LONGAUSSIAN: TRACKING FOUNDATION PRIORS FOR LONG-CONTEXT FEEDFORWARD GAUSSIAN RECONSTRUCTION

Anonymous authors
Paper under double-blind review

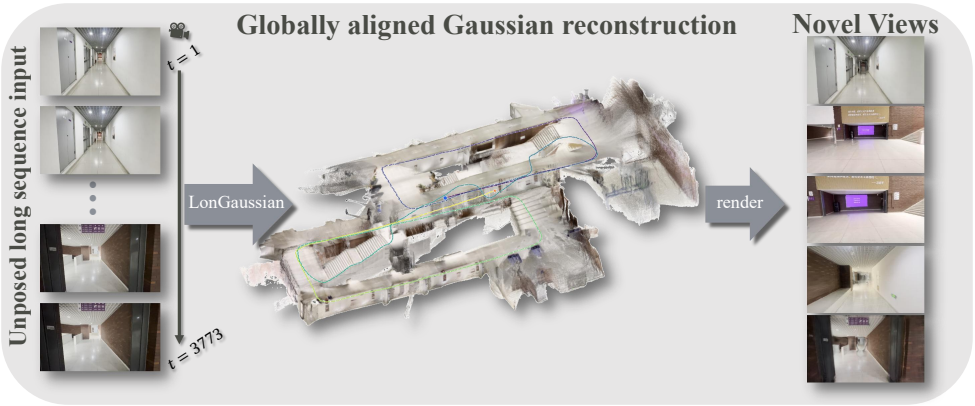


Figure 1: Ultra-long reconstruction with known intrinsics and unknown camera poses. LonGaussian accumulates Gaussians and camera poses in a shared coordinate system to support novel-view rendering.

ABSTRACT

Reconstructing high-quality, renderable 3D scenes from long videos is essential for robotic perception and embodied intelligence. Many feed-forward Gaussian reconstruction methods rapidly predict 3D scenes from a small set of images, but extending them to longer sequences presents two bottlenecks: jointly processing all views increases GPU memory requirements, while accumulating pixel-aligned predictions redundantly represents the same surfaces and increases storage and rendering costs. We present **LonGaussian**, a streaming feed-forward framework that reconstructs compact, globally renderable Gaussian scenes from long image sequences with known intrinsics and unknown camera poses. Built on a pretrained streaming 3D reconstruction backbone, LonGaussian decodes long-context geometric features into dense Gaussians. Within each chunk, multi-frame point tracks link repeated observations of the same surface and guide a learned merger to consolidate the corresponding Gaussian predictions. Rendered-depth consistency filtering further suppresses Gaussians from other chunks that incorrectly occlude locally reconstructed surfaces. On DL3DV and RealEstate10K benchmarks, LonGaussian improves novel-view rendering quality while using substantially fewer Gaussians than the evaluated feed-forward baselines.

1 INTRODUCTION

Reconstructing renderable 3D scenes from long videos supports applications in robotic perception, embodied intelligence, and immersive media. 3D Gaussian Splatting (3DGS) represents scenes with explicit Gaussian primitives, enabling high-quality novel-view synthesis and real-time rendering (Kerbl et al., 2023). Optimization-based reconstruction methods produce high-quality scene representations but require time-consuming per-scene iterative optimization (Mildenhall et al., 2021;

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



Figure 2: Gaussian redundancy from repeated observations. Pixel-aligned predictions accumulate overlapping primitives for the same surface, increasing storage and rendering costs; matching colors denote corresponding observations and Gaussians.

Kerbl et al., 2023; Huang et al., 2024; Wu et al., 2024; Zhang et al., 2025). Feed-forward Gaussian reconstruction improves efficiency by directly predicting renderable 3D representations from multiple images through network inference (Charatan et al., 2024; Chen et al., 2024). However, many existing models jointly process the input views (Charatan et al., 2024; Chen et al., 2024; Xu et al., 2025a; Jiang et al., 2025), causing computational and GPU memory requirements to grow rapidly with sequence length. These demands limit long-video inference, whose memory requirements may exceed GPU capacity.

Scaling to long sequences is constrained by both the inference backbone and the explicit scene representation. Online approaches alleviate joint-view computational bottlenecks by updating an accumulated Gaussian scene or a compressed historical state (Chen et al., 2026a; Cheng et al., 2026; Song et al., 2026; Huang et al., 2026). However, many feed-forward Gaussian predictors still associate one primitive with each input pixel. Accumulating predictions from repeated surface observations retains overlapping Gaussians for the same geometry, increasing representation size and rasterization workload with sequence length. Figure 2 illustrates this redundancy. Although LoGeR enables streaming long-context geometric reconstruction through chunk-wise processing and hybrid memory (Zhang et al., 2026a), it predicts dense geometry and camera trajectories rather than maintaining a compact renderable scene. Extending feed-forward reconstruction to long videos therefore requires both streaming geometric processing and explicit control of redundancy caused by repeated observations.

Tracking foundation models offer a natural prior for addressing this redundancy. A point trajectory links image observations of the same physical surface across frames and records surface visibility (Harley et al., 2022; Doersch et al., 2023; Karaev et al., 2024; 2025; Xiao et al., 2024; Harley et al., 2025). These trajectories can guide Gaussian grouping across repeated observations. In contrast to spatial strategies that aggregate primitives according to voxel occupancy or Euclidean proximity, track-based grouping uses explicit cross-frame correspondences, providing a more direct and interpretable basis for consolidation.

To this end, we propose **LonGaussian**, a streaming feed-forward Gaussian reconstruction framework for long image sequences. LoGeR provides the backbone for predicting renderable 3D Gaussians, using chunk-wise hybrid memory to process local chunks while retaining long-range geometric context. An independent tracker within each chunk groups corresponding pixels into multi-frame trajectories. A learned merger consolidates their Gaussian predictions into representative primitives, reducing redundancy while integrating geometric and appearance evidence. Rendered-depth consistency filtering further suppresses residual cross-chunk floaters. Together, these components reconstruct compact, globally renderable scenes from long sequences while preserving detail for high-quality rendering. Figure 1 provides a qualitative example of scene reconstruction and novel-view rendering from an ultra-long unposed sequence.

Our contributions are summarized as follows:

- We propose **LonGaussian**, a streaming feed-forward framework that extends LoGeR's long-context geometric representation to the reconstruction of compact, globally renderable Gaussian scenes.
- We introduce a Gaussian consolidation mechanism that uses explicit cross-frame correspondences from multi-frame tracks to merge redundant predictions within each chunk, together with rendered-depth consistency filtering for residual cross-chunk artifacts.
- Experiments on DL3DV and RealEstate10K demonstrate high-quality novel-view synthesis with substantially fewer Gaussians than existing feed-forward approaches; an ultra-long real-world sequence further demonstrates qualitative applicability at scale.

2 Related Work

2.1 Feed-Forward and Long-Context 3D Reconstruction

Feed-forward 3D reconstruction has progressed from pairwise point-map regression and matching models such as DUSt3R and MASt3R (Wang et al., 2024; Leroy et al., 2024) to architectures that process larger image collections or streams. Offline models such as VGGT and π^3 jointly reason over all input views through global attention, whose computational and memory costs grow rapidly with sequence length (Wang et al., 2025a; 2026b; Yang et al., 2025; Cabon et al., 2025; Tang et al., 2025; Lin et al., 2026; Keetha et al., 2026; Wang et al., 2026a). CUT3R instead processes observations sequentially while maintaining a persistent recurrent state (Wang et al., 2025b; Wang & Agapito, 2025; Liu et al., 2025; Chen et al., 2025; Lan et al., 2026). TTT3R reinterprets CUT3R's recurrent update as test-time training and derives a confidence-aware learning rate to improve generalization to longer sequences (Chen et al., 2026b). Separately, LaCT improves the hardware efficiency and state capacity of test-time training by updating fast weights over large token chunks (Zhang et al., 2026b; Elflein et al., 2026; Jin et al., 2026). LoGeR combines chunk-wise bidirectional reasoning with sliding-window attention for uncompressed local context and TTT fast weights for compressed global context (Zhang et al., 2026a). These models primarily predict point maps, depth, and camera parameters; a suitable bridge to compact, explicit Gaussian scenes remains missing. We build on LoGeR's long-context geometry to address this complementary problem of feed-forward reconstruction of compact, globally renderable scenes.

