PixelLoop: Shortcut Topological Navigation with Pixel-Level Loops

Sarthak Chittawar[1†], Vansh Garg[1], Aditya Vadali[1], Krish Pandya[1]
Rohit Jayanti[1], Sourav Garg[1], and Madhava Krishna[1]

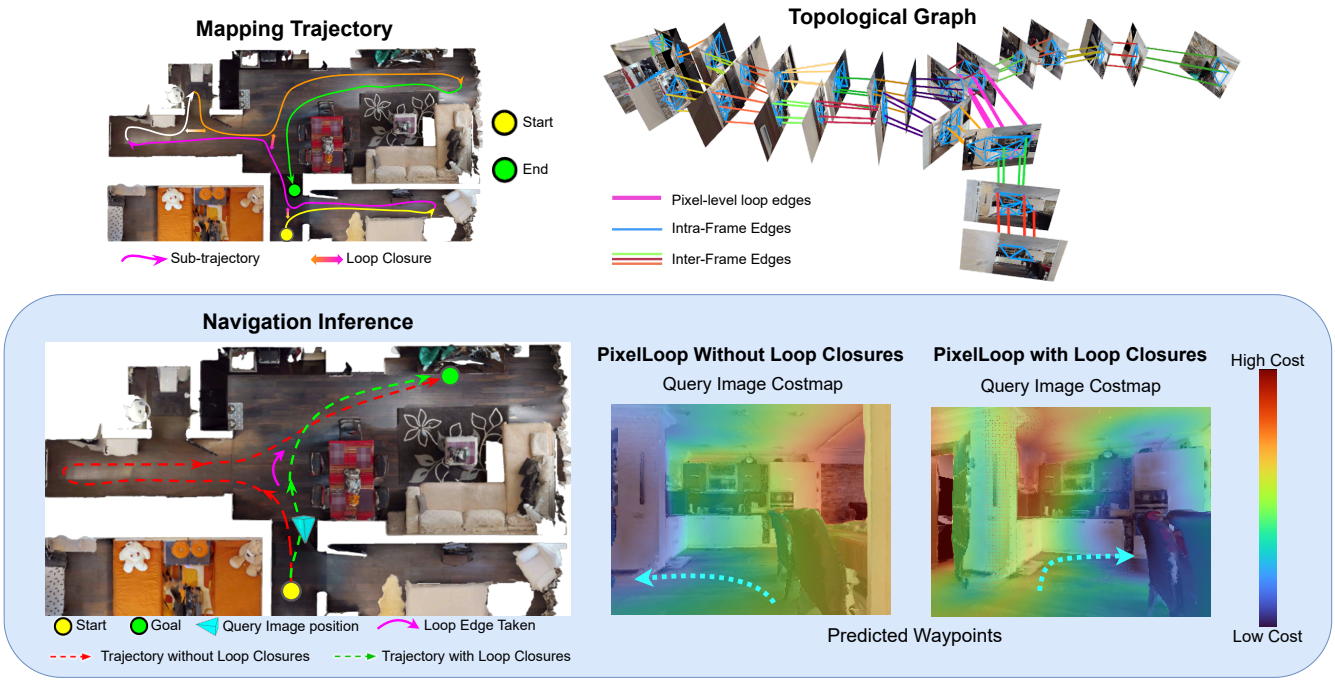

Fig. 1: **Top:** Start-to-goal navigation over large-scale mapping runs composed of multiple teach-and-repeat sub-trajectories (shown in different colours) is challenging without loop closures. PixelLoop detects revisited regions and establishes pixel-level loop edges, integrating them into a densely loop-connected (shown as magenta edges) topological graph. **Bottom:** The resulting costmaps enable direct start-to-goal paths (green) via loop closures, whereas the no-loop configuration (red) produces longer detours. *(Absolute poses are used for visualization only; our method operates entirely in relative 3D space.)*

