PERM4D: PERSISTENT MEMORY FOR STREAMING 4D GAUSSIAN SPLATTING FROM MONOCULAR VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming 4D reconstruction from monocular videos aims to reconstruct dynamic scenes using only past and current frames. Gaussian splatting provides an explicit representation for this task, but its incremental updates require reliable cross-frame association. Distinct regions may share motion, while reappearing content may lack recent correspondences, potentially causing incorrect updates or redundant initialization. To address these challenges, we introduce PerM4D, a persistent memory framework for streaming 4D Gaussian reconstruction from posed monocular videos. Our framework constructs per-frame Gaussians and integrates them through memory-guided association. Persistent static memory fuses static Gaussians across frames, while short-term motion memory stores recent trajectories and long-term surface memory preserves the identities, geometry, and appearance of dynamic Gaussians. Together, these memories support static accumulation, surface-validated dynamic updates, and the reuse of historical attributes when previously observed surfaces reappear. Experiments on the NVIDIA, DyCheck, and RealEstate10K datasets demonstrate competitive novel-view synthesis performance. Code will be released.

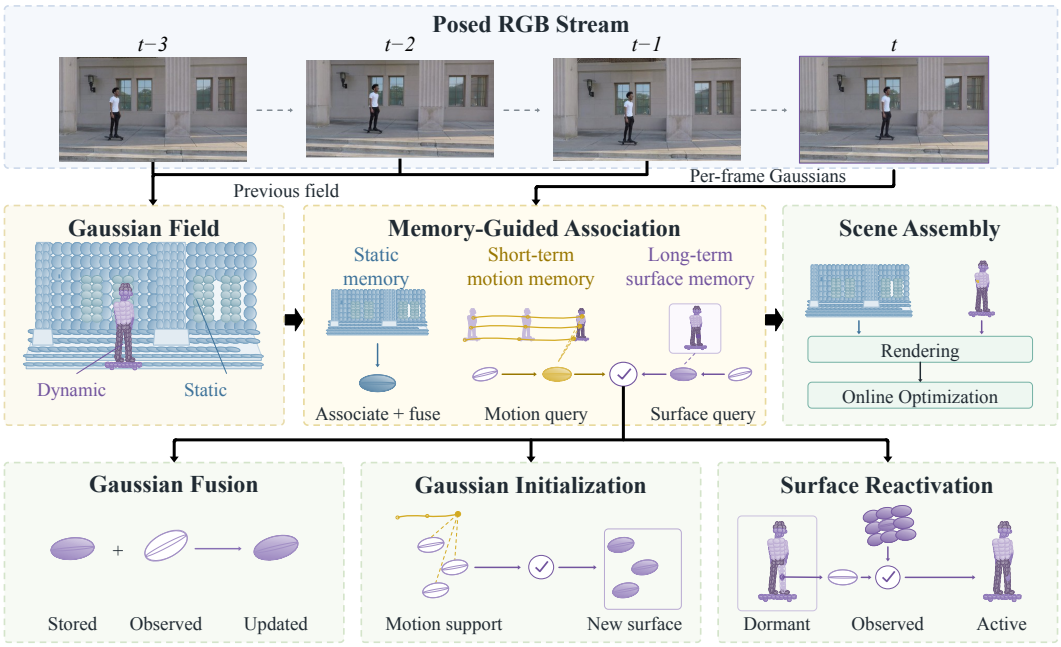

Figure 1: **Per-frame Gaussians from posed monocular videos update the previous field through memory-guided association.** Persistent static memory accumulates background geometry, while short-term motion and long-term surface memories jointly guide dynamic Gaussian fusion, initialization, and surface reactivation. The static field and active dynamic Gaussians supported by current observations are then assembled for rendering and online optimization.

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.

1 INTRODUCTION

Dynamic 3D reconstruction, also known as 4D reconstruction, is important for applications such as robotic manipulation (Huang et al., 2023; 2024a; Qian et al., 2025; Liu et al., 2026), augmented and virtual reality (AR/VR) (Furht, 2011; Gafni et al., 2021; Orts-Escolano et al., 2016), and autonomous driving (Sun et al., 2020; Peng et al., 2025; Chen et al., 2026; Wang et al., 2026). Specifically, streaming 4D reconstruction from video streams aims to reconstruct dynamic scenes using only past and current frames. It requires accumulating static geometry and updating dynamic content as new observations arrive, while accounting for motion, occlusion, and newly visible regions.

Recent Gaussian-based methods have advanced online dynamic scene reconstruction. For monocular video, GFlow (Wang et al., 2025c) alternates camera pose estimation and Gaussian optimization using depth and optical-flow priors, with pixel-wise densification for new content. DynOMo (Seidenschwarz et al., 2025) updates dynamic Gaussians for online point tracking using visual-feature reconstruction and feature-similarity-weighted 3D motion regularization. ProDyG (Chen et al., 2025) maintains separate static and dynamic Gaussian components and extends its motion scaffold and dynamic Gaussians as new frame batches arrive. StreamSplat (Wu et al., 2026) performs feed-forward reconstruction from uncalibrated video streams using bidirectional deformation and opacity-based adaptive fusion to propagate persistent Gaussians.

A key challenge in streaming reconstruction is to integrate incoming observations with accumulated scene information under motion and changing visibility. Repeated observations should refine the existing scene, while dynamic updates must account for both recent motion and previously reconstructed surfaces. Newly exposed regions may share motion with known content, and previously observed surfaces may reappear after recent correspondences are lost. These ambiguities affect whether historical information should be updated, recovered, or supplemented with new content. To address these challenges, our key idea is to fuse static Gaussians in persistent memory and associate dynamic observations using complementary short-term motion and long-term surface memories. Motion memory stores current Gaussian motion from recent trajectories, while surface memory preserves persistent identities and accumulated geometry and appearance beyond trajectory interruptions.

We introduce **PerM4D**, a persistent memory framework for streaming 4D Gaussian reconstruction from posed monocular videos. As illustrated in Figure 1, each incoming frame is converted into temporary static and dynamic Gaussians. Memory-based association guides dynamic Gaussian fusion, initialization, and dormant-surface reactivation, while observations that fail validation are deferred. The static field and active dynamic Gaussians supported by accepted current observations are then assembled for rendering and online refinement.

Persistent memory supports static accumulation, dynamic updates checked by motion and surface evidence, and the reuse of historical shape and appearance upon reappearance. These operations integrate observations into an evolving scene representation for novel-view synthesis. Comprehensive experiments on NVIDIA (Yoon et al., 2020), DyCheck (Gao et al., 2022), and RealEstate10K (Zhou et al., 2018) demonstrate the competitive novel view synthesis performance of **PerM4D**. In summary, our main contributions are as follows:

- We introduce **PerM4D**, a persistent memory framework that integrates incoming observations with accumulated static and dynamic scene information for streaming 4D reconstruction from posed monocular videos.
- We develop a memory-guided association mechanism that jointly evaluates matches from short-term motion memory and long-term surface memory to guide dynamic Gaussian fusion, initialization, and reactivation.
- We demonstrate competitive novel view synthesis performance on three datasets and validate the effectiveness of our persistent memory mechanism through ablation studies.

2 RELATED WORK

Dynamic 3D Reconstruction. Dynamic neural radiance fields represent scene evolution through canonical-space deformation (Pumarola et al., 2021; Park et al., 2021), scene flow (Li et al., 2021),

or factorized spatiotemporal representations (Cao & Johnson, 2023; Fridovich-Keil et al., 2023). Gaussian-based methods represent scene dynamics using deformation fields (Yang et al., 2024b; Wu et al., 2024), sparse control points (Huang et al., 2024b), or explicit spatiotemporal 4D Gaussian primitives (Yang et al., 2024a). For monocular videos, MoSca (Lei et al., 2025) combines motion scaffolds with Gaussian fusion, while Shape of Motion (Wang et al., 2025b) uses long-range trajectories and shared low-dimensional motion bases. SplineGS (Park et al., 2025) further models Gaussian trajectories with motion-adaptive splines.

Feed-Forward Reconstruction. Feed-forward methods predict geometry or renderable representations directly from images (Wang et al., 2025a). DUSt3R (Wang et al., 2024) predicts pointmaps, while pixelSplat (Charatan et al., 2024), Splatt3R (Smart et al., 2024) and MVSplat (Chen et al., 2024) predict Gaussians from sparse views. For dynamic scenes, MonST3R (Zhang et al., 2025) combines time-varying pointmap prediction with global alignment. BTimer (Liang et al., 2024) aggregates video context into a Gaussian scene at a target time, and DGS-LRM (Lin et al., 2025a) predicts deformable Gaussians. 4DGT (Xu et al., 2025) processes posed video in rolling windows, while MoVieS (Lin et al., 2026) jointly predicts Gaussian appearance, geometry, and motion.

Online Dynamic Reconstruction. Online methods reuse and update scene representations as observations arrive. For calibrated multi-view video, Dynamic 3D Gaussians (Luiten et al., 2024) optimizes persistent primitives per timestep, and 3DGStream (Sun et al., 2024) combines motion updates with new Gaussian insertion. With monocular input, GFlow (Wang et al., 2025c) alternates camera pose and Gaussian optimization using depth and optical-flow priors, with densification for newly visible content. DynOMo (Seidenschwarz et al., 2025) incorporates feature reconstruction and motion regularization for online point tracking. 4D-Fly (Wu et al., 2025) propagates Gaussians through motion anchors and expands a global 4D map. ProDyG (Chen et al., 2025) extends recently visible tracks and motion scaffolds, adding dynamic Gaussians in newly observed regions. StreamSplat (Wu et al., 2026) propagates persistent Gaussians from uncalibrated video streams through feed-forward bidirectional deformation and opacity-based soft fusion. In this context, PerM4D focuses on the acceptance of persistent updates: motion-selected targets are checked against surface records before fusion, while dormant records remain available for recovery after trajectory loss.

3 METHOD

Overview. As illustrated in Fig. 2, given a posed monocular RGB stream, **PerM4D** incrementally updates a persistent Gaussian representation using only past and current frames. Each incoming frame is converted into temporary static and dynamic Gaussians from estimated depth and dynamic masks (Section 3.1). Static Gaussians are fused into persistent static memory, while dynamic Gaussians are updated under the joint guidance of short-term motion memory and long-term surface memory (Sections 3.2 and 3.3). Finally, the static field and accepted active dynamic Gaussians are assembled and refined through online optimization (Section 3.4).

3.1 PER-FRAME GAUSSIAN CONSTRUCTION

Inputs and Dynamic Mask Estimation. At time t, the inputs are RGB I_t, intrinsics K_t, and world-to-camera pose $\mathbf{P}_t$. To separate scene motion from camera motion, we compare observed forward flow $F^{\text{obs}}_{t-1\to t}$ (e.g., RAFT (Teed & Deng, 2020)) with pose-induced rigid flow $F^{\text{rig}}_{t-1\to t}$. Reliable residual-flow endpoints prompt a video segmentation model (e.g., SAM2 (Ravi et al., 2025)) in current-image coordinates, yielding the dynamic mask Y_t.

Per-Frame Gaussian Initialization. A frozen depth predictor (e.g., Depth Anything 3 (Lin et al., 2025b)) supplies D_t. We back-project pixels using $(D_t, K_t, \mathbf{P}_t)$ and split the world-space points by Y_t. Cross-frame geometric comparisons assume depth and pose translations share a consistent scale; errors in either can corrupt association. Static points receive nearest-neighbor scale estimates and are aggregated within world-space voxels. To separate dynamic regions with different motion or depth, we estimate surface normals and group points by motion node, image-space cell, and log-depth bin. A motion node supplies local support for grouping observations and attaching Gaussians; a trajectory records a tracked point over time. Neither is a persistent surface record. Each group aggregates its points' geometry, appearance, confidence, and motion support into a temporary Gaussian.

