ActiveLoopSplat: Active Indoor Exploration with Loop-Closing Gaussian SLAM

Nandan Natesan[1], Kornvik Tanpipat[1], Akshunn Trivedi[1], Saumit Vedula[1]

***Abstract*— LoopSplat provides dense 3D Gaussian Splatting (3DGS) SLAM with loop closure but operates only offline over pre-recorded sequences. ActiveSplat enables live active exploration but lacks loop closure, leaving it vulnerable to accumulated camera drift. We present ActiveLoopSplat, a live system running entirely within the Habitat simulator [1] that unifies these capabilities to allow an autonomous agent to actively explore indoor environments while maintaining a globally consistent map. Our architecture introduces three key improvements: (1) an optimized LoopSplat backend featuring linear-time loop closures and full SE(3) corrections to both poses and Gaussian orientations; (2) a live integration where a Voronoi-based active planner navigates directly using the growing Gaussian map with closures triggered at submap boundaries; and (3) a novel "Backtrack Revisit" strategy that detects camera drift via height shifts and autonomously navigates the agent back to reliably mapped regions to force loop closures. Evaluated on Matterport3D scenes, our system achieves average PSNR gains of 3.2–7.5 dB and substantially lower ATE RMSE compared to the baseline, demonstrating the effectiveness of actively scheduled loop closures in online 3DGS mapping. [Code] [Presentation]**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is a foundational capability for autonomous mobile robots, allowing them to navigate and understand unknown environments. Classical SLAM systems typically rely on sparse feature representations or basic occupancy grids. While effective for navigation, these maps are insufficient for the rich, photo-realistic reconstruction demanded by emerging downstream applications, such as embodied AI, scene understanding, and neural rendering. The introduction of 3D Gaussian Splatting (3DGS) has revolutionized this domain, enabling high-quality, real-time novel view synthesis via differentiable, tile-based rasterization. However, deploying an autonomous agent that can actively explore a scene while generating a dense, globally consistent 3DGS map remains a highly challenging open problem due to the compounded effects of occlusion, incomplete coverage, and long-term camera drift.

Recent advancements in 3DGS SLAM have sought to address the challenge of global consistency. LoopSplat [2] integrates 3DGS into a robust SLAM front-end, maintaining a sequence of local submaps that are stitched together via loop closure and pose graph optimization. By identifying previously visited locations, LoopSplat successfully eliminates accumulated drift and aligns the global map. However, this system is inherently passive and operates purely offline over pre-recorded trajectory sequences. It lacks the intelligence to make decisions about where a robot should move next to optimally discover unmapped regions of an environment.

Conversely, ActiveSplat [3] tackles the problem of autonomous exploration directly. It proposes an active exploration strategy operating iteratively over a live, continuously growing 3DGS map. By utilizing a Voronoi-based active planner, ActiveSplat computes optimal navigation actions to maximize geometric and visual scene coverage within a fixed step budget. Yet, ActiveSplat is strictly an odometry-based mapping system; it does not perform loop closure. Over extended exploration trajectories, inevitable and accumulated camera drift irreparably degrades the global consistency of the map, limiting the system's viability in expansive indoor spaces.

Recognizing that these two paradigms are complementary, we present **ActiveLoopSplat**, a unified live system that bridges active coverage with globally consistent reconstruction. Operating entirely within a single interactive process in the Habitat simulator, our system actively explores an environment while aggressively managing drift. We introduce three core contributions: (1) an optimized LoopSplat backend featuring linear-time loop closures and full SE(3) orientation corrections applied explicitly to the Gaussian primitives; (2) a live integration where the Voronoi-based planner navigates over the evolving map with loop closures triggered strategically at submap boundaries; and (3) a novel "Backtrack Revisit" exploration policy that autonomously detects camera tracking drift and forces the agent to navigate back to reliably mapped regions to close loops.