***Abstract*— Although topological mapping and navigation have been studied extensively, the specific role and downstream effect of loop closures in purely topological representations has received relatively little attention. Importantly, loop closure over topological maps is distinct from loop closure over globally referenced trajectories and metric maps. Building on recent denser topologies grounded in pixel-level, relative 3D geometry, we propose PixelLoop which introduces loop closures directly in pixel space. Unlike sparse image-level edges or pose-graph corrections in SLAM, our pixel-level closures act as dense topological shortcuts that alter planning connectivity and cost propagation rather than merely aligning coordinates. This dense connectivity enables stable any-point-to-any-point navigation and produces costmaps that align accurately with geometric shortest paths. In particular, we showcase the distinct advantage of applying loop closures to fine-grained pixel topologies rather than image-level topologies. Across extensive simulated experiments, PixelLoop achieves over 35% absolute improvement in both Success Rate and SPL compared to image-relative baselines, with the largest gains in scenarios requiring shortcut exploitation. Results are further validated through real-world mobile robot deployments, demonstrating that dense pixel-level loop closures provide a practical and robust foundation for topological visual navigation.**
Project Page: https://pixelloop-nav.github.io/

The authors acknowledge the support of Ati Motors for this project.
† Corresponding Author.
[1] Robotics Research Center, IIIT-Hyderabad, India

I. INTRODUCTION

Visual navigation has recently advanced through image-conditioned policies and trajectory-centric topological representations that bypass explicit, globally consistent 3D maps [1], [2], [3], [4]. A more recent method, MASt3R-Nav [5], has demonstrated that pixel-level geometric correspondences can be leveraged to construct topological graphs grounded in 3D relative mapping, enabling robust path planning and control. However, topological navigation is typically constrained to *teach-and-repeat* settings through a one-dimensional representation of a route as an image sequence, restricting navigation to replaying or locally adapting a single trajectory. In the absence of globally grounded reference

frames, relative distances between two topologies become constrained or biased based on the order in which they appear in the reference run. These sequential biases carry forward to subsequent inference runs, making trajectories to goal queries unnaturally long with increasing failure rates.

The core limitation lies in topological connectivity. Topology induced maps, such as a sequential collection of images (without a grounded pose reference), encode sequential adjacency but lack mechanisms to structurally link regions that are geometrically proximate but temporally distant. Consequently, naively planning paths along such a sequence forces the agent to retrace convoluted paths rather than taking obvious physical shortcuts. As illustrated in Fig. 1, multiple teach-and-repeat sub-trajectory runs (shown in different colours) remain disconnected without loop closures. Two rooms separated by a single wall (or common space) might be adjacent in 3D space but hundreds of frame-nodes apart in a purely sequential graph, forcing a severe detour (red trajectory). The fundamental problem is that the geometry of the planning graph reflects historical trajectory rather than true shortest-path structure.

Furthermore, the role of loop closures in topological representations remains largely unexplored. Unlike globally-referenced metric SLAM systems, loop closures in topological maps strongly affect the planning structure. In this work, we study this phenomenon in the context of fine-grained pixel-level topology and demonstrate that loop closures applied at the pixel level provide substantially stronger navigation performance than closures applied over image-level topologies.

We propose **PixelLoop**, an arbitrary start-to-goal navigation stack built upon 3D relative mapping. Using a pixel-relative topological map representation, we propose ***pixel-level* loop closures** that *directly* enable relative-geometry grounded pixel-level planning of shorter paths. We establish these closures directly in pixel space, inserting them as zero-cost edges between geometrically grounded 3D pixel nodes. This seamlessly stitches overlapping topological regions into a unified spatial manifold, defined in a local relative 3D space, without requiring globally consistent reconstruction or pose graph optimization, as illustrated in Fig. 1.

This is in stark contrast with the typical role of loop closures in existing literature, where *a)* in metric SLAM pipelines they typically account for drift in globally-referenced geometry, and *b)* in image-topological pipelines they provide shortcuts but without any form of geometric leverage. As our topological shortening is densely grounded in pixel space, it yields dense pixel-level costmaps that encode accurate path planning costs. These costmaps are then used to condition a learnt-controller, providing the continuous geometric signals necessary for precise control prediction, an affordance that sparse image-relative baselines fundamentally lack [2], [6], [7], [8], [9]. Through extensive simulated experiments, we demonstrate that PixelLoop achieves an absolute improvement of over 35% in both Success Rate and Success weighted by Path Length (SPL) compared to trajectory-bound baselines. Crucially, these simulation gains translate directly to physical hardware, with real-world mobile robot deployments demonstrating the practical efficacy of our approach.