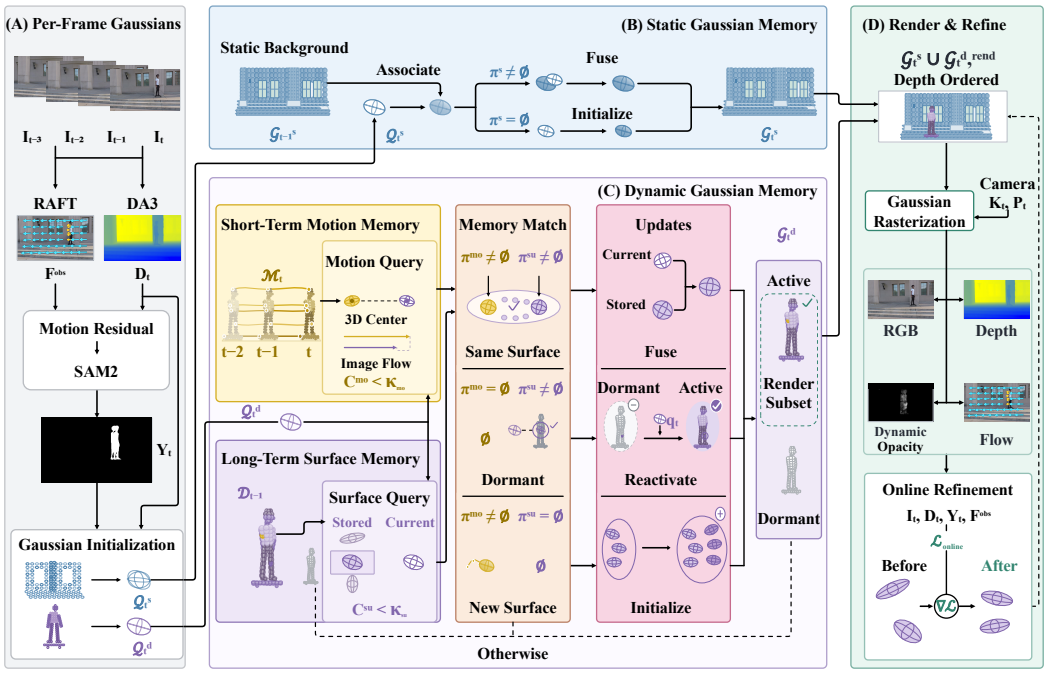

Figure 2: **Overview of PerM4D.** (A) Posed RGB, depth, and dynamic masks produce temporary static and dynamic Gaussians (Section 3.1). (B) Static memory accumulates geometry and appearance across views (Section 3.2). (C) Motion proposals and surface-record checks guide dynamic fusion, new-content initialization, and historical attribute reuse through reactivation (Section 3.3). (D) The static field and individually accepted active dynamic Gaussians undergo rendering and online refinement (Section 3.4).

Let $\mathcal{Q}_t^a = \{q_j^{a,t}\}_{j=1}^{M_t^a}$ denote temporary Gaussians and $\mathcal{G}_{t-1}^a = \{g_i^{a,t-1}\}_{i=1}^{N_{t-1}^a}$ the pre-update persistent field, for static/dynamic branch $a \in \{s, d\}$. Both use the parameterization $(\boldsymbol{\mu}, \mathbf{R}, \mathbf{s}, \tau, \mathbf{c})$: a center $\boldsymbol{\mu} \in \mathbb{R}^3$, orientation $\mathbf{R} \in \mathrm{SO}(3)$, positive principal-axis scales $\mathbf{s}$, renderer opacity parameter τ, and color $\mathbf{c}$. The covariance is $\boldsymbol{\Sigma} = \mathbf{R}\,\mathrm{diag}(\mathbf{s}^2)\mathbf{R}^\top$. Dynamic memory additionally retains observation support and an optical-thickness mass, converted to τ for rendering (Appendix A). We initialize from the first two frames and produce the first reconstruction at $t = 2$; subsequent updates use only the available input prefix (Appendix B.1).

3.2 Persistent Static Gaussian Memory

Persistent static memory integrates repeated observations into accumulated geometry and appearance. Each observation in $\mathcal{Q}_t^s$ is associated with at most one Gaussian in $\mathcal{G}_{t-1}^s$. Several observations may refine the same stored Gaussian; unmatched observations initialize new content.

Static Gaussian Association. For temporary Gaussian $q_j^{s,t}$, the K_s nearest persistent centers define the candidate set $\mathcal{N}_j^s$. Candidate i has cost

$$C_{ji}^s = \lambda_\mu \left\|\boldsymbol{\mu}_j^t - \boldsymbol{\mu}_i^{t-1}\right\|^2_{\left(\boldsymbol{\Sigma}_j^t + \boldsymbol{\Sigma}_i^{t-1} + \epsilon \mathbf{I}\right)^{-1}} + \lambda_R \left\|\mathrm{Log}\left(\mathbf{R}_i^{t-1\top} \mathbf{R}_j^t\right)\right\|_2^2 + \lambda_c \left\|\mathbf{c}_j^t - \mathbf{c}_i^{t-1}\right\|_1 + \lambda_s \left\|\log \mathbf{s}_j^t - \log \mathbf{s}_i^{t-1}\right\|_1, \qquad i \in \mathcal{N}_j^s. \tag{1}$$

Center distance locates nearby geometry; orientation, color, and scale assess its compatibility with the observation. Here, $\|\mathbf{v}\|_\mathbf{A}^2 = \mathbf{v}^\top \mathbf{A} \mathbf{v}$, Log is the rotation-vector logarithm on $\mathrm{SO}(3)$, and $\epsilon > 0$ stabilizes the covariance inverse. For nonempty $\mathcal{N}_j^s$, let $i_j^\star = \arg\min_{i \in \mathcal{N}_j^s} C_{ji}^s$. We assign $\pi_s(j) = i_j^\star$ if $C_{ji_j^\star}^s < \kappa_s$; an empty candidate set or a failed threshold test gives $\pi_s(j) = \varnothing$.

Static Field Update. For each persistent Gaussian, we collect its matched current observations as $\mathcal{Q}_{i,t}^{s,\mathrm{match}} = \{q_j^{s,t} \in \mathcal{Q}_t^s \mid \pi_s(j) = i\}$. The static field is then updated by

$$\mathcal{G}_t^s = \left\{ \mathrm{Update}_s \left(g_i^{s,t-1}, \mathcal{Q}_{i,t}^{s,\mathrm{match}} \right) \right\}_{i=1}^{N_{t-1}^s} \cup \left\{ \mathrm{Init}_s(q_j^{s,t}) \;\middle|\; q_j^{s,t} \in \mathcal{Q}_t^s,\ \pi_s(j) = \varnothing \right\}. \tag{2}$$

Update_s integrates assigned observations with stored geometry and appearance and retains unmatched Gaussians. The resulting field carries information from earlier viewpoints into scene assembly; Init_s adds newly observed content. Support statistics are refreshed, with low-support Gaussians pruned when the static memory budget is exceeded.

3.3 Persistent Dynamic Gaussian Memory

Complementary Memory Roles. Dynamic memory uses recent motion to propose Gaussian updates and accumulated surface attributes to check record agreement. Shared motion does not establish surface identity, and trajectory loss does not imply new content. Gaussians receive numerical updates; surface records group them for persistent association. We use short-term motion memory $\mathcal{M}_{t-1} = \{\mathcal{T}_\ell\}_\ell$ and a long-term surface memory $\mathcal{D}_{t-1} = \{d_\iota^{t-1}\}_\iota$. Each $\mathcal{T}_\ell$ stores a recent 3D trajectory and its image-plane coordinates. Surface memory retains committed records $d_\iota^{t-1} = (\iota, \mathcal{G}_\iota^{d,t-1}, z_\iota^{t-1})$, where ι is a non-reused record ID, $\mathcal{G}_\iota^{d,t-1} \subseteq \mathcal{G}_{t-1}^d$ is the set of persistent dynamic Gaussians assigned to the record, and $z_\iota^{t-1} \in \{\mathsf{active}, \mathsf{dormant}\}$ is its lifecycle state. A record groups nearby Gaussians initialized together around a motion node, representing a local surface hypothesis rather than a semantic object. Each Gaussian belongs to exactly one record, with $\mathrm{sid}_{t-1}(i) = \iota$ for $g_i^{d,t-1} \in \mathcal{G}_\iota^{d,t-1}$.

Short-Term Motion Memory. Reliable optical-flow correspondences extend trajectories; their back-projected endpoints provide current world-space positions. For Gaussian $g_i^{d,t-1}$, $\mathcal{N}_{i,\mathrm{mo}}^t$ indexes trajectories observed in both frames and spatially near $\boldsymbol{\mu}_i^{t-1}$ at $t-1$. As in Flow4DGS-SLAM (Wang & Lee, 2026), flow guides propagation; here, surface memory additionally checks the proposal. Local trajectory displacements predict the center and image displacement:

$$\widehat{\boldsymbol{\mu}}_i^t = \boldsymbol{\mu}_i^{t-1} + \sum_{\ell \in \mathcal{N}_{i,\mathrm{mo}}^t} \eta_{i\ell}^t \left(\mathbf{x}_\ell^t - \mathbf{x}_\ell^{t-1} \right), \quad \widehat{\mathbf{m}}_i^t = \sum_{\ell \in \mathcal{N}_{i,\mathrm{mo}}^t} \eta_{i\ell}^t \left(\mathbf{p}_\ell^t - \mathbf{p}_\ell^{t-1} \right). \tag{3}$$

Here, $\mathbf{x}_\ell^r$ and $\mathbf{p}_\ell^r$ are trajectory ℓ's world-space and image-plane positions at frame r. Without valid consecutive-frame support, a Gaussian retains its stored position but is excluded from motion matching. The interpolation weights satisfy $\eta_{i\ell}^t \geq 0$ and $\sum_{\ell \in \mathcal{N}_{i,\mathrm{mo}}^t} \eta_{i\ell}^t = 1$. For a temporary Gaussian $q_j^{d,t} \in \mathcal{Q}_t^d$, the motion compatibility cost is

$$C_{ji}^{\mathrm{mo}} = \lambda_\mu^{\mathrm{mo}} \left\| \boldsymbol{\mu}_j^t - \widehat{\boldsymbol{\mu}}_i^t \right\|_2^2 + \lambda_m^{\mathrm{mo}} \left\| \mathbf{m}_j^t - \widehat{\mathbf{m}}_i^t \right\|_1, \qquad i \in \mathcal{E}_{j,\mathrm{mo}}^t. \tag{4}$$

Here, $\mathcal{E}_{j,\mathrm{mo}}^t$ indexes trajectory-supported candidates, and $\mathbf{m}_j^t$ is the observation's inter-frame image displacement. The predicted center is also the prior for accepted fusion. We set $\pi_j^{\mathrm{mo}} = i$ when i is the lowest-cost candidate in $\mathcal{E}_{j,\mathrm{mo}}^t$ and $C_{ji}^{\mathrm{mo}} < \kappa_{\mathrm{mo}}$; otherwise, $\pi_j^{\mathrm{mo}} = \varnothing$.

Long-Term Surface Memory. For observation $q_j^{d,t}$, $\mathcal{E}_{j,\mathrm{su}}^t$ indexes nonempty active or dormant records, including those without recent trajectory support. Each record is scored by its smallest Gaussian attribute discrepancy:

$$\begin{aligned} C_{j\iota}^{\mathrm{su}} = \min_{g_i^{d,t-1} \in \mathcal{G}_\iota^{d,t-1}} \Big[& \lambda_R^{\mathrm{su}} \left\| \mathrm{Log}\left(\mathbf{R}_i^{t-1\top} \mathbf{R}_j^t \right) \right\|_2^2 \\ & + \lambda_s^{\mathrm{su}} \left\| \log \mathbf{s}_j^t - \log \mathbf{s}_i^{t-1} \right\|_1 + \lambda_c^{\mathrm{su}} \left\| \mathbf{c}_j^t - \mathbf{c}_i^{t-1} \right\|_1 \Big], \qquad \iota \in \mathcal{E}_{j,\mathrm{su}}^t. \end{aligned} \tag{5}$$

Excluding position keeps stored attributes available for retrieval when centers become outdated. The same score checks ongoing motion proposals and retrieves dormant records. Its minimum requires one compatible record member, which may differ from the motion-selected target. Similar attributes can make retrieval ambiguous; large rotations or inconsistent full orientation frames, not determined by normals alone, can disrupt it. We set $\pi_j^{\mathrm{su}} = i$ when the lowest-cost record $\iota \in \mathcal{E}_{j,\mathrm{su}}^t$ satisfies $C_{j\iota}^{\mathrm{su}} < \kappa_{\mathrm{su}}$, where i is the Gaussian index attaining its within-record minimum; otherwise, $\pi_j^{\mathrm{su}} = \varnothing$.

Dynamic Gaussian Field Update. Motion and surface evidence determine how current observations enrich the persistent dynamic field. Both memories are queried before any update; reactivation validation and competing-proposal resolution precede acceptance (Appendix B.3). The resulting update policy is:

- Fuse: both matches identify the same record, $\mathrm{sid}_{t-1}(\pi_j^{\mathrm{mo}}) = \mathrm{sid}_{t-1}(\pi_j^{\mathrm{su}}) = \iota$. The motion match selects the update target; the surface match checks record membership.
- Reactivate: only the surface match exists, its record is dormant, and $C_{j\iota}^{\mathrm{su}} < \kappa_{\mathrm{re}}$. Reliable current geometry and compatible current motion-node support are also required. The surface match selects the target.
- Initialize: a motion match exists without a surface match, or both are absent and temporal validation succeeds. Only the latter requires consecutive dynamic observations with locally consistent motion; at $t = 2$, validation uses the first two frames (Appendix B.2).
- Otherwise: the observation is not committed to the field; its motion evidence may be retained for later validation.