2.2 Feed-Forward Gaussian Reconstruction

Feed-forward Gaussian reconstruction predicts image-conditioned 3D Gaussians from multi-view inputs (Charatan et al., 2024; Chen et al., 2024; Xu et al., 2025a; Smart et al., 2024; Ye et al., 2025; Kang et al., 2025; Xu et al., 2025b; Huang & Mikolajczyk, 2025; Ye et al., 2026), with AnySplat further supporting uncalibrated image collections (Jiang et al., 2025). Because these predictions are commonly pixel-aligned, overlapping views may assign redundant or inconsistent primitives to the same physical surface, causing the accumulated representation to grow with sequence length (Park et al., 2026; Zhang et al., 2026c; Miao et al., 2025; Xia et al., 2025; Guo et al., 2026; Smart et al., 2026). Recent approaches support online or long-sequence reconstruction through recursive fusion, rendering feedback, or persistent scene memory (Chen et al., 2026a; Cheng et al., 2026; Song et al., 2026; Huang et al., 2026; Li et al., 2025), but preserving geometry and rendering quality remains challenging as sequences grow. Our framework builds on LoGeR's established long-context backbone. Bidirectional within-chunk reasoning and global TTT memory establish coherent geometry before Gaussian consolidation, separating long-context inference from explicit scene maintenance.

2.3 Compact Gaussian Representations

Compression of optimization-based 3DGS commonly relies on pruning, quantization, codebooks, or entropy coding (Lee et al., 2024; Fan et al., 2024; Hanson et al., 2025). These methods operate on per-scene optimized representations and often need training-view statistics or refinement, complicating integration into feed-forward reconstruction without additional per-scene processing. Feed-forward approaches instead control the number of primitives through geometry-adaptive allocation (Fei et al., 2024; Park et al., 2026; Moreau et al., 2026), anchor- or voxel-aligned representations (Zhang et al., 2026c; Guo et al., 2026), and matching-based fusion (Xia et al., 2025).

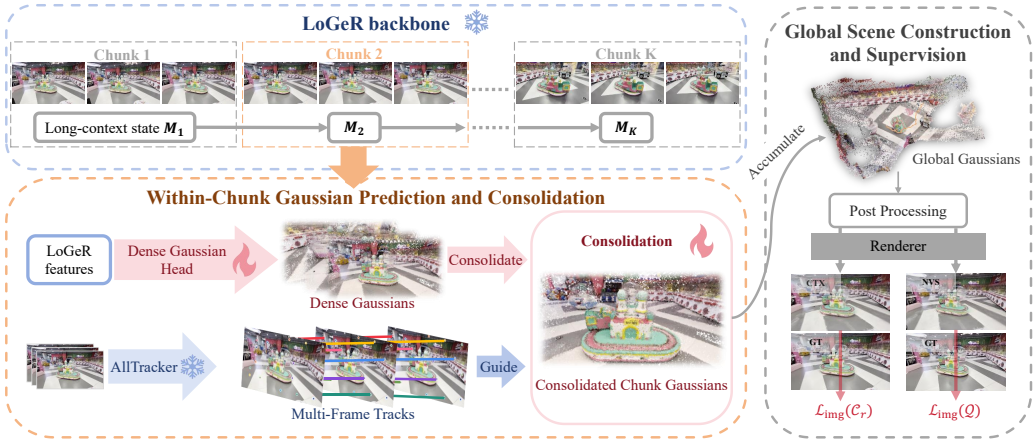


Figure 3: Overview of **LonGaussian**. LoGeR processes overlapping chunks, and AllTracker correspondences guide Gaussian consolidation before cross-chunk filtering and far-field modeling. Snowflakes and flames denote frozen and trainable modules, respectively.

Such strategies typically consolidate Gaussians according to importance, spatial proximity, or local matches (Smart et al., 2026), but do not explicitly organize repeated observations of the same surface through consistent multi-frame correspondences. In contrast, LonGaussian uses tracker-produced trajectories as explicit consolidation units, grouping track-supported observations across an entire chunk before fusing their Gaussian predictions. Resolving correspondence before consolidation targets the source of long-sequence redundancy rather than a post-hoc density problem, while preserving efficient feed-forward inference without per-scene refinement.

3 Method

3.1 Preliminaries

3D Gaussian Splatting. 3DGS represents a scene as a set of anisotropic Gaussian primitives $\mathcal{G} = \{g_i\}_{i=1}^{M}$ (Kerbl et al., 2023). Each primitive is parameterized as

$$g_i = \left(\boldsymbol{\mu}_i,\, \mathbf{q}_i,\, \mathbf{s}_i,\, \alpha_i,\, \boldsymbol{\theta}_i\right), \tag{1}$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ is the center, $\mathbf{q}_i \in \mathbb{R}^4$ is a unit quaternion encoding rotation, $\mathbf{s}_i \in \mathbb{R}^3_{>0}$ is the anisotropic scale, $\alpha_i \in (0,1)$ is the opacity, and $\boldsymbol{\theta}_i$ contains spherical-harmonic (SH) coefficients for view-dependent color. A differentiable rasterizer $\mathcal{R}$ renders an image from viewpoint $(\mathbf{T}, \mathbf{K})$ as $\hat{\mathbf{I}} = \mathcal{R}(\mathcal{G}, \mathbf{T}, \mathbf{K})$. We therefore adopt 3DGS as our scene representation: its explicit primitives from repeated observations can be consolidated directly, while differentiable rendering provides image-space supervision for the resulting global scene.

Long-Context Geometric Reconstruction. Long-Context Geometric Reconstruction (LoGeR) reconstructs long image sequences through chunk-wise processing and hybrid memory (Zhang et al., 2026a). It partitions a sequence into overlapping chunks of length L with O shared frames and a stride of $L - O$. Within each chunk, bidirectional interaction captures local geometry; across chunks, Sliding-Window Attention (SWA) preserves detailed recent context, while persistent TTT fast weights carry compressed long-range information. For chunk $\mathcal{I}_k$ with frame-index set $\mathcal{J}_k$, we summarize the backbone's prediction interface as

$$(\mathbf{H}_k, \{\mathbf{P}_t, \mathbf{T}_t\}_{t\in\mathcal{J}_k}) = \mathcal{B}(\mathcal{I}_k; \mathbf{M}_{k-1})\,, \tag{2}$$

where $\mathbf{H}_k$ denotes long-context features, $\mathbf{P}_t$ and $\mathbf{T}_t$ are the dense point map and camera pose for frame t, and $\mathbf{M}_{k-1}$ comprises the SWA cache and TTT fast weights carried from preceding chunks. LoGeR updates memory per chunk and registers successive chunks using overlapping-frame pose estimates. LonGaussian inherits this pretrained backbone, memory, and alignment without modification. It extends LoGeR's dense geometry and camera motion to compact, globally renderable

scenes by decoding long-context features into 3D Gaussians and consolidating repeated surface observations.

3.2 LonGaussian Framework

Given an unposed RGB sequence $\mathcal{I} = \{\mathbf{I}_t\}_{t=1}^{N}$ with known intrinsics $\mathbf{K}$, LonGaussian estimates camera poses and reconstructs a compact global Gaussian scene:

$$\mathcal{G} = F(\mathcal{I}, \mathbf{K}). \tag{3}$$

We decompose F into dense prediction, track-guided consolidation, and scene-level artifact handling (Figure 3). For each overlapping chunk $\mathcal{I}_k$, the frozen LoGeR backbone $\mathcal{B}$ extracts geometry and features using the long-context state $\mathbf{M}_{k-1}$ from preceding chunks. A trainable Gaussian branch $\mathcal{D}_\psi$ decodes these outputs, and the inherited alignment operator $\mathcal{W}_k$ maps the predictions to the first chunk's coordinate system:

$$\mathcal{G}_k^d = \mathcal{W}_k[\mathcal{D}_\psi(\mathcal{B}(\mathcal{I}_k, \mathbf{K}_k; \mathbf{M}_{k-1}))]\,. \tag{4}$$

Each context pixel produces one Gaussian, centered at its predicted 3D point plus a learned offset. Frozen AllTracker $\mathcal{A}$ extracts multi-frame tracks directly from $\mathcal{I}_k$, linking repeated surface observations. The consolidation operator $\mathcal{C}_\phi$ replaces track-associated predictions with representative Gaussians while retaining uncovered predictions. After the aligned chunk representations are accumulated, the scene operator $\mathcal{S}$ applies rendered-depth consistency filtering and far-field shell construction:

$$\mathcal{G} = \mathcal{S}\left(\bigcup_{k=1}^{K} \mathcal{C}_\phi\big(\mathcal{G}_k^d, \mathcal{A}(\mathcal{I}_k)\big)\right). \tag{5}$$

Here, globally renderable denotes a single Gaussian representation in a shared coordinate system, without guaranteeing drift-free alignment on long-range revisits.