II. Related Work

A. 3D Gaussian Splatting SLAM

KinectFusion [4] demonstrated real-time dense reconstruction by fusing depth frames into a TSDF voxel grid with ICP-based tracking, establishing the template for dense SLAM. However, its fixed voxel volume, lack of photorealistic color, and absence of loop closure cause inconsistency over long trajectories. Neural implicit methods addressed some of these gaps: iMAP [5] encoded full rooms in a single MLP but suffered from catastrophic forgetting, while NICE-SLAM [6] introduced hierarchical grid-based encoding for scalable local updates. Both, however, rely on expensive volumetric ray marching unsuitable for real-time robotic planning. 3D Gaussian Splatting (3DGS) [7] replaced volume rendering with tile-based rasterization of anisotropic

This work was completed as part of the University of Michigan Robotics course ROB 530: Mobile Robotics.

[1]Department of Robotics, University of Michigan, Ann Arbor, MI, 48109, USA {nandann, akshunn, kornvik, savedula}@umich.edu

Gaussian primitives, enabling real-time photorealistic synthesis. SplaTAM [8] first applied 3DGS to dense RGB-D SLAM with silhouette-guided optimization; GS-SLAM [9] added adaptive Gaussian expansion with coarse-to-fine pose estimation; and MonoGS [10] extended the paradigm to monocular inputs. RTG-SLAM [11] further improved efficiency through an opaque/transparent Gaussian decomposition and a stable/unstable categorization that freezes converged primitives, achieving 17.9 fps at half the memory of prior methods. These systems collectively demonstrated that explicit Gaussians offer faster convergence, superior rendering, and—crucially—direct primitive manipulation for downstream correction, but none incorporated loop closure or active exploration.

LoopSplat [2] builds on this foundation by integrating 3DGS natively into a real-time RGB-D SLAM front-end. Tracking and mapping operate concurrently: the tracker continuously estimates incoming camera poses by performing differentiable rasterization against the existing map to minimize photometric loss, while the mapper refines the positions, colors, and opacities of the underlying Gaussian primitives. To prevent catastrophic forgetting and boundless memory scaling, LoopSplat dynamically divides the environment into a sequence of independent, localized submaps. When the system detects significant camera motion, the active submap is frozen and a new one is initialized. While this local strategy preserves high-frequency structural details within active regions, the disconnected submap coordinate frames inherently accumulate structural drift over time.
To restore global consistency, LoopSplat identifies visual overlaps with historical submaps using NetVLAD [12], registers the geometry, and executes Pose Graph Optimization (PGO). However, the baseline implementation exhibits significant scaling and fidelity bottlenecks. The place recognition module scales quadratically as it continuously queries redundant historical submaps. Furthermore, resulting PGO corrections primarily execute spatial translations rather than computing rigorous full SE(3) orientation corrections (quaternions) for the Gaussians, nor do they uniformly align every camera pose within a submap's timeframe, occasionally leaving artifacts during large drift corrections.

B. Active Exploration and Reconstruction

Frontier-based exploration [13] established the foundational paradigm for autonomous coverage: the robot iteratively navigates toward boundaries between explored and unexplored space. While simple and effective, classical frontier methods rely on discrete occupancy grids and lack adaptive granularity for geometrically complex environments. Recent work integrated neural representations into the exploration loop. ActiveNeRF [14] augmented a NeRF MLP with learned variance to greedily select views that maximally reduce rendering uncertainty, demonstrating that active learning over neural scenes substantially outperforms passive sampling. However, both operate in computationally expensive batch settings—re-optimizing the neural field after each image takes minutes—making them unsuitable for real-time robotic use.