Let $\mathcal{Q}_{t,\iota}^{d,\mathrm{upd}}$ collect accepted fusion/reactivation observations for record ι, and $\mathcal{Q}_t^{d,\mathrm{init}}$ collect accepted births. The update is

$$\mathcal{G}_t^d = \bigcup_{d_\iota^{t-1} \in \mathcal{D}_{t-1}} \mathrm{Update}_d\Big(\mathcal{G}_\iota^{d,t-1}, \mathcal{Q}_{t,\iota}^{d,\mathrm{upd}}\Big) \cup \Big\{ \mathrm{Init}_d(q_j^{d,t}) \;\Big|\; q_j^{d,t} \in \mathcal{Q}_t^{d,\mathrm{init}} \Big\}. \tag{6}$$

Update_d preserves identifiers and record membership. Fusion uses the motion-predicted center as a prior for reliability-weighted geometric updates from motion-compatible observations. Reactivation recovers the surface-selected Gaussian's center from current observations instead of averaging with its outdated position, restores its record, and refreshes its motion attachment; old trajectories need not survive. Both actions retain accumulated shape and appearance through reliability-weighted fusion with accepted current attributes (Appendix B.3). Init_d creates Gaussians and records without modifying the motion-matched historical Gaussian (Appendix B.2).

For records without accepted observations, $\mathrm{Update}_d(\mathcal{G}_\iota^{d,t-1}, \varnothing) = \mathcal{G}_\iota^{d,t-1}$ preserves their Gaussians for later retrieval. Accepted observations refresh motion attachments and surface records; stale trajectories expire and persistently unsupported records become dormant.

3.4 Online Gaussian Assembly and Optimization

Static–Dynamic Field Assembly and Rendering. We assemble the pre-refinement scene as $\widetilde{\mathcal{G}}_t = \mathcal{G}_t^s \cup \mathcal{G}_t^{d,\mathrm{rend}}$, where $\mathcal{G}_t^{d,\mathrm{rend}} \subseteq \mathcal{G}_t^d$ contains individually supported Gaussians: each must have an accepted current observation and belong to an active record. Unsupported members remain stored but are not rendered, even in active records. This avoids unsupported historical geometry but can omit surfaces hidden in the source view and visible from novel views. Joint depth-ordered alpha compositing produces RGB $\widehat{I}_t$, opacity-normalized depth $\widehat{D}_t$, and visible dynamic opacity $\widehat{A}_{d,t}$. Image-plane motions from $\mathcal{M}_t$ supply rendered dynamic flow $\widehat{F}_{t-1\to t}^d$ (Appendix B).

Finite-Step Online Optimization. **PerM4D** performs 15 gradient steps on the visible Gaussians $\widetilde{\mathcal{G}}_t^{\mathrm{vis}} \subseteq \widetilde{\mathcal{G}}_t$ of the assembled scene. Associations, identifiers, record memberships, lifecycle states, and trajectories remain fixed during refinement. We minimize

$$\begin{aligned}\mathcal{L}_{\mathrm{online}} = {} & \rho_I\Big(\widehat{I}_t - I_t\Big) + \lambda_D \rho_D\Big(\log \widehat{D}_t - \log D_t\Big) + \lambda_M \,\mathrm{BCE}\Big(\widehat{A}_{d,t}, Y_t\Big) \\ & + \lambda_F \rho_F\Big(\widehat{F}_{t-1\to t}^d - F_{t-1\to t}^{\mathrm{obs}}\Big) + \lambda_{\mathrm{mem}} \sum_{g \in \widetilde{\mathcal{G}}_t^{\mathrm{vis}}} \omega_g \, d_{\mathrm{G}}\Big(\boldsymbol{\theta}_g, \boldsymbol{\theta}_g^{(0)}\Big).\end{aligned} \tag{7}$$

The data terms fit RGB, depth, dynamic opacity, and flow, using residual penalties ρ_I, ρ_D, ρ_F and binary cross-entropy (BCE). The memory term anchors refinement to the state formed by accepted updates, penalizing attribute drift without reconsidering discrete matches. Here, $\boldsymbol{\theta}_g$ contains the optimized attributes, $\boldsymbol{\theta}_g^{(0)}$ their fixed post-association reference, ω_g a fixed support weight, and d_{G} an attribute discrepancy. Refined attributes are written back to the same Gaussian identities in $\mathcal{G}_t^s$ or $\mathcal{G}_t^d$, yielding the next frame's persistent state.

Table 1: **Main results on the NVIDIA, DyCheck, and RealEstate10K datasets. Stream** indicates whether the evaluated method uses only past and current frames to generate an incremental reconstruction, and **Time** reports the average processing time per frame in seconds. The best, second-best, and third-best results are highlighted.

Method	Stream	NVIDIA (Yoon et al., 2020)			DyCheck (Gao et al., 2022)			RealEstate10K (Zhou et al., 2018)			Time ↓
		PSNR ↑	SSIM ↑	LPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	
4DGT (Xu et al., 2025)	×	17.28	0.402	0.577	11.26	0.606	0.472	20.38	0.720	0.265	**0.13s**
MoVieS (Lin et al., 2026)	×	19.16	0.514	0.315	13.96	0.664	**0.337**	26.98	0.817	0.111	0.15s
StreamSplat (Wu et al., 2026)	✓	14.23	0.314	0.656	11.31	0.456	0.714	18.68	0.474	0.394	0.38s
Flow4DGS-SLAM (Wang & Lee, 2026)	×	21.32	0.648	0.399	13.79	0.646	0.408	29.33	0.865	0.130	0.87s
4DGS-SLAM (Li et al., 2025)	×	18.11	0.557	0.592	13.85	**0.673**	0.415	21.04	0.684	0.199	80.9s
DynOMo (Seidenschwarz et al., 2025)	✓	21.96	0.669	0.301	12.88	0.604	0.452	26.23	0.857	0.214	34.1s
Ours	✓	**23.69**	**0.743**	**0.281**	**14.27**	0.653	0.404	**29.76**	**0.929**	**0.104**	0.49s

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets. We evaluate **PerM4D** on three real-world video datasets: RealEstate10K (Zhou et al., 2018), NVIDIA (Yoon et al., 2020), and DyCheck (Gao et al., 2022). RealEstate10K is used to evaluate novel view synthesis in static scenes, while NVIDIA and DyCheck are used for dynamic-scene reconstruction. For DyCheck, we follow the data sampling setup of MoVieS (Lin et al., 2026), sampling 13 input frames at a resolution of 518×518 and using views from the other two cameras for evaluation. For NVIDIA, 12 input views at 379×672 resolution are sampled in a round-robin style, where the i-th frame comes from the i-th camera at timestep i (Gao et al., 2021). For RealEstate10K, we evaluate novel-view synthesis using 12 distinct posed input frames and three predefined held-out target frames, with all images center-cropped to 256×256 and identical input and target frame selections across all methods.

Evaluation Settings. We follow prior works (Wu et al., 2026; Lin et al., 2026) and report peak signal-to-noise ratio (PSNR) (Huynh-Thu & Ghanbari, 2008), structural similarity index (SSIM) (Wang et al., 2004), and learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018). For DyCheck (Gao et al., 2022), we adapt its covisibility-mask computation procedure to the streaming setting. At each evaluation time t, we recompute the covisibility mask using only the source frames available up to t as support, rather than the full input sequence. The resulting streaming covisibility mask selects target-view pixels satisfying the covisibility criterion with respect to this causal input prefix. We report the corresponding masked metrics, mPSNR, mSSIM, and mLPIPS, and apply the same masks to all compared methods. We conduct all experiments on a single NVIDIA H100 GPU. For all methods, the reported average processing time is measured end-to-end; for PerM4D, this includes RAFT, DA3, and SAM2. We render the Gaussian representation using the differentiable, tile-based 3D Gaussian splatting rasterizer (Kerbl et al., 2023), and optimize the source-view reconstruction loss with AdamW (Loshchilov & Hutter, 2017).

Baseline Methods. To evaluate the effectiveness of **PerM4D**, we compare against representative methods from three categories. (i) *Non-streaming feed-forward 4D reconstruction methods*, such as MoVieS (Lin et al., 2026) and 4DGT (Xu et al., 2025), directly predict dynamic Gaussian representations from input video clips. (ii) *Dynamic SLAM methods*, such as Flow4DGS-SLAM (Wang & Lee, 2026) and 4DGS-SLAM (Li et al., 2025), jointly estimate camera poses and reconstruct static and dynamic scene components through online optimization. We categorize these methods as non-streaming because their pipelines include sequence-end refinement. (iii) *Streaming 4D reconstruction methods*, such as StreamSplat (Wu et al., 2026) and DynOMo (Seidenschwarz et al., 2025), incrementally update Gaussian representations as new frames arrive. For a fair comparison, all compared methods use ground-truth camera poses.

4.2 NOVEL VIEW SYNTHESIS

Dynamic Scene View Synthesis. We compare **PerM4D** with representative open-source methods for 4D Gaussian reconstruction on NVIDIA (Yoon et al., 2020) and DyCheck (Gao et al., 2022)



Figure 3: **Qualitative novel-view synthesis on the NVIDIA dataset.**

datasets. As shown in Table 1, **PerM4D** achieves the best results across all three image-quality metrics on NVIDIA, reaching 23.69 dB PSNR, 0.743 SSIM, and 0.281 LPIPS. Compared with DynOMo, the strongest baseline on this dataset across the three reported metrics, our method improves PSNR by 1.73 dB and SSIM by 0.074, while reducing LPIPS by 0.020. These improvements extend beyond pixel-wise fidelity to structural and perceptual similarity. On DyCheck, **PerM4D** achieves the highest mPSNR of 14.27 dB, exceeding the next-best method, MoVieS, by 0.31 dB, while ranking third in mSSIM and second in mLPIPS overall. The lower mSSIM and mLPIPS rankings indicate that the gain in pixel-wise fidelity does not translate into the best structural or perceptual scores.

Among the evaluated streaming methods, it achieves the best scores across all three masked metrics. Across both dynamic datasets, **PerM4D** outperforms StreamSplat, Flow4DGS-SLAM, and DynOMo on all reported image-quality metrics. The average per-frame processing time reported in Table 1 is 0.49 s, compared with 0.87 s for Flow4DGS-SLAM and 34.1 s for DynOMo. Thus, relative to these two baselines, our persistent-memory framework improves reconstruction quality while reducing processing time. Qualitative comparisons in Figures 3 and 4 further illustrate these differences. On NVIDIA, **PerM4D** preserves more complete foreground silhouettes with less fragmentation than Flow4DGS-SLAM and DynOMo in the umbrella and balloon examples. In the last example, our method renders a single coherent foreground subject, whereas MoVieS produces multiple displaced copies. On DyCheck, we focus on regions selected by the streaming covisibility mask. In the first example, **PerM4D** better preserves the manipulated object's outline and local structure than Flow4DGS-SLAM, with fewer scattered artifacts than DynOMo. In the last example, it renders the pinwheel with clearer blade boundaries and less ghosting than MoVieS.

Static Scene View Synthesis. We further evaluate **PerM4D** on RealEstate10K (Zhou et al., 2018) to assess its performance in static scenes. As shown in Table 1, our method achieves the best PSNR, SSIM, and LPIPS among the compared methods, reaching 29.76 dB, 0.929, and 0.104, respectively. Compared with Flow4DGS-SLAM, it improves PSNR by 0.43 dB and SSIM by 0.064, while also achieving a lower LPIPS than MoVieS (0.104 vs. 0.111). These results support the effectiveness of our persistent-memory framework in static novel-view synthesis, where repeated observations are fused into an accumulated Gaussian representation. This indicates that persistent accumulation improves reconstruction quality even without dynamic-scene association, supporting its broader role in maintaining cross-view scene consistency. Qualitative comparisons in Figures 5–7 in Appendix D further illustrate these differences. In the indoor examples, **PerM4D** preserves cleaner wall surfaces and clearer window and door boundaries, with fewer scattered artifacts than DynOMo.

