MoPe: Motion Permanence for Robust Monocular Gaussian Mapping in Dynamic Environments

Qixin Xiao
University of Michigan
qxiaocs@umich.edu

***Abstract*—Robust robot autonomy depends on scene representations that remain stable enough to support localization, navigation, and downstream decision making in dynamic environments. Monocular Gaussian Splatting SLAM provides high-fidelity mapping, but current uncertainty-aware methods still treat dynamic regions largely as per-frame observations. This makes the representation effectively memoryless: when a pedestrian slows, pauses, or reappears after occlusion, the current frame may look static, allowing dynamic content to be absorbed into the map and leaving persistent ghosting artifacts. We argue that this failure reflects a representation-level mismatch. Dynamic-ness is not an instantaneous appearance property, but a temporal property defined by motion history. Building on this view, we introduce Motion Permanence: the principle that an object's dynamic identity should persist over time rather than be re-decided from each frame independently. We realize this principle in MoPe, a memory-aware uncertainty filter for monocular Gaussian mapping. MoPe propagates the historical dynamic posterior through geometry-consistent SE(3) warping and fuses it with current-frame evidence using bounded Bayesian log-odds updates. The resulting persistent posterior guides tracking, mapping, dynamic-aware Gaussian insertion, and Gaussian-level post-cleanup. On Wild-SLAM, Bonn, and TUM sequences, MoPe improves tracking robustness and reduces residual ghosting, with the strongest gains on dynamic-human scenes that most directly violate the memoryless assumption. These results show that maintaining temporal dynamic state inside the scene representation is a practical step toward more reliable representation-centric autonomy in changing real-world environments.**

***Index Terms*—Representation-Centric Autonomy, Dynamic SLAM, Gaussian Splatting, Motion Permanence, Temporal Uncertainty**

I. Introduction

Robust monocular SLAM in dynamic environments is a long-standing prerequisite for real-world robotic deployment. Recent advances in 3D Gaussian Splatting (3DGS) have substantially improved scene representation and rendering quality, but dynamic distractors remain a fundamental obstacle. When a moving object repeatedly occludes the same background region, even brief intervals of apparent stillness can cause its appearance to be baked into the static map, leaving residual ghosting and afterimages that corrupt both geometry and novel-view rendering. Crucially, for robotic deployment this is not merely a rendering artifact: a static map corrupted by intermittent distractors can mislead any downstream planner or navigation stack that relies on it.

A representative state-of-the-art approach is WildGS-SLAM [17], which replaces brittle binary masking with dense uncertainty weighting. From monocular RGB input, it predicts a continuous uncertainty field from DINOv2 features [10] and uses it to modulate both dense bundle adjustment [13] during tracking and photometric/depth supervision during mapping. This soft-weighting design is well-motivated and avoids the hard, error-prone decisions of binary masks. Its one structural blind spot is that the uncertainty field is estimated independently at every frame, purely from instantaneous image evidence. The system is therefore effectively *memoryless*: the moment a distractor slows, pauses, or re-emerges in a previously occluded region, the current frame alone may look sufficiently static for it to be assigned low uncertainty, and dynamic content silently contaminates both pose and map. In a single frame, a paused pedestrian and a wall are nearly indistinguishable; what separates them is not their current appearance, but their history. For autonomous robots, this contamination is not only a rendering artifact: it can corrupt the static world model used by localization, navigation, and downstream decision-making modules.

This observation points to a conceptual mismatch. Whether a region is dynamic is treated as an *appearance* property—something to be read off each frame—when it is in fact a *temporal* property, defined over an object's motion history rather than its momentary look. Resolving this requires an object's dynamic identity to persist over time, rather than being re-decided from scratch at every frame. We call this principle **Motion Permanence**, by analogy to object permanence in cognitive development: just as an object continues to exist when momentarily unseen, a moving object retains its dynamic identity when momentarily still. Motion Permanence is thus not an added heuristic, but the property the problem structurally demands.

We realize Motion Permanence in **MoPe**, a memory-aware uncertainty filter for monocular Gaussian mapping. Rather than treating uncertainty as a transient per-frame prediction, MoPe promotes it into a temporally propagated dynamic state. It warps the historical dynamic posterior into the current frame through geometry-consistent SE(3) propagation, then fuses it with current-frame evidence via a bounded Bayesian log-odds update. The resulting posterior provides a stable estimate of persistently unreliable regions under intermittent motion and repeated occlusion, and is used online to guide tracking, mapping, dynamic-aware Gaussian insertion, and a conservative Gaussian-level post-cleanup.