In summary, this paper makes the following key contributions: *i)* We introduce ***pixel-level* loop closures** within a 3D relative mapping framework, seamlessly stitching disjoint topologies into a unified spatial manifold without global metric reconstruction. *ii)* We present a first-of-its-kind analysis of ground truth/simulator based loop detection and closures for the navigation task across image-, object- and pixel-level topological map representations.

II. RELATED WORK

Visual navigation research spans various representations, but our focus lies on topological frameworks that relax the need for explicit global metric SLAM [10], [11]. We position our work by examining how existing map-less paradigms operate based on their node representations—image, object, and pixel—how they attempt to scale to arbitrary start-to-goal (A→B) navigation, and why our dense loop closures provide a fundamentally superior scaling mechanism.

a) Image-Relative Topological Methods: To avoid the computational overhead of globally consistent mapping, recent navigation models have widely adopted purely image-relative topological representations [6]. Methods such as ViNG [12] and General Navigation Model (GNM) [2] completely discard explicit metric poses and global 3D maps. Instead, they abstract the environment into a graph where nodes are discrete images and edges denote visual similarity or predicted temporal proximity [13]. To scale beyond simple teach-and-repeat trajectories [14], [15] and support arbitrary A→B navigation, these map-less frameworks attempt to establish loop closures by adding discrete scalar edges between visually similar but temporally distant views [16], [17]. However, because connectivity is defined purely at the image level, these closures lack explicit geometric grounding. Handing the controller an entire discrete image as a sub-goal creates a severe information bottleneck, as it lacks the fine-grained, continuous geometric context necessary to smoothly steer through novel junctions [18], [19].

b) Object-Relative Topological Methods: An alternative line of work seeks to ground navigation in semantically meaningful entities [20], [21]. Methods like ObjectReact [1] and RoboHop [3] construct topological maps where nodes correspond to distinct objects or semantic regions, rather than whole images. These formulations can naturally link the same object observed across different traversals, forming semantic loop closures that aid in global route planning [22], [23]. While object-relative reasoning provides more localized sub-goals than full images [24], [25], it inherently abstracts away the underlying structural geometry. The resulting path planning costs remain coarse, discarding the fine-grained geometric gradients (e.g., free space around the object) required for precise, collision-free control prediction through complex intersections.

c) Pixel-Relative Navigation and Loop-Augmented Scaling: Recent geometry-aware approaches, such as

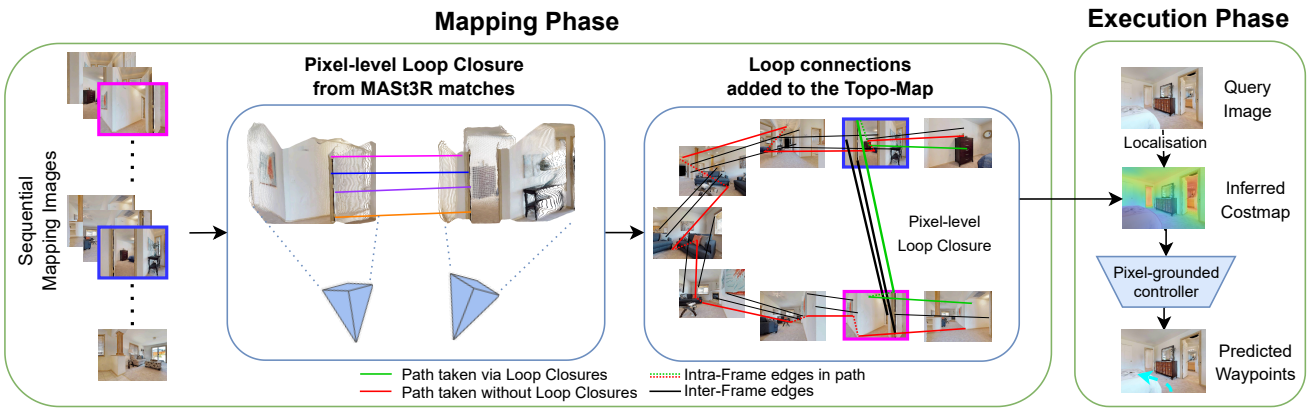

Fig. 2: **Overview of the PixelLoop navigation pipeline. Left:** Offline mapping. Sequential images are processed with MASt3R to construct a topological graph with pixel-level loop closures. **Right:** Online execution. The query image is localised and matched against the graph to generate a costmap that conditions the controller for waypoint prediction. *Figure best viewed when zoomed in.*