FisherRF [15] addressed the speed gap by using Fisher Information to quantify per-Gaussian observation uncertainty, selecting next-best views at approximately 70 fps, though it operates as a standalone view selector rather than a full SLAM system. The computational bottleneck of NeRF-based active methods is the central motivation for moving to explicit Gaussian representations, where coverage and novelty can be estimated geometrically in milliseconds without neural inference.
ActiveSplat [3] extends 3DGS-based SLAM into the active domain. Operating continuously within simulated environments like Habitat, the system queries the live, evolving Gaussian map to dynamically plan navigation actions, aiming to maximize geometric and visual scene coverage within a strict exploration step budget. To bridge 3D photorealistic mapping with 2D navigation, ActiveSplat projects the 3D map into a 2D occupancy grid to extract safe traversal paths. The core logic relies on classical frontier-based exploration, detecting boundaries that separate safely mapped territory from unknown space. The active planner scores these frontier nodes by weighing the expected information gain (novelty) against the physical traversal distance, utilizing Dijkstra's algorithm across a Voronoi graph to rapidly and aggressively pursue unseen regions. However, prioritizing maximum scene coverage introduces a severe exploration-revisitation trade-off. By constantly directing the camera toward unfamiliar frontiers, the SLAM tracker is regularly starved of the established, dense geometric features required for stable pose optimization. In differentiable rendering frameworks, navigating into high-uncertainty regions induces unrecoverable camera tracking drift, often presenting as unphysical shifts in estimated camera height. Since ActiveSplat lacks a loop closure backend, this drift irreparably warps the global reconstruction over time, highlighting the need for a paired system capable of intentionally sacrificing short-term exploration to revisit known regions and anchor the map.

III. METHODOLOGY

Our system integrates two coordinated components running inside the Habitat simulator on Matterport3D scenes: (1) a live LoopSplat tracker, mapper, and loop closer, and (2) a Voronoi-based ActiveSplat frontier planner with a drift-triggered revisit mode. See Fig. 1 for a high level system overview.

A. Live Integration of LoopSplat and ActiveSplat

At each timestep t, the agent captures an RGB-D observation (I_t, D_t) from the Habitat simulator. The system processes each frame through three tightly coupled stages:
Tracking. The LoopSplat tracker estimates the camera-to-world pose $\mathbf{T}_t \in SE(3)$ by minimizing a weighted photometric and geometric loss against the current submap's Gaussians:

$$\mathcal{L}_{\text{track}} = w_c \cdot \mathcal{L}_{\text{color}} + (1 - w_c) \cdot \mathcal{L}_{\text{depth}} \tag{1}$$

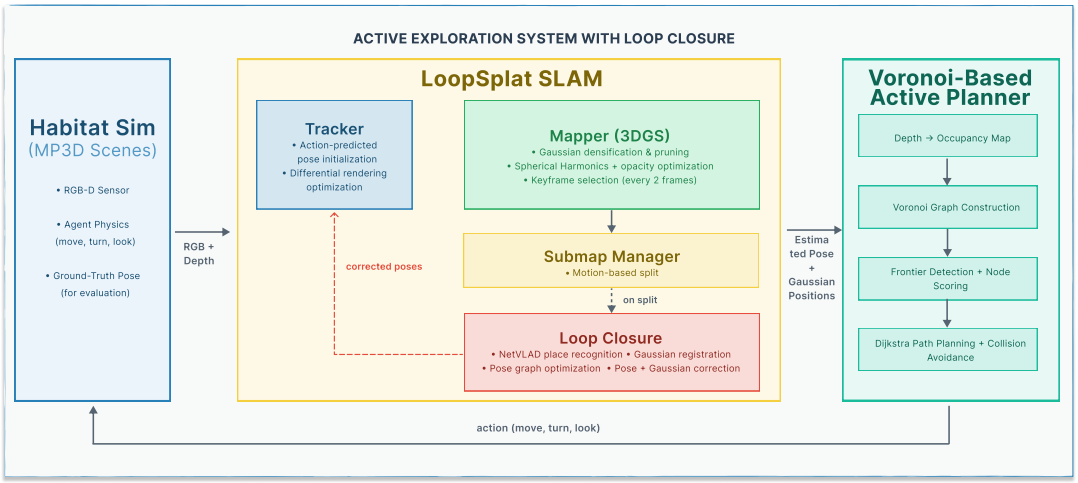

Fig. 1: ActiveLoopSplat System Architecture

where $\mathcal{L}_{\text{color}} = \|\hat{I}_t - I_t\|_1$ and $\mathcal{L}_{\text{depth}} = \|\hat{D}_t - D_t\|_1$ are computed over valid pixels, with $w_c = 0.95$. The camera pose is parameterized as a rotation quaternion $\mathbf{q} \in \mathbb{R}^4$ and translation $\mathbf{t} \in \mathbb{R}^3$, optimized for 60 gradient descent iterations per frame. Rather than relying on constant-speed extrapolation for pose initialization, which fails when the action type changes (e.g., turn → move_forward), we exploit the known discrete action a_{t-1} to compute the predicted pose $\hat{\mathbf{T}}_t$ analytically from the previous estimate $\mathbf{T}_{t-1}$ and the simulator's exact kinematic parameters (step size, turn angle). This action-predicted initialization is then added into LoopSplat's existing constant-speed extrapolation interface by constructing a synthetic previous pose $\mathbf{T}_{\text{fake}}$:

$$\mathbf{T}_{\text{fake}} = \mathbf{T}_{t-1} \cdot \hat{\mathbf{T}}_t^{-1} \cdot \mathbf{T}_{t-1} \tag{2}$$

such that the extrapolation $\mathbf{T}_{t-1} \cdot \mathbf{T}_{\text{fake}}^{-1} \cdot \mathbf{T}_{t-1} = \hat{\mathbf{T}}_t$.

Mapping. On every k-th frame (k=2 by default), the mapper refines the current submap's Gaussian primitives, positions $\boldsymbol{\mu}_i \in \mathbb{R}^3$, colors (Spherical Harmonics coefficients), scales $\mathbf{s}_i$, opacities α_i, and orientations $\mathbf{q}_i$, via differentiable rasterization. Each mapping step runs 300 optimization iterations with periodic densification (splitting and cloning under-reconstructed Gaussians based on accumulated view-space gradients) and pruning of low-opacity primitives ($\alpha < 0.1$).

Active Planning. Simultaneously, the Voronoi-based planner reads the current Gaussian positions $\{\boldsymbol{\mu}_i\}$ and projects them onto a 2D top-down occupancy grid $\mathcal{O} \in \{0,1\}^{G \times G}$ at 5 cm resolution. A Voronoi graph is constructed from obstacle contours, and frontier nodes—boundaries between mapped and unknown space—are scored by a weighted combination of flags:

$$S(n) = w_U \cdot \mathbb{1}_{\text{unarr}}(n) + w_H \cdot \mathbb{1}_{\text{horiz}}(n) + w_C \cdot C(n) + w_F \cdot \mathbb{1}_{\text{fail}}(n) \tag{3}$$

where w_U=20, w_H=10, w_C=2, w_F=−60 in explore mode, and $C(n)$ is a frontier density proxy for expected information gain. The planner selects the highest-scoring node, plans a collision-free path via Dijkstra's algorithm with fast-forward line-of-sight optimization, and converts waypoints into discrete Habitat actions (`move_forward`, `turn_left`, `turn_right`, `look_up`, `look_down`).

B. Submap Management and Loop Closure

Submap Splitting. The system monitors the cumulative camera motion since the last submap boundary. A new submap is triggered when both a minimum frame count ($N_{\min}$=50) and a motion threshold (rotation $>$ 120° or translation $>$ 1.0 m) are exceeded. We deliberately avoid drift-based splitting (e.g., triggering on high tracking loss), because active exploration naturally directs the camera toward novel, poorly-mapped regions that cause temporarily elevated loss without indicating real drift.

Loop Closure. At every submap boundary, the current submap is frozen and loop closure is attempted. NetVLAD descriptors are extracted for each keyframe in each submap. Cross-submap similarity is computed as:

$$\text{sim}(i,j) = \mathbf{d}_i^\top \mathbf{d}_j \tag{4}$$

and a match is accepted when the cross-similarity exceeds the minimum within-submap self-similarity. Matched submaps undergo pairwise geometric registration via Gaussian-based registration, and a Pose Graph Optimization (PGO) distributes the corrective transforms $\{\mathbf{C}_s\}_{s=0}^{S}$ across all S+1 submaps using Levenberg-Marquardt optimization [16].