3.3 Track-Guided Gaussian Consolidation

The consolidation operator groups predictions by correspondence before merging them. For each chunk, frozen AllTracker (Harley et al., 2025) associates image locations across frames. Each location indexes the corresponding pixel-aligned Gaussian g_{mt}. We retain observations whose tracking confidence exceeds τ_{trk} and assign each frame–pixel prediction to at most one track. Let Ω_m denote the retained observations of track m. A track is accepted only if it contains at least $n_{\min}$ observations, its mean opacity is at least τ_α, and the root-mean-square distance of its centers from their arithmetic mean is at most τ_{geo}.

The geometric acceptance test is

$$\left(\frac{1}{|\Omega_m|}\sum_{t\in\Omega_m} \|\boldsymbol{\mu}_{mt} - \tilde{\boldsymbol{\mu}}_m\|_2^2\right)^{1/2} \le \tau_{\text{geo}}, \tag{6}$$

where $\tilde{\boldsymbol{\mu}}_m$ is the mean center of the retained observations. This test rejects tracks whose associated predictions do not agree on a common 3D location.

For each accepted track, we construct a base Gaussian $\bar{g}_m$ using opacity-weighted attribute averages, alpha compositing, and projected rotation averaging. A permutation-invariant network refines this base using pooled information from the associated observations:

$$\hat{g}_m = \bar{g}_m \oplus F_\phi(\text{Pool}\{g_{mt} : t \in \Omega_m\})\,. \tag{7}$$

The pooled descriptor contains the mean and standard deviation of the centers, scales, opacities, and SH coefficients, together with the visible-frame fraction, summarizing observation agreement and track support. The operator $\oplus$ applies additive corrections to the center and SH coefficients and bounded corrections in log-scale and logit-opacity space, while retaining the base rotation. Zero-initialized output layers make the merger initially reproduce deterministic aggregation before learning residual corrections through rendering supervision. Each accepted track contributes one refined representative in place of its member Gaussians. Uncovered predictions from the chunk's non-overlapping contribution range are retained to preserve scene coverage. Together, these representatives and uncovered predictions form the output of $\mathcal{C}_\phi$.

3.4 Scene-Level Artifact Handling

The scene operator $\mathcal{S}$ addresses residual cross-chunk occlusions and unreliable sky geometry. To identify foreground floaters, we render each chunk's scene to obtain reference depth and opacity, then project Gaussians from other chunks into that chunk's views. Comparisons exclude neighborhoods with insufficient opacity support or excessive depth spread. For Gaussian i projected into frame t of chunk k, we measure

$$\delta_{ikt} = \frac{\widetilde{D}_{kt}(\mathbf{u}_{ikt}) - z_{ikt}}{\widetilde{D}_{kt}(\mathbf{u}_{ikt})}, \tag{8}$$

where z_{ikt} is the Gaussian's camera-space depth, $\mathbf{u}_{ikt}$ is its projected position, and $\widetilde{D}_{kt}$ is the local median reference depth. Positive values indicate a Gaussian in front of the reference surface. A reliable comparison contributes an occlusion vote when $\delta_{ikt} \geq \tau_{\text{gap}}$, and an agreement vote when $|\delta_{ikt}| \leq \tau_{\text{agree}}$.

Removal requires at least n_{vote} occlusion votes, an occlusion fraction of at least ρ_{vis} among all reliable comparisons, and an occlusion fraction of at least ρ_{dec} among the combined occlusion and agreement votes. This conservatively suppresses persistent foreground floaters; a chunk never votes against its own predictions. For sky pixels identified by a fixed segmenter, we replace finite-depth Gaussians with viewing-direction observations on a shell of radius $r_{\text{sky}} = \gamma_{\text{sky}} r_{\text{scene}}$. Angular binning yields one isotropic, degree-zero-SH Gaussian per occupied bin, compactly representing the far-field appearance.

3.5 Training Strategy and Objectives

We train the Gaussian branch and merger in two stages while keeping LoGeR and AllTracker frozen. For stage $s \in \{1, 2\}$, the complete objective is

$$\mathcal{L}^{(s)} = \mathcal{L}_{\text{img}}(\mathcal{C}_r) + \lambda_{\text{tgt}} \mathcal{L}_{\text{img}}(\mathcal{Q}) + \lambda^{(s)}_{\text{aniso}} \mathcal{L}_{\text{aniso}} + \lambda^{(s)}_{\text{off}} \mathcal{L}_{\text{off}}. \tag{9}$$

The set $\mathcal{C}_r$ contains a small subset of context views; target views $\mathcal{Q}$ never contribute Gaussians. For either view set $\mathcal{V}$, the image objective is

$$\mathcal{L}_{\text{img}}(\mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{t \in \mathcal{V}} \left(\lambda_{\text{rgb}} \ell^t_{\text{rgb}} + \lambda_{\text{lpips}} \ell^t_{\text{lpips}} + \lambda_{\text{ssim}} \ell^t_{\text{ssim}} \right). \tag{10}$$

The three per-view terms measure RGB mean-squared error, LPIPS, and $1 - \text{SSIM}$, respectively, between the rendered image and its ground truth.

Stage I trains the dense Gaussian branch on short sequences with nonzero $\lambda^{(1)}_{\text{aniso}}$ and $\lambda^{(1)}_{\text{off}}$, so the corresponding terms regularize anisotropic scale and the learned offset from the point map. Stage II initializes from Stage I and freezes the feature decoder. Stage IIa trains only the merger on mixed-length DL3DV clips; Stage IIb then jointly adapts the merger and, on short clips only, the attribute head using DL3DV and RE10K. We set $\lambda^{(2)}_{\text{aniso}} = \lambda^{(2)}_{\text{off}} = 0$, so Stage II directly supervises the consolidated scene through the context- and target-view image losses in Eq. equation 9. The final training stage includes depth filtering and the far-field shell. Correspondence, ownership, filtering, and sky assignment are treated as non-differentiable routing decisions; gradients pass through the retained learned Gaussian attributes and differentiable rendering.

4 Experiments

4.1 Experimental Setup

Baselines. We compare LonGaussian with the offline methods AnySplat (Jiang et al., 2025) and WorldMirror (Liu et al., 2026), and the streaming methods OF3GS (Chen et al., 2026a) and ReCoSplat (Cheng et al., 2026). For ReCoSplat, we report both its released low-resolution checkpoint and a locally resolution-adapted checkpoint; the latter is an accommodation for our evaluation resolution rather than an official training reproduction. StreamSplat (Song et al., 2026), SaLon3R (Guo et al., 2026), and StreamGS (Li et al., 2025) are omitted because runnable official implementations were unavailable at the time of evaluation.

Table 1: Main results on DL3DV and RealEstate10K using complete reconstructed scenes for target-view rendering. #G is the average Gaussian count in millions; pink / yellow mark best/second-best. "OOM" exceeds 24-GB GPU memory.

Dataset	Method	64 frames				96 frames				128 frames			
		PSNR↑	SSIM↑	LPIPS↓	#G↓	PSNR↑	SSIM↑	LPIPS↓	#G↓	PSNR↑	SSIM↑	LPIPS↓	#G↓
DL3DV	WorldMirror	OOM				OOM				OOM			
	AnySplat	OOM				OOM				OOM			
	ReCoSplat (official)	13.62	0.234	0.647	8.78	13.67	0.237	0.654	13.17	13.70	0.241	0.660	17.56
	ReCoSplat (adapted)	15.95	0.288	0.599	8.78	15.74	0.282	0.606	13.17	15.48	0.276	0.612	17.56
	OF3GS	16.29	0.438	0.489	7.10	15.31	0.404	0.545	10.45	14.83	0.316	0.574	13.67
	LonGaussian	19.57	0.617	0.381	2.36	18.88	0.575	0.420	3.54	18.31	0.536	0.450	4.80
RealEstate10K	WorldMirror	OOM				OOM				OOM			
	AnySplat	OOM				OOM				OOM			
	ReCoSplat (official)	13.66	0.358	0.609	8.78	13.72	0.359	0.618	13.17	13.67	0.358	0.627	17.56
	ReCoSplat (adapted)	16.34	0.431	0.589	8.78	15.85	0.421	0.601	13.17	15.34	0.412	0.612	17.56
	OF3GS	18.47	0.580	0.331	6.04	17.17	0.518	0.384	8.74	16.19	0.473	0.431	11.37
	LonGaussian	19.73	0.636	0.339	1.12	18.91	0.595	0.372	1.66	18.23	0.561	0.397	2.15

Datasets and Metrics. We train on DL3DV and use 150 sequences from the official RE10K training split for final adaptation. We evaluate on DL3DV-140 and 180 eligible sequences from the official RE10K test split; identity-level audits confirm zero overlap between training and evaluation sequences. Images are resized to 280×490; each scene is constructed from up to 128 context frames, and held-out views serve as targets. LonGaussian uses two-pass unposed evaluation: the first pass processes context and target images to estimate camera poses, while the second resets the TTT states and SWA cache and constructs the scene from context images only. We align the first-pass trajectory to the second-pass scene using shared context cameras and render held-out views at the aligned target poses (Appendix D.2). Methods share frame indices whenever their interfaces permit, and ReCoSplat retains its prescribed first-chunk scale matching. We average PSNR, SSIM, LPIPS, and the final Gaussian count over all held-out views and scenes.