Our aim is not to solve every form of dynamic SLAM, but

to address a specific and common failure mode of uncertainty-based systems: temporarily stationary or intermittently moving distractors that corrupt the static map. Accordingly, MoPe yields its strongest gains on dynamic-human scenes, the regime that most directly violates the memoryless assumption. Our contributions are threefold:

- We identify the memoryless nature of single-frame uncertainty estimation as a **representation-level** limitation in uncertainty-aware monocular Gaussian mapping, and reframe dynamic-ness as a persistent temporal state rather than a per-frame appearance cue.
- We introduce **Motion Permanence**, realized through geometry-consistent SE(3) propagation and bounded Bayesian log-odds fusion, which maintains a temporally consistent dynamic posterior under intermittent motion and repeated occlusion.
- We integrate this posterior across tracking, mapping, dynamic-aware Gaussian insertion, and Gaussian-level post-cleanup, improving robustness to residual ghosting and map contamination, with consistent gains on representative dynamic-human sequences.

Code: The implementation and reproduction scripts are available at https://github.com/chloeqxq/MoPe.

II. Related Work

A. Visual SLAM in Dynamic Environments

Classical visual SLAM assumes static scenes and treats dynamic objects as outliers to be rejected [1, 11, 12]. Dense, differentiable alignment architectures—most notably DROID-SLAM [13]—popularized recurrent update operators and dense bundle adjustment for accurate tracking under challenging appearance variation. To handle dynamic interference within such dense frameworks, a complementary line of work models heteroscedastic noise and per-residual confidence [5, 2] to softly down-weight unreliable regions during photometric minimization. These foundations establish the value of continuous soft-weighting, but maintaining the *temporal* consistency of such weights during online pose-and-map optimization remains an unsolved failure mode in the wild.

B. Dynamic 3DGS-SLAM

While 3DGS enables real-time, high-fidelity SLAM [8, 4, 3], deploying it in dynamic scenes requires robust filtering to suppress ghosting and map contamination. BDGS-SLAM [16] introduces a Gaussian-level Bayesian framework driven by YOLO-based semantic priors, which restricts it to known object classes. UP-SLAM [18] and RGD-SLAM [15] operate in 2D image space but threshold motion residuals or semantic probabilities into brittle binary masks, discarding useful boundary information and struggling with open-set distractors. WildGS-SLAM [17] instead predicts a continuous 2D uncertainty map from foundation features to jointly weight tracking and mapping; this flexible per-frame estimate, however, remains oblivious to history—precisely the gap Motion Permanence is designed to close.

C. Temporal Modeling in Dynamic 3DGS-SLAM

Existing attempts at temporal consistency operate at the object, mask, or 3D-representation level. DGS-SLAM [6] and DyGS-SLAM [19] enforce consistency through discrete object-track association or multiview geometric refinement. RGD-SLAM applies an Extended Kalman Filter, but its filtering is confined to discrete segmentation masks. UP-SLAM includes a "temporal encoder" that merely injects frame-index positional embeddings to condition rendering—a timestamp for view synthesis, not a mechanism for propagating dynamic state across frames. VarSplat [14] bakes uncertainty into the 3D map as a passive byproduct of a slowly evolving static representation, while 4D Gaussian Splatting [7] explicitly models moving geometry, addressing a fundamentally different, non-static-map problem. A critical gap therefore remains: continuous, image-space soft-weighting pipelines lack any explicit notion of persistent dynamic identity. MoPe fills this gap by elevating a memoryless 2D prediction into a trackable temporal state, performing cross-frame geometric propagation and Bayesian fusion directly on the dense uncertainty field—supplying the temporal memory that Motion Permanence requires.

III. Methodology

A. Overview

MoPe augments a monocular Gaussian mapping backbone with a single principle: a region's dynamic identity should *persist* rather than be re-estimated at each frame. Concretely, we do not treat dynamic estimation as an instantaneous per-frame decision; we accumulate dynamic evidence over time and reuse the resulting temporal posterior throughout tracking, mapping, and final cleanup. The method comprises three tightly coupled components: (i) a temporally propagated dynamic posterior that operationalizes Motion Permanence, (ii) posterior-guided optimization with dynamic-aware Gaussian insertion, and (iii) a conservative Gaussian-level post-cleanup. Together they suppress dynamic contamination before it enters the map, while retaining a final safeguard for residual errors that survive the online process. The full per-frame procedure is summarized in Algorithm 1.