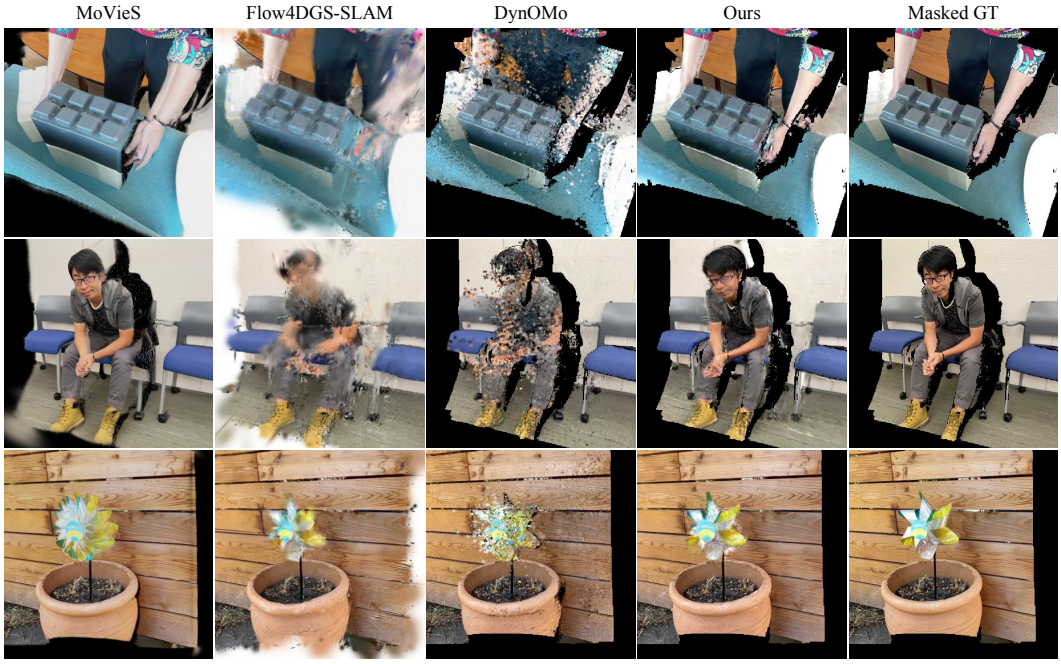

Figure 4: **Qualitative novel-view synthesis on the DyCheck dataset.**

4.3 Ablation Studies

We evaluate four variants: (i) *w/o surface memory*, using only short-term motion-based association; (ii) *w/o motion memory*, using only long-term surface-based association; (iii) *w/o static memory*, using only current-frame static Gaussians; (iv) *w/o reactivation*, disabling dormant-record reactivation while retaining both dynamic memory branches. As shown in Table 2, all ablations degrade PSNR, SSIM, and LPIPS, indicating that these components improve not only pixel-wise fidelity but also structural and perceptual quality. Removing static memory causes the largest PSNR drop (7.22 dB), highlighting the importance of cross-frame static accumulation. Removing surface memory and motion memory reduces PSNR by 4.46 dB and 2.24 dB, respectively, supporting the complementary roles of historical surface evidence and motion-guided association. The larger degradation without surface memory suggests that long-term surface evidence provides constraints that cannot be fully replaced by short-term motion correspondence alone. Disabling reactivation further reduces PSNR by 1.56 dB and increases LPIPS from 0.281 to 0.301, demonstrating the additional benefit of reusing dormant surface records. Notably, *w/o reactivation* still achieves 2.90 dB higher PSNR than *w/o surface memory*. This suggests that surface memory contributes beyond dormant-surface reuse, consistent with its role in validating motion-based associations during ongoing updates.

Table 2: Ablation study on the NVIDIA dataset.

Variant	PSNR ↑	SSIM ↑	LPIPS ↓
w/o surface memory	19.23	0.614	0.341
w/o motion memory	21.45	0.632	0.325
w/o static memory	16.47	0.604	0.430
w/o reactivation	22.13	0.712	0.301
Ours	**23.69**	**0.743**	**0.281**

5 Conclusion

We presented **PerM4D**, a persistent-memory framework for streaming 4D Gaussian reconstruction from posed monocular videos. Our method constructs per-frame Gaussians, accumulates static geometry in persistent memory, and combines short-term motion memory with long-term surface memory to update dynamic Gaussians, followed by scene assembly and online refinement. Comprehensive experiments demonstrate competitive novel-view synthesis performance. Nevertheless, our framework still requires camera poses as input, leaving reliable pose estimation under challenging streaming conditions to future work. Further acceleration is also needed to reduce per-frame processing latency. Limitations and future work are discussed in the supplementary material.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript editing, literature retrieval, and research-related feedback. For manuscript preparation, these tools assisted with translating and polishing author-written text, improving grammar and clarity, and LaTeX formatting. For literature retrieval, they helped identify relevant studies and summarize their methods and findings. For research-related feedback, we used these tools to obtain comments on research methodology and mathematical formulations, including the clarity of methodological assumptions, the consistency of notation, and the correspondence between equations and their accompanying explanations. We also used them to discuss possible interpretations of quantitative comparisons, qualitative observations, and ablation results.

We have not used generative AI tools to draft sections of the paper, generate synthetic datasets, or prove mathematical claims.

We reviewed all AI-assisted work. Specifically, we verified bibliographic information and literature summaries against the original sources, examined mathematical formulations for correctness and consistency with the methodological descriptions, and checked numerical comparisons and interpretations against the reported experimental results. We also reviewed AI-assisted translations and revisions to ensure that they preserved the intended technical meaning. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI tools.

ETHICS STATEMENT

Our experiments use three publicly available datasets: NVIDIA (Yoon et al., 2020), DyCheck (Gao et al., 2022), and RealEstate10K (Zhou et al., 2018). These datasets are used to evaluate scene reconstruction and novel-view synthesis. We use these datasets in accordance with their respective licenses and terms of use. This paper does not involve collecting new data from human participants, and our experiments do not seek to identify individuals or infer sensitive personal attributes. To the best of our knowledge, our use of these datasets in this paper does not introduce additional ethical concerns.

REPRODUCIBILITY STATEMENT

We describe the reconstruction pipeline and online optimization in Section 3, and the datasets, evaluation settings, and computational setup in Section 4.1. Appendix A specifies the notation and representation, while Appendix B provides additional implementation details, including causal initialization, new-surface admission, dynamic Gaussian updates, rendering eligibility, and the streaming evaluation protocol. We include a subset of our source code in the supplementary material. The full source code will be made publicly available upon publication.

REFERENCES

Ang Cao and Justin Johnson. HexPlane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 130–141, 2023.

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

Shi Chen, Erik Sandström, Sandro Lombardi, Siyuan Li, and Martin R Oswald. ProDyG: Progressive dynamic scene reconstruction via Gaussian splatting from monocular videos. In *Advances in Neural Information Processing Systems*, volume 38, pp. 88054–88074, 2025.

Xiaoxue Chen, Ziyi Xiong, Yuantao Chen, Gen Li, Nan Wang, Hongcheng Luo, Long Chen, Haiyang Sun, Bing Wang, Guang Chen, et al. DGGT: Feedforward 4D reconstruction of dynamic driving scenes using unposed images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1265–1276, 2026.

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 *Computer Vision – ECCV 2024*, volume 15079 of *Lecture Notes in Computer Science*, pp. 370–386. Springer, 2024.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-Planes: Explicit radiance fields in space, time, and appearance. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12479–12488. IEEE, 2023.

Borko Furht (ed.). *Handbook of augmented reality*. Springer Science & Business Media, 2011.

Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8649–8658, 2021.

Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5692–5701. IEEE, 2021.

Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *Advances in Neural Information Processing Systems*, volume 35, pp. 33768–33780, 2022.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3D value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024a.

Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. SC-GS: Sparse-controlled Gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4220–4230, 2024b.

Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

Jiahui Lei, Yijia Weng, Adam W Harley, Leonidas Guibas, and Kostas Daniilidis. MoSca: Dynamic Gaussian fusion from casual videos via 4D motion scaffolds. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6165–6177. IEEE, 2025.

Yanyan Li, Youxu Fang, Zunjie Zhu, Kunyi Li, Yong Ding, and Federico Tombari. 4D Gaussian splatting SLAM. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25019–25028. IEEE, 2025.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6498–6508, 2021.

Hanxue Liang, Jiawei Ren, Ashkan Mirzaei, Antonio Torralba, Ziwei Liu, Igor Gilitschenski, Sanja Fidler, Cengiz Oztireli, Huan Ling, Zan Gojcic, and Jiahui Huang. Feed-forward bullet-time reconstruction of dynamic scenes from monocular videos. *arXiv preprint arXiv:2412.03526*, 2024.

Chenguo Lin, Yuchen Lin, Panwang Pan, Yifan Yu, Tao Hu, Honglei Yan, Katerina Fragkiadaki, and Yadong Mu. MoVieS: Motion-aware 4D dynamic view synthesis in one second. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–306, 2026.

Chieh Hubert Lin, Zhaoyang Lv, Songyin Wu, Zhen Xu, Thu Nguyen-Phuoc, Hung-Yu Tseng, Julian Straub, Numair Khan, Lei Xiao, Ming-Hsuan Yang, Yuheng Ren, Richard Newcombe, Zhao Dong, and Zhengqin Li. DGS-LRM: Real-time deformable 3D Gaussian reconstruction from monocular videos. In *Advances in Neural Information Processing Systems*, volume 38, 2025a.

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

Zeyi Liu, Shuang Li, Eric Cousineau, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Geometry-aware 4D video generation for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 140751–140773, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pp. 800–809. IEEE, 2024.

Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. Holoportation: Virtual 3D teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp. 741–754, 2016.

Jongmin Park, Minh-Quan Viet Bui, Juan Luis Gonzalez Bello, Jaeho Moon, Jihyong Oh, and Munchurl Kim. SplineGS: Robust motion-adaptive spline for real-time dynamic 3D Gaussians from monocular video. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26866–26875. IEEE, 2025.

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5865–5874, 2021.

Chensheng Peng, Chengwei Zhang, Yixiao Wang, Chenfeng Xu, Yichen Xie, Wenzhao Zheng, Kurt Keutzer, Masayoshi Tomizuka, and Wei Zhan. DeSiRe-GS: 4D street Gaussians for static-dynamic decomposition and surface reconstruction for urban driving scenes. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6782–6791. IEEE, 2025.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10318–10327, 2021.

Zezhong Qian, Xiaowei Chi, Yuming Li, Shizun Wang, Zhiyuan Qin, Xiaozhu Ju, Sirui Han, and Shanghang Zhang. WristWorld: Generating wrist-views via 4D world models for robotic manipulation. *arXiv preprint arXiv:2510.07313*, 2025.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Jenny Seidenschwarz, Qunjie Zhou, Bardienus P Duisterhof, Deva Ramanan, and Laura Leal-Taixé. DynOMo: Online point tracking by dynamic online monocular Gaussian reconstruction. In *2025 International Conference on 3D Vision (3DV)*, pp. 1012–1021. IEEE, 2025.

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.

Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 3DGStream: On-the-fly training of 3D Gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20675–20685. IEEE, 2024.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo Open Dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2451. IEEE, 2020.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pp. 402–419. Springer, 2020.

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

Lening Wang, Wenzhao Zheng, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, and Jiwen Lu. OccSora: 4D occupancy generation models as world simulators for autonomous driving. *IEEE Transactions on Image Processing*, 2026.

Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of Motion: 4D reconstruction from a single video. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9660–9672. IEEE, 2025b.

Shizun Wang, Xingyi Yang, Qiuhong Shen, Zhenxiang Jiang, and Xinchao Wang. GFlow: Recovering 4D world from monocular video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7862–7870, 2025c.

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

Yunsong Wang and Gim Hee Lee. Flow4DGS-SLAM: Optical flow-guided 4D Gaussian splatting SLAM. *arXiv preprint arXiv:2604.22339*, 2026.

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.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, Yue Qian, Xiaohang Zhan, and Yueqi Duan. 4D-Fly: Fast 4D reconstruction from a single monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16663–16673, 2025.

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, 2024.

Zike Wu, Qi Yan, Xuanyu Yi, Lele Wang, and Renjie Liao. StreamSplat: Towards online dynamic 3D reconstruction from uncalibrated video streams. In *International Conference on Learning Representations*, 2026.

Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4DGT: Learning a 4D Gaussian transformer using real-world monocular videos. *arXiv preprint arXiv:2506.08015*, 2025.

Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4D Gaussian splatting. In *International Conference on Learning Representations*, volume 2024, pp. 9142–9159, 2024a.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20331–20341, 2024b.

Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5335–5344. IEEE, 2020.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations*, 2025.

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

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Appendices

The appendices are organized as follows:

- **Appendix A: Notation and Representation.** Gaussian fields, persistent identifiers, and parameter conventions.
- **Appendix B: Implementation Details.** Scene initialization, dynamic Gaussian updates, rendering, and streaming evaluation.
- **Appendix C: Limitations.** Dependence on externally provided camera poses, per-frame processing latency, and reconstruction challenges under complex motion.
- **Appendix D: Additional Experimental Results.** Qualitative novel-view synthesis comparisons on RealEstate10K.