Implementation Details. We initialize and freeze the public LoGeR backbone and set $L = 12$, $O = 3$. With a global batch size of four, Stage IIa trains only the merger on DL3DV for 900 steps; Stage IIb adapts for 300 steps, updating the attribute head only at 16 frames and the merger at every length. Both uniformly sample $|\mathcal{C}| \in \{16, 64, 96, 128\}$. Training uses AdamW on 24-GB RTX 3090 GPUs. Depth filtering and the far-field shell are enabled in final-stage training and evaluation; supplementary material provides the remaining settings.

4.2 Results

Table 1 and Figure 4 compare reconstruction across sequence lengths. WorldMirror and AnySplat exceed the GPU memory budget even at the shortest evaluated length. Reconstructions from both ReCoSplat variants remain blurred or geometrically distorted, while OF3GS preserves coarse layout but loses fine detail and accumulates many Gaussians. LonGaussian instead achieves the best PSNR and SSIM at every length, produces sharper and more coherent renderings, and uses the fewest Gaussians. OF3GS retains a slight LPIPS advantage on RE10K at the shortest evaluated sequence length. As sequences grow, LonGaussian adds substantially fewer Gaussians than the streaming baselines on both datasets, supporting more compact reconstruction while preserving rendering quality. Supplementary videos provide side-by-side novel-view rendering comparisons for all evaluated methods.

4.3 Long-Sequence Scalability

We evaluate six cumulative prefixes of the scene in Figure 1, extending to 3,768 input frames. Unlike the complete-scene rendering in Table 1, image metrics here average 15 held-out views per prefix, each rendered from its owner chunk and up to three adjacent chunks on either side. This seven-chunk scope reduces interference from distant chunks misregistered by LoGeR's loop-closure errors. Gaussian count and construction resources measure the complete accumulated prefix.

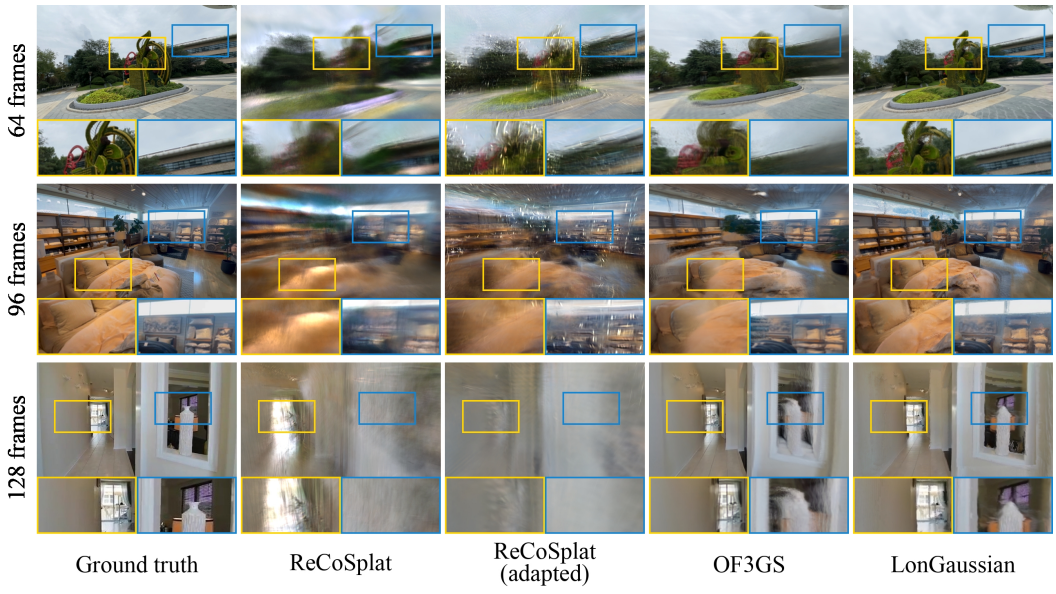


Figure 4: Representative novel-view comparisons across sequence lengths. Colored boxes show that LonGaussian better preserves structure and detail; complete comparisons appear in the Appendix.

Table 2: Single-scene long-sequence scalability. Quality uses seven-chunk rendering (mean $\pm$ standard deviation across target views); #G counts the complete filtered representation. Memory is peak allocated GPU memory during streaming construction. Time excludes model loading, metric rendering, and offline prefix filtering.

Input N	PSNR ↑	SSIM ↑	LPIPS ↓	#G (M) ↓	Memory	Time
126	20.33 ± 1.87	0.711 ± 0.068	0.404 ± 0.061	7.28	19.99 GB	55 s
250	19.35 ± 2.50	0.681 ± 0.083	0.438 ± 0.073	15.04	20.05 GB	101 s
507	18.40 ± 2.83	0.626 ± 0.118	0.485 ± 0.097	28.16	20.05 GB	197 s
1,021	17.73 ± 2.52	0.624 ± 0.094	0.506 ± 0.091	57.72	20.05 GB	392 s
2,040	18.24 ± 2.31	0.586 ± 0.172	0.530 ± 0.111	111.42	20.05 GB	781 s
3,768	17.17 ± 2.89	0.556 ± 0.125	0.531 ± 0.113	208.29	20.05 GB	1,445 s

Table 2 shows approximately linear growth in construction time and representation size, with peak construction memory remaining nearly constant on this sequence. Local rendering quality generally decreases as the prefix expands. Appendix D.4 provides protocol details and scaling curves.

4.4 Ablation Studies

We ablate the components of LonGaussian on DL3DV-140 using 96 context frames and 13 target views per scene, with the same checkpoint, two-pass protocol, resolution, and split as Table 1. Dense accumulation retains pixel-aligned predictions, deterministic merging replaces the learned merger with fixed within-track aggregation, and each remaining variant removes one component.

Table 3 shows that the full model substantially reduces Gaussian count while improving PSNR and SSIM, with a slight LPIPS trade-off relative to dense accumulation. Learned refinement improves rendering quality over deterministic merging at a similar representation size. Removing the dense complement yields a smaller representation but degrades reconstruction quality, demonstrating the importance of preserving content without reliable track support. Depth filtering provides the largest PSNR gain by suppressing cross-chunk occluders, while the far-field shell addresses unstable sky geometry (Figure 5). Disabling either reduces reconstruction and evaluation time but lowers rendering quality: despite retaining more Gaussians, these variants run faster because they omit processing stages. Thus, Gaussian count alone does not determine total runtime.

Table 3: Component ablation on DL3DV-140 at 96 frames. #G is in millions; time is the per-scene reconstruction and evaluation time.

Configuration	PSNR↑	SSIM↑	LPIPS↓	#G↓	Time (s)↓
Dense accumulation	18.48	0.567	**0.416**	10.02	155.8
Deterministic merging	18.63	0.569	0.432	3.60	138.6
w/o dense complement	17.53	0.550	0.424	**0.95**	124.8
w/o depth filtering	17.36	0.482	0.486	4.86	121.5
w/o far-field shell	18.59	0.569	0.425	3.71	**103.4**
Full model	**18.88**	**0.575**	0.420	3.54	132.6

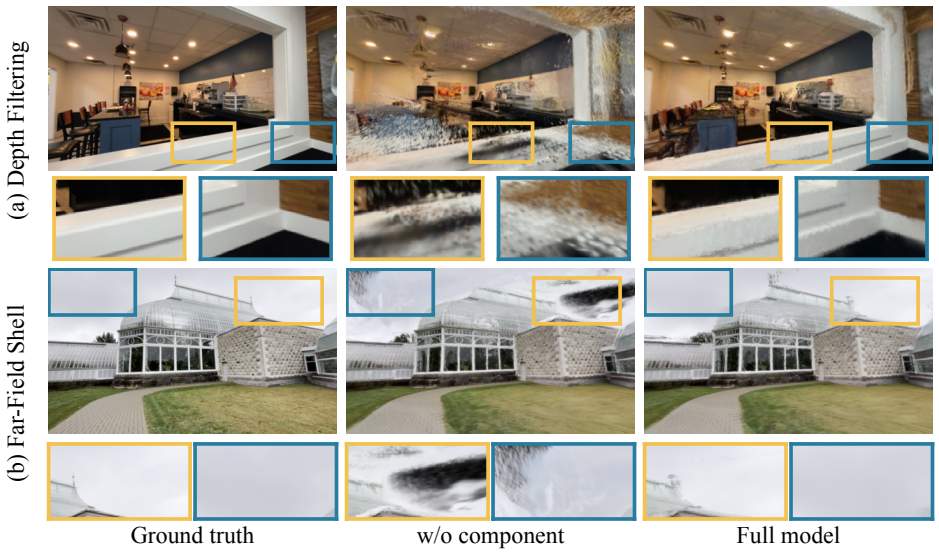


Figure 5: Qualitative artifact ablations. Rendered-depth filtering removes cross-chunk floaters, while far-field modeling avoids unstable sky geometry; colored boxes mark enlarged regions.