B. Motion Permanence via Temporal Evidence Accumulation

For each frame t, the uncertainty branch predicts a per-pixel uncertainty map $\beta_t(u)$, which we convert into a confidence weight

$$w_t(u) = \mathrm{clip}\left(\frac{0.5}{\beta_t(u)^2},\, w_{\min},\, 1\right), \tag{1}$$

and a frame-wise dynamic evidence score

$$d_t^{\mathrm{unc}}(u) = 1 - w_t(u). \tag{2}$$

To improve robustness in ambiguous regions, we optionally add a weak semantic prior $d_t^{\mathrm{sem}}(u)$ from an off-the-shelf segmentation model—not as a hard replacement for geometry,

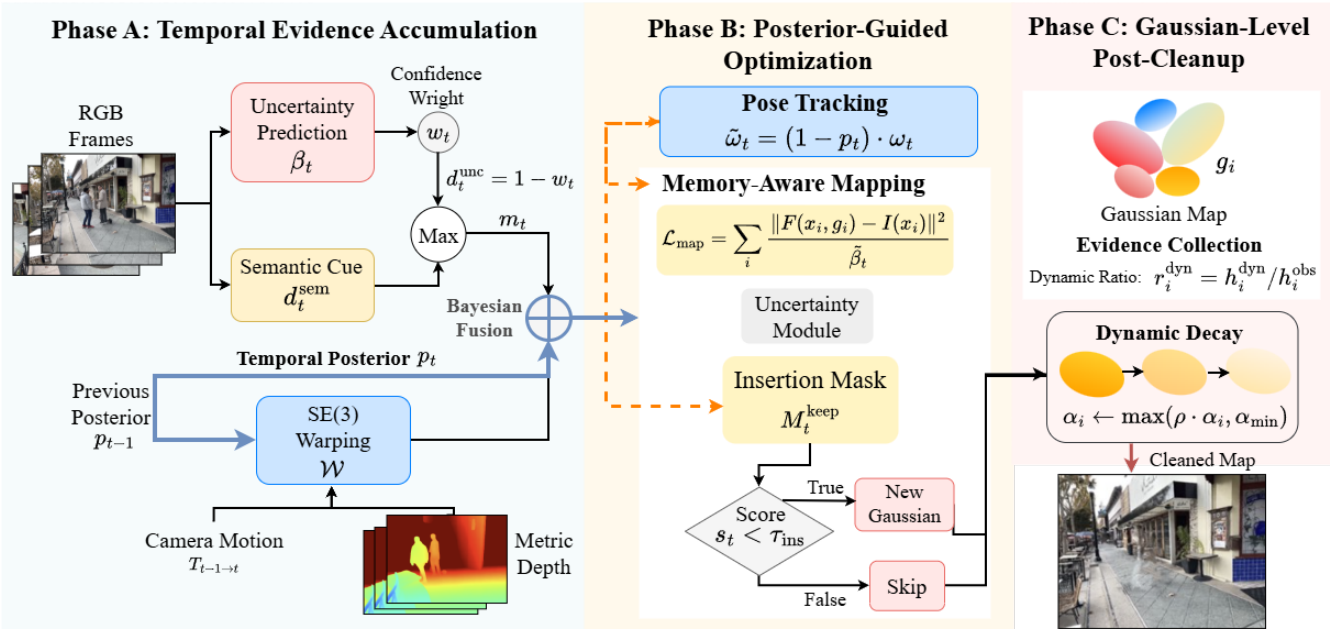

Fig. 1. **MoPe** architecture. **Phase A (Motion Permanence):** instantaneous uncertainty β_t and an optional semantic cue are fused with the SE(3)-warped historical posterior P_{t-1} via bounded Bayesian log-odds updates, yielding a temporally consistent dynamic posterior P_t. **Phase B (Posterior-Guided Optimization):** the posterior down-weights tracking residuals, modulates the mapping objective, and gates Gaussian insertion through a keep-mask M_t^{keep}. **Phase C (Post-Cleanup):** Gaussians with persistent dynamic evidence undergo opacity decay, yielding a clean static reconstruction.

Algorithm 1 MoPe: Per-Frame Motion Permanence Update

Require: frame I_t, previous log-odds L_{t-1}, motion $T_{t-1\to t}$, depth D_t

Ensure: dynamic posterior p_t, updated map

1: $m_t \leftarrow \max\left(1 - w_t(\beta_t),\, d_t^{\text{sem}}\right)$ {measurement from uncertainty β_t}

2: $\tilde{L}_{t-1\to t} \leftarrow \text{Warp}(L_{t-1}; T_{t-1\to t}, D_t)$ {SE(3) prop. (across viewpoint)}

3: $L_t \leftarrow \text{clip}(\tilde{L}_{t-1\to t} + \gamma \,\text{logit}(m_t), -c, c)$; $p_t \leftarrow \sigma(L_t)$ {fusion $\to$ posterior (across time)}