MASt3R-Nav [5], introduce denser topologies grounded in pixel-level relative 3D geometry [26], [27]. These representations preserve local spatial structure without demanding global metric consistency, but prior formulations are strictly constrained to single-run, teach-and-repeat trajectories, lacking any mechanism to reconcile multiple runs or exploit spatial overlap. **PixelLoop** bridges this critical gap. We argue that loop closures are a superior mechanism for scaling topological maps to A→B navigation, provided they are executed with geometric fidelity. By introducing loop closures directly in pixel space, we establish dense zero-cost edges between geometrically grounded 3D pixel nodes across image pairs. Unlike image-level loop closures that act as sparse proxy links, or object-level loop closures that discard structural nuances, our pixel-level loop closures directly reshape the geometric planning costmap. This fundamental structural shift provides the continuous geometric signals necessary for precise control prediction, enabling the robust discovery and exploitation of physically meaningful shortcuts across large environments.

III. APPROACH

A. Background: Pixel Relative Navigation

Recent work introduced MASt3R-Nav, a visual navigation framework that leverages dense 3D correspondences to construct a relative-geometry grounded, pixel-level topological graph. Our complete pipeline (Fig. 2) is fundamentally built upon this architecture. By explicitly relying on this pixel-level 3D map representation, MASt3R-Nav preserves fine grained local geometry without demanding global metric consistency.

1) Pixel-Based Topological Mapping: The foundation of this framework is the offline construction of a pixel-level topological graph from a sequence of reference images. During this offline mapping phase (Fig. 2 (Left)), incoming images are processed by MASt3R [26], a relative-geometry grounded image matching model. This model gives dense pixel correspondences and 3D pointmaps in a relative frame of reference between sequentially connected frames.

Specifically, for a given frame I_t, correspondences are computed within a temporal window W. These matches form the basis of the topological map $G = \{N, E\}$:

1) **Nodes (N):** Each pixel successfully matched to another image is instantiated as a node in the graph.
2) **Inter-frame Edges:** Pixels matched across different frames are connected by zero-cost edges, as they represent the same underlying 3D point observed from varying viewpoints. These edges are represented in black colour in Fig. 2 (Left).
3) **Intra-frame Edges:** To allow path traversal through pixels within a single image's coordinate frame, edges are formed between distinct pixel nodes belonging to the same image. The cost assigned to these edges is the 3D Euclidean distance between their corresponding relative 3D coordinates as calculated using MASt3R pointmaps. To maintain computational tractability while preserving geometric structure, these connections are pruned to a Euclidean Minimum Spanning Tree (EMST), denoted as dotted lines in Fig. 2 (Left) and light-blue edges in Fig. 3 (Middle).

2) Execution Phase: During execution (Fig. 2 (Right)), the agent matches the current observation against the topological graph to generate a WayPixel [5] costmap. A learned controller conditioned on this costmap then predicts a rollout of future 2D waypoints defined by their relative position and yaw in bird's-eye-view (BEV) space.

B. PixelLoop

In this section, we present our proposed pixel-level loop closure method, dubbed *PixelLoop*, where we first describe *loop detection*, which is composed of sequential descriptors based global retrieval and covisibility based verification using UniFlowMatch [29] (UFM), and then present *loop closure* and discuss in detail how pixel-level closures differ from the typical use and purpose of closures in globally-registered metric maps and image-level topological maps.