5 Discussion and Limitations

LonGaussian assumes static scenes and reliable tracks. Motion, geometric errors, and fragmented tracks may cause reconstruction errors, while within-chunk tracking may duplicate surfaces revisited after long gaps. Depth voting and shell construction add overhead. On long-range revisits, LoGeR pose errors can misalign accumulated Gaussians and cause global rendering ghosting. The seven-chunk metrics reduce interference from distant misregistered chunks and therefore do not measure full-scene global rendering fidelity. Multi-scene benchmarks cover at most 128 frames; Section 4.3 extends a single-scene quantitative case study to 3,768 frames and a qualitative example to 3,773 frames. Future work could address dynamic scenes, cross-chunk association, uncertainty-aware filtering, and adaptive far-field representations.

6 Conclusion

We present LonGaussian, a streaming feed-forward framework for reconstructing compact, globally renderable Gaussian scenes from long image sequences. Building on LoGeR's long-context geometry, LonGaussian leverages tracker-derived correspondences to consolidate redundant Gaussian predictions from repeated surface observations. Rendered-depth filtering and far-field modeling further suppress cross-chunk and sky artifacts. Experiments on DL3DV and RealEstate10K demonstrate improved novel-view rendering quality with substantially fewer Gaussians than existing feed-forward baselines. An ultra-long sequence experiment further demonstrates nearly constant peak construction memory and approximately linear growth in representation size and construction time. Future work will explore cross-chunk consolidation and improved long-range geometric alignment to enhance global scene consistency.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the implementation of selected components of the proposed method and the translation of manuscript content into English. We also used these tools for literature search and summarization, code debugging, and language refinement.

Generative AI tools were not used to develop the research methodology, design experiments, generate or process datasets, formulate mathematical claims or proofs, create scientific figures, or interpret experimental results.

All AI-assisted work was reviewed by the authors, and the authors take full responsibility for the final content of this paper, including all text, scientific claims, code, and other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies the reconstruction of static 3D scenes from image sequences and does not involve human-subject experiments or the collection of new personal data. We use existing benchmark datasets and pretrained models for research purposes and follow their applicable terms of use. As with other scene-reconstruction systems, LonGaussian may reproduce identifiable or private visual content present in its inputs and could be misused for unauthorized reconstruction or surveillance. Its deployment should therefore respect privacy, consent, copyright, dataset licenses, and applicable regulations. The method may also inherit geographic and visual biases from its training data, and we do not evaluate it for safety-critical deployment.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the model components, two-stage training procedure, objectives, datasets, evaluation protocol, and principal implementation settings. The appendix and supplementary material provide additional architecture and optimization details, hyperparameters, data-processing procedures, sequence construction rules, baseline adaptations, and evaluation settings. The supplementary material also includes our code and side-by-side novel-view rendering comparison videos. Together, these materials are intended to support reproduction of the reported results.

REFERENCES

Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1050–1060. IEEE, 2025.

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

Ruiyang Chen, Feiran Li, Chu Zhou, Zonglin Li, Zhanyu Ma, and Heng Guo. Of^{3}gs: On-the-fly feed-forward 3d gaussian splatting from unposed images, 2026a. URL https://arxiv.org/abs/2606.03254.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. In *International Conference on Learning Representations*, volume 2026, pp. 50694–50718, 2026b.

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

Zhuoguang Chen, Minghui Qin, Tianyuan Yuan, Zhe Liu, and Hang Zhao. Long3r: Long sequence streaming 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5273–5284. IEEE, 2025.

Freeman Cheng, Botao Ye, Xueting Li, Junqi You, Fangneng Zhan, and Ming-Hsuan Yang. Recosplat: Autoregressive feed-forward gaussian splatting using render-and-compare. *arXiv preprint arXiv:2603.09968*, 2026.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10027–10038. IEEE, 2023.

Sven Elflein, Ruilong Li, Sérgio Agostinho, Zan Gojcic, Laura Leal-Taixé, Qunjie Zhou, and Aljosa Osep. VGG-T^3: Offline feed-forward 3d reconstruction at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36464–36474, June 2026.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 37:140138–140158, 2024.

Xin Fei, Wenzhao Zheng, Yueqi Duan, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Jiwen Lu. Pixelgaussian: Generalizable 3d gaussian reconstruction from arbitrary views. *arXiv preprint arXiv:2410.18979*, 2024.

Jiaxin Guo, Tongfan Guan, Wenzhen Dong, Wenzhao Zheng, Wenting Wang, Yue Wang, Yeung Yam, and Yun-Hui Liu. Salon3r: Structure-aware long-term feedforward 3d reconstruction from unposed images. *International Journal of Computer Vision*, 134(8):383, 2026.

Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. Pup 3d-gs: Principled uncertainty pruning for 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5949–5958. IEEE, 2025.

Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pp. 59–75. Springer, 2022.

Adam W. Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Suya You, Rares Ambrus, Katerina Fragkiadaki, and Leonidas Guibas. Alltracker: Efficient dense point tracking at high resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5253–5262, October 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.

Guichen Huang, Ruoyu Wang, Xiangjun Gao, Che Sun, Yuwei Wu, Shenghua Gao, and Yunde Jia. Longsplat: Online generalizable 3d gaussian splatting from long sequence images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4994–5002, 2026.

Ranran Huang and Krystian Mikolajczyk. No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27947–27957. IEEE, 2025.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics*, 44(6):1–16, 2025.

Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T Barron, Noah Snavely, and Aleksander Hołyński. Zipmap: Linear-time stateful 3d reconstruction via test-time training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21748–21759, 2026.

Gyeongjin Kang, Jisang Yoo, Jihyeon Park, Seungtae Nam, Hyeonsoo Im, Sangheon Shin, Sangpil Kim, and Eunbyung Park. Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22012–22022. IEEE, 2025.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pp. 18–35. Springer, 2024.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labeling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6013–6022. IEEE, 2025.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. MapAnything: Universal feed-forward metric 3D reconstruction. In *2026 International Conference on 3D Vision (3DV)*, pp. 499–509. IEEE, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. ISSN 0730-0301. doi: 10.1145/3592433. URL https://hal.science/hal-04088161.

Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Bo Dai, Shuai Yang, Chen Change Loy, and Xingang Pan. Stream3r: Scalable sequential 3d reconstruction with causal transformer. In *International Conference on Learning Representations*, volume 2026, pp. 42746–42768, 2026.

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21719–21728. IEEE, 2024.

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.

Yang Li, Jinglu Wang, Lei Chu, Xiao Li, Shiu-hong Kao, Ying-Cong Chen, and Yan Lu. Streamgs: Online generalizable gaussian splatting reconstruction for unposed image streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25841–25850. IEEE, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yirunib8l8.

Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. WorldMirror: Universal 3D world reconstruction with any-prior prompting. In *International Conference on Machine Learning*, 2026.

Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16651–16662. IEEE, 2025.

Sheng Miao, Jiaxin Huang, Dongfeng Bai, Xu Yan, Hongyu Zhou, Yue Wang, Bingbing Liu, Andreas Geiger, and Yiyi Liao. Evolsplat: Efficient volume-based gaussian splatting for urban view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11286–11296. IEEE, 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. *Communications of the ACM*, 65(1):99–106, 2021.

Arthur Moreau, Richard Shaw, Michal Nazarczuk, Jisu Shin, Thomas Tanay, Zhensong Zhang, Songcen Xu, and Eduardo Pérez-Pellitero. Off the grid: Detection of primitives for feed-forward 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11756–11766, 2026.

Jongmin Park, Minh-Quan Viet Bui, Juan Luis Gonzalez, Jaeho Moon, Jihyong Oh, and Munchurl Kim. Ecosplat: Efficiency-controllable feed-forward 3d gaussian splatting from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26010–26020, 2026.

Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024.

Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Learning compact 3d gaussians via feed-forward point fusion. In *2026 International Conference on 3D Vision (3DV)*, pp. 554–564. IEEE, 2026.

