel view synthesis. *Advances in Neural Information Processing Systems*, 37:97328–97352, 2024.

Adil Qureshi, Dongki Jung, Jaehoon Choi, and Dinesh Manocha. PanoPlane: Plane-aware panoramic completion for sparse-view indoor 3D gaussian splatting. *arXiv preprint arXiv:2605.14135*, 2026. URL https://arxiv.org/abs/2605.14135.

Xingjian Ran, Xiaoye Mo, Sihao Liu, Jianyu Zhang, Li Luo, and Bo Dai. SceneMosaic: Efficient and diverse simulation-ready scene generation via hybrid agentic layout evolution. *arXiv preprint arXiv:2609.05594*, 2026. URL https://arxiv.org/abs/2609.05594.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3C: 3D-informed world-consistent video generation with precise camera control. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6121–6132, 2025.

Katharina Schmid, Nicolas von Lützow, Jozef Hladký, Angela Dai, and Matthias Nießner. GenRecon: Bridging generative priors for multi-view 3D scene reconstruction. *arXiv preprint arXiv:2605.23888*, 2026. URL https://arxiv.org/abs/2605.23888.

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

Atakan Topaloglu, Kunyi Li, Michael Niemeyer, Nassir Navab, A. Murat Tekalp, and Federico Tombari. OracleGS: Grounding generative priors for sparse-view gaussian splatting. *arXiv preprint arXiv:2509.23258*, 2025. URL https://arxiv.org/abs/2509.23258.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27171–27183, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/e41e164f7485ec4a28741a2d0ea41c74-Paper.pdf.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26024–26035, 2025a. doi: 10.1109/CVPR52734.2025.02424.

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. ReconFusion: 3D reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2312.02981.

Sibo Wu, Congrong Xu, Binbin Huang, Andreas Geiger, and Anpei Chen. GenFusion: Closing the loop between reconstruction and generation via videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6078–6088, 2025b. doi: 10.1109/CVPR52734.2025.00570.

Antonia Wüst, Wolfgang Stammer, Hikaru Shindo, Lukas Helff, Devendra Singh Dhami, and Kristian Kersting. Synthesizing visual concepts as vision-language programs. *arXiv preprint arXiv:2511.18964*, 2025.

Tianling Xu, Shengzhe Gan, Leslie Gu, Yuelei Li, Fangneng Zhan, and Hanspeter Pfister. AREA3D: Active reconstruction agent with unified feed-forward 3D perception and vision-language guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2512.05131.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *Advances in Neural Information Processing Systems*, volume 37, pp. 21875–21911, 2024. doi: 10.52202/079017-0688.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12–22, 2023.

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.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19447–19456. IEEE, 2024a. doi: 10.1109/CVPR52733.2024.01839.

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 43(6):1–13, 2024b. doi: 10.1145/3687937.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Yingji Zhong, Zhihao Li, Dave Zhenyu Chen, Lanqing Hong, and Dan Xu. Taming video diffusion prior with scene-grounding guidance for 3D gaussian splatting from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6133–6143, 2025. doi: 10.1109/CVPR52734.2025.00575.

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

We expand the definitions and update rules in Section 3, then present implementation details and additional experiments.

A.1 ADDITIONAL METHOD DETAILS

A.1.1 GAUSSIAN RENDERING AND PARAMETER UPDATES

For camera $(\mathbf{K}_i, \mathbf{T}_i)$, we use the world-to-camera rotation $\mathbf{R}_i$ and translation $\mathbf{t}_i$ defined in Section 3.1. Let $\pi([x, y, z]^\mathsf{T}) = [x/z, y/z]^\mathsf{T}$ denote perspective division, and let $\mathbf{J}_{i,j}$ be the jacobian of $\pi(\mathbf{K}_i \mathbf{x}_{\text{cam}})$ at $\mathbf{x}_{\text{cam}} = \mathbf{R}_i \boldsymbol{\mu}_j + \mathbf{t}_i$. The projected center and first-order local covariance approximation of gaussian j follow the gaussian-splatting projection (Kerbl et al., 2023):

$$\boldsymbol{\mu}'_{i,j} = \pi(\mathbf{K}_i(\mathbf{R}_i \boldsymbol{\mu}_j + \mathbf{t}_i)), \qquad \boldsymbol{\Sigma}'_{i,j} = \mathbf{J}_{i,j} \mathbf{R}_i \boldsymbol{\Sigma}_j \mathbf{R}_i^\mathsf{T} \mathbf{J}_{i,j}^\mathsf{T}. \tag{A.1}$$

For rendering, the 2DGS backend (Huang et al., 2024) evaluates perspective-correct ray–surfel intersections with low-pass filtering. For pixel $\mathbf{u}$, let $G_{i,j}^{\text{2D}}(\mathbf{u})$ denote this filtered surfel response, and let $\prec_i$ order the contributing primitives from front to back. The opacity and transmittance for primitive j are:

$$\alpha_{i,j}(\mathbf{u}) = \alpha_j G_{i,j}^{\text{2D}}(\mathbf{u}), \qquad T_{i,j}(\mathbf{u}) = \prod_{\ell \prec_i j} \big(1 - \alpha_{i,\ell}(\mathbf{u})\big). \tag{A.2}$$

Let $\boldsymbol{\nu}_{i,j}$ denote the viewing direction for surfel j in camera i, and let $\mathbf{c}_j(\boldsymbol{\nu}_{i,j})$ be the RGB color obtained by evaluating its spherical-harmonic coefficients in that direction. The rendered RGB value is:

$$\widehat{I}_i(\mathbf{u}; \mathcal{G}_t) = \sum_{j:\, \mathbf{u} \in \text{supp}(G_{i,j}^{\text{2D}})} T_{i,j}(\mathbf{u}) \alpha_{i,j}(\mathbf{u}) \mathbf{c}_j(\boldsymbol{\nu}_{i,j}). \tag{A.3}$$

Let $d_{i,j}(\mathbf{u})$ and $\boldsymbol{n}_{i,j}(\mathbf{u})$ denote the camera-space depth and world-space normal of primitive j. With $w_{i,j} = T_{i,j}\alpha_{i,j}$ and accumulated opacity $A_i = \sum_j w_{i,j}$, the expected depth is $D_i^{\text{exp}} = \sum_j w_{i,j} d_{i,j} / A_i$ at supported pixels. Let D_i^{med} be the rasterizer's median depth. The refinement backend computes depth and normals as follows:

$$\widehat{D}_i(\mathbf{u}; \mathcal{G}_t) = (1 - \eta) D_i^{\text{exp}}(\mathbf{u}) + \eta D_i^{\text{med}}(\mathbf{u}), \tag{A.4}$$

$$\widehat{N}_i(\mathbf{u}; \mathcal{G}_t) = \sum_j w_{i,j}(\mathbf{u}) \boldsymbol{n}_{i,j}(\mathbf{u}), \tag{A.5}$$

$$\widehat{N}_i^{\text{d}}(\mathbf{u}; \mathcal{G}_t) = \text{sg}(A_i(\mathbf{u}))\, \boldsymbol{n}[\widehat{D}_i](\mathbf{u}), \tag{A.6}$$

where $\eta = 0.5$ during refinement, sg denotes stop-gradient, and $\boldsymbol{n}[\widehat{D}_i]$ computes world-space normals from surface depth. Both normal maps retain the renderer's opacity scaling in the losses. Real and generated cameras use the same differentiable renderer, with trainable scene parameters:

$$\Theta_t = \{(\boldsymbol{\mu}_j, \mathbf{q}_j, \mathbf{s}_j, \alpha_j, \mathbf{c}_j)\}_{j=1}^{M_t}. \tag{A.7}$$

A.1.2 CONSTRUCTION OF THE SHARED SCENE STATE

All state channels share the domain $\Omega \subset \mathbb{R}^3$. Generated records retain their source, VLP request, viewpoint, and arbitration decision. For real view i, let $\omega_{i,t}^{\text{real}} : \Omega \to [0, 1]$ denote its support in the current scene. The aggregate real support and its complementary hole mask are:

$$q_t^{\text{real}}(\mathbf{x}) = 1 - \prod_{i=1}^{N} \big(1 - \omega_{i,t}^{\text{real}}(\mathbf{x})\big), \qquad m_t^{\text{real}}(\mathbf{x}) = \mathbf{1}\big[q_t^{\text{real}}(\mathbf{x}) \geq \tau_{\text{real}}\big],$$
$$m_t^{\text{hole}}(\mathbf{x}) = 1 - m_t^{\text{real}}(\mathbf{x}). \tag{A.8}$$

The binary anchor-support test below makes any $\tau_{\text{real}} \in (0, 1]$ equivalent to taking the union of supported regions. We derive the visibility channel from the complete current field, $\text{Vis}_t = \Phi_{\text{vis}}(\mathcal{G}_t)$,

and the coverage channel combines real support with accepted generated support. Let $\omega^{\mathrm{acc}}_{s,k,t} : \Omega \to [0,1]$ denote the support attributed to accepted record (s,k) in the current state:

$$\mathrm{Cov}_t(\mathbf{x}) = 1 - \left[\prod_{i=1}^{N}\big(1-\omega^{\mathrm{real}}_{i,t}(\mathbf{x})\big)\prod_{(s,k)\in\mathcal{H}_t}\big(1-\omega^{\mathrm{acc}}_{s,k,t}(\mathbf{x})\big)\right]. \tag{A.9}$$

Here, $\mathcal{H}_t$ indexes records accepted before stage t:

$$\mathcal{H}_t = \{(s,k) \mid 0 \le s < t,\ z_{s,k} = \texttt{accepted}\}. \tag{A.10}$$

By contrast, $\mathcal{A}_t$ denotes the planner's candidate pool. We define confidence as a state-derived field:

$$\mathrm{Conf}_t = \Phi_{\mathrm{conf}}(\mathrm{Vis}_t, \mathrm{Cov}_t, \{(e_{s,k}, z_{s,k})\}_{s<t}) : \Omega \to [0,1]. \tag{A.11}$$

The evidence ledger Evid_t is append-only:

$$\mathrm{Evid}_{t+1} = \mathrm{Evid}_t \cup \{(e_{t,k}, z_{t,k})\}_k, \qquad \mathrm{Evid}_0 = \varnothing. \tag{A.12}$$

For real support in Equation A.8, we infer depth and confidence priors from the real RGB inputs and freeze them as anchors before the first generation stage. For a point projected into a real camera with positive depth z, we sample the anchor depth d at the nearest pixel. A reliable anchor supports the point when $|z-d| \le \epsilon_{\mathrm{d}} z$ and excludes it as an observed-free-space conflict when $d - z > \epsilon_{\mathrm{d}} z$. We implement $\omega^{\mathrm{real}}_{i,t}$ as this binary support test, so any reliable real view can establish support or veto generated geometry. The stored state separates persistent records from view-dependent render samples of these fields. Persistent facts carry a type, support status, source identifiers, creation and last-check epochs, and conflict links. The planner and arbiter receive a read-only summary of these records and the current render statistics. The scalar coverage entries in the record schema are bookkeeping fields; planning measures missing area from rendered masks. The field equations describe the shared-state representation, while the view-dependent masks and statistics provide its planning and validation cues. Admission adds a generated record to the supervision archive while retaining its generated status.