Full SE(3) Gaussian Correction. Critically, we apply each correction transform $\mathbf{C}_s \in SE(3)$ not only to the Gaussian positions but also to their orientations. For each Gaussian i in submap s, the corrected position and orientation are:

$$\boldsymbol{\mu}_i' = \mathbf{C}_s \begin{bmatrix} \boldsymbol{\mu}_i \\ 1 \end{bmatrix}, \quad \mathbf{q}_i' = \mathbf{q}_{\mathbf{C}_s} \otimes \mathbf{q}_i \tag{5}$$

where $\mathbf{q}_{\mathbf{C}_s}$ is the quaternion extracted from $\mathbf{C}_s[:3,:3]$ and $\otimes$ denotes the Hamilton product. Additionally, every estimated camera pose within the corrected submap's frame range $[t^s_{\min}, t^s_{\max}]$ is updated:

$$\mathbf{T}_t' = \mathbf{C}_s \cdot \mathbf{T}_t, \quad \forall\, t \in [t^s_{\min}, t^s_{\max}] \tag{6}$$

C. Backtrack Revisit Mode

Pure frontier-driven exploration devoids the tracker of familiar features, causing progressive drift. We detect this via two complementary signals on the estimated pose $\mathbf{T}_t$:

1) **Vertical drift:** The agent operates on flat ground in Habitat, so the camera height y_t should remain approximately constant. A vertical shift $|\Delta y_t| = |y_t - y_{t-1}| > 15\,\text{cm}$ triggers a drift alert, this is a scene-invariant, physics-grounded signal that is immune to tracker overconfidence.
2) **Pose convergence instability:** If the tracker's pose standard deviation exceeds 1 cm in 2 out of 3 recent frames, a secondary drift alert is activated.

When drift is detected, the planner switches from explore mode to **backtrack revisit mode**. Rather than target a submap centroid which may require navigating through unverified geometry, we target the agent's own pose from $N{=}40$ frames earlier: $\mathbf{T}_{target} = \mathbf{T}_{t-N}$. This pose is by construction reachable, since the agent was just there. The planner routes toward it with the same Voronoi machinery used in explore mode but with `REVISIT_WEIGHTS` that favour already-visited cells ($w_U = -20$, $w_C = -2$). Upon arrival (within 1.0 m) the agent performs a narrow $\pm 45°$ scan—wide enough to expose the old viewpoint to hloc for place recognition, but short enough to avoid the parallax-free rotation that destabilised the tracker in earlier ablations. The next submap boundary then triggers loop closure with the agent sitting on overlap with a prior submap, so PGO produces a correction. If the target is not reached within 50 frames, or the scan completes without a new submap boundary, the planner exits revisit with a cooldown period so it does not immediately re-fire.

D. Tracker Robustness

To prevent the tracker from diverging when physically stuck against walls, we add two depth-based safety mechanisms:

1) **Collision guard:** Before executing a `move_forward` command, the planner checks the center-patch depth. If the minimum depth in the center 10×10 pixel patch satisfies $d_{\min} < d_{\text{step}} + 5\,\text{cm}$, the action is replaced with a random turn.
2) **Stuck-on-wall detector:** If the previous action was `move_forward` but the center depth did not decrease by at least half the expected step distance, the agent is physically stuck and a turn is forced, regardless of what the (potentially drifted) estimated position suggests.

IV. Experiments

We evaluate ActiveLoopSplat across two experimental setups : (1) a random walk baseline comparison against vanilla LoopSplat, and (2) an ablation study of the full Voronoi-based active exploration pipeline with and without loop closure and backtrack revisit.

A. Experimental Setup

Datasets. All experiments are conducted on four scenes from the Matterport3D (MP3D) dataset [17], rendered in real time using the Habitat simulator: `gZ6f7yhEvPG`, `pLe4wQe7qrG`, `YmJkqBEsHnH`, and `GdvgFV5R1Z5`. Refer Fig 2. MP3D provides photorealistic, large-scale indoor environments captured from real-world building scans, making them well-suited for evaluating SLAM systems under realistic visual conditions.

Simulator Configuration. The agent collects posed RGB-D data at a resolution of 512×512 with a horizontal field of view of $90°$. The agent height is set to 1.25 m. The discrete action space consists of five actions: `move_forward` by 6.5 cm (10 cm for the random walk protocol), `turn_left` and `turn_right` by $10°$, and `look_up` and `look_down` by $10°$. Ground-truth camera poses from the simulator are recorded for evaluation but are *not* used by the SLAM system during tracking.