A Notation and Representation

Gaussian Fields. We retain the notation introduced in the main text. The temporary static and dynamic Gaussians constructed at time t are $q_j^{s,t} \in \mathcal{Q}_t^s$ and $q_j^{d,t} \in \mathcal{Q}_t^d$, respectively, and their persistent fields are denoted by $\mathcal{G}_t^s$ and $\mathcal{G}_t^d$. The dynamic field contains all committed dynamic Gaussians retained in surface memory. Its current-frame renderable subset, $\mathcal{G}_t^{d,\mathrm{rend}} \subseteq \mathcal{G}_t^d$, is selected at the Gaussian level as specified in Appendix B.4.

Persistent Identifiers. Each persistent dynamic Gaussian $g_i^{d,t}$ occupies an individually addressable row of the dynamic atlas and belongs to exactly one surface record,

$$d_\iota^t = (\iota, \mathcal{G}_\iota^{d,t}, z_\iota^t) \in \mathcal{D}_t,$$

where $\mathcal{G}_\iota^{d,t} \subseteq \mathcal{G}_t^d$ and $\mathrm{sid}_t(i) = \iota$. Thus, i identifies an individual Gaussian, whereas ι identifies a surface record that may contain multiple Gaussians. The corresponding implementation fields are `surface_id` for the Gaussian identifier and `surface_record_id` for the shared record identifier. The mapping sid_t associates each Gaussian identifier with its surface-record identifier.

Parameter Conventions. Dynamic Gaussians retain the full orientation $\mathbf{R}_i^t \in \mathrm{SO}(3)$ used in the main-text representation. In the dynamic-atlas update equations, $\zeta_i^t \geq 0$ and $\zeta_j^t \geq 0$ denote the optical-thickness masses of persistent and temporary dynamic Gaussians, respectively. These quantities are distinct from the renderer opacity parameter τ. When a persistent Gaussian is prepared for rendering, ζ_i^t is converted to that opacity parameter.

Motion Memory and Lifecycle States. The short-term motion memory is $\mathcal{M}_t = \{\mathcal{T}_\ell\}_\ell$. Throughout the appendix, j indexes a temporary Gaussian, i a persistent Gaussian, ι a surface record, and ℓ a short-term trajectory. The main-text *active* category comprises the implementation's emerging and persistent surface states, whereas *dormant* denotes retained surfaces in the vanishing state.

B Implementation Details

This section describes scene initialization, temporal validation of new surfaces, dynamic Gaussian updates, renderable-subset selection, static–dynamic compositing, and the streaming evaluation protocol.

B.1 Causal Field Initialization

Scene Initialization. We initialize the scene representation from the first two input frames ($B = 2$), using their inter-frame correspondences for motion estimation and static–dynamic separation. The initial reconstruction represents the scene at $t = 2$; subsequent frames update the representation incrementally using only observations available up to the current time. Before initialization, all persistent fields and memories are empty:

$$\mathcal{G}_0^s = \mathcal{G}_0^d = \mathcal{M}_0 = \mathcal{D}_0 = \varnothing. \tag{8}$$

At $t = 1$, we cache I_1, D_1, K_1, and $\mathbf{P}_1$ without constructing a scene output. Inter-frame motion is not yet available, so no motion-derived mask is assigned to this frame. In particular, the absence of motion evidence is not sufficient to commit its pixels to the static field.

At $t = 2$, we estimate $F^{\text{obs}}_{1\to 2}$, the pose-induced rigid flow, and the current dynamic mask Y_2 following Section 3.1. We construct $\mathcal{Q}^s_2$ and $\mathcal{Q}^d_2$ using the per-frame Gaussian construction procedure and initialize short-term trajectories from reliable correspondences between the two frames. This initialization uses only the first two frames and their depth maps and camera parameters.

Short-Term Motion Memory. Short-term trajectories are initialized from reliable image correspondences independently of persistent Gaussian and surface identifiers. Given a valid trajectory observation at frame $r-1$, we propagate its image position with optical flow and back-project the resulting position into world coordinates:

$$\begin{aligned} \mathbf{p}^r_\ell &= \mathbf{p}^{r-1}_\ell + F^{\text{obs}}_{r-1\to r}\big(\mathbf{p}^{r-1}_\ell\big)\,, \\ \mathbf{x}^r_\ell &= \left[\mathbf{P}^{-1}_r \begin{bmatrix} D_r(\mathbf{p}^r_\ell) K_r^{-1} \widetilde{\mathbf{p}}^r_\ell \\ 1 \end{bmatrix}\right]_{1:3}, \qquad \widetilde{\mathbf{p}}^r_\ell = \begin{bmatrix} (\mathbf{p}^r_\ell)^\top & 1 \end{bmatrix}^\top. \end{aligned} \tag{9}$$

Here, $[\cdot]_{1:3}$ extracts the Euclidean coordinates of a homogeneous world-space point. A correspondence is retained only if both image endpoints are valid, both depths are finite and positive, and the correspondence passes the reliability checks of the main method.

At time t, the retained trajectory state is

$$\mathcal{T}_\ell = \left\{ (r, \mathbf{p}^r_\ell, \mathbf{x}^r_\ell) \;\middle|\; r \in \mathcal{O}^t_\ell \right\}, \qquad \mathcal{O}^t_\ell \subseteq \{\max(1, t - L_{\text{mo}} + 1), \ldots, t\}\,, \tag{10}$$

where L_{mo} denotes the short-term memory length. A trajectory with a single valid observation can support current-frame grouping, but cannot provide an inter-frame displacement.

Trajectories whose observations have not been assigned a persistent surface identity remain tentative. They provide temporal evidence for new-surface validation in Appendix B.2, but are excluded from the motion-prediction neighborhoods $\mathcal{N}^t_{i,\text{mo}}$ of existing persistent Gaussians. They become eligible for regular motion prediction only after their supporting observation is committed to a surface record.

Static Field Initialization. The static field is initialized with the association and fusion rule used for subsequent static updates:

$$\mathcal{G}^s_1 = \varnothing, \qquad \mathcal{G}^s_r = \text{Fuse}_s\left(\mathcal{G}^s_{r-1}, \mathcal{Q}^s_r\right), \qquad 2 \le r \le B. \tag{11}$$

For an empty previous field, every valid Gaussian in $\mathcal{Q}^s_r$ is unmatched and is inserted through the initialization branch of Equation 2. The association cost is unchanged during scene initialization.

Dynamic Field Initialization. Before the first persistent dynamic update, the initial trajectories are stored in $\mathcal{M}_B$, while $\mathcal{G}^d_{B-1} = \mathcal{D}_{B-1} = \varnothing$. Both historical assignments are therefore empty:

$$\pi^{\text{mo}}_j = \pi^{\text{su}}_j = \varnothing, \qquad q^{d,B}_j \in \mathcal{Q}^d_B. \tag{12}$$

At $B = 2$, new-surface validation uses the initialization case in Appendix B.2: reliable two-frame correspondences, current dynamic-mask membership in Y_2, and local motion consistency. With both assignments empty, Equation 19 admits an observation exactly when $b^2_j = 1$. Each accepted observation initializes a persistent Gaussian through Equation 20 and is assigned to an active surface record. It is eligible for rendering at $t = 2$ under the selection rule in Appendix B.4. If no observation passes validation, the dynamic field and long-term surface memory remain empty.

The initialized static field and accepted dynamic Gaussians are then assembled and refined to obtain the reconstruction at $t = 2$. For $t > 2$, observations unmatched by both memories are subject to the regular K_{birth}-frame validation window in Appendix B.2. Subsequent updates do not revise previously generated reconstructions.

B.2 Initialization of Newly Appearing Surfaces

Unmatched Observations. The motion and surface assignments, π^{mo}_j and π^{su}_j, query only previously committed Gaussians. A newly appearing surface may therefore be unmatched by both

memories. We collect these observations in

$$\mathcal{U}_t^d = \left\{ q_j^{d,t} \in \mathcal{Q}_t^d \;\middle|\; \pi_j^{\mathrm{mo}} = \varnothing,\ \pi_j^{\mathrm{su}} = \varnothing \right\}. \tag{13}$$

An observation in $\mathcal{U}_t^d$ remains tentative until sufficient temporal evidence is available. Persistent Gaussian and surface identifiers are allocated only after validation.

Temporal Support. For $t > 2$, an observation in $\mathcal{U}_t^d$ requires $K_{\mathrm{birth}} \geq 2$ consecutive dynamic observations, with $L_{\mathrm{mo}} \geq K_{\mathrm{birth}}$. Scene initialization at $t = 2$ instead uses correspondences from the first two frames together with the current dynamic mask Y_2. For $t \geq 2$, the validation window is

$$a_t = \begin{cases} 1, & t = 2, \\ t - K_{\mathrm{birth}} + 1, & t > 2, \end{cases} \qquad \mathcal{W}_t = \{a_t, \ldots, t\}. \tag{14}$$

Let Ω_j^t denote the current-frame pixels aggregated into $q_j^{d,t}$ during per-frame Gaussian construction. For $t = 2$ or $a_t \geq 2$, its supporting trajectories are

$$\mathcal{A}_j^t = \left\{ \ell \;\middle|\; \begin{array}{ll} \mathbf{p}_\ell^t \in \Omega_j^t, & \mathcal{W}_t \subseteq \mathcal{O}_\ell^t, \\ Y_r(\mathbf{p}_\ell^r) = 1 & \text{for every } r \in \{\max(2, a_t), \ldots, t\} \end{array} \right\}. \tag{15}$$

For $t > 2$ with $a_t < 2$, we set $\mathcal{A}_j^t = \varnothing$. All supporting correspondences satisfy the endpoint, depth-validity, and reliability requirements in Appendix B.1.

At initialization, a supporting trajectory must have valid observations in both frames, but dynamic-mask membership is evaluated only at $t = 2$ because no first-frame dynamic mask is assumed. For subsequent validation, a trajectory must remain in dynamic regions throughout the complete K_{birth}-frame window.

Local Motion Consistency. For each supporting trajectory $\ell \in \mathcal{A}_j^t$, the world-space displacement between consecutive frames is

$$\boldsymbol{\delta}_\ell^r = \mathbf{x}_\ell^r - \mathbf{x}_\ell^{r-1}. \tag{16}$$

When $|\mathcal{A}_j^t| \geq n_{\mathrm{birth}}$, we estimate the dominant local displacement separately for each frame pair and measure the residual motion dispersion:

$$\begin{aligned} \overline{\boldsymbol{\delta}}_j^r &= \mathrm{median}_{\ell \in \mathcal{A}_j^t}\, \boldsymbol{\delta}_\ell^r, \\ e_j^t &= \max_{r = a_t + 1, \ldots, t} \frac{\mathrm{median}_{\ell \in \mathcal{A}_j^t} \left\| \boldsymbol{\delta}_\ell^r - \overline{\boldsymbol{\delta}}_j^r \right\|_2}{\mathrm{median}_{\ell \in \mathcal{A}_j^t}\, D_r(\mathbf{p}_\ell^r) + \epsilon_D}. \end{aligned} \tag{17}$$

The vector median is evaluated component-wise. The depth normalization makes the residual approximately invariant to a common scene scale, up to the stabilizing constant $\epsilon_D > 0$. If $|\mathcal{A}_j^t| < n_{\mathrm{birth}}$, we set $e_j^t = +\infty$ without evaluating the medians. The test requires locally coherent motion within each frame pair, but does not assume zero motion or constant velocity over the window. At initialization, $a_2 = 1$, so only the frame pair $1 \to 2$ is evaluated.

The resulting birth-validation indicator is

$$b_j^t = \mathbf{1} \begin{bmatrix} q_j^{d,t} \in \mathcal{U}_t^d, & \big(t = 2 \ \vee \ a_t \geq 2\big), \\ |\mathcal{A}_j^t| \geq n_{\mathrm{birth}}, & e_j^t \leq \epsilon_{\mathrm{birth}} \end{bmatrix}, \tag{18}$$

where $n_{\mathrm{birth}} \geq 2$ is the minimum number of supporting trajectories and $\epsilon_{\mathrm{birth}} > 0$ is the motion-consistency threshold. Observations that fail any condition have $b_j^t = 0$. Because validation uses observation trajectories rather than matches to persistent Gaussians, it remains applicable when both historical candidate sets are empty. In particular, it permits validated dynamic initialization at $t = 2$ without a pre-existing Gaussian or a dynamic mask for frame 1.

Complete Initialization Condition. The two initialization cases in the main text are expressed jointly as

$$\pi_j^{\mathrm{su}} = \varnothing \quad \wedge \quad \big(\pi_j^{\mathrm{mo}} \neq \varnothing \ \vee \ b_j^t = 1\big). \tag{19}$$

An observation is initialized either when it has a motion match but no compatible surface match, or when both matches are absent and temporal validation succeeds. These cases are disjoint because $b_j^t = 1$ requires $q_j^{d,t} \in \mathcal{U}_t^d$. Equation 19 specifies the initialization branch of Equation 6; the fusion and reactivation conditions are unchanged.