We realize the local confidence cue with binary image maps. Let $\mathcal{X}_t$ be the union of points backprojected from the current real and accepted generated depth priors, and let D^{prior}_v be the prior used to construct the map for camera v. For a point $\mathbf{x}$, write $z_v(\mathbf{x})$ for its camera-axis depth and $\mathbf{q}_v(\mathbf{x})$ for its projected integer pixel. The depth-consistent visibility test is:

$$\chi_v(\mathbf{x}) = \mathbf{1}\left[z_v(\mathbf{x}) > 0, \quad \mathbf{q}_v(\mathbf{x}) \text{ in bounds}, \quad \frac{|z_v(\mathbf{x}) - D^{\mathrm{prior}}_v(\mathbf{q}_v(\mathbf{x}))|}{z_v(\mathbf{x}) + 10^{-6}} < 0.1\right]. \tag{A.13}$$

For an accepted generated view g, we collect pixels reached by points visible in both g and at least one real input camera:

$$\begin{aligned} \mathcal{Z}_{t,g} &= \{\mathbf{q}_g(\mathbf{x}) : \mathbf{x} \in \mathcal{X}_t,\ \chi_g(\mathbf{x}) = 1,\ \max_{1\le i\le N}\chi_i(\mathbf{x}) = 1\}, \\ C^{\mathrm{bin}}_{t,g}(\mathbf{u}) &= \mathbf{1}[\mathbf{u} \notin \mathcal{Z}_{t,g}]. \end{aligned} \tag{A.14}$$

All maps start at one, and real-view maps retain this value. Generated-view maps set the pixels in $\mathcal{Z}_{t,g}$ to zero. These maps supply the sampled confidence cue in Equation A.11 and the local factor $\gamma_{t,k}$ in Equation 15; they are support indicators, not calibrated correctness probabilities. The final geometric mask additionally applies the frozen real-anchor support and free-space tests, so a unit confidence value alone cannot authorize an update.

A.1.3 Exploration gain and candidate selection

For a candidate view v, let $M_{t,v}(\mathbf{u}) \in \{0,1\}$ be the rendered conditioning-support mask on its $H \times W$ image. In stages I–II, this mask thresholds the current training-view visibility grid at 0.5 and the rendered opacity at 0.99; the grid includes real inputs and previously admitted views. Stage III instead tests visibility in the fixed real input cameras. This planning mask is distinct from the frozen real-anchor test that restricts generated geometry during optimization. For the four cells $\{\Omega_{r,s}\}$ of a 2×2 image grid, we compute:

$$\begin{aligned} \rho_{\mathrm{stable}}(v) &= \frac{1}{HW}\sum_{\mathbf{u}} M_{t,v}(\mathbf{u}), & \rho_{\mathrm{miss}}(v) &= 1 - \rho_{\mathrm{stable}}(v), \\ n_{\mathrm{occ}}(v) &= \sum_{r,s}\mathbf{1}\left[\sum_{\mathbf{u}\in\Omega_{r,s}} M_{t,v}(\mathbf{u}) > 0\right]. \end{aligned} \tag{A.15}$$

A candidate is admissible when $\rho_{\text{stable}} \geq 0.20$, $n_{\text{occ}} \geq 3$, and $0.02 \leq \rho_{\text{miss}} \leq 0.80$. The gain in Equation 8 is the deterministic ranking key computed from these masks and the stored viewing-sector identifier $\sigma(v)$, rather than a learned quality score. Starting with $\mathcal{C}_t = \varnothing$, the selector prioritizes unrepresented sectors, then larger missing-area rates, and finally smaller candidate indices. After each selection it adds $\sigma(v)$ to $\mathcal{C}_t$ and re-ranks the remaining candidates.

The planner explores 15 virtual viewpoints at each refinement stage. Its selected queries and their VLP requests are:

$$\mathcal{Q}_t = \operatorname{Top}_{15}^{\text{lex}} \{v \in \mathcal{A}_t : \operatorname{Gain}_t(v)\}, \qquad \mathcal{P}_t = \{\Pi_{\text{VLP}}(S_t, q) : q \in \mathcal{Q}_t\}. \tag{A.16}$$

A.1.4 VLP MESSAGE SCHEMA

Let $\mathcal{V} = \mathrm{SE}(3)$ denote the space of target poses. Within each stage, each pose $v = \mathbf{T}_v$ is paired with intrinsic matrix $\mathbf{K}_v$ and pixel lattice $\mathcal{U}_v$ at the specified image resolution. The stage-specific field of view fixes the target camera's focal lengths. Projection therefore uses both intrinsics and extrinsics: $\pi_v(\mathbf{x}) = \pi(\mathbf{K}_v(\mathbf{R}_v\mathbf{x} + \mathbf{t}_v))$. Let $\mathcal{R}_v \subseteq 2^{\mathcal{U}_v} \setminus \{\varnothing\}$ denote the admissible nonempty image-plane target regions for camera v. The available output channels are:

$$\mathcal{O} = \{\text{rgb}, \text{depth}, \text{normal}\}. \tag{A.17}$$

The VLP program space is:

$$\mathcal{P} = \{(r, v, \mathcal{O}') \mid v \in \mathcal{V},\ r \in \mathcal{R}_v,\ \varnothing \neq \mathcal{O}' \subseteq \mathcal{O}\}. \tag{A.18}$$

The Qwen3-VL-8B-Instruct module instantiates the typed constructor:

$$\Pi_{\text{VLP}} : \mathcal{S} \times \mathcal{Q} \longrightarrow \mathcal{P}, \qquad p_{t,k} = \Pi_{\text{VLP}}(S_t, q_{t,k}) = (r_{t,k}, v_{t,k}, \mathcal{O}_{t,k}). \tag{A.19}$$

The core spatial and channel validity condition is:

$$\text{Valid}_{\text{core}}(p_{t,k}) \iff v_{t,k} \in \mathcal{V}_{\text{cand}}(S_t), \quad r_{t,k} \in \mathcal{R}_{v_{t,k}}, \quad \varnothing \neq \mathcal{O}_{t,k} \subseteq \mathcal{O}. \tag{A.20}$$

Full message validity also requires the identifiers, provenance, ordered channels, and textual fields to pass the schema checks below. For the pipeline shown in Figure 1, $\mathcal{O}_{t,k} = \mathcal{O}$. Qwen3-VL-8B-Instruct receives the registered real RGB images, the current RGB and depth renderings at the selected viewpoint, the current coverage mask, and the previous stages' admission decisions. It describes the missing region and produces a concise completion instruction in a typed JSON response. The selected viewpoint, region identifier, completion mask, and requested channels remain fixed. We validate these fields and reject malformed responses before generation. Each message uses schema version 2 and carries a stage identifier, message and query identifiers, and a nonempty list of source-record identifiers. The pose contains finite 3×3 rotation and three-dimensional translation arrays. We check rotation orthogonality and unit determinant with absolute tolerance 10^{-4} and reject duplicated or unknown output channels. The model must return the supplied region and viewpoint identifiers and the ordered list `[rgb, depth, normal]`, together with nonempty region-description, instruction, and reason strings. The spatial target is stored as a binary image mask at the fixed camera. For the serialized coverage image $C_{t,v}^{\text{img}}$ with values in $\{0, \ldots, 255\}$, the requested pixels are:

$$M_{t,v}^{\text{target}}(\mathbf{u}) = \mathbf{1}[C_{t,v}^{\text{img}}(\mathbf{u}) < 230]. \tag{A.21}$$

The corresponding target is $r_{t,k} = \{\mathbf{u} \in \mathcal{U}_{v_{t,k}} : M_{t,v_{t,k}}^{\text{target}}(\mathbf{u}) = 1\}$. An empty target invalidates the request. The region description annotates this mask without changing its pixels. Planner and arbiter contexts retain the compact state summary, decision counts by stage, and at most six recent records within an 8,000-character structured-context limit.

A.1.5 GENERATOR OUTPUT AND PREDICTION PROVENANCE

For generated maps of resolution $H \times W$, we define the output spaces:

$$\mathcal{I} = [0,1]^{H \times W \times 3}, \qquad \mathcal{D}_{\text{depth}} = (\mathbb{R}_{>0} \cup \{\perp\})^{H \times W}, \qquad \mathcal{N} = (\mathbb{S}^2 \cup \{\perp\})^{H \times W}, \tag{A.22}$$

and let $\mathcal{Y} = \mathcal{I} \times \mathcal{D}_{\text{depth}} \times \mathcal{N}$, $\mathcal{E} = \mathcal{Y} \times \mathcal{V} \times \mathcal{P}$. Here, $\perp$ denotes an invalid entry, including a nonfinite or nonpositive depth. Geometric losses exclude invalid entries and use only pixels where

Table 2: Arbiter decision rules used in the final experiments. Structural conflicts and severe visual degradation trigger rejection; diagnostic findings, unresolved assessments, and execution errors have distinct outcomes.