Configurations. We evaluate three system configurations in the ablation study:

- **Without LC**: Voronoi-based active planner with Loop-Splat mapping and tracking, but loop closure disabled (`--no-lc`).
- **LC**: Full system with loop closure enabled at every submap boundary.
- **LC + Backtrack**: Full system with both loop closure and the backtrack revisit mode enabled (`--revisit`).

B. Evaluation Metrics

We adopt two standard metrics to evaluate trajectory accuracy and reconstruction quality:

ATE RMSE. The Absolute Trajectory Error (ATE) root-mean-square error measures the global consistency of the estimated camera trajectory against the ground-truth trajectory. For each frame t, the translational error is computed as:

$$\text{ATE}_t = \|\mathbf{p}_t^{\text{est}} - \mathbf{p}_t^{\text{gt}}\|_2 \tag{7}$$

and the overall ATE RMSE is:

$$\text{ATE RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N} \|\mathbf{p}_t^{\text{est}} - \mathbf{p}_t^{\text{gt}}\|_2^2} \tag{8}$$

Lower values indicate better global pose accuracy.

Average PSNR. Peak Signal-to-Noise Ratio (PSNR) measures the photometric fidelity of the rendered Gaussian map against the ground-truth RGB observations at each mapping keyframe:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{1}{\text{MSE}}\right) \tag{9}$$

where $\text{MSE} = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} \|\hat{I}(p) - I(p)\|^2$ over all pixels $\mathcal{P}$. Higher values indicate better reconstruction quality.

Fig. 2: Indoor scenes from Matterport3D dataset rendered in Habitat simulator

TABLE I: Random walk evaluation on 4 MP3D scenes (300 steps). Our improved LoopSplat backend achieves substantial PSNR gains and lower ATE RMSE across all scenes.

Scene	LoopSplat		Ours	
	Avg PSNR	ATE RMSE	Avg PSNR	ATE RMSE
gZ6f7	17.24 dB	4.41 m	**24.71 dB**	**0.33 m**
pLe4w	22.35 dB	3.29 m	**28.07 dB**	**1.01 m**
YmJkq	23.28 dB	**0.61** m	**28.48 dB**	0.68 m
GdvgF	21.21 dB	1.42 m	**26.28 dB**	**0.75 m**

C. Baseline Comparison: Random Walk

To isolate the effect of our improved LoopSplat backend from the active planning, we first evaluate under a *random walk* policy where both systems receive identical, randomized trajectories. The agent executes 300 steps, randomly selecting from `move_forward`, `turn_left`, `turn_right`, `look_up`, `look_down` at each step. As shown in Table I, our improved backend achieves consistent PSNR gains of 5.1–7.5 dB across all four scenes, demonstrating that the enhanced mapping pipeline (densification, scale tuning, and trainable color features) produces substantially higher-fidelity Gaussian maps. Trajectory accuracy also improves dramatically: ATE RMSE drops from 4.41 m to 0.33 m on `gZ6f7` (a 92% reduction) and from 3.29 m to 1.01 m on `pLe4w`. On `YmJkq`, the ATE RMSE increases slightly (0.61 m $\rightarrow$ 0.68 m), which we attribute to the aggressive densification introducing more Gaussians in poorly-textured regions that momentarily confuse the tracker. Overall, the random walk results confirm that the backend improvements alone, independent of the active planner, yield major gains in both metrics.

D. Ablation: Active Exploration with Loop Closure

We next evaluate the full active exploration system using the Voronoi-based planner over 700 steps, ablating the contributions of loop closure and backtrack revisit. Table II reports several important findings:

Loop Closure Impact. Enabling loop closure produces dramatic improvements on `gZ6f7` and `GdvgF`: ATE RMSE drops from 1.52 m to 0.05 m on `gZ6f7` (a 97% reduction) and from 0.78 m to 0.03 m on `GdvgF` (a 96% reduction), accompanied by PSNR gains of 8.6 dB and 1.8 dB respectively.