4: track with $\tilde{\omega}_t = (1 - p_t)\,\omega_t$ {posterior-guided tracking}

5: $M_t^{\text{keep}} \leftarrow \mathbf{1}[\max(p_t, \eta M_t^{\text{sem}}) < \tau_{\text{ins}}]$; update map {fail-closed insertion + mapping}

6: if keyframe, decay opacity of persistent dynamic Gaussians {post-cleanup}

7: **return** p_t, updated map

but as a soft floor on the dynamic likelihood of likely-transient classes. The per-frame measurement is

$$m_t(u) = \max(d_t^{\text{unc}}(u),\, d_t^{\text{sem}}(u))\,. \tag{3}$$

Motion Permanence is realized in two stages. First, to preserve dynamic identity *across changing viewpoints*, we perform SE(3) kinematic propagation, warping the previous posterior into the current frame using the estimated camera motion and available geometry:

$$\tilde{p}_{t-1\to t}(u) = \mathcal{W}(p_{t-1};\, T_{t-1\to t},\, D_t)\,(u), \tag{4}$$

where $T_{t-1\to t}$ is the relative camera motion, D_t the current depth support, and $\mathcal{W}(\cdot)$ projective warping. Second, to preserve dynamic identity *across time*, we fuse the propagated state with the current measurement in a bounded log-odds space:

$$L_t(u) = \text{clip}(L_{t-1\to t}(u) + \gamma \,\text{logit}(m_t(u)),\, -c,\, c)\,, \tag{5}$$

$$p_t(u) = \sigma(L_t(u))\,, \tag{6}$$

where γ is a measurement gain, c a clipping bound, and $\sigma(\cdot)$ the sigmoid. Regions that repeatedly behave dynamically accumulate durable evidence, while a single misleading "looks static" frame can no longer erase that history. We accumulate evidence in log-odds rather than probability space for three reasons: Bayesian updates reduce to simple additions, dynamic and static evidence are treated symmetrically, and the bound c keeps the posterior from saturating—so a region long believed dynamic can still recover to static once it consistently stops moving, which matters for objects that become permanently stationary. The posterior $p_t(u)$ is thus a persistent estimate of dynamic identity—the concrete form Motion Permanence takes.

C. Posterior-Guided Optimization and Gaussian Insertion

The persistent posterior is used throughout the online pipeline. For **robust tracking**, we down-weight residuals in high-posterior regions so that likely-dynamic observations contribute less to pose estimation,

$$\tilde{\omega}_t(u) = (1 - p_t(u))\,\omega_t(u), \tag{7}$$

making the tracker insensitive to slowly moving or temporarily stationary distractors. For **memory-aware mapping**, the ac-

cumulated posterior is folded back into the rendering objective through a temporally informed uncertainty $\tilde{\beta}_t(u)$,

$$\mathcal{L}_{\text{map}} = \sum_u \frac{\mathcal{L}_{\text{rgb}}(u) + \lambda_d \mathcal{L}_{\text{depth}}(u)}{\tilde{\beta}_t(u)^2} + \lambda_u \mathcal{L}_{\text{unc}}, \quad (8)$$

so that regions with persistent dynamic evidence receive lower optimization weight. Most importantly, the posterior gates **Gaussian insertion**: a pixel is admitted only if its dynamic insertion score stays below a threshold,

$$s_t(u) = \max(p_t(u),\, \eta\, M_t^{\text{sem}}(u))\,, \quad (9)$$

$$M_t^{\text{keep}}(u) = \mathbf{1}[s_t(u) < \tau_{\text{ins}}]\,, \quad (10)$$

where M_t^{sem} is an optional semantic support mask and η a conservative floor. This rule is deliberately fail-closed: when the dynamic region dominates or reliable support is scarce, MoPe skips insertion rather than committing ambiguous content to the static map. We deliberately bias this gate toward skipping rather than inserting: a missed insertion can be recovered in a later frame once the region is confidently static, whereas a dynamic Gaussian baked into the map produces persistent ghosting that the downstream cleanup must then undo. Under uncertainty, omission is cheaper to fix than contamination. Because the system *remembers* that a region is dynamic, it refuses to bake that region into the map in the first place.