Persistent Surface Creation. Each observation satisfying Equation 19 is assigned a new, non-reused Gaussian identifier ν_j^t. The initialized observations are partitioned into local groups $\{\mathcal{B}_\beta^t\}_\beta$: nearby observations initialized at the same frame around the same primary motion node share a group. We allocate a distinct surface-record identifier ι_β^t to each group, allowing multiple Gaussians to share one record:

$$\begin{aligned} \nu_j^t &= \mathrm{NewGID}(), & \iota_\beta^t &= \mathrm{NewSID}(), \\ g_{\nu_j^t}^{d,t} &\leftarrow q_j^{d,t}, & \mathrm{sid}_t(\nu_j^t) &= \iota_\beta^t, \qquad j \in \mathcal{B}_\beta^t, \\ \mathcal{G}_{\iota_\beta^t}^{d,t} &= \{g_{\nu_j^t}^{d,t} \mid j \in \mathcal{B}_\beta^t\}, & d_{\iota_\beta^t}^t &= (\iota_\beta^t, \mathcal{G}_{\iota_\beta^t}^{d,t}, \mathsf{active}). \end{aligned} \tag{20}$$

The new Gaussian inherits its orientation, color, linear scales, and optical-thickness mass from the current observation. Denoting the observation support by $S_j^{\mathrm{obs},t}$ and its primary motion-node identifier by o_j^t, the initial state is

$$\begin{aligned} \overline{\boldsymbol{\mu}}_{\nu_j^t} &= \boldsymbol{\mu}_{\nu_j^t}^t = \boldsymbol{\mu}_j^t, & \mathbf{v}_{\nu_j^t}^t &= \mathbf{0}, \\ \mathbf{R}_{\nu_j^t}^t &= \mathbf{R}_j^t, & \mathbf{s}_{\nu_j^t}^t &= \mathbf{s}_j^t, \\ \zeta_{\nu_j^t}^t &= \zeta_j^t, & \mathbf{c}_{\nu_j^t}^t &= \mathbf{c}_j^t, \\ N_{\nu_j^t}^t &= 1, & S_{\nu_j^t}^t &= S_j^{\mathrm{obs},t}, \\ t_{\nu_j^t}^{\mathrm{birth}} &= t_{\nu_j^t}^{\mathrm{last}} = t, & o_{\nu_j^t} &= o_j^t. \end{aligned} \tag{21}$$

Here, $\overline{\boldsymbol{\mu}}$ denotes the canonical position, $\mathbf{v}$ the motion velocity, N the observation count, and S the accumulated support. Each Gaussian starts in the implementation's emerging state. Its position variance is initialized from the current depth or observation uncertainty; member offsets, source keys, and skinning weights are inherited from the observation.

The new record is appended to $\mathcal{D}_t$, and its member Gaussians enter $\mathcal{G}_t^d$ through the initialization branch of Equation 6 according to Equation 19. Initialization uses the current per-frame Gaussians rather than replaying earlier ones. For a temporally validated observation, the supporting trajectories in $\mathcal{A}_j^t$ remain available in $\mathcal{M}_t$ and become eligible for regular motion prediction after commitment, as described in Appendix B.1. For a motion-supported initialization, the new Gaussian inherits the current observation's local motion support. The historical Gaussian selected by π_j^{mo} is neither updated nor reassigned to the new surface record.

Deferral and Causality. An observation unmatched by both memories with $b_j^t = 0$ is not added to the persistent field. Its tentative trajectories may continue to accumulate evidence within the L_{mo}-frame memory window. Invalid or interrupted correspondences break the consecutive support window, and expired tentative trajectories are removed.

Nonempty motion and surface matches to different records constitute a cross-memory conflict and are ineligible for birth validation. Subsequent observations can trigger fusion or reactivation only when the corresponding acceptance criteria are met. A surface-only match to an active record remains tentative, while a match to a dormant record must satisfy the stricter reactivation threshold.

Equations 15–18 depend exclusively on observations available by time t. Temporal validation can defer the initialization of newly appearing content, but introduces neither future-frame access nor retrospective changes to earlier reconstructions.

B.3 Gaussian-Level Dynamic Updates

Action Validation. Although Equation 6 organizes the persistent field by surface record, numerical updates are applied to individual Gaussians. Assignments are computed from the pre-update memories, and accepted updates precede online optimization. We denote the action assigned to $q_j^{d,t}$ by A_j^t. Two nonempty assignments to the same surface record specify fusion; assignments to different records specify a conflict. A surface-only match is a reactivation candidate only when the record is dormant and $C_{j\iota}^{\mathrm{su}} < \kappa_{\mathrm{re}}$; the reported configuration uses $\kappa_{\mathrm{re}} = 0.10$. Initialization follows

Equation 19. An active surface-only match or a dormant match that fails the reactivation threshold remains tentative.

Reactivation also requires a finite current Gaussian center, observation reliability $\rho_j^t > \rho_{\min}$, and a valid primary motion node o_j^t. Candidates that fail these geometric checks remain tentative and do not update the persistent Gaussian.

When multiple geometrically valid candidates propose to reactivate Gaussian i, we select the primary motion node of the candidate with the highest reliability, breaking ties by observation index. Let $o_i^{\mathrm{re},t}$ denote the selected node. Only candidates associated with this node are accepted for reactivation. Competing proposals targeting the same Gaussian through a different primary node, including fusion proposals, are deferred at the current step.

These checks precede the finalization of accepted observations and association responsibilities. Each accepted target has positive responsibility. The finalized actions—fuse, reactivate, initialize, conflict, and tentative—are mutually exclusive. Motion-node reassignment does not alter the current-frame matches, which are determined from the pre-update memories.

Update Targets and Responsibilities. For updates to existing Gaussians, the target index and its associated observation set are

$$i_j^t = \begin{cases} \pi_j^{\mathrm{mo}}, & \mathsf{A}_j^t = \mathsf{fuse}, \\ \pi_j^{\mathrm{su}}, & \mathsf{A}_j^t = \mathsf{reactivate}, \\ \varnothing, & \text{otherwise.} \end{cases} \qquad \mathcal{J}_i^t = \{j \mid i_j^t = i\}. \tag{22}$$

Fusion updates the motion-selected Gaussian; the surface-selected Gaussian verifies surface-record membership and need not have the same index. Reactivation instead updates the surface-selected Gaussian. Both operations preserve the target Gaussian's identifier and surface-record membership.

Let $r_{ji}^t \in [0,1]$ denote the association responsibility of observation j for Gaussian i, with zero responsibility outside its accepted target. In the usual single-target setting, the accepted responsibility is one. Multiple accepted observations of the same Gaussian are combined by the weighted sums below. Initialized observations follow Equation 20 and do not update their motion-matched historical Gaussians.

Gaussian Attributes and Motion Prior. Each persistent Gaussian stores its center $\boldsymbol{\mu}_i$, orientation $\mathbf{R}_i \in \mathrm{SO}(3)$, principal-axis scales $\mathbf{s}_i$, color $\mathbf{c}_i$, and nonnegative optical-thickness mass ζ_i, together with the observation count N_i, accumulated support S_i, and motion-owner identifier o_i. The quantity ζ_i corresponds to the implementation field `opacity_mass`, rather than the projected alpha used for compositing. A temporary Gaussian carries the corresponding $\mathbf{R}_j^t$, $\mathbf{s}_j^t$, $\mathbf{c}_j^t$, and ζ_j^t, as well as reliability $\rho_j^t \in [0,1]$, observation support $S_j^{\mathrm{obs},t}$, and primary motion node o_j^t.

Before observation fusion, the position prior is computed as

$$\widehat{\boldsymbol{\mu}}_i^t = \begin{cases} \boldsymbol{\mu}_i^{t-1} + \displaystyle\sum_{\ell \in \mathcal{N}_{i,\mathrm{mo}}^t} \eta_{i\ell}^t \left(\mathbf{x}_\ell^t - \mathbf{x}_\ell^{t-1}\right), & \mathcal{N}_{i,\mathrm{mo}}^t \neq \varnothing, \\ \boldsymbol{\mu}_i^{t-1}, & \mathcal{N}_{i,\mathrm{mo}}^t = \varnothing. \end{cases} \tag{23}$$

With valid trajectory support, the prior follows Equation 3. Otherwise, the last committed center is retained as a fallback. This fallback is a stored geometric state, not a valid motion estimate or an assumption of stationarity. Gaussians without trajectory support are excluded from motion-based association but remain available for surface-based retrieval.

The orientation prior is $\mathbf{R}_i^{t-1}$ because the motion predictor does not estimate a separate rotation. Superscripts $t-1$ and t denote the retained pre-update state and the updated state before online refinement, respectively. Without an accepted current observation, a Gaussian retains its stored state; the fallback prior alone does not make it renderable.

Attribute Fusion. Fusion and reactivation use a common reliability-weighted update for color, scale, optical-thickness mass, and orientation. Their position and motion-ownership updates are

specified separately. Define

$$w_i^t = \max(N_i^{t-1}, 1), \qquad a_{ji}^t = r_{ji}^t \max(\rho_j^t, 0.1), \qquad A_i^t = \sum_{j \in \mathcal{J}_i^t} a_{ji}^t. \tag{24}$$

For a Gaussian with a nonempty accepted observation set, the attributes are updated by

$$\begin{aligned}
\mathbf{c}_i^t &= \frac{w_i^t \mathbf{c}_i^{t-1} + \sum_{j \in \mathcal{J}_i^t} a_{ji}^t \mathbf{c}_j^t}{w_i^t + A_i^t}, \\
\mathbf{s}_i^t &= \frac{w_i^t \mathbf{s}_i^{t-1} + \sum_{j \in \mathcal{J}_i^t} a_{ji}^t \mathbf{s}_j^t}{w_i^t + A_i^t}, \\
\zeta_i^t &= \frac{w_i^t \zeta_i^{t-1} + \sum_{j \in \mathcal{J}_i^t} a_{ji}^t \zeta_j^t}{w_i^t + A_i^t}, \\
\mathbf{R}_i^t &= \underset{\mathbf{R} \in \mathrm{SO}(3)}{\arg\min} \left[w_i^t \left\| \mathrm{Log}\big((\mathbf{R}_i^{t-1})^\top \mathbf{R}\big) \right\|_2^2 + \sum_{j \in \mathcal{J}_i^t} a_{ji}^t \left\| \mathrm{Log}\big((\mathbf{R}_j^t)^\top \mathbf{R}\big) \right\|_2^2 \right].
\end{aligned} \tag{25}$$

Scales are averaged in linear space. Orientations are combined by a weighted geodesic mean on $\mathrm{SO}(3)$ in a local neighborhood of $\mathbf{R}_i^{t-1}$, preserving the full rotation, including its twist degree of freedom. The attribute-weight floor allows an accepted low-reliability observation to make a limited contribution, without relaxing the action-level acceptance criteria.

Gated Position Correction. Position updates distinguish motion-guided fusion from observation-based position recovery during reactivation. For Gaussian i, let

$$\mathcal{R}_i^t = \left\{ j \in \mathcal{J}_i^t \;\middle|\; \mathsf{A}_j^t = \mathsf{reactivate} \right\} \tag{26}$$

be its accepted reactivation observations, and let o_i^{t-1} and o_i^t denote its retained and updated motion-owner identifiers. The geometry gate is

$$\chi_{ji}^t = \begin{cases} \mathbf{1}\left[o_j^t = o_i^{\mathrm{re},t}\right], & \mathcal{R}_i^t \neq \varnothing, \\ \mathbf{1}\left[o_j^t \neq \varnothing\right] \mathbf{1}\left[o_i^{t-1} = \varnothing \;\vee\; o_i^{t-1} = o_j^t\right], & \mathcal{R}_i^t = \varnothing. \end{cases} \tag{27}$$

For ordinary fusion, an observation cannot displace a Gaussian assigned to a different motion node. For validated reactivation, compatibility is checked against the selected current owner $o_i^{\mathrm{re},t}$ rather than the historical owner. A recovered Gaussian can therefore acquire new short-term motion support while preserving its persistent identity.