Check	Condition	Outcome
Structure	A valid response reports scene inconsistency, a nonempty unexpected-object list, or an unexpected object, duplicated structure, implausible layout, or geometry break.	Reject the candidate.
Diagnostics	A response reports only missing surfaces, appearance seams (boundary discontinuity), or appearance mismatch with references.	Record the findings; these alone do not trigger rejection.
Visual quality	Severe defocus, broad smearing, melted shapes, or ghosting makes a substantial structured region indiscernible.	Reject for severe generation collapse. Mild softness, low texture, or small artifacts alone do not qualify.
Response format	The structural reply cannot be parsed or contains missing, ill-typed, or conflicting fields.	Retry once; if still invalid, retain an unresolved assessment with a protocol-failure status.
Image citations	Optional image citations in a defect report are malformed or refer to image IDs outside the supplied set.	Retry once; if still invalid, retain an unresolved assessment marked for review.
Quality response	The separate quality reply cannot be parsed or lacks a valid usability label and nonempty region and reason.	Retry once; if still invalid, record a quality-assessment failure and retain the preceding admission decision.
Execution	Required inputs are missing, corrupt, or changed; the request or cached assessment is stale; context construction or file access fails.	Stop arbitration with an execution error; no admission fallback applies.

the quantities required by the corresponding residual are valid. We use the following per-request notation for the generator's output:

$$\text{Generator}_\psi : \mathcal{S} \times \mathcal{P} \longrightarrow \mathcal{E}, \qquad e_{t,k} = \big((\widetilde{I}_{t,k}, \widetilde{D}_{t,k}, \widetilde{N}_{t,k}), v_{t,k}, p_{t,k}\big). \tag{A.23}$$

See3D synthesizes the selected RGB views jointly, conditioned on their masked renderings and the real reference images. Qwen3-VL-8B-Instruct summarizes the validated instructions into one short text condition for the batch. Depth and normal estimation then supply the geometric channels. $\text{Generator}_\psi(S_t, p_{t,k})$ indexes the output for request $p_{t,k}$ within this joint batch. The retained $(v_{t,k}, p_{t,k})$ and per-request masks preserve the requested region, viewpoint, and channels for per-record arbitration.

A.1.6 Arbiter and prediction admission

The arbiter assigns a binary decision to each candidate record using the current state, as shown in Figure 2:

$$\text{Arbiter} : \mathcal{E} \times \mathcal{S} \longrightarrow \{\texttt{accepted}, \texttt{rejected}\}, \qquad z_{t,k} = \text{Arbiter}(e_{t,k}, S_t). \tag{A.24}$$

The Qwen3-VL-8B-Instruct arbiter receives the real RGB observations, the current target-view RGB and depth renderings, the coverage and requested-region masks, and the candidate RGB, depth, and normal maps. We include the shared-state summary and previous decisions as structured context. Depth previews carry their numerical ranges, and normal previews encode vector components as color. The model checks cross-channel agreement and scene consistency, returning structured consistency fields, unexpected-object reports, and defect predicates. A fixed admission rule derives the binary decision from these fields. A separate call inspects the candidate RGB image for severe visual degradation, without the planner text or geometric channels. Either check can trigger rejection under Table 2. The structural check accounts for viewpoint and occlusion; unobserved regions and incomplete targets alone do not establish a structural conflict.

Each check permits at most two attempts on the same visual inputs. The retry requests a valid response without prescribing an admission outcome; valid rejections are not retried. The final failed

attempt determines the unresolved status. If the structural assessment remains unresolved, we mark it invalid, leave scene consistency unset, and provisionally admit the candidate unless the visual-quality check rejects it. Thus, absent a valid rejection from either check, a complete candidate remains eligible for masked supervision. The ledger stores the binary decision, assessment validity, failure category, and retry history separately; an unresolved assessment is distinct from a valid acceptance.

Let $m_t^{\text{free}} : \Omega \to \{0,1\}$ denote the free-space admissibility field and let $\gamma_{t,k} : \Omega \to [0,1]$ denote the local confidence of generated geometry. These fields, together with the acceptance decision and real-support exclusion, define the mask in Equation 15. After depth alignment, we evaluate this mask at points unprojected from the final generated depth prior $D_v^\star$. Invalid or nonpositive depths receive zero weight in geometric supervision. The local confidence cue is a binary geometric-support indicator; a uniform unit cue leaves the support and free-space tests as the active constraints.

The appearance target already lies in the image plane, so its mask is:

$$m_{t,k}^{\text{app}}(\mathbf{u}) = \mathbf{1}[\mathbf{u} \in r_{t,k}]\,. \tag{A.25}$$

The masks satisfy the following constraints:

$$m_{t,k}^{\text{geo}}(\mathbf{x}) m_t^{\text{real}}(\mathbf{x}) = 0, \qquad z_{t,k} = \texttt{rejected} \Longrightarrow m_{t,k}^{\text{geo}} \equiv 0. \tag{A.26}$$

Record-level admission determines which generated cameras enter training; the masks restrict their supervision to channel-specific admissible regions. We define the aligned depth targets in **Appendix A.1.7**.

A.1.7 Generated-depth alignment at every refinement stage

We align generated depths to the frozen real-input geometry before each stage's gaussian reinitialization. At each stage $t \in \{0,1,2\}$, we fit a separate inverse-depth affine transform for every accumulated accepted view, including earlier-stage views. Each fit starts from that stage's uncorrected prior; corrections from previous stages are not composed. The real-input depth and confidence maps remain fixed from before the first generation stage, and the initial real-only optimization requires no such alignment.

Let D_g be a generated view's uncorrected depth and $(D_a^{\text{R}}, C_a^{\text{R}})$ a frozen real-view anchor. All depths measure distance along the camera's optical axis. Using the camera conventions of Section 3.1, we define backprojection, camera-axis depth, and projection as follows:

$$\begin{aligned} \mathcal{B}_j(\mathbf{u}, d) &= \mathbf{R}_j^{\mathsf{T}}\big(d\mathbf{K}_j^{-1}\bar{\mathbf{u}} - \mathbf{t}_j\big), \quad z_j(\mathbf{x}) = [\mathbf{R}_j\mathbf{x} + \mathbf{t}_j]_3, \\ \pi_j(\mathbf{x}) &= \pi(\mathbf{K}_j(\mathbf{R}_j\mathbf{x} + \mathbf{t}_j))\,, \qquad \bar{\mathbf{u}} = (u_1, u_2, 1)^{\mathsf{T}}. \end{aligned} \tag{A.27}$$

Here, π denotes perspective division. For generated pixel $\mathbf{u}$, we project $\mathbf{x} = \mathcal{B}_g(\mathbf{u}, D_g(\mathbf{u}))$ into anchor a and sample its depth and confidence at the nearest integer pixel $\mathbf{q}_a$. We backproject this sample as $\mathbf{x}_a = \mathcal{B}_a(\mathbf{q}_a, D_a^{\text{R}}(\mathbf{q}_a))$. Both projections must be finite, in bounds, and have positive camera-axis depth. Correspondences must satisfy:

$$\begin{aligned} C_a^{\text{R}}(\mathbf{q}_a) &\geq \tau_{\text{a}}, \\ \big|z_a(\mathbf{x}) - D_a^{\text{R}}(\mathbf{q}_a)\big| &\leq \epsilon_{\text{d}} z_a(\mathbf{x}), \\ e_{\mathbf{u},a} := \|\pi_g(\mathbf{x}_a) - \mathbf{u}\|_2 &\leq \tau_{\text{px}}. \end{aligned} \tag{A.28}$$

We exclude borders and depth discontinuities in both maps: the center and its four axial neighbors must have finite positive depths, with each neighbor differing from the center by at most ϵ_{edge} times the center depth. Among qualifying anchors, we choose the smallest $e_{\mathbf{u},a}$, breaking ties by fixed real-view order. Its depth in the generated camera, $T_g(\mathbf{u}) = z_g(\mathbf{x}_a)$, supplies one target per pixel. The resulting support Ω_g must meet the sample-count and spatial-coverage requirements in Table 3.

For $\mathbf{u} \in \Omega_g$, set $x_{\mathbf{u}} = 1/D_g(\mathbf{u})$ and $y_{\mathbf{u}} = 1/T_g(\mathbf{u})$. We normalize the predictor using $m_x = \text{median}(x)$ and the population standard deviation $s_x = \text{std}(x)$, rejecting fits at or below the spread threshold in Table 3. With design rows $\mathbf{A}_{\mathbf{u}} = [1, (x_{\mathbf{u}} - m_x)/s_x]$, we fit the affine relation using Huber iteratively reweighted least squares (Huber, 1964). Starting from unit weights, each iteration solves the two-parameter system:

$$(\mathbf{A}^{\mathsf{T}}\mathbf{W}^{(k)}\mathbf{A})\boldsymbol{\theta}^{(k)} = \mathbf{A}^{\mathsf{T}}\mathbf{W}^{(k)}\mathbf{y}, \qquad \mathbf{W}^{(k)} = \text{diag}(w^{(k)}). \tag{A.29}$$

We solve directly in double precision and reject nonfinite or ill-conditioned systems. For residuals $r_{\mathbf{u}}^{(k)} = y_{\mathbf{u}} - \mathbf{A}_{\mathbf{u}}\boldsymbol{\theta}^{(k)}$, the median absolute deviation determines the scale and subsequent weights:

$$\begin{aligned} \sigma^{(k)} &= \max\Big(1.4826\,\mathrm{median}_{\mathbf{u}}\,|r_{\mathbf{u}}^{(k)} - \mathrm{median}(r^{(k)})|, 10^{-6}|\,\mathrm{median}(y)|\Big)\,, \\ w_{\mathbf{u}}^{(k+1)} &= \min\left(1, \frac{1.345\sigma^{(k)}}{\max(|r_{\mathbf{u}}^{(k)}|, 10^{-12})}\right). \end{aligned} \tag{A.30}$$