D. Gaussian-Level Post-Cleanup

Some dynamic Gaussians may still survive due to imperfect observations, repeated occlusions, or early mistakes. As a final safeguard, we project each Gaussian g_i into multiple keyframes and collect evidence on whether it repeatedly behaves dynamically. With h_i^{dyn} dynamic observations out of h_i^{obs} valid ones, we define the dynamic ratio

$$r_i^{\text{dyn}} = \frac{h_i^{\text{dyn}}}{h_i^{\text{obs}}}. \quad (11)$$

A Gaussian with sufficient multi-view support and a high dynamic ratio is a cleanup candidate. Rather than hard-deleting it—an aggressive operation that can damage valid background—we progressively attenuate its opacity,

$$\alpha_i \leftarrow \max(\rho\, \alpha_i,\, \alpha_{\min})\,, \quad (12)$$

with decay factor $\rho \in (0, 1)$ and lower bound $\alpha_{\min}$. This gradual decay suppresses persistent dynamic residuals and afterimages while preserving uncertain-but-valid structure.

IV. Experiments

A. Experimental Setup

Datasets and metrics. We evaluate MoPe on the Wild-SLAM MoCap benchmark [17] for camera tracking (ATE RMSE) and novel-view synthesis (PSNR, SSIM, LPIPS), and extend the tracking evaluation to the Bonn and TUM RGB-D dynamic benchmarks to assess generalization. To directly probe residual ghosting, we additionally evaluate the challenging `iphone_wandering` sequence from the Wild-SLAM iPhone dataset; lacking reliable trajectory ground truth, we report background-only PSNR and MAE, with tail-only variants that isolate the hardest segment where afterimages are most likely to persist.

Baselines. We compare against representative dense and Gaussian SLAM systems—DROID-SLAM [13], MonoGS [8], Splat-SLAM [4], and MonST3R-SW [9]—and against the uncertainty-aware WildGS-SLAM [17], our primary baseline and the method MoPe most directly extends.

Implementation. MoPe is built on a monocular Gaussian SLAM backbone with dense uncertainty weighting. Across all evaluations, the temporal posterior is fused in both tracking and mapping, dynamic evidence is accumulated in bounded log-odds space, semantic-aware insertion gating is active, and the Gaussian-level cleanup is applied as a final stage. The key hyperparameters—measurement gain γ, log-odds bound c, insertion threshold τ_{ins}, and opacity decay ρ—are held fixed across all sequences, with no per-sequence tuning.

B. Results

Tracking robustness. As shown in Table I, MoPe achieves the best tracking accuracy across all three dynamic datasets, improving over the WildGS-SLAM baseline. The gain is most pronounced in sequences with intermittently moving objects, where Motion Permanence prevents the tracker from locking onto a temporarily stationary distractor. The relative gains are consistent across benchmarks—17.4% on Wild-SLAM ($0.46 \rightarrow 0.38$), 15.6% on Bonn ($2.31 \rightarrow 1.95$), and 15.2% on TUM ($1.51 \rightarrow 1.28$)—and are largest on Wild-SLAM, which contains the most frequent intermittent human motion. This is exactly what Motion Permanence predicts: the more often a distractor stops and resumes, the more a persistent dynamic state helps over a memoryless per-frame estimate.

TABLE I
TRACKING PERFORMANCE ACROSS DYNAMIC DATASETS (ATE RMSE ↓ [CM]). MOPE'S IMPROVEMENTS OVER BASELINES ARE HIGHLIGHTED IN BLUE.

Method	Wild-SLAM	Bonn	TUM
DROID-SLAM [13]	16.17	4.91	30.4
MonoGS [8]	47.99	22.80	26.4
Splat-SLAM [4]	8.71	8.80	1.71
MonST3R-SW [9]	12.12	7.30	1.51
WildGS-SLAM [17]	0.46	2.31	1.51
MoPe (Ours)	0.38	1.95	1.28

Map quality and artifact suppression. Beyond tracking, MoPe improves reconstruction by suppressing the ghosting that memoryless systems leave behind. On the MoCap comparison (Table II), MoPe improves over the baseline on the representative human-dynamic sequences, most clearly on *Crowd* and *Person*, where both tracking and rendering metrics improve. On the targeted artifact benchmark (Table III), MoPe achieves the cleanest background reconstruction on `iphone_wandering`, with the clearest advantage in the tail

(a) WildGS (b) Ours

Fig. 2. Rendered RGB on the iPhone-wandering sequence at frame 387. MoPe yields a cleaner pedestrian region and reduces persistent afterimages.

segment—exactly where persistent afterimages accumulate. The qualitative results corroborate this: Figure 2 shows MoPe preventing dynamic pedestrians from being baked into the static map, and Figure 3 shows clean dynamic-object removal across MoCap scenes without floating residuals.