The gated, reliability-weighted geometric contribution is

$$\gamma_{ji}^t = r_{ji}^t \,\mathrm{clip}\left(\frac{\rho_j^t - \rho_{\min}}{1 - \rho_{\min}}, 0, 1 \right) \chi_{ji}^t, \qquad \Gamma_i^t = \sum_{j \in \mathcal{J}_i^t} \gamma_{ji}^t, \qquad 0 \leq \rho_{\min} < 1. \tag{28}$$

For $\mathcal{J}_i^t \neq \varnothing$, the center update is

$$\boldsymbol{\mu}_i^t = \begin{cases} \dfrac{\sum_{j \in \mathcal{J}_i^t} \gamma_{ji}^t \boldsymbol{\mu}_j^t}{\Gamma_i^t}, & \mathcal{R}_i^t \neq \varnothing, \\ \dfrac{w_i^t \widehat{\boldsymbol{\mu}}_i^t + \sum_{j \in \mathcal{J}_i^t} \gamma_{ji}^t \boldsymbol{\mu}_j^t}{w_i^t + \Gamma_i^t}, & \mathcal{R}_i^t = \varnothing. \end{cases} \tag{29}$$

The reactivation checks ensure $\Gamma_i^t > 0$ in the first branch. Reactivation estimates the current center directly from accepted observations, without averaging it with a potentially outdated historical position. Ordinary fusion combines the motion prior with accepted geometric evidence; if $\Gamma_i^t = 0$, it retains $\widehat{\boldsymbol{\mu}}_i^t$.

Reactivation preserves the Gaussian identifier and surface-record membership. Appearance and shape attributes continue to follow Equation 25, while Gaussians without accepted observations retain their pre-update states.

Support and Ownership. The observation count and accumulated support are updated as

$$N_i^t = N_i^{t-1} + \sum_{j \in \mathcal{J}_i^t} r_{ji}^t, \qquad S_i^t = S_i^{t-1} + \sum_{j \in \mathcal{J}_i^t} r_{ji}^t S_j^{\mathrm{obs},t}. \tag{30}$$

The last-seen time is set to t if at least one accepted responsibility satisfies $r_{ji}^t \geq 0.1$; otherwise, its previous value is retained. Ordinary fusion preserves an existing motion owner. If no owner is assigned, the accepted observation with the largest positive geometry weight γ_{ji}^t supplies the owner, with ties resolved by observation index. The owner remains unassigned if no positive geometry weight is available.

After successful reactivation, we set $o_i^t = o_i^{\mathrm{re},t}$ and rebuild the short-term motion attachments from accepted current observations. Obsolete motion-node links are removed, and the motion-dependent offsets and skinning weights are updated consistently with the recovered center and current support. This reassignment affects only the short-term motion association; persistent identifiers, surface-record membership, and accumulated appearance and support statistics are retained.

Inter-frame motion is estimated exclusively from valid consecutive-frame correspondences. The displacement between a recovered center and a historical center across an unobserved interval is not interpreted as a one-frame velocity or image-plane displacement.

Record-Level Aggregation. Let $\mathcal{Q}_{t,\iota}^{d,\mathrm{upd}}$ contain the accepted observations assigned to the record-level update in Equation 6. The corresponding Gaussian-level update is

$$\begin{aligned} &\mathrm{Update}_d\Big(\mathcal{G}_\iota^{d,t-1}, \mathcal{Q}_{t,\iota}^{d,\mathrm{upd}}\Big) \\ &\quad = \Big\{ g_i^{d,t} \;\Big|\; g_i^{d,t-1} \in \mathcal{G}_\iota^{d,t-1},\ \mathcal{J}_i^t \neq \varnothing \Big\} \cup \Big\{ g_i^{d,t-1} \;\Big|\; g_i^{d,t-1} \in \mathcal{G}_\iota^{d,t-1},\ \mathcal{J}_i^t = \varnothing \Big\}. \end{aligned} \tag{31}$$

Only targeted Gaussians are replaced by their updated states; untargeted members retain their previous attributes. Fusion and reactivation do not insert additional Gaussians or propagate one observation's update to every member of the record. With no accepted observations, the operator reduces to the identity update used in the main text.

Fusion and reactivation preserve Gaussian identities. Successful reactivation restores a dormant record to the active category, corresponding to an emerging or persistent implementation state. New Gaussians and their shared records are created according to Equation 20. Retained records without observation support may subsequently become dormant under the lifecycle policy.

Rejected Observations and Rendering Interface. For tentative and conflict observations, $r_{ji}^t = 0$ for every existing Gaussian. Rejected observations modify no persistent Gaussian, allocate no Gaussian or surface-record identifier, and make no contribution to current-frame rendering. Their short-term trajectory or motion-node evidence may be retained for subsequent validation. Rejection of one observation does not prevent another accepted observation from updating the same candidate record.

For rendering, each accepted dynamic Gaussian is instantiated from $(\boldsymbol{\mu}_i^t, \mathbf{R}_i^t, \mathbf{s}_i^t, \zeta_i^t, \mathbf{c}_i^t)$. The stored orientation is passed directly to the renderer without conversion from a surface normal. The optical-thickness mass ζ_i^t is converted to the renderer's opacity parameter rather than used directly as projected alpha. Visible Gaussians then undergo the finite-step online optimization in Section 3.4, with update targets, Gaussian identifiers, and surface-record memberships held fixed.

B.4 Persistent Storage and Rendering Eligibility

Persistent Dynamic Storage. After updating the long-term surface memory, the persistent dynamic field comprises all committed Gaussians in its surface records:

$$\mathcal{G}_t^d = \bigcup_{\iota:\, d_\iota^t \in \mathcal{D}_t} \mathcal{G}_\iota^{d,t}. \tag{32}$$

This field includes Gaussians in both active and dormant records. For a record without accepted current observations, the identity update

$$\mathrm{Update}_d\left(\mathcal{G}_\iota^{d,t-1}, \varnothing\right) = \mathcal{G}_\iota^{d,t-1} \tag{33}$$

preserves its member Gaussians, identifiers, and accumulated attributes for future association or reactivation. Persistent retention alone does not establish current-frame rendering eligibility.

Accepted Current Gaussians. Rendering eligibility is evaluated independently for each Gaussian, rather than assigned to an entire surface record. The indices supported by accepted current observations are

$$\mathcal{I}_t^{\mathrm{acc}} = \left\{ i \;\middle|\; \mathcal{J}_i^t \neq \varnothing \right\} \cup \left\{ \nu_j^t \;\middle|\; \mathsf{A}_j^t = \mathsf{initialize} \right\} \tag{34}$$

The first set contains existing Gaussians updated by fusion or reactivation; the second contains newly initialized Gaussians, including validated births at $t = 2$. A new Gaussian is eligible in the frame in which it is initialized. Motion prediction alone does not establish membership in $\mathcal{I}_t^{\mathrm{acc}}$, and rejected observations contribute no indices to this set.

Renderable Dynamic Subset. The dynamic field eligible for current-frame rendering is

$$\mathcal{G}_t^{d,\mathrm{rend}} = \left\{ g_i^{d,t} \in \mathcal{G}_t^d \;\middle|\; i \in \mathcal{I}_t^{\mathrm{acc}},\ z^t_{\mathrm{sid}_t(i)} = \mathsf{active} \right\}. \tag{35}$$

A fused, reactivated, or newly initialized Gaussian is eligible when its surface record is active. Other members of the same record are excluded unless they independently receive accepted current evidence. Unobserved Gaussians remain in persistent storage, and dormant Gaussians remain excluded until successfully reactivated. Thus,

$$\mathcal{G}_t^{d,\mathrm{rend}} \subseteq \mathcal{G}_t^d. \tag{36}$$

Acceptance of an observation for one member does not make the unchanged historical members of its record renderable.

Scene Assembly. The pre-refinement scene combines the static field with the selected dynamic subset:

$$\widetilde{\mathcal{G}}_t = \mathcal{G}_t^s \cup \mathcal{G}_t^{d,\mathrm{rend}}. \tag{37}$$

Only Gaussians in $\mathcal{G}_t^{d,\mathrm{rend}}$ contribute to the dynamic opacity $\widehat{A}_{d,t}$ and dynamic flow $\widehat{F}^d_{t-1\to t}$. Projection and visibility further restrict $\widetilde{\mathcal{G}}_t$ to the view-dependent set $\widetilde{\mathcal{G}}_t^{\mathrm{vis}}$. This visibility selection is distinct from the observation-based selection of persistent Gaussians in Equation 35.

B.5 Static–Dynamic Compositing

Gaussian-Level Selection. We render the assembled scene $\widetilde{\mathcal{G}}_t = \mathcal{G}_t^s \cup \mathcal{G}_t^{d,\mathrm{rend}}$ using the dynamic selection rule in Appendix B.4. A dynamic Gaussian must have an accepted current observation and an active surface record. Unsupported Gaussians, including other members of an active record, remain stored but are excluded from rasterization.

Joint Depth-Ordered Compositing. For pixel $\mathbf{p}$, let $\mathcal{V}_t(\mathbf{p}) = (g_1, \ldots, g_{L_\mathbf{p}})$ contain the projected Gaussians from $\widetilde{\mathcal{G}}_t$, sorted by camera-space depth in front-to-back order, $z_1 \leq \cdots \leq z_{L_\mathbf{p}}$. Their compositing weights are $w_n(\mathbf{p}) = \alpha_n(\mathbf{p}) \prod_{m<n}(1 - \alpha_m(\mathbf{p}))$, and $W_t(\mathbf{p}) = \sum_n w_n(\mathbf{p})$ is the accumulated foreground opacity. Suppressing the pixel argument and summing over $\mathcal{V}_t(\mathbf{p})$, the rendered quantities are

$$\begin{aligned} \widehat{I}_t &= \sum_n w_n \mathbf{c}_n + (1 - W_t)\mathbf{c}_{\mathrm{bg}}, \\ \widehat{D}_t &= \frac{\sum_n w_n z_n}{W_t + \epsilon}, \\ \widehat{A}_{d,t} &= \sum_{n:\, g_n \in \mathcal{G}_t^{d,\mathrm{rend}}} w_n. \end{aligned} \tag{38}$$

Here, $\alpha_n(\mathbf{p})$ is the effective projected opacity of g_n, and $w_n(\mathbf{p})$ includes the transmittance of all closer Gaussians. Static and dynamic Gaussians share one depth ordering, so mutual occlusion is handled by the same compositing rule. The depth $\widehat{D}_t$ is the opacity-normalized expected depth, while $\widehat{A}_{d,t}$ sums the visible dynamic contributions under the joint ordering.

Dynamic Flow Rendering. The short-term motion memory $\mathcal{M}_t$ supplies image-plane motions for dynamic flow rendering. The renderer composites the projected displacements of Gaussians in $\mathcal{G}_t^{d,\mathrm{rend}}$ under the same visibility ordering to obtain $\widehat{F}^d_{t-1\to t}$. Thus, $\mathcal{D}_t$ and $\mathcal{G}_t^d$ retain the persistent dynamic representation, while only the selected subset contributes to current-frame rasterization.

B.6 Streaming Evaluation Protocol

Causal Source Processing. Our method processes sampled source frames in chronological order. The index t refers to the sampled input sequence, not the original video. At reconstruction step $t \geq 2$, scene construction, memory updates, and online refinement use only source frames $1, \ldots, t$ and their supplied camera parameters. Depth, optical-flow, and dynamic-mask estimation obey the same temporal restriction. The first two source frames initialize the scene at $t = 2$, including any dynamic Gaussians admitted by Equation 18; no reconstruction is produced for source frame 1. Subsequent frames update the scene incrementally without accessing future source observations or revising earlier predictions. Target RGB images are excluded from scene construction, Gaussian association, and online refinement. Target camera parameters specify only the novel views to be rendered.

NVIDIA. We use 12 source frames with round-robin camera selection: source frame t is acquired by camera t at timestep t. One source view is available at each timestep, yielding a chronological monocular input sequence. At each evaluated timestep, we render the held-out target views at the same scene time from the state obtained after processing source frame t. The remaining source frames, $t + 1, \ldots, 12$, are excluded from that prediction.

DyCheck. Following the MoVieS data-sampling protocol, we select 13 source frames from a 65-frame clip. Let $s_1 < \cdots < s_{13}$ denote their indices in the original video. At step t, reconstruction uses the selected iPhone frames at $s_1, \ldots, s_t$; the target views are the synchronized images from the other two cameras at s_t. We render these views from the state obtained after processing the source frame at s_t, without incorporating the selected frames at $s_{t+1}, \ldots, s_{13}$. The covisibility mask is recomputed from the same source prefix and applied to all compared methods when evaluating mPSNR, mSSIM, and mLPIPS.

RealEstate10K. The 12 selected posed source frames are processed chronologically. After the final update, the accumulated static Gaussian scene is used to render the three predefined held-out target views. All methods use identical source and target selections. This protocol evaluates static novel-view synthesis after incremental scene accumulation, rather than causal prediction at the original capture time of each target image.

C Limitations

While PerM4D demonstrates competitive novel-view synthesis performance, several limitations remain. First, our framework requires externally provided camera poses, and all experiments use ground-truth poses. Reliable camera pose estimation is itself particularly challenging in dynamic monocular streams, where only past and current observations are available. Our framework does not estimate camera poses within the reconstruction pipeline, limiting its direct applicability to unposed video streams. Second, although PerM4D processes frames incrementally using only past and current observations, its per-frame processing latency does not yet support real-time reconstruction. Per-frame Gaussian construction, memory-guided association, and online refinement incur computational overhead, requiring further acceleration for latency-sensitive applications. Third, complex motion remains challenging, particularly when reliable inter-frame correspondences cannot be maintained. In such cases, motion estimation and memory-guided association may become less reliable, leading to inaccurate Gaussian updates and rendering artifacts. Future work will explore reliable online camera pose estimation, more efficient per-frame processing, and more robust motion estimation and memory-guided association under challenging dynamics.