Correspondences stay fixed during the eight solves; anchor confidence filters correspondences without adding a continuous regression weight. From the final coefficients $\boldsymbol{\theta}^{(7)}$, we recover the transform and apply it to all finite positive input depths:

$$a_g = \frac{\theta_1^{(7)}}{s_x}, \qquad b_g = \theta_0^{(7)} - a_g m_x, \qquad D_g^{\mathrm{cand}}(\mathbf{u}) = \frac{1}{a_g/D_g(\mathbf{u}) + b_g}. \tag{A.31}$$

We accept the correction only if $a_g > 0$ and the transformed inverse depth is finite and positive at every valid input pixel. Let $\mathcal{V}_g$ be this full set of valid pixels. The correction must also satisfy a global change bound and preserve or reduce the median support residual:

$$\begin{aligned} &\max_{\mathbf{u}\in\mathcal{V}_g}\left|\frac{D_g^{\mathrm{cand}}(\mathbf{u})}{D_g(\mathbf{u})} - 1\right| \le \epsilon_{\mathrm{chg}}, \\ &\mathrm{median}_{\mathbf{u}\in\Omega_g}\,\frac{|D_g^{\mathrm{cand}}(\mathbf{u}) - T_g(\mathbf{u})|}{T_g(\mathbf{u})} \le \mathrm{median}_{\mathbf{u}\in\Omega_g}\,\frac{|D_g(\mathbf{u}) - T_g(\mathbf{u})|}{T_g(\mathbf{u})} + 10^{-8}. \end{aligned} \tag{A.32}$$

The final target $D_g^\star$ equals D_g^{cand} if all checks pass; otherwise, we retain the entire original map D_g. Invalid input pixels remain unchanged. This fallback preserves the view's admission decision. The final depths feed gaussian initialization, depth-derived supervision, and multiview correspondences. They retain their generated provenance: overlap with real anchors supports calibration, while the geometric mask still restricts generated updates to admissible regions without real support.

Table 3: **Generated-depth alignment parameters.** The same settings apply at all three refinement stages; the change bound covers every valid input pixel.

Parameter	Value
Anchor confidence τ_{a}	0.5
Relative-depth tolerance ϵ_{d}	0.10
Round-trip tolerance τ_{px}	1.5 pixels
Depth-edge tolerance ϵ_{edge}	0.05
Minimum support	512 pixels in at least 4 cells of a 4×4 grid
Predictor-spread rejection threshold	$\max(10^{-4}\lvert m_x\rvert, 10^{-12})$
IRLS solves / Huber threshold	8 / 1.345σ
MAD scale factor / minimum scale	1.4826 / $10^{-6}\lvert\,\mathrm{median}(y)\rvert$
Weight-denominator floor	10^{-12}
Maximum normal-matrix condition number	10^{10}
Maximum relative change ϵ_{chg}	0.10
Median-residual tolerance	10^{-8}

A.1.8 Authority-aware Update objective

ReconAgent builds on MAtCha's geometric priors (Guédon et al., 2025) for gaussian refinement within a 2DGS backend (Huang et al., 2024), adding prediction admission and spatially masked supervision. At stage t, the training archive combines real input cameras with the newly accepted set $\mathcal{E}_t^+$ and previously accepted records in S_t. We align all generated depth priors to the frozen real anchors, then initialize the stage's gaussian parameters and optimizer state afresh. At iteration i of stage t, we sample a camera v_i without replacement from this archive and replenish the sampling pool after visiting every camera. Each retained camera has an RGB target $I_v^\star$ and a depth target $D_v^\star$: a frozen inferred prior for a real view or the final aligned target for a generated view. The

corresponding normal prior $N_v^\star$ is derived from this depth, and the curvature prior is $C_v^\star = \mathcal{K}(N_v^\star)$. For a residual f and binary mask m, we use the masked average:

$$\langle f \rangle_m = \frac{\sum_{\mathbf{u},c} m(\mathbf{u}) f_c(\mathbf{u})}{\max\Big(1, \sum_{\mathbf{u},c} m(\mathbf{u})\Big)}. \tag{A.33}$$

The channel index c is omitted for scalar residuals. An empty mask contributes zero loss. Real cameras use unit masks. For an accepted generated camera, m_v^{app} is the requested-region mask and m_v^{geo} is Equation 15 evaluated at its unprojected depth-prior points. We write $E_r(m)$ for erosion by a square neighborhood of radius r, which retains a residual only when its entire stencil lies inside the admissible domain. We erode generated masks and retain unit masks for real cameras, including their boundary pixels.

The refinement objective combines photometric fitting with four geometric prior terms, normal consistency, depth ordering, and gaussian anisotropy:

$$\begin{aligned}\mathcal{L}_{t,i}(\Theta; v_i) = a_{v_i}\mathcal{L}_{\mathrm{rgb}} \\ + r(i)\left[0.375\mathcal{L}_{\mathrm{D}} + 0.5\mathcal{L}_{\mathrm{DN}} + 0.5\mathcal{L}_{\mathrm{N}} + 0.25\mathcal{L}_{\mathrm{C}}\right] \\ + 0.05\,\mathbf{1}[i > 3500]\mathcal{L}_{\mathrm{NC}} + w_{\mathrm{ord}}(i)\mathcal{L}_{\mathrm{ord}} + 0.1\mathcal{L}_{\mathrm{ani}}.\end{aligned} \tag{A.34}$$

We evaluate all image terms at (Θ, v_i) and apply the anisotropy term to the current gaussian scales. The factors $r(i)$ and $w_{\mathrm{ord}}(i)$ schedule the prior and depth-order terms, respectively. The distortion coefficient is zero.

Real and generated cameras use the same RGB residual, with different photometric multipliers. The residual combines L_1 with structural similarity (Wang et al., 2004):

$$\begin{aligned}\mathcal{L}_{\mathrm{rgb}} = 0.8\left\langle |\widehat{I}_v - I_v^\star| \right\rangle_{m_v^{\mathrm{app}}} \\ + 0.2\left\langle 1 - \mathrm{SSIM}_{\mathbf{u}}(\widehat{I}_v, I_v^\star) \right\rangle_{E_5(m_v^{\mathrm{app}})},\end{aligned} \tag{A.35}$$

$$a_v = \begin{cases} 1, & v \text{ is a real input camera,} \\ 0.01, & v \text{ is an accepted generated camera.} \end{cases}$$

The multiplier a_v prioritizes real observations. For generated views, the appearance mask limits L_1 supervision to the requested pixels and SSIM supervision to 11×11 windows contained entirely within the requested region. Geometric terms use their own admissibility mask.

Let $\widehat{D}_v$ be the rendered surface depth, $\widehat{N}_v$ the rendered normal map, and $\widehat{N}_v^{\mathrm{d}}$ the depth-derived surface normal map. We define the depth-fitting loss, the two normal-prior alignment losses, and the curvature-fitting loss as follows:

$$\mathcal{L}_{\mathrm{D}} = \left\langle \log\left(1 + |\widehat{D}_v - D_v^\star|\right) \right\rangle_{m_v^{\mathrm{geo}}}, \tag{A.36}$$

$$\mathcal{L}_{\mathrm{DN}} = \left\langle 1 - \widehat{N}_v^{\mathrm{d}} \cdot N_v^\star \right\rangle_{E_1(m_v^{\mathrm{geo}})}, \qquad \mathcal{L}_{\mathrm{N}} = \left\langle 1 - \widehat{N}_v \cdot N_v^\star \right\rangle_{E_1(m_v^{\mathrm{geo}})}, \tag{A.37}$$

$$\mathcal{L}_{\mathrm{C}} = \left\langle |\mathcal{K}(\widehat{N}_v) - C_v^\star| \right\rangle_{E_2(m_v^{\mathrm{geo}})}. \tag{A.38}$$

The operator $\mathcal{K}$ sums the four-neighbor normal differences and takes an L_1 norm over normal components, with replicated values at image boundaries. The coefficient $0.375r(i)$ combines the depth-prior multiplier $0.75r(i)$ with the fixed factor 0.5 in the logarithmic depth residual.

The normal-consistency term aligns the two rendered normal maps:

$$\mathcal{L}_{\mathrm{NC}} = \left\langle 1 - \widehat{N}_v \cdot \widehat{N}_v^{\mathrm{d}} \right\rangle_{E_1(m_v^{\mathrm{geo}})}. \tag{A.39}$$

For depth ordering, we pair each pixel $\mathbf{u}$ with a randomly shifted pixel $\mathbf{u}'$ whose coordinates are clipped to the image boundary. Let ℓ_{scene} be the scene extent used by the gaussian backend. We define the normalized depth differences and prior ordering as follows:

$$\delta = \frac{\widehat{D}_v(\mathbf{u}) - \widehat{D}_v(\mathbf{u}')}{\ell_{\mathrm{scene}}}, \qquad \delta^\star = \frac{D_v^\star(\mathbf{u}) - D_v^\star(\mathbf{u}')}{\ell_{\mathrm{scene}}}, \qquad \xi^\star = \frac{\delta^\star}{\max(|\delta^\star|, 10^{-8})}. \tag{A.40}$$

The loss penalizes disagreement with the prior depth ordering:

$$\mathcal{L}_{\mathrm{ord}} = \langle \log\left(1 + 20\max(-\delta\xi^{\star}, 0)\right)\rangle_{m_v^{\mathrm{ord}}}. \tag{A.41}$$

Here, $m_v^{\mathrm{ord}}(\mathbf{u}) = m_v^{\mathrm{geo}}(\mathbf{u})m_v^{\mathrm{geo}}(\mathbf{u}')$ requires both endpoints to be admissible. The maximum integer displacement is $\mathrm{round}(0.05\max(H, W))$ along each image axis.