We stress that MoPe is a *targeted* rather than uniform improvement. On harder cluttered sequences such as ANYmal2, Stones, and Table1, a few visual metrics fall slightly below the baseline, marking the boundary of the current design. The accurate claim is therefore not blanket superiority, but preserved performance on most scenes together with a clear gain on the dynamic-human failure mode we target. The per-sequence breakdown in Table II sharpens this picture. The clearest gains are on human-dynamic sequences with stop-and-go motion: *Person* improves in both ATE ($0.800 \rightarrow 0.650$) and PSNR ($20.31 \rightarrow 20.85$), and *Crowd* similarly. The small regressions instead concentrate on *ANYmal2*, *Stones*, and *Table1*, where motion is either continuous (a walking robot) or near-static clutter—regimes in which a persistent dynamic prior offers little benefit while conservative insertion mildly reduces fine detail. MoPe helps precisely where its assumption holds, and is largely neutral elsewhere.

TABLE III
ARTIFACT-FOCUSED COMPARISON ON `IPHONE_WANDERING`. BACKGROUND-ONLY METRICS MEASURE HOW CLEANLY THE STATIC SCENE IS RECOVERED AFTER DYNAMIC OCCLUSIONS.

Method	Mean BG PSNR ↑	Mean BG MAE ↓	Tail BG PSNR ↑	Tail BG MAE ↓
Semantic-prior Baseline	18.401	19.187	18.102	20.649
MoPe (Ours)	18.410	19.183	18.418	20.021

Efficiency. Table IV reports per-frame runtime and memory. The temporal posterior adds only 0.3 ms to tracking ($24.5 \rightarrow 24.8$ ms), confirming that a single SE(3) warp plus an element-wise log-odds update is lightweight. Strikingly, MoPe is not merely free of net overhead but is *faster* overall (65.3 vs. 69.7 ms) and more memory-efficient (5.3 vs. 5.8 GB) than the baseline. This is a direct consequence of Motion Permanence: by refusing to insert content it remembers as dynamic, fail-closed gating keeps the Gaussian map leaner, which accelerates mapping ($45.2 \rightarrow 40.5$ ms) and lowers memory. A cleaner static representation is thus also a cheaper one—temporal filtering pays for itself.

TABLE IV
RUNTIME AND MEMORY. PER-FRAME AVERAGES ON THE WILD-SLAM DATASET, MEASURED ON AN NVIDIA RTX 4090 GPU.

Method	Track (ms)	Map (ms)	Total (ms)	Mem (GB)
WildGS-SLAM [17]	24.5	45.2	69.7	5.8
MoPe (Ours)	24.8	40.5	65.3	5.3

C. Ablation Study

Table V isolates the contribution of each component on `iphone_wandering`. Starting from temporal fusion—the log-odds accumulation at the core of Motion Permanence—we progressively add the semantic prior, fail-closed insertion gating, and opacity decay. The full pipeline attains the best values across all background metrics. The progression is not strictly monotonic on every metric: adding insertion gating or opacity decay in isolation can slightly perturb mean PSNR, since each trades a small amount of reconstruction density for cleanliness. The full configuration nonetheless attains the best values on all four background metrics, indicating the components are complementary rather than independently additive—fusion supplies the persistent state, gating keeps dynamic content out, and decay removes what survives, with the strongest effect on the tail segment ($20.862 \rightarrow 20.021$ in MAE) where afterimages are hardest to suppress.

TABLE V
PROGRESSIVE ABLATION ON `IPHONE_WANDERING`. EACH ROW ADDS ONE COMPONENT. THE FULL PIPELINE YIELDS THE BEST BACKGROUND PRESERVATION.

Variant	Mean BG PSNR ↑	Mean BG MAE ↓	Tail BG PSNR ↑	Tail BG MAE ↓
Temporal Fusion	18.364	19.365	18.026	20.862
+ Semantic Prior	18.396	19.367	18.102	20.649
+ Insertion Gating	18.371	19.405	18.072	20.759
+ Opacity Decay	18.354	19.441	18.045	20.829
MoPe (Full)	18.410	19.183	18.418	20.021

Hyperparameter sensitivity. Table VI varies the insertion threshold τ_{ins}. MoPe is stable across a broad range $[0.3, 0.7]$, confirming the method does not rely on careful tuning. Performance degrades only at the extremes: overly small τ_{ins} starves insertion and harms reconstruction, while overly large values relax the fail-closed guard and let ghosting return. Concretely, performance is flat across $\tau_{\text{ins}} \in [0.3, 0.7]$ (mean BG PSNR within 0.03 of the 18.410 optimum) and degrades only at the extremes: at τ_{ins}=0.1 insertion is starved and PSNR drops to 17.852, while at 0.9 the fail-closed guard is effectively disabled and tail MAE rises to 20.612.