Changhao Song, Yuxuan Wang, Qibiao Li, Youcheng Cai, and Ligang Liu. Streamsplat: Streaming feed-forward 3d gaussian splatting. *arXiv preprint arXiv:2608.01659*, 2026.

Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5283–5293. IEEE, 2025.

Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pp. 78–89. IEEE, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21486–21499, 2026a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522. IEEE, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026b.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20310–20320, June 2024.

Yan Xia, Wenbo Ji, Weirong Chen, and Daniel Cremers. Csg-fusion: Consistent sparse-view gaussian splatting via matching-based fusion. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2653–2662. IEEE, 2025.

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20406–20417. IEEE, 2024.

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

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

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21924–21935. IEEE, 2025.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *International Conference on Learning Representations*, volume 2025, pp. 54009–54033, 2025.

Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. Yonosplat: You only need one model for feedforward 3d gaussian splatting. In *International Conference on Learning Representations*, volume 2026, pp. 39852–39871, 2026.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Chen Sun, Ming-Hsuan Yang, Forrester Cole, Trevor Darrell, and Deqing Sun. LoGeR: Long-context geometric reconstruction with hybrid memory. In *European Conference on Computer Vision*. Springer, 2026a. URL https://eccv.ecva.net/virtual/2026/poster/3286.

Shuai Zhang, Guanjun Wu, Zhoufeng Xie, Xinggang Wang, Bin Feng, and Wenyu Liu. Dynamic 2d gaussians: Geometrically accurate radiance fields for dynamic objects. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8144–8153, 2025.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William Freeman, and Hao Tan. Test-time training done right. In *International Conference on Learning Representations*, volume 2026, pp. 157604–157638, 2026b.

Xiaoxue Zhang, Xiaoxu Zheng, Yixuan Yin, Tiao Zhao, Kaihua Tang, Michael Bi Mi, Zhan Xu, and Dave Zhenyu Chen. Anchorsplat: Feed-forward 3d gaussian splatting with 3d geometric priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18924–18933, June 2026c.

A Architecture and Consolidation Details

A.1 Dense Gaussian Prediction Head

The Gaussian branch follows a DPT-style dense prediction design. It reassembles four intermediate decoder features at complementary spatial scales, fuses them from coarse to fine, and produces a 256-channel full-resolution feature map. Separate two-layer convolutional heads predict a 3D offset, anisotropic scale, unit-quaternion rotation, opacity, and degree-2 spherical-harmonic coefficients for every context pixel. The heads therefore predict 3, 3, 4, 1, and 27 values per pixel for these attributes, respectively. The opacity-head bias is initialized to -2, while the final offset layer is initialized close to zero, preventing opaque or strongly displaced primitives at the start of training.

A.2 Track Validation and Ownership

AllTracker is run independently in each LoGeR chunk with four recurrent inference iterations and a confidence threshold $\tau_{\text{trk}} = 0.1$. Tracked coordinates are rounded to the dense Gaussian grid. When multiple tracks map to the same frame–pixel location, only the observation with the highest opacity-weighted visibility is retained, so a dense prediction cannot be consumed by more than one representative. A track is eligible for consolidation when it contains at least two retained observations, has a mean opacity of at least $\tau_\alpha = 0.05$, and has a root-mean-square distance of its 3D

centers from their arithmetic mean of at most $\tau_{\text{geo}} = 0.3$. The threshold τ_{geo} is an absolute distance in the backbone reconstruction coordinates, not a calibrated distance in meters or a scene-normalized quantity. Intrinsics calibration preserves predicted depth, and SE(3) alignment does not change this scale. We additionally restrict observations to an adaptive neighborhood around the query-frame anchor. Its radius is

$$r_{\text{trk}} = \text{clip}(6d_{\text{1NN}}, 0.02, 0.5)\,, \tag{11}$$

where d_{1NN} is the median nearest-neighbor distance estimated from at most 5,000 points in the chunk. This neighborhood gate adapts to local point spacing within the clipping bounds; it does not make the separate absolute threshold τ_{geo} scale invariant. We keep τ_{geo} fixed across scenes and do not independently evaluate its sensitivity.

A.3 Learned Track Merger

For every accepted track, the base center, scale, and SH coefficients are opacity-weighted averages of its observations. The base opacity is obtained by alpha compositing, and the base rotation is computed by weighted quaternion averaging. The pooled descriptor defined in Section 3.3 has 69 dimensions and is processed by a two-layer MLP with hidden dimension 128, LayerNorm, and GELU activations. Four zero-initialized output layers predict bounded corrections:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_m &= \bar{\boldsymbol{\mu}}_m + 0.2\tanh(\Delta\boldsymbol{\mu}_m), \\ \hat{\mathbf{s}}_m &= \bar{\mathbf{s}}_m \odot \exp(0.5\tanh(\Delta\mathbf{s}_m))\,, \end{aligned} \tag{12}$$

while the opacity correction is applied in logit space and clipped to $[-4, 4]$, and the SH correction is scaled by $0.25\tanh(\cdot)$.

B Scene-Level Artifact Handling

B.1 Rendered-Depth Consistency Filtering

For each chunk, we render its Gaussian predictions in its contributing context views and use the resulting depth and opacity as local references. Gaussians originating from other chunks are projected into these views and compared with the median depth in a 3×3 neighborhood. A comparison is ignored when fewer than five neighbors have sufficient opacity or when their relative depth spread exceeds 0.15. Otherwise, the comparison contributes an occlusion vote when the projected Gaussian lies at least 1% in front of the reference surface and an agreement vote when the relative depth difference is at most 1%. A Gaussian is removed only if it receives at least one occlusion vote, occlusion votes constitute at least 30% of all reliable comparisons, and they also constitute at least 30% of the combined occlusion and agreement votes. The reference-opacity threshold is 0.5. A chunk never votes against its own predictions. These conservative ownership and support tests prevent a single uncertain depth sample from deleting locally consistent geometry.

B.2 Filtering-Threshold Selection

We select the reported filtering decision thresholds exclusively on a subset of the DL3DV training split, disjoint from DL3DV-140. Specifically, we randomly sample 20 training scenes and reconstruct each using 96 context and 13 held-out target views. We evaluate nine predefined joint threshold configurations rather than an exhaustive Cartesian grid. Candidate filters are replayed on the same cached, unfiltered reconstructions, so their differences arise only from thresholding rather than separate model training or reconstruction. We rank candidates by scene-equal mean target-view PSNR. The selected setting uses a 1% relative depth gap, at least one occlusion vote, a 30% reliable-view ratio, and a 30% decision ratio. It achieves the highest training-subset PSNR among the candidates, improves all 20 scenes over the matched no-filter reconstruction, and is fixed before evaluation on the test sets. The DL3DV-140 ablation in Table 3 evaluates this fixed filter and is not used for threshold selection. Across the tested joint configurations, filtering improves mean PSNR, SSIM, and LPIPS over the matched unfiltered reconstruction. Nearby candidates yield similar mean PSNR, suggesting that the benefit is not confined to the selected setting. This comparison varies depth gaps and voting criteria jointly; it does not isolate each parameter's effect or establish robustness outside the tested configurations.

Table 4: Rendered-depth threshold selection on 20 DL3DV training scenes. Setting tuples denote (relative gap, minimum votes, reliable-view ratio, decision ratio). Metrics are averaged equally across scenes; ΔPSNR is paired against the same reconstruction without filtering.

Setting	PSNR ↑	SSIM ↑	LPIPS ↓	ΔPSNR
$(0.10, 2, 0.50, 0.65)$	17.418	0.514	0.464	$+0.550$
$(0.05, 2, 0.50, 0.65)$	17.631	0.524	0.457	$+0.763$
$(0.05, 1, 0.50, 0.50)$	17.688	0.527	0.456	$+0.820$
$(0.05, 1, 0.40, 0.40)$	17.795	0.534	0.452	$+0.926$
$(0.02, 2, 0.50, 0.50)$	17.967	0.541	0.449	$+1.099$
$(0.02, 1, 0.40, 0.40)$	18.137	0.551	0.443	$+1.269$
$(0.02, 1, 0.30, 0.30)$	18.271	0.560	0.438	$+1.403$
$(0.01, 1, 0.40, 0.40)$	18.279	0.559	0.440	$+1.411$
Selected $(0.01, 1, 0.30, 0.30)$	**18.356**	**0.566**	**0.435**	$+1.488$
No filtering	16.868	0.484	0.487	—

B.3 Far-Field Gaussian Shell

We use the public `skyseg.onnx` checkpoint hosted by Jianyuan Wang.[1] Each RGB frame is resized to 320×320, normalized with ImageNet statistics, and processed by the fixed ONNX model. Its score map is min–max normalized per image to unsigned values in $[0, 255]$ and bilinearly resized to the input resolution; pixels with scores greater than 160 are classified as sky. The resulting mask first suppresses finite-depth Gaussian predictions. We define the shell center as the coordinate-wise median of the input camera centers and estimate the scene radius as the 90th percentile of the distances from scene Gaussians to this center. The shell radius is $r_{\text{sky}} = 50 r_{\text{scene}}$. Sky pixels are sampled with stride 2, converted to world-space rays, and quantized into 512×256 azimuth–elevation bins. Each occupied bin produces one isotropic Gaussian with opacity 0.98. Its direction and RGB color are averaged from the observations in the bin; RGB is represented by degree-zero SH, while the remaining coefficients are zero. The Gaussian scale equals the shell radius times the angular bin size, with an overlap factor of 1.0.