The final regularizer controls gaussian shape. Let $\mathcal{J}_v$ contain all gaussians on a real-view step. On a generated-view step, it contains gaussians whose centers pass the frozen real-anchor exclusion tests, project into m_v^{geo}, and agree with the generated depth prior within the relative tolerance ϵ_{d}. The anisotropy penalty is:

$$\mathcal{L}_{\mathrm{ani}} = \frac{1}{\max(1, |\mathcal{J}_v|)} \sum_{j\in\mathcal{J}_v} \max\left(\frac{s_{j,\max}}{s_{j,\min}} - 5, 0\right), \tag{A.42}$$

where $s_{j,\max}$ and $s_{j,\min}$ are the larger and smaller tangent-plane scales of gaussian j. Generated-view steps update only the gaussians in $\mathcal{J}_v$: protected parameter rows, their first and second moments, and their adam clocks remain unchanged. We reject a proposed row update if its new center fails the same admissibility tests. This restriction applies jointly to position, rotation, scale, opacity, and appearance parameters, including gradients from RGB supervision. Real-view steps retain full-scene supervision and control densification, pruning, and opacity resets. Each refinement call runs 10,000 iterations with row-wise adam (Kingma & Ba, 2015):

$$\begin{aligned} \Theta^{(i)} &= \mathrm{Adam}_{\mathcal{J}_{v_i}}\left(\Theta^{(i-1)}, \nabla_{\Theta}\mathcal{L}_{t,i}(\Theta; v_i)|_{\Theta=\Theta^{(i-1)}}\right), \\ \mathcal{G}_{t+1} &= \mathcal{G}(\Theta^{(10000)}). \end{aligned} \tag{A.43}$$

The constraint preserves protected Gaussian parameters during generated-view steps; composited pixel values also depend on the footprints and visibility of other Gaussians. We recompute the shared scene state from the refined field for the next stage. We detail rendering in **Appendix A.1.1**, depth-target construction in **Appendix A.1.7**, and loss schedules in **Appendix A.2.3**.

A.1.9 Multi-stage state transition

The stage index set is $\mathcal{T} = \{0, 1, 2\}$, corresponding to coarse, occluded, and detailed refinement. We identify $\mathcal{G}_0 = \mathcal{G}^{(0)}$ and initialize $S_0 = \mathrm{InitState}(\mathcal{G}^{(0)}, \mathcal{D}_{\mathrm{real}})$ and execute the following operations at stage $t \in \mathcal{T}$:

$$\begin{aligned} \mathcal{Q}_t &= \mathrm{Planner}(S_t), & p_{t,k} &= \Pi_{\mathrm{VLP}}(S_t, q_{t,k}), \\ e_{t,k} &= \mathrm{Generator}_{\psi}(S_t, p_{t,k}), & z_{t,k} &= \mathrm{Arbiter}(e_{t,k}, S_t), \\ \mathcal{E}_t^{+} &= \{e_{t,k} : z_{t,k} = \texttt{accepted}\}, & \mathcal{U}_t(\mathcal{G}_t, S_t) &:= \mathcal{U}\big(\mathcal{G}_t, S_t, \mathcal{D}_{\mathrm{real}}, \mathcal{E}_t^{+}\big), \\ & & (\mathcal{G}_{t+1}, S_{t+1}) &= \mathcal{U}_t(\mathcal{G}_t, S_t). \end{aligned} \tag{A.44}$$

Here, $\mathcal{U}_t$ specializes Equation 16 to stage t. The ledger retains all decisions through Equation A.12, while Update adds $\mathcal{E}_t^{+}$ to the accepted archive in S_t. The resulting geometry and unresolved targets guide acquisition at stage $t + 1$. The three stages differ in candidate construction and conditioning support. The coarse stage adds cameras directed toward the scene center and surrounding regions; the occluded stage adds angular sweeps around the input cameras. The detailed stage uses surface-directed candidates with a wider field of view and conditioning support restricted to the real inputs. The vertical field of view is 80° in stages I–II and 100° in stage III. All stages apply the same gain ordering, VLP schema, and admission policy to their refreshed candidates.

At each stage t, Update reinitializes the Gaussian field from the fixed real inputs and accumulated accepted priors after depth alignment. It resets optimizer moments and uses a stage-local iteration schedule. Thus, $\mathcal{G}_t$ supplies the planning renderings, while the supervision archive and decision history carry information into the new initialization. The implementation checks the stage decision, ledger, training-record index, and frozen real anchors before optimization and verifies that these records remain unchanged when the stage completes.

A.1.10 Overall reconstruction algorithm

Algorithm 1 consolidates these operations, with joint generation followed by per-record arbitration and scene refinement. Within $\mathcal{U}$, we align accumulated accepted depths to the frozen real anchors, reinitialize the gaussians, and optimize with spatially restricted generated supervision.

Algorithm 1 Sparse-view indoor surface reconstruction with ReconAgent

Require: Sparse posed real views $\mathcal{D}_{\text{real}}$
$\mathcal{G}^{(0)} \leftarrow \text{DenseInitAndTrain}(\mathcal{D}_{\text{real}})$
$S_0 \leftarrow \text{InitState}(\mathcal{G}^{(0)}, \mathcal{D}_{\text{real}})$
for $t = 0, 1, 2$ (coarse, occluded, and detailed stages, respectively) **do**
  $\mathcal{Q}_t \leftarrow \text{Planner}(S_t)$
  $\mathcal{P}_t \leftarrow \{\Pi_{\text{VLP}}(S_t, q) : q \in \mathcal{Q}_t\}$
  Generate $\mathcal{E}_t$ jointly from $(S_t, \mathcal{P}_t)$
  for each record $e_{t,k}$ in $\mathcal{E}_t$ **do**
    $z_{t,k} \leftarrow \text{Arbiter}(e_{t,k}, S_t)$
  end for
  $\mathcal{E}_t^+ \leftarrow \{e_{t,k} : z_{t,k} = \texttt{accepted}\}$
  $(\mathcal{G}_{t+1}, S_{t+1}) \leftarrow \mathcal{U}(\mathcal{G}_t, S_t, \mathcal{D}_{\text{real}}, \mathcal{E}_t^+)$
end for
return final gaussian scene $\mathcal{G}_{\text{final}} = \mathcal{G}_3$

A.2 Experimental Details

We build ReconAgent on MAtCha (Guédon et al., 2025), which serves as our primary reconstruction baseline. Our additional comparisons include 3DGS (Kerbl et al., 2023), 2DGS (Huang et al., 2024), FSGS (Zhu et al., 2024), InstantSplat (Fan et al., 2025), and G4Splat (Ni et al., 2026).

A.2.1 Data preparation and view selection

We evaluate on eight Replica scenes (Straub et al., 2019) and six ScanNet++ scenes (Yeshwanth et al., 2023). For each scene, we uniformly sample 100 frames and use five posed frames as reconstruction inputs. The remaining frames and reference geometry define the evaluation set. Generated views provide supplemental training supervision, while real test views define the evaluation targets.

A.2.2 Training configuration

We run ReconAgent and all controlled variants on an NVIDIA H20 GPU. We use Qwen3-VL-8B-Instruct (Bai et al., 2025) as the foundation model for the multi-agent framework and See3D (Ma et al., 2025) for RGB generation. The planner explores 15 virtual viewpoints per refinement stage. The MAtCha-based reconstruction pipeline (Guédon et al., 2025) begins with posed MASt3R-SfM (Duisterhof et al., 2025), chart alignment, and initial gaussian optimization. ReconAgent then applies three agent-guided refinement stages before mesh extraction and evaluation. We optimize the 2D gaussian field for 7,000 iterations at initialization and 10,000 iterations in each subsequent refinement stage. Mesh extraction combines gaussian-adaptive tetrahedralization (Yu et al., 2024b) with MAtCha's adaptive TSDF fusion (Guédon et al., 2025). Table 4 summarizes the input and agent settings, and Table 5 gives the optimization and mesh-extraction parameters. Component ablations retain the same stage schedule, five-view split, initial field, rendering resolution, gaussian backend, optimizer, evaluator, and authority order while removing the designated module.

In the no-planner ablation, the random selector samples without replacement from the same admissible candidate pool as the full planner. It matches the corresponding full-system run's candidate count at stage t, isolating the selection rule under a matched generation budget. The sampled candidates pass through the same VLP, generator, arbiter, and update stages.

Within each stage, See3D jointly completes RGB views before the geometric estimators produce depth and normal maps. The arbiter evaluates all three channels together with the VLP request. We align the accumulated accepted depth priors before each gaussian reinitialization and recompute the supervision masks from the final depths. Each refinement stage starts with fresh optimizer state; generated-view steps preserve protected rows within that stage. We detail candidate selection in **Appendix A.1.3** and depth alignment and its fixed parameters in **Appendix A.1.7**.

Table 4: **ReconAgent input setup and agent configuration.** The full system and controlled ablations use these settings unless an ablation removes the relevant component.

Component	Setting
Input setup	Five posed input views; 100 uniformly sampled frames per scene; three stages: coarse, occluded, and detailed.
Planner exploration	15 virtual viewpoints explored at each refinement stage. Candidate admissibility and ranking follow the renderer-side gain procedure.
Multi-agent foundation model	`Qwen/Qwen3-VL-8B-Instruct` weights; `bfloat16`; one request per call; images resized to at most 448 pixels per side while preserving aspect ratio; greedy decoding with at most 256 new tokens. We use the checkpoint's chat template for VLP construction and arbitration. Each request specifies RGB, depth, and normal channels.
Generator	See3D sparse multi-view diffusion generator; 512×512 RGB completion; random seed 12345; classifier-free guidance scale 2.0; 50 denoising steps; super-resolution disabled.
Geometry cues	Depth and normal maps are estimated after See3D RGB completion and before arbitration. The initial depth predictor is Depth Anything V2 (Yang et al., 2024) with the ViT-L encoder.
Spatial authority	Confidence cutoff $\tau_{\text{geo}} = 0.5$; relative depth tolerance $\epsilon_{\text{d}} = 0.1$ for real-anchor and generated-depth agreement. Anchors inferred from real input images remain fixed across stages.

Table 5: **Gaussian optimization, initialization, and mesh-extraction configuration.** Initial Gaussian optimization uses 7,000 iterations; each subsequent refinement call uses 10,000 iterations.