TABLE II
COMPARISON ON WILD-SLAM MOCAP. LOWER IS BETTER FOR ATE AND LPIPS; HIGHER IS BETTER FOR PSNR AND SSIM. BETTER VALUES ARE HIGHLIGHTED IN BLUE.

Sequence	Baseline ATE ↓	MoPe ATE ↓	Baseline PSNR ↑	MoPe PSNR ↑	Baseline SSIM ↑	MoPe SSIM ↑	Baseline LPIPS ↓	MoPe LPIPS ↓
ANYmal1	0.200	0.175	21.850	22.120	0.8070	0.8210	0.2110	0.1920
ANYmal2	0.300	0.265	21.460	21.320	0.8320	0.8250	0.2300	0.2050
Ball	0.200	0.210	20.060	20.350	0.7540	0.7680	0.1910	0.1780
Crowd	0.300	0.250	21.280	21.650	0.8020	0.8150	0.1760	0.1650
Person	0.800	0.650	20.310	20.850	0.8010	0.8180	0.1890	0.1720
Racket	0.400	0.340	20.870	21.150	0.7850	0.7980	0.1860	0.1710
Stones	0.300	0.285	20.520	20.450	0.7680	0.7650	0.1850	0.1680
Table1	0.600	0.520	20.330	20.210	0.7880	0.7810	0.2090	0.1850
Table2	1.300	1.150	19.160	19.550	0.7280	0.7450	0.3030	0.2750
Umbrella	0.200	0.190	20.030	20.250	0.7660	0.7750	0.2100	0.1980
Crowd+Person Avg.	0.550	0.450	20.795	21.250	0.8015	0.8165	0.1825	0.1685
Full 10-Seq Avg.	0.460	0.404	20.587	20.790	0.7831	0.7911	0.2090	0.1909

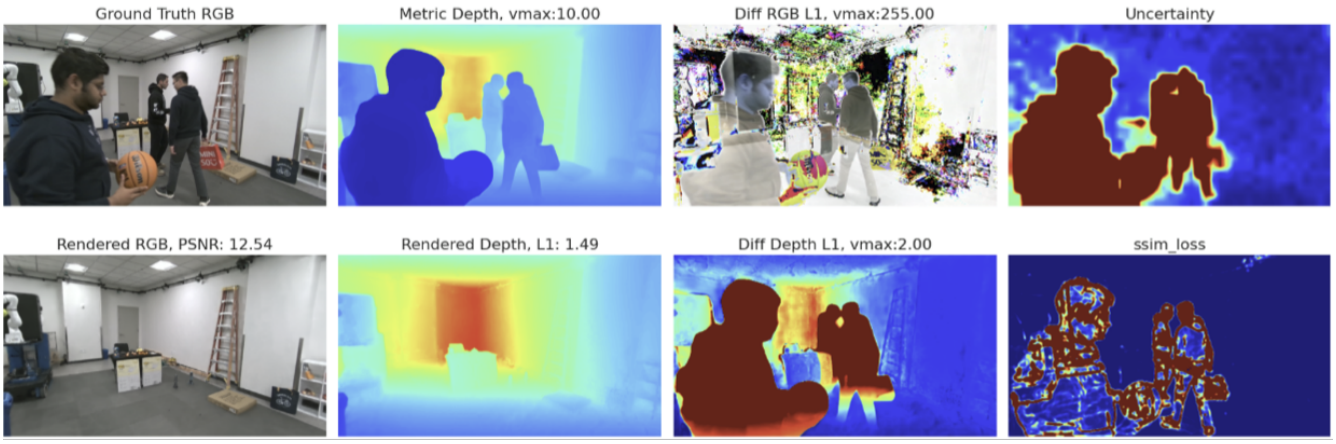

Fig. 3. Dynamic-object removal on MoCap sequences. Input RGB (top) and MoPe output (bottom). Example: *Crowd* (frame 238).

TABLE VI
SENSITIVITY TO THE INSERTION THRESHOLD τ_{ins} ON IPHONE_WANDERING.

τ_{ins}	0.1	0.3	**0.5**	0.7	0.9
Mean BG PSNR ↑	17.852	18.385	18.410	18.398	18.245
Tail BG MAE ↓	20.315	20.054	20.021	20.108	20.612

V. LIMITATIONS AND FUTURE WORK

Motion Permanence buys temporal consistency at the price of conservatism. By suppressing or skipping uncertain regions to keep dynamic content out of the map, MoPe can reduce Gaussian insertion density and limit fine-detail reconstruction, which slightly lowers pixel-level metrics such as PSNR and SSIM on some sequences—a deliberate trade of completeness for cleanliness. The propagation step also relies on reasonably accurate depth. Future work will relax both constraints: adaptive noise parameters for the Bayesian update to tolerate depth error, and lightweight attention to complement purely geometric warping, extending Motion Permanence to more unstructured environments.