Loop Closure Failure Cases. On `pLe4w` and `YmJkq`, enabling loop closure *increases* ATE RMSE (0.31 m $\rightarrow$ 0.92 m on `pLe4w`; 0.02 m $\rightarrow$ 0.78 m on `YmJkq`). This occurs when the PGO applies an incorrect correction due to ambiguous place recognition matches, a known challenge in repetitive indoor environments with visually similar corridors. In these cases, the Without LC configuration benefits from naturally low drift and avoids the erroneous corrections entirely.

Backtrack Revisit. The LC + Backtrack configuration resolves the failure case on `pLe4w`, reducing ATE RMSE from 0.92 m (LC only) to 0.09 m while simultaneously achieving the highest PSNR of 36.1 dB. The backtrack revisit mode forces the agent to return to a well-mapped region before the next submap boundary, ensuring the loop closure receives a high-quality visual match rather than an ambiguous one. On `gZ6f7`, `YmJkq`, and `GdvgF`, the backtrack revisit did not trigger (marked with *), as the vertical drift threshold was not exceeded during those runs.

E. Discussion

Across both experimental protocols, the results demonstrate two clear trends. First, the improved mapping backend (densification, extended optimization, trainable SH colors) consistently improves PSNR by 3–9 dB over the vanilla LoopSplat baseline, regardless of the exploration policy. Fig 3 compares one of the rendered 3DGS image with the ground truth image in Habitat simulator. Second, loop closure is highly effective when the agent's trajectory naturally revisits mapped regions (`gZ6f7` and `GdvgF`), but can degrade performance when place recognition produces false matches in visually ambiguous environments (`pLe4w` and `YmJkq`). The backtrack revisit mode mitigates these failures by ensuring high-quality visual overlap before triggering loop closure, as demonstrated by the substantial recovery on `pLe4w`. These findings highlight the fundamental tension in active exploration with loop closure: aggressive frontier-seeking improves coverage but reduces the natural revisitation that loop closure depends on. Our backtrack revisit strategy offers a principled resolution by monitoring physical drift signals and autonomously scheduling revisits when needed, without sacrificing overall exploration efficiency.

Our presentation is available at [Link]

TABLE II: Indoor exploration with Voronoi-based planner (4 MP3D scenes, 700 steps). LC = Loop Closure. *Backtrack revisit did not trigger in this run.

Scene	Without LC		LC		LC + Backtrack	
	Avg PSNR	ATE RMSE	Avg PSNR	ATE RMSE	Avg PSNR	ATE RMSE
gZ6f7	25.3 dB	1.52 m	33.9 dB	0.05 m	**33.9 dB**	**0.05 m***
pLe4w	35.4 dB	0.31 m	34.2 dB	0.92 m	**36.1 dB**	**0.09 m**
YmJkq	**34.3 dB**	**0.02 m**	33.3 dB	0.78 m	33.3 dB	0.78 m*
GdvgF	32.9 dB	0.78 m	34.7 dB	0.03 m	**34.7 dB**	**0.03 m***

Fig. 3: Rendered 3DGS scene (left) vs Ground Truth scene (right)

V. Conclusion and Future Work

We presented ActiveLoopSplat, a unified system that combines active frontier-based exploration with globally consistent 3D Gaussian Splatting SLAM via online loop closure. By tightly integrating LoopSplat's submap-based mapping and tracking with ActiveSplat's Voronoi-based planner inside the Habitat simulator, our system enables an autonomous agent to explore unknown indoor environments while continuously correcting accumulated drift. We introduced three key contributions: (1) an improved LoopSplat backend with full SE(3) Gaussian corrections that jointly transforms both positions and orientations; (2) a live integration where loop closure is triggered at every submap boundary during active exploration; and (3) a backtrack revisit strategy that detects camera drift via physically grounded signals, such as unphysical vertical motion, and autonomously navigates the agent to well-mapped regions to force high-quality loop closures.

Evaluated on four Matterport3D scenes, ActiveLoopSplat achieves PSNR gains of 5–9 dB over vanilla LoopSplat under random walk and reduces ATE RMSE by up to 97% when loop closure fires correctly. Our ablation study further demonstrates that the backtrack revisit mode recovers from erroneous loop closures caused by ambiguous place recognition, reducing ATE RMSE from 0.92 m to 0.09 m on pLe4w while simultaneously achieving the highest PSNR of 36.1 dB.