C Training Details

C.1 Two-Stage Optimization

All stages keep LoGeR and AllTracker frozen and use a global batch size of four. Stage I trains the Gaussian decoder and attribute heads for one complete pass over the 8,590 eligible scenes in the DL3DV training split after excluding all scenes that overlap with DL3DV-140; each sample is a dense 16-frame clip. Stage II initializes from Stage I and freezes the feature decoder. In Stage IIa, we additionally freeze the Gaussian attribute head and train only the learned merger on DL3DV for 900 optimizer steps. Stage IIb initializes from the Stage IIa checkpoint and performs a 300-step low-learning-rate adaptation with a balanced 1:1 schedule over 150 sequences from the official RE10K training split and 118 DL3DV training scenes; neither adaptation set contains an evaluation identity. In both substages, we sample uniformly from four sequence-length buckets: a 16-frame sparse sequence sampled at interval 8 and contiguous sequences of 64, 96, and 128 frames. During Stage IIb, the merger is updated in every step, whereas the attribute head is updated only for the 16-frame bucket; this corresponds to approximately 75 head updates under uniform sampling. At most three context views and three held-out target views are rendered in each update.

We use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and weight decay 0.01, without learning-rate scheduling, and clip the gradient norm at 1.0. Training uses four GPUs with one sequence per GPU. The Stage-I learning rate is 10^{-5}. In Stage IIa, the merger learning rate is 1.25×10^{-5}. Stage IIb retains this merger learning rate and uses 2×10^{-6} for the attribute head. The image-loss weights are $\lambda_{\text{rgb}} = 1$, $\lambda_{\text{lpips}} = 0.25$, $\lambda_{\text{ssim}} = 0.15$, and $\lambda_{\text{tgt}} = 1$. Stage I additionally uses $\lambda^{(1)}_{\text{aniso}} = 0.1$ with anisotropy threshold 4 and $\lambda^{(1)}_{\text{off}} = 0.01$. Because Stage II supervises only the consolidated rendering branch, $\lambda^{(2)}_{\text{aniso}} = \lambda^{(2)}_{\text{off}} = 0$. All experiments use random seed 0.

[1] https://huggingface.co/JianyuanWang/skyseg

For dense predictions indexed by $i = 1, \ldots, M$, with scale vector $\mathbf{s}_i$ and predicted center offset $\Delta\boldsymbol{\mu}_i$, the two regularizers are

$$\mathcal{L}_{\text{aniso}} = \frac{1}{M}\sum_{i=1}^{M} \max\left(0, \frac{\max_j s_{ij}}{\min_j s_{ij} + 10^{-6}} - \tau_{\text{aniso}}\right), \qquad \mathcal{L}_{\text{off}} = \frac{1}{M}\sum_{i=1}^{M} \|\Delta\boldsymbol{\mu}_i\|_2, \tag{13}$$

where $\tau_{\text{aniso}} = 4$. These terms discourage needle-like Gaussians and large displacements from the LoGeR point-map predictions, respectively.

Table 5: Principal architecture, training, and inference hyperparameters.

Hyperparameter	Value
Image resolution	280×490
LoGeR chunk length / overlap (L, O)	$(12, 3)$
Gaussian SH degree	2
Merger hidden dimension	128
AllTracker confidence threshold τ_{trk}	0.1
Minimum track length $n_{\min}$	2
Minimum mean opacity τ_α	0.05
Maximum RMS center deviation τ_{geo}	0.3
Head / merger learning rate	2×10^{-6} / 1.25×10^{-5}
Stage-I / Stage-II schedule	8,590 scenes (one pass) / 900+300 steps
AdamW weight decay	0.01
Context / target renders per update	at most 3 / 3
Depth gap / visibility ratio	0.01 / 0.30
Depth neighborhood / minimum support	1 / 5
Maximum local depth spread	0.15
Sky radius factor / sampling stride	50 / 2
Sky angular bins	512×256

D Evaluation Protocol and Baselines

D.1 Dataset Preparation and View Sampling

DL3DV uses its downsampled images, while RealEstate10K frames are center-cropped to the target aspect ratio and resized to 280×490 with corresponding updates to the intrinsics. For evaluation with C context views, every eighth frame in a contiguous source prefix is held out as a target. The prefix therefore contains $C + \lfloor C/7 \rfloor$ frames: the 64-, 96-, and 128-frame context settings contain 9, 13, and 18 target views, respectively. Scene lists and frame indices are fixed before evaluation and shared across methods whenever their interfaces permit.

D.2 Two-Pass Unposed Evaluation

Our unposed evaluation separates camera estimation from scene construction. The first pass processes the complete packed sequence to estimate a camera trajectory for both context and target views. The second pass reinitializes the TTT fast-weight states and SWA cache, receives only context images, and constructs the Gaussian scene with tracking, consolidation, filtering, and shell generation. The first-pass camera trajectory is aligned to the second-pass scene coordinate system using the shared context cameras, after which the aligned held-out target poses are used for rendering. Consequently, target images provide camera-pose observations in the first pass but never contribute Gaussian primitives, tracking correspondences, appearance attributes, depth-filter references, or sky-shell samples. This protocol evaluates renderable reconstruction from unposed context images without using target appearance in the reconstructed scene.

All LonGaussian experiments use SE(3), enable LoGeR's TTT and sliding-window attention, and calibrate local point maps with the dataset intrinsics. We use orientation-aware two-pass alignment and disable Sim(3) rescaling. PSNR, SSIM, and LPIPS-VGG are computed with the same implementation for all methods. The Gaussian count is measured after all scene construction and filtering steps specific to each method. "OOM" denotes failure to complete a method's native inference on a 24-GB GPU; we do not introduce method-specific chunking solely to obtain a result.

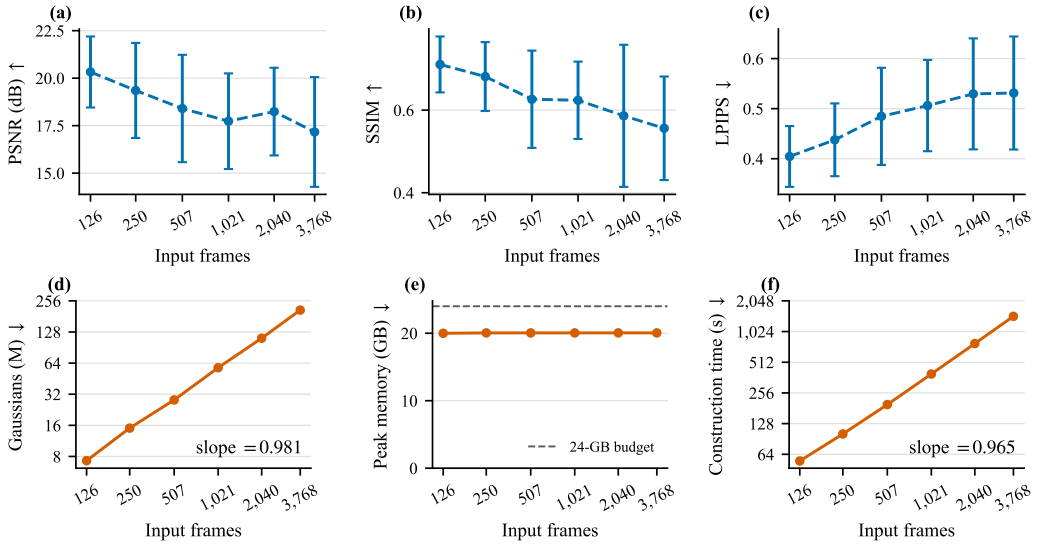


Figure 6: Single-scene long-sequence scaling. Local rendering quality remains usable as the sequence grows (a–c), while complete representation size and construction time grow approximately linearly and peak memory remains bounded (d–f). Error bars show variation across held-out views.