VI. CONCLUSION

We addressed the memoryless limitation of uncertainty-aware monocular Gaussian mapping by introducing **Motion Permanence**: the principle that an object's dynamic identity should persist across time rather than be re-decided at every frame. Realized in MoPe through SE(3) propagation and bounded Bayesian log-odds fusion, this turns a transient per-frame uncertainty into a persistent dynamic state that guides tracking, mapping, insertion, and cleanup. By treating dynamic-ness as a temporal property rather than an instantaneous appearance cue, MoPe filters intermittent distractors and produces cleaner reconstructions, with its strongest gains on the dynamic-human scenes that most violate the memoryless assumption. More broadly, MoPe targets a representation bottleneck that precedes action: by keeping dynamic content out of the static map, Motion Permanence offers a step toward more reliable robot autonomy in dynamic environments.

References

[1] Berta Bescos, Jose M. Facil, Javier Civera, and Jose Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, October 2018. ISSN 2377-3774. doi: 10.1109/lra.2018.2860039. URL http://dx.doi.org/10.1109/LRA.2018.2860039.

[2] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J. Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5(2):721–728, April 2020. ISSN 2377-3774. doi: 10.1109/lra.2020.2965415. URL http://dx.doi.org/10.1109/LRA.2020.2965415.

[3] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras, 2024. URL https://arxiv.org/abs/2311.16728.

[4] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam, 2024. URL https://arxiv.org/abs/2312.02126.

[5] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision?, 2017. URL https://arxiv.org/abs/1703.04977.

[6] Mangyu Kong, Jaewon Lee, Seongwon Lee, and Euntai Kim. Dgs-slam: Gaussian splatting slam in dynamic environment, 2024. URL https://arxiv.org/abs/2411.10722.

[7] Yanyan Li, Youxu Fang, Zunjie Zhu, Kunyi Li, Yong Ding, and Federico Tombari. 4d gaussian splatting slam, 2025. URL https://arxiv.org/abs/2503.16710.

[8] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian splatting slam, 2024. URL https://arxiv.org/abs/2312.06741.

[9] Riku Murai, Eric Dexheimer, and Andrew J. Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors, 2025. URL https://arxiv.org/abs/2412.12392.

[10] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

[11] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguère, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals, 2019. URL https://arxiv.org/abs/1905.02082.

[12] Michael Strecke and Joerg Stueckler. Em-fusion: Dynamic object-level slam with probabilistic data association. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 5864–5873. IEEE, October 2019. doi: 10.1109/iccv.2019.00596. URL http://dx.doi.org/10.1109/ICCV.2019.00596.

[13] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras, 2022. URL https://arxiv.org/abs/2108.10869.

[14] Anh Thuan Tran and Jana Kosecka. Varsplat: Uncertainty-aware 3d gaussian splatting for robust rgb-d slam, 2026. URL https://arxiv.org/abs/2603.09673.

[15] Haocheng Wang, Yejun Shou, Lingfeng Shen, Shuai Li, and Yanlong Cao. Rgd-slam: Robust gaussian splatting slam for dynamic environments. *Pattern Recognition*, 175:113071, 2026. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2026.113071. URL https://www.sciencedirect.com/science/article/pii/S0031320326000348.

[16] Tianyu Yang, Shuangfeng Wei, Jingxuan Nan, Mingyang Li, and Mingrui Li. Bdgs-slam: A probabilistic 3d gaussian splatting framework for robust slam in dynamic environments. *Sensors*, 25(21), 2025. ISSN 1424-8220. doi: 10.3390/s25216641. URL https://www.mdpi.com/1424-8220/25/21/6641.

[17] Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments, 2025. URL https://arxiv.org/abs/2504.03886.

[18] Wancai Zheng, Linlin Ou, Jiajie He, Libo Zhou, Xinyi Yu, and Yan Wei. Up-slam: Adaptively structured gaussian slam with uncertainty prediction in dynamic environments, 2025. URL https://arxiv.org/abs/2505.22335.

[19] Fan Zhu, Yifan Zhao, Ziyu Chen, Chunmao Jiang, Hui Zhu, and Xiaoxi Hu. Dygs-slam: Realistic map reconstruction in dynamic scenes based on double-constrained visual slam. *Remote Sensing*, 17(4), 2025. ISSN 2072-4292. doi: 10.3390/rs17040625. URL https://www.mdpi.com/2072-4292/17/4/625.