D Additional Experimental Results

Static Scene Novel-View Synthesis. Figures 5–7 present additional qualitative comparisons on RealEstate10K. The panels compare novel views synthesized by MoVieS, Flow4DGS-SLAM, DynOMo, and PerM4D with the corresponding target images; methods and targets are identified by the column headings.

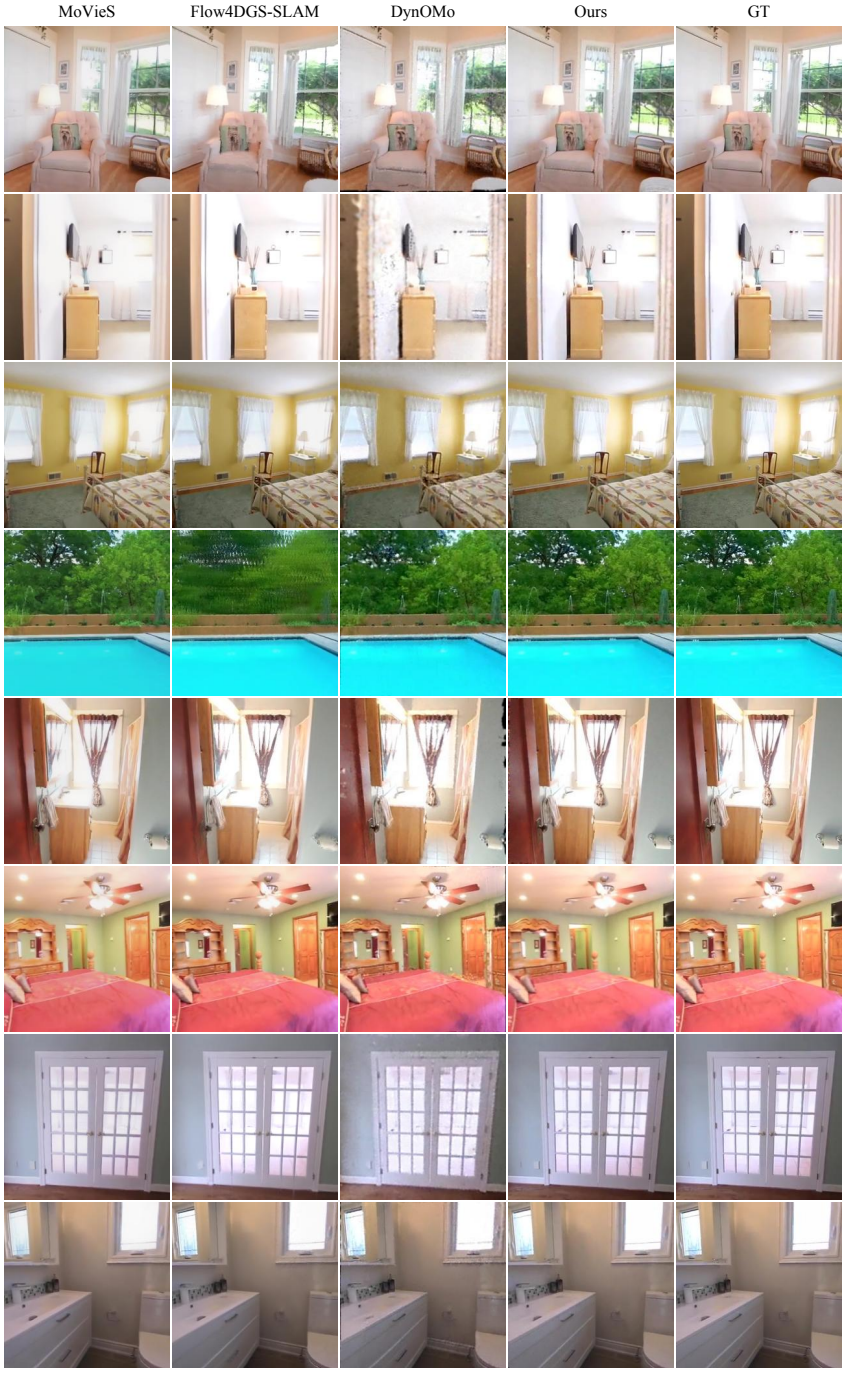

Figure 5: **Novel-view synthesis on RealEstate10K (set 1 of 3).** Additional qualitative comparisons of MoVieS, Flow4DGS-SLAM, DynOMo, and PerM4D.

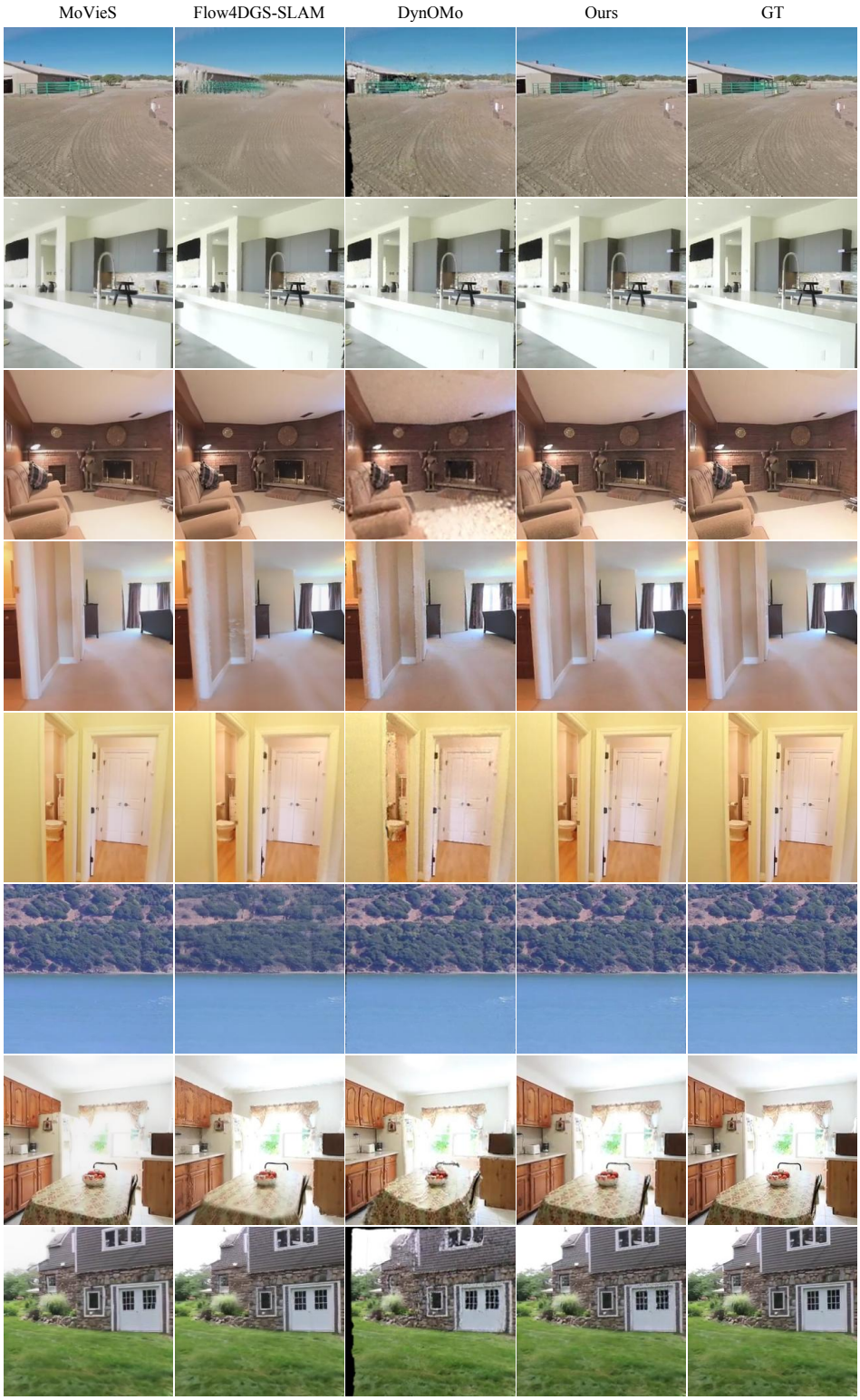

Figure 6: **Novel-view synthesis on RealEstate10K (set 2 of 3).** Additional qualitative comparisons of MoVieS, Flow4DGS-SLAM, DynOMo, and PerM4D.

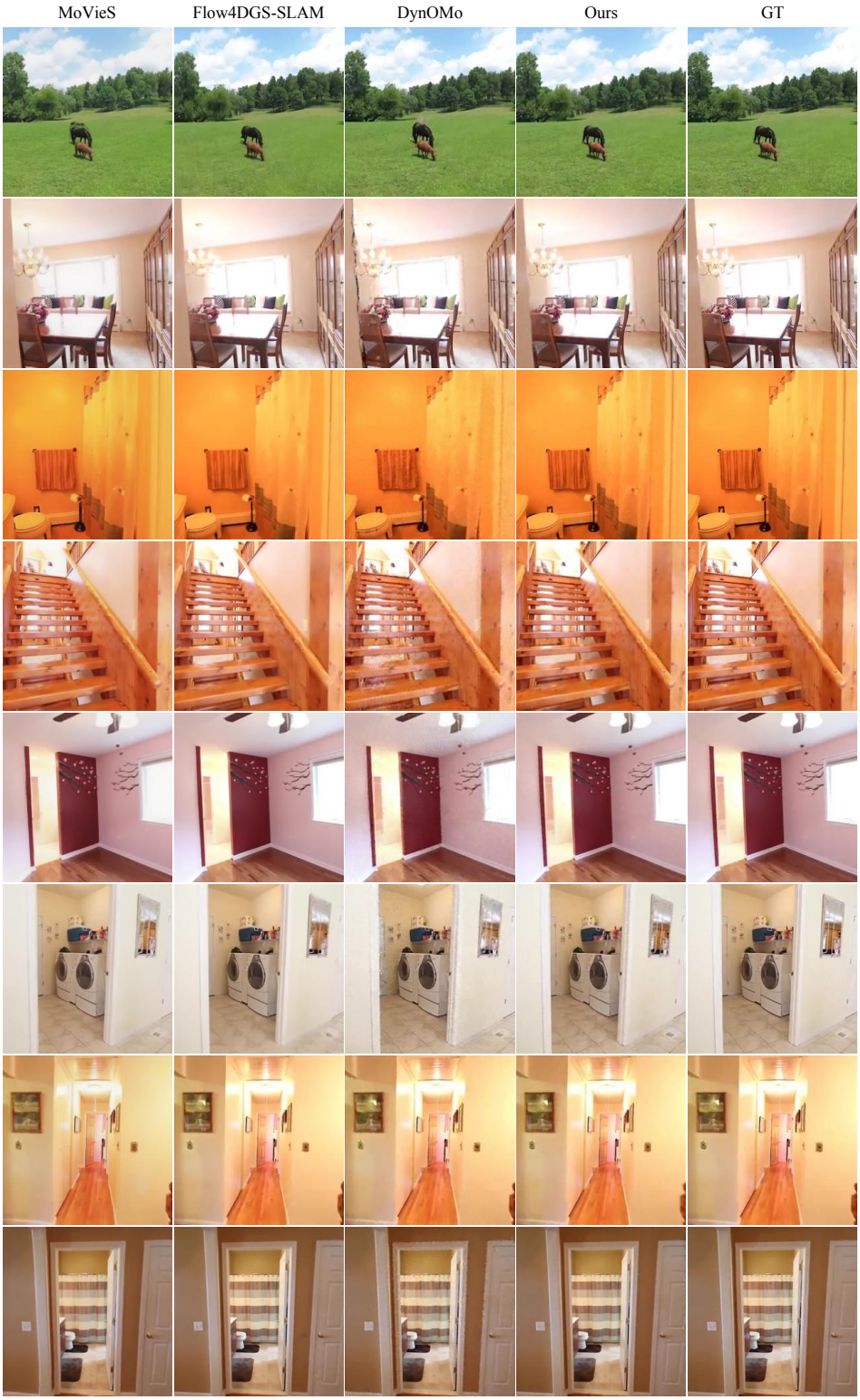

Figure 7: **Novel-view synthesis on RealEstate10K (set 3 of 3).** Additional qualitative comparisons of MoVieS, Flow4DGS-SLAM, DynOMo, and PerM4D.