Limitations and Future Work. Several directions remain open for future investigation. First, we conducted test runs only once for each of the four scenes, since each one of them took five to six hours on compute available to us. This means we do not account for stochasticity in the results due to trajectory initialization and executed plans. More exhaustive evaluations with higher number of runs for each scene can help us get more statistically rigorous results. Second, our system currently operates on static Matterport3D scenes. Extending the pipeline to dynamic environments, for instance, by incorporating a YOLO-based object detector [18] coupled with FastSAM [19] for semantic segmentation and optical flow for motion confirmation, would enable robust dynamic region suppression and broaden applicability to real-world settings with moving occupants. Third, the NetVLAD-based place recognition occasionally produces false positive matches in visually repetitive environments, as observed on pLe4w and YmJkq. Replacing or augmenting it with learned visual place recognition methods such as MixVPR [20] could improve loop closure reliability. Fourth, deploying ActiveLoopSplat on a physical robot platform with noisy sensor inputs and real-time computational constraints remains an important step toward practical embodied mapping. Finally, the current backtrack revisit trigger relies on a fixed vertical drift threshold; an adaptive, learned threshold that accounts for scene geometry and tracker confidence could further improve the exploration-revisitation balance.

References

[1] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, "Habitat 2.0: Training home assistants to rearrange their habitat," 2022. [Online]. Available: https://arxiv.org/abs/2106.14405

[2] L. Zhu, Y. Li, E. Sandström, S. Huang, K. Schindler, and I. Armeni, "Loopsplat: Loop closure by registering 3d gaussian splats," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2025.

[3] Y. Li *et al.*, "Activesplat: High-fidelity scene reconstruction through active gaussian splatting," *IEEE Robotics and Automation Letters*, vol. 10, no. 8, pp. 8099–8106, 2025.

[4] S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. J. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time dynamic 3d surface reconstruction and interaction," in *ACM SIGGRAPH 2011 Talks*, ser. SIGGRAPH '11. New York, NY, USA: ACM, 2011, pp. 23:1–23:1. [Online]. Available: http://doi.acm.org/10.1145/2037826.2037857

[5] E. Sucar, S. Liu, J. Ortiz, and A. Davison, "iMAP: Implicit mapping and positioning in real-time," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[6] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 776–12 786.

[7] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, 2023.

[8] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat, track map 3d gaussians for dense rgb-d slam," 2024. [Online]. Available: https://arxiv.org/abs/2312.02126

[9] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," in *CVPR*, 2024.

[10] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian splatting slam," 2024. [Online]. Available: https://arxiv.org/abs/2312.06741

[11] Z. Peng, Y. Yang, T. Shao, C. Jiang, and K. Zhou, "X-slam: Scalable dense slam for task-aware optimization using csfd," *ACM Transactions on Graphics*, vol. 43, no. 4, p. 1–15, 2024. [Online]. Available: http://dx.doi.org/10.1145/3658233

[12] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," 2016. [Online]. Available: https://arxiv.org/abs/1511.07247

[13] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, 1997, pp. 146–151.

[14] X. Pan, Z. Lai, S. Song, and G. Huang, "Activenerf: Learning where to see with uncertainty estimation," 2022. [Online]. Available: https://arxiv.org/abs/2209.08546

[15] W. Jiang, B. Lei, and K. Daniilidis, "Fisherrf: Active view selection and uncertainty quantification for radiance fields using fisher information," 2024. [Online]. Available: https://arxiv.org/abs/2311.17874

[16] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.

[17] A. Chang *et al.*, "Matterport3D: Learning from RGB-D data in indoor environments," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2017.

[18] Ultralytics, "YOLO11," https://github.com/ultralytics/ultralytics, 2024, accessed: 2024.

[19] X. Zhao *et al.*, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.

[20] A. Ali-bey, B. Chaib-draa, and P. Giguère, "Mixvpr: Feature mixing for visual place recognition," 2023. [Online]. Available: https://arxiv.org/abs/2303.02190