Component	Setting
Gaussian optimization	2D Gaussian Splatting; 7,000 iterations at initialization and 10,000 per refinement stage; densification through iteration 3,500; opacity reset interval 1,000; depth ratio 0.5; normal consistency after iteration 3,500; distortion disabled (weight 0); MIP filtering (Yu et al., 2024a) enabled.
Gaussian optimizer	Position learning rate $1.6 \times 10^{-4} \rightarrow 1.6 \times 10^{-6}$ with delay multiplier 0.01; feature, opacity, scaling, and rotation learning rates 2.5×10^{-3}, 5×10^{-2}, 5×10^{-3}, and 10^{-3}, respectively. Densification begins at iteration 500, occurs every 100 iterations, and uses a gradient threshold of 2×10^{-4}. The DSSIM and normal-loss weights are 0.2 and 0.05, respectively.
Initialization and alignment	Posed MASt3R uses 512-pixel images, 1,000 coarse and 1,000 refinement iterations, a maximum window size of 20, and a maximum reference ID of 10. Chart alignment uses 512-pixel point maps, a target scale of 5, and 1,000 iterations. Its learning rates for encoding, MLP, and confidence are $(10^{-2}, 10^{-3}, 10^{-3})$, and its normal, curvature, and matching loss weights are $(4, 1, 5)$.
Training options	Dense supervision, Gaussian downsampling, input-view color-loss downweighting, and post-hoc mesh filtering are disabled. Interpolated views remain enabled for mesh extraction.
Adaptive tetrahedralization	Gaussian flatness 2×10^{-4}; tetrahedral downsampling ratio 0.25; depth ratio 1.0; bilinear depth interpolation with truncation margin 0.005; 2 neighbors and 10 interpolated cameras per neighbor; mesh filtering enabled within the tetrahedral extractor.

A.2.3 Loss weights and schedules

Equation A.34 gives the gaussian refinement objective. Its photometric coefficient a_v is 1 for real views and 0.01 for generated views. The L_1 and $1 - \text{SSIM}$ terms have weights 0.8 and 0.2, respectively. Normal consistency has weight 0.05 for $i > 3500$ and 0 otherwise. The distortion coefficient is 0 throughout optimization. Here, i counts iterations within each 10,000-iteration refinement call.

The geometric priors share a decaying factor:

$$r(i) = \max\left(\frac{0.5}{2^{\lfloor i/1000 \rfloor}},\ 0.015\right). \tag{A.45}$$

The coefficients of $(\mathcal{L}_{\mathrm{D}}, \mathcal{L}_{\mathrm{DN}}, \mathcal{L}_{\mathrm{N}}, \mathcal{L}_{\mathrm{C}})$ are $(0.375, 0.5, 0.5, 0.25)r(i)$, including the fixed depth-residual factor 0.5. The initial effective coefficients are therefore $(0.1875, 0.25, 0.25, 0.125)$, and the final coefficients are $(0.005625, 0.0075, 0.0075, 0.00375)$. The two normal-prior losses use one minus the normal dot product, while the curvature loss uses an absolute residual.

Depth-order regularization uses the piecewise coefficient:

$$w_{\mathrm{ord}}(i) = \begin{cases} 0, & 1 \le i \le 1500, \\ 1, & 1500 < i \le 3000, \\ 0.1, & 3000 < i \le 4500, \\ 0.01, & 4500 < i \le 6000, \\ 0.001, & 6000 < i \le 10000. \end{cases} \tag{A.46}$$

The depth-order residual scales ordering violations by 20 inside the logarithm and uses a maximum pixel displacement of $\mathrm{round}(0.05 \max(H, W))$, as defined in Equation A.41. The coefficient of $\mathcal{L}_{\mathrm{ani}}$ is 0.1, and its scale-ratio penalty activates above 5. All these schedules restart for each refinement call. Before gaussian refinement, chart alignment uses normal, curvature, and matching loss weights $(4, 1, 5)$.

A.2.4 Evaluation metrics and qualitative analysis

We compute Chamfer-L_1 (CD) as the mean of the two directed nearest-neighbor distances, in centimeters (cm), and F-score from precision and recall at a 5 cm threshold, after voxel-downsampling mesh vertices at 2 cm resolution. Normal consistency (NC) averages absolute normal dot products between nearest-neighbor correspondences in both directions, using $200{,}000$ surface samples per mesh. We report F-score and NC as percentages. For novel-view synthesis, PSNR measures pixel-level fidelity, SSIM (Wang et al., 2004) measures local structural similarity, and LPIPS (Zhang et al., 2018) measures perceptual distance using VGG features. Lower CD and LPIPS and higher values of the other metrics indicate better results. Together, the six metrics in Table 1 and Figure 5 capture surface proximity, orientation, and rendered appearance.

The visual comparisons localize differences summarized by the aggregate metrics. Figure 3 shows surface continuity and local geometry around furniture, ceilings, and window frames, whereas Figure 4 shows appearance completeness and boundary artifacts. ReconAgent reduces prominent gaps and folded surfaces in these examples, while the ScanNet++ chair and window regions retain the discrepancies discussed in the main text. The stage-wise RGB, normal, and depth comparisons in Figure 11 further distinguish appearance completion from changes in geometric continuity.

A.3 Additional Experimental Results

A.3.1 Surface reconstruction with more input views

We extend the Replica comparison to seven and nine input views to examine whether ReconAgent retains its surface reconstruction advantage as more real observations become available. Table 6 compares ReconAgent with its MAtCha baseline (Guédon et al., 2025), with 2DGS (Huang et al., 2024) and G4Splat (Ni et al., 2026) as additional comparison methods. ReconAgent achieves the lowest CD in both settings: 5.18 with seven views and 4.86 with nine. Relative to its MAtCha baseline, these results reduce CD by 3.54 (40.6%) and 2.33 (32.4%), respectively. The reductions over G4Splat, the strongest competing method by CD in both settings, are 1.30 (20.1%) and 1.05 (17.8%).

Together with the five-view result in Table 1, ReconAgent's CD decreases from 6.35 to 5.18 to 4.86 across the three tested input counts. The seven- and nine-view settings reduce its error by 18.4% and 23.5% relative to five views. The additional reduction from seven to nine views is 0.32, compared with 1.17 from five to seven, indicating a smaller incremental benefit in the denser setting. MAtCha and G4Splat also benefit from more input views, narrowing the absolute gap to ReconAgent. The

persistent advantage supports the use of complementary view generation and selective admission alongside additional real observations.

A.3.2 Computational cost and rendering speed

Table 7 reports scene-averaged peak GPU memory, total runtime, and rendering speed on Replica with five input views. Runtime includes initialization and all three refinement stages, while memory is the mean of the per-scene peaks. Relative to its MAtCha backbone, ReconAgent's multi-agent framework adds model inference and coordination for viewpoint planning, generation, and arbitration, increasing end-to-end runtime and peak memory. These additional operations direct generated supervision toward unresolved regions and screen inconsistent predictions before refinement, trading higher reconstruction cost for substantially improved surface completeness and geometric consistency. Compared with G4Splat, ReconAgent uses 14 GB more peak memory and takes 3 additional minutes per scene, with quality gains concentrated in local surface continuity and appearance consistency. The additional computation supports explicit coordination between viewpoint selection and prediction admission throughout reconstruction. After refinement, novel-view synthesis uses the optimized Gaussian scene: ReconAgent matches G4Splat's 49 FPS, although its rendering throughput remains below MAtCha's 71 FPS.

Table 6: **Surface reconstruction on Replica with seven and nine input views.** CD is reported in centimeters. Underlined values identify MAtCha, the reconstruction backbone and baseline of ReconAgent.

Method	7 views CD↓	9 views CD↓
2DGS	10.84	9.49
G4Splat	6.48	5.91
MAtCha	8.72	7.19
ReconAgent	5.18	4.86

Table 7: **Computational cost on Replica with five input views.** Scene averages of peak memory, total runtime, and rendering speed. Underlined values identify MAtCha, ReconAgent's backbone and baseline.

Method	GPU memory (GB)↓	Runtime (min)↓	FPS↑
G4Splat	32	87	49
MAtCha	**28**	**22**	**71**
ReconAgent	46	90	49

A.3.3 Quantitative progression across refinement stages

Table 8 tracks surface reconstruction and novel-view synthesis quality for the full ReconAgent model on Replica with five input views. Stages I, II, and III correspond to coarse, occluded, and detailed refinement. From Stage I to Stage II, CD decreases from 6.55 to 6.42, F-score rises from 67.14 to 67.85, and NC rises from 81.45 to 82.23. Novel-view synthesis quality improves concurrently: PSNR increases by 0.27 dB, SSIM rises from 0.841 to 0.846, and LPIPS falls from 0.198 to 0.192.

Stage III brings the largest stage-to-stage gains in PSNR and NC, increasing them by 1.63 dB and 1.88 points, respectively, over Stage II. CD decreases further to 6.35 and SSIM increases to 0.849. The final stage achieves the best CD, NC, PSNR, and SSIM, while LPIPS remains 0.192. F-score peaks at Stage II and then decreases by 0.17 points to 67.68, so improved average surface proximity and orientation do not coincide with a further gain in thresholded geometric agreement.

Across Stages I–III, CD falls from 6.55 to 6.35, NC rises by 2.66 points, and PSNR rises from 22.38 to 24.28 dB. The final transition accounts for most of the PSNR and NC gains, showing further improvements in appearance and surface orientation after the earlier coverage-oriented stages. Figure 11 localizes the accompanying changes in RGB, normal, and depth predictions.

A.3.4 Additional surface reconstruction comparisons

Figures 6 and 7 extend the five-view surface reconstruction comparison with four examples from each dataset. Each row compares 2DGS, MAtCha, G4Splat, ReconAgent, and ground truth from the same viewpoint, with identical crop coordinates across methods. The underlined MAtCha label identifies ReconAgent's reconstruction backbone and baseline.

Table 8: **Stage-wise surface reconstruction and novel-view synthesis on Replica with five input views.**

Stage	Surface reconstruction			Novel-view synthesis		
	CD↓	F-score↑	NC↑	PSNR↑	SSIM↑	LPIPS↓
I (coarse)	6.55	67.14	81.45	22.38	0.841	0.198
II (occluded)	6.42	**67.85**	82.23	22.65	0.846	**0.192**
III (detailed)	**6.35**	67.68	**84.11**	**24.28**	**0.849**	**0.192**

Replica. Figure 6 shows improvements in both room surfaces and furniture geometry. In the first row, ReconAgent replaces the large folded ceiling patches visible in 2DGS and MAtCha with a continuous surface and reduces the roughness remaining in G4Splat. The second row examines the foreground chair back: ReconAgent recovers a flatter surface and a more regular upper outline, closer to the reference than MAtCha's fragmented reconstruction or G4Splat's rounded shape. The lower two rows extend this comparison to a back wall and the floor. ReconAgent suppresses wall undulations and recovers a connected floor where MAtCha produces large spurious facets; the floor also exhibits fewer ripples than G4Splat. Fine furniture details and the whiteboard boundary remain less distinct than in the reference, while the highlighted regions show gains in surface continuity and local shape.