D.3 Baseline Configuration

For the baselines with numerical results in Table 1, target poses are predicted independently by each method rather than shared with LonGaussian or taken directly from the dataset. Both ReCoSplat checkpoints use a first pass over context and target images for pose estimation and a separate context-only pass for Gaussian construction. Their target translations are rescaled using the predicted context-camera scale ratio between passes, with a shared first-camera coordinate frame; the prescribed first-chunk scale matching is retained. OF3GS likewise predicts poses from context and target images, constructs Gaussians from context images, and applies its native context-based translation-scale matching before target rendering. These image metrics therefore reflect both scene reconstruction and each method's pose-estimation errors, and are not equivalent to evaluation with fixed ground-truth target poses. WorldMirror and AnySplat yield only OOM entries at the reported lengths.

WorldMirror (Liu et al., 2026) and AnySplat use their native joint-view inference. OF3GS uses its native streaming construction. For ReCoSplat, we report both the released low-resolution checkpoint and a locally adapted checkpoint trained at 280×490 on a single 24-GB GPU. Starting from the released weights, adaptation comprises 1,000 updates of the Gaussian decoder and output head, 2,000 updates that additionally train the recurrent-guidance projections, and 1,000 mixed-context updates over 32, 64, 96, and 128 frames. All three phases use a learning rate of 10^{-5} and approximately 975 DL3DV training scenes disjoint from DL3DV-140; the backbone and camera and point prediction paths remain frozen. The final checkpoint is selected using non-RE10K DL3DV development evidence before RE10K evaluation. This procedure preserves the released architecture and inference protocol but is a local resolution adaptation rather than an official ReCoSplat reproduction. All available methods use the same prepared scene and target-view lists.

D.4 Single-Scene Long-Sequence Scalability

The prefixes in Table 2 preserve complete chunk boundaries; the final five frames are included only in the qualitative 3,773-frame reconstruction in Figure 1. We process the sequence once at 280×490 with $L = 12$ and $O = 3$ and record these six streaming snapshots. Context frames follow the same one-in-eight held-out split as the main evaluation. The 15 target views per prefix are sampled uniformly from the held-out views and use the seven-chunk rendering scope defined in Section 4.3. Filtering is recomputed within each prefix, and the rendered representation contains the prefix-filtered compact component and dense complement with an 8% compact-depth surface-

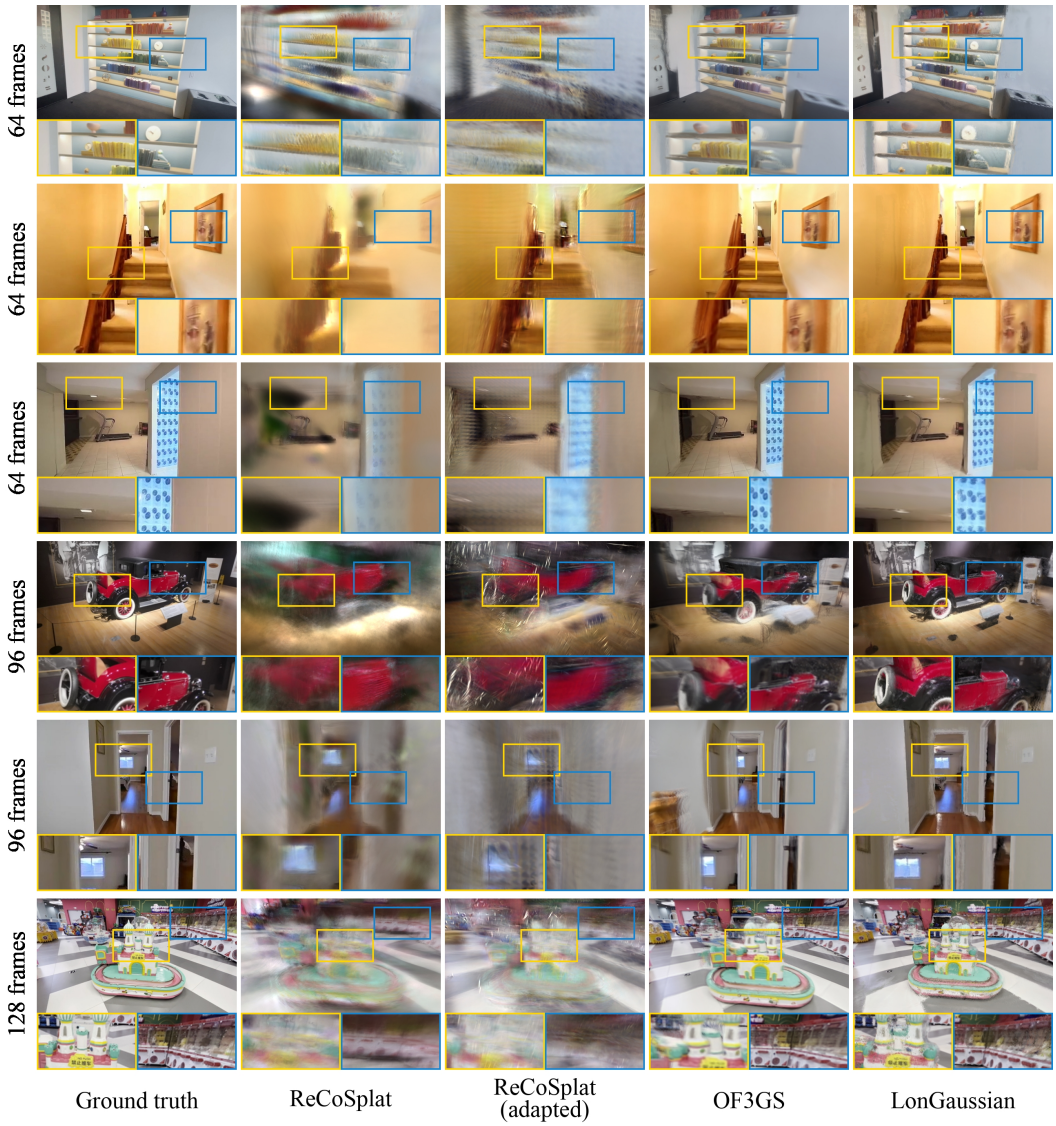
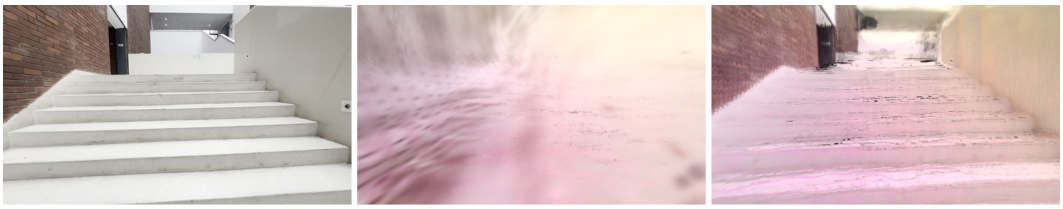


Figure 7: Additional novel-view comparisons on DL3DV and RealEstate10K. LonGaussian consistently preserves sharper structure and finer detail across scenes and sequence lengths.

culling margin. Because the snapshots share the full run's first-pass pose trajectory, this experiment measures scaling under a fixed trajectory rather than strictly causal prefix-only camera estimation.

Table 2 reports the combined quality and resource measurements. For its six prefixes, the context counts are 111, 219, 444, 894, 1,785, and 3,297, corresponding to 12, 24, 49, 99, 198, and 366 chunks. Construction time fits $T \propto N^{0.965}$ ($R^2 = 0.999$), while representation size follows $G \propto C^{0.981}$ ($R^2 = 1.000$). Time per input frame and Gaussian count per context view remain approximately stable. Figure 6 visualizes these trends and the target-view variation. These single-scene fits do not establish cross-scene scaling rates.

Because each longer prefix introduces new scene regions and different held-out views, this diagnostic combines increased content difficulty with local pose and alignment error and should not be interpreted as a controlled fixed-content degradation curve.



Ground truth | Global rendering | Seven-chunk rendering

Figure 8: Long-range revisit failure in the ultra-long sequence. Rendering the complete accumulated scene produces severe ghosting, whereas restricting the same target view to seven neighboring chunks recovers much of the scene structure. The improvement is consistent with interference from distant misregistered chunks; residual local artifacts remain.

E Additional Analysis and Failure Cases

The Gaussian counts in Table 1 include track representatives, uncovered dense predictions, and far-field shell primitives. Runtime also includes frozen tracking, two LoGeR passes, depth voting, and sky segmentation.

Tracks on moving objects may link observations that do not share a single 3D location, while textureless, reflective, or repeatedly occluded surfaces may fragment tracks and leave duplicated dense predictions. Geometric errors that are consistent within a LoGeR chunk can pass the track test. The depth filter is deliberately conservative but can still remove thin foreground structures when their local rendered depth is unstable, and an erroneous sky mask may replace finite scene content with shell Gaussians.