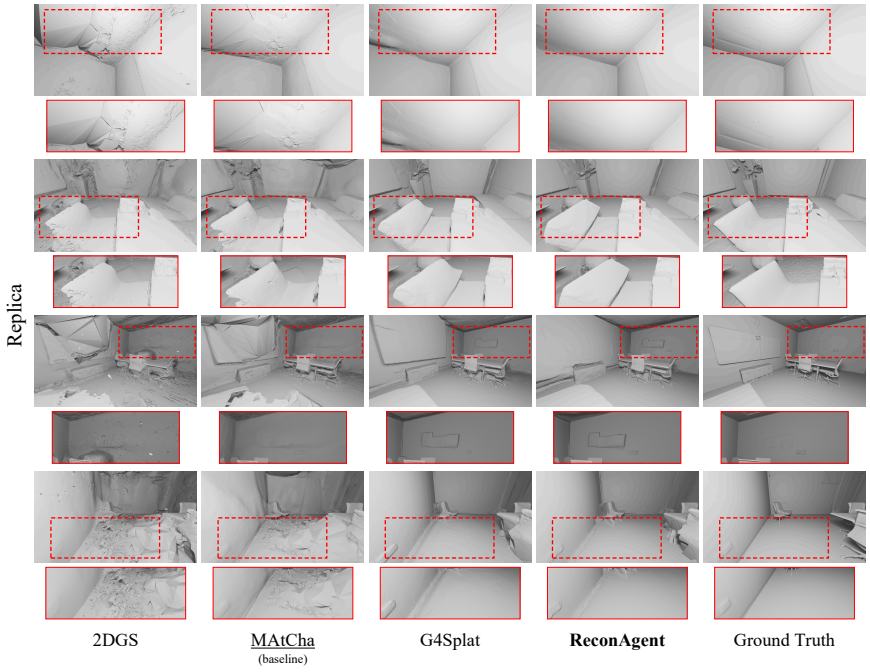

Figure 6: **Additional surface reconstruction comparisons on Replica.** Four five-view examples highlight ceiling continuity, chair-back shape, wall smoothness, and floor completion. Red dashed boxes mark the regions enlarged below each full view, using the same crop across methods. Underlined MAtCha identifies our reconstruction baseline. ReconAgent reduces folded surfaces and local irregularities in the highlighted regions.

ScanNet++. Figure 7 shows how these improvements extend to captured indoor scenes. In the upper three rows, 2DGS and MAtCha produce folded sheets or fragmented surfaces near the ceiling, doorway, and cabinet. ReconAgent reconstructs these regions as more continuous surfaces. Compared with G4Splat, it reduces irregularities along the upper wall in the first row, the broken surface above the doorway in the second, and the rough patch above the cabinet in the third. In the last row, ReconAgent produces a more continuous chair-back surface and surrounding floor than MAtCha, with fewer local irregularities than G4Splat. Fine chair details, small shelf contents, and wall fixtures remain simplified relative to ground truth. Across both datasets, the enlarged regions

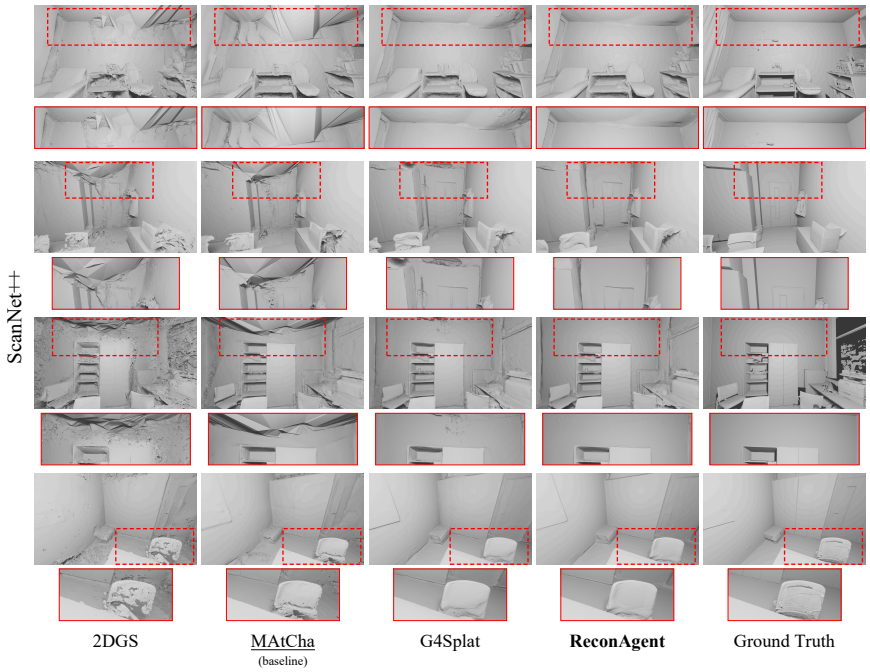

Figure 7: **Additional surface reconstruction comparisons on ScanNet++.** Four five-view examples focus on upper walls, the doorway, the region above the cabinet, and the chair with its surrounding floor. Red dashed boxes and the corresponding enlargements use identical coordinates across methods. Underlined MAtCha identifies our reconstruction baseline. ReconAgent reduces folded sheets, fragmented regions, and residual surface ripples.

show that the qualitative gains extend beyond rendered appearance to the continuity and local shape of the extracted surfaces.

A.3.5 Additional novel-view synthesis comparisons

Figures 8–10 extend the novel-view synthesis comparison to eight Replica scenes and four ScanNet++ scenes with five input views. Each row compares the same camera view across 2DGS, MAtCha, G4Splat, ReconAgent, and ground truth; the enlarged regions use identical crop coordinates. MAtCha is the reconstruction backbone and baseline for ReconAgent.

Completion of missing regions. Figure 8 focuses on ceilings that remain largely unrendered by MAtCha. ReconAgent fills much of these regions in all four examples. Relative to G4Splat, it reduces residual ceiling gaps in the first and last rows and avoids the bright yellow patch in the second row. Completion and appearance fidelity remain distinct: in the final ScanNet++ example, ReconAgent fills the ceiling but does not recover the reference light fixtures.

Ceiling and wall boundaries. Figure 9 examines the transition between completed ceilings and surrounding walls. MAtCha leaves broad dark regions along these boundaries. ReconAgent restores a continuous ceiling–wall junction in the first Replica example and fills the upper-right region in the second. G4Splat also completes these regions, but introduces a bright patch and an irregular edge, respectively. On ScanNet++, ReconAgent reduces the remaining gaps visible in the G4Splat renderings. The enlarged views also expose residual appearance differences: the completed surfaces are uneven in the upper ScanNet++ example, and the light fixture in the lower example remains elongated relative to ground truth.

Room surfaces and foreground structure. Figure 10 shows four further Replica scenes. ReconAgent fills the missing upper-wall regions in the top two rows relative to MAtCha. In the first example, its ceiling appearance is more uniform than G4Splat's brightly patterned completion. The lower two rows examine regions closer to the camera: ReconAgent reduces the streaking around the foreground furniture and fills most of the floor gap. These regions remain blurred or distorted

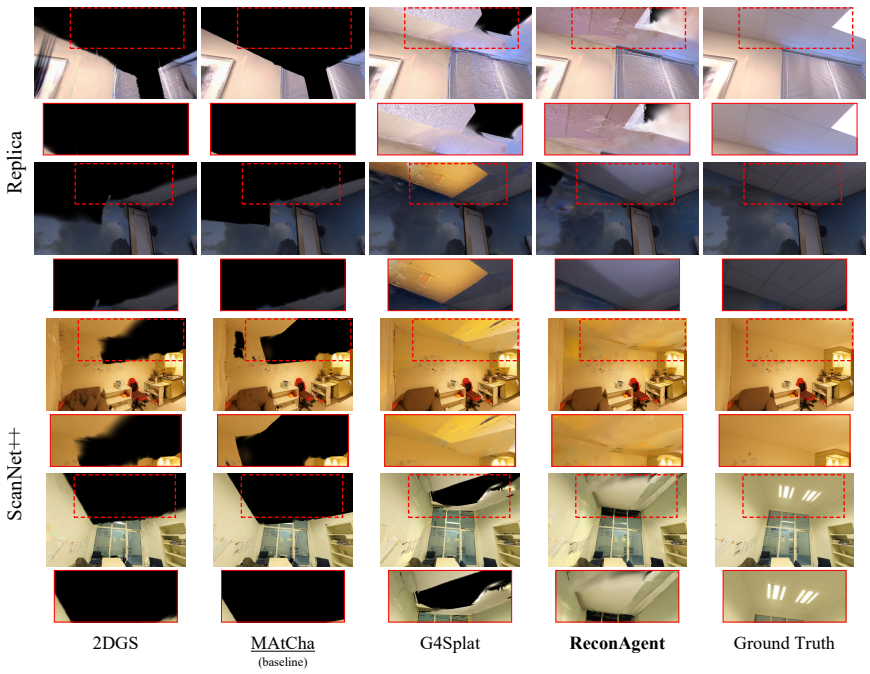

Figure 8: **Additional novel-view synthesis comparisons of ceiling completion.** The upper two examples are from Replica and the lower two from ScanNet++, all with five input views. Red dashed boxes mark the regions enlarged below each full view. The underlined MAtCha label identifies our reconstruction baseline. ReconAgent reduces large missing regions and several localized appearance artifacts, while the reference images reveal remaining differences in ceiling texture and lighting detail.

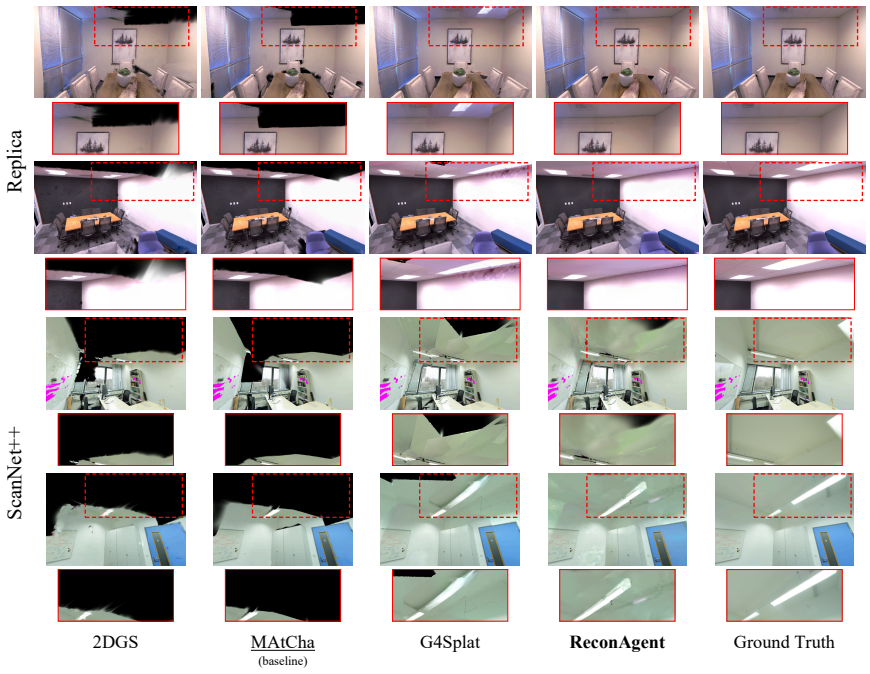

Figure 9: **Additional novel-view synthesis comparisons of ceiling–wall boundaries.** The five-view examples compare identical camera views and cropped regions. ReconAgent fills gaps left by its MAtCha baseline and reduces several discontinuities visible in the G4Splat renderings. The reference column shows the target boundaries and lighting structures.

relative to ground truth, but the comparisons show that the gain in rendered completeness extends beyond ceilings to foreground structure and floors.

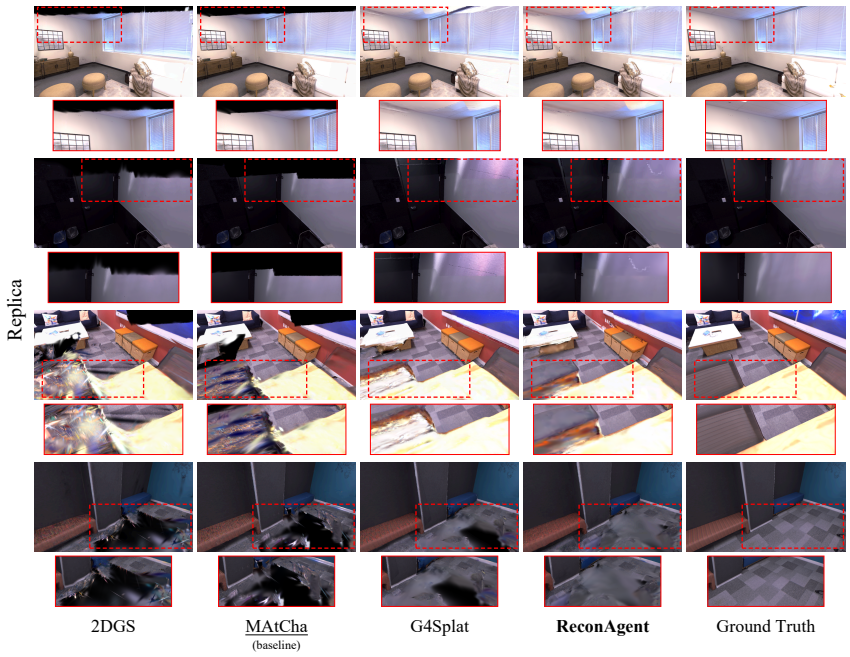

Figure 10: **Additional novel-view synthesis comparisons of room surfaces and foreground regions.** Four Replica examples under the five-view setting highlight upper walls, foreground furniture, and a floor region. ReconAgent reduces missing regions and streaking relative to its MAtCha baseline; enlarged crops reveal the remaining discrepancies from ground truth.

A.3.6 Planner behavior across refinement stages

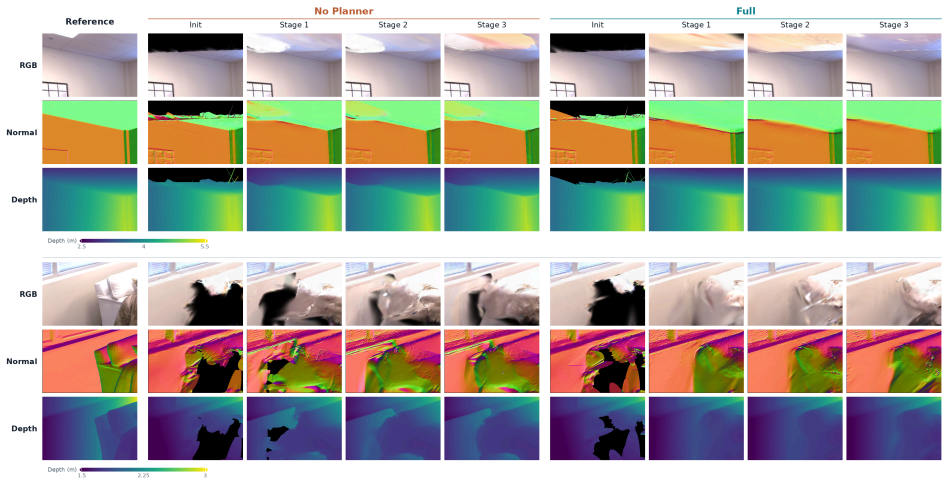

Figure 11: **Planner ablation across refinement stages.** Two examples trace reconstruction from initialization through three refinement stages under random view selection and gain-aware selection. Rows show RGB, normal, and depth predictions; the leftmost column provides the reference. Depth color bars are in meters. Gain-aware selection progressively fills the wall and ceiling regions in the upper example and recovers more coherent chair geometry in the lower example.

Figure 11 compares gain-aware selection with random selection. Both variants use the same initial field, admissible candidate pool, and selected view count at each stage. RGB, normal, and depth predictions trace appearance and geometry from initialization through three refinement stages.

In the upper example, gain-aware selection fills missing wall and ceiling regions that remain incomplete under random selection. By the final stage, the Full variant produces a more continuous RGB image and larger connected regions in both normal and depth maps. The accompanying geometric maps show that the improvement extends beyond RGB completion to more continuous surface predictions. Random selection leaves gaps visible in the final geometric maps despite using the same view budget.

The lower example emphasizes local structure around the chair. The Full variant recovers a more coherent outline and surface orientation, with fewer interruptions in the corresponding depth prediction. Together, the examples show improvements in both extended room surfaces and partially observed furniture.

These local differences accompany the aggregate gains in Figure 5: relative to random selection, the full model reduces CD from 6.49 to 6.35, increases PSNR from 23.82 to 24.28 dB, and reduces LPIPS from 0.222 to 0.192, while SSIM remains 0.849. We report the full model's quantitative progression across stages in **Appendix A.3.3**.

A.3.7 ARBITRATION EXAMPLES

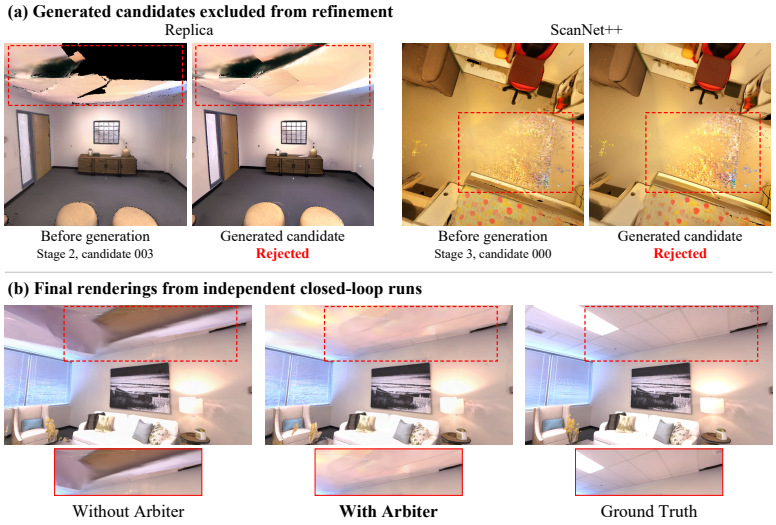

Figure 12: **Candidate rejection and novel-view synthesis with the Arbiter.** (a) Two generated candidates rejected by the arbiter, paired with their pre-generation renderings. Red boxes locate persistent appearance defects. (b) Final renderings from independent runs with and without arbitration, shown at the same Replica camera with a shared ceiling crop and ground truth. Arbitration accompanies a more uniform ceiling in this example.

Figure 12 illustrates selective admission by the Qwen3-VL-8B-Instruct arbiter. Panel (a) pairs each pre-generation rendering with its generated candidate. In the Replica example, generation fills part of the ceiling gap but leaves fragmented ceiling appearance. In the ScanNet++ example, a mottled floor region persists after generation. The recorded Arbiter decisions reject both candidates, excluding them from refinement. These examples distinguish producing a more complete RGB image from deciding whether to use it as supervision.

Panel (b) compares final Replica renderings at the same camera from independent closed-loop runs with and without the arbiter. The run with arbitration produces a more uniform ceiling, whereas the run without arbitration contains a large brown ceiling patch. The two runs generate different candidate sets as their reconstruction states evolve, so the comparison illustrates their final trajectories rather than isolating the effect of either displayed rejection.