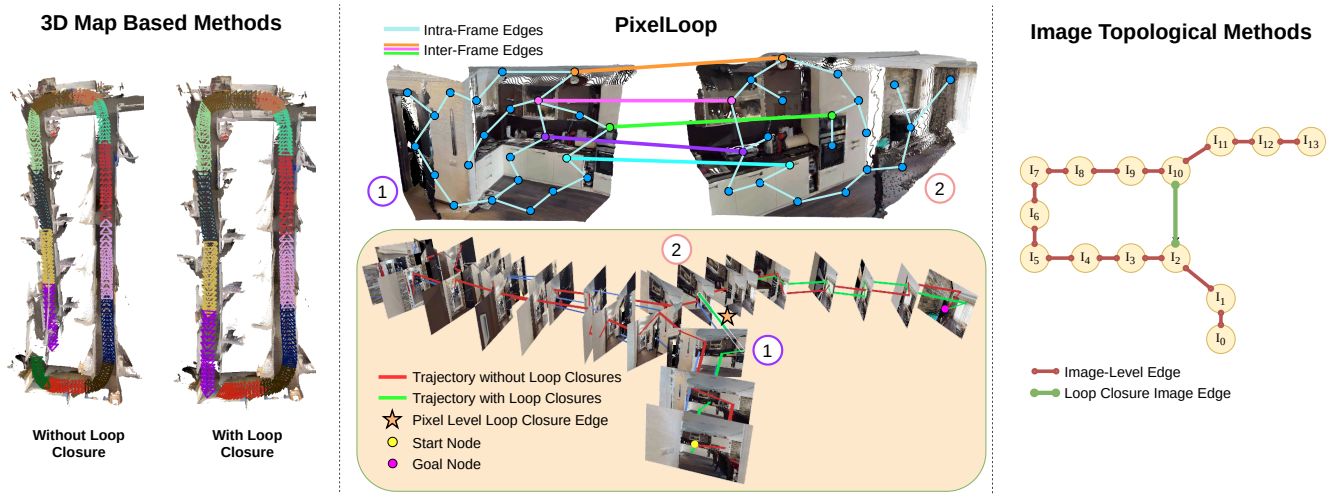

Fig. 3: **Impact of loop closures across different map representations. Left:** 3D map-based methods use loop closures primarily to correct accumulated drift (visualized using VGGT-SLAM 2.0 [28].) **Middle:** Our pixel-topological method establishes dense geometric shortcuts between loop closure candidates (① and ②), enabling the agent to bypass the indirect sequential route (red) and take a shorter, direct path (green). **Right:** Image-topological methods establish a single discrete edge between image nodes (e.g., I_2 and I_{10}), lacking fine-grained geometric cues. *(Absolute poses are used for visualization only; our method operates entirely in relative 3D space.)*

1) Loop Detection: While the standard mapping process connects sequential frames locally, it inherently lacks the ability to recognize when the camera revisits a previously mapped area. To address this, we identify temporally distant but geometrically overlapping views and establish loops between them.

a) Sequence Image Descriptors and Candidate Selection: The first stage identifies broad structural similarities across the environment. Instead of relying on single-image descriptors, which are highly susceptible to local visual ambiguity, we extract robust sequence representations. We aggregate visual features over a sliding temporal window of 5 frames using a pretrained SeqVLAD [30] model and compute a pairwise cosine similarity matrix between all frames. We define the set of initial loop closure candidates $\mathcal{C}_{cand}$ by applying a similarity threshold of 0.4. To avoid matches between temporally adjacent frames that are already connected in the mapping window, we apply a temporal exclusion window of ± 3 frames around each query frame.

b) Dense Covisibility Check: To verify the candidate pairs in $\mathcal{C}_{cand}$, we perform a bidirectional covisibility evaluation using UFM [29]. Let I_i denote the i^{th} frame. First, UFM predicts a dense probability map assigning a visibility confidence to each pixel. A pixel in one image is classified as visible in the other if this confidence exceeds a fixed threshold of 0.9. We then calculate the fraction of these visible pixels in both directions: from I_i to I_j and from I_j to I_i. The final symmetric covisibility score, $\nu_{i,j}$, is defined as the minimum of these two directional fractions. A pair is successfully verified only if $\nu_{i,j} > 0.1$, ensuring sufficient 3D overlap between the two views.

This process yields a reliable set of loop closure candidates. Importantly, these detected loop image pairs are agnostic to the underlying mapping method. They establish a set of spatial connections that can be directly applied to any map representation, whether it is a global 3D metric map, an image-topological graph, or a pixel-topological graph.

2) Loop Closure: Once valid candidate pairs are identified, they must be integrated into the underlying map. While the detected loops themselves are representation-agnostic, the act of closing a loop alters different map representations in distinct ways. Consequently, these structural modifications dictate how the loops ultimately impact downstream path planning and navigation.

To contextualize our approach, we contrast how loop closures manifest across three distinct mapping paradigms (illustrated in Fig. 3):

a) Global 3D Maps: In traditional metric SLAM, loop closures primarily serve to reduce accumulated drift (Fig. 3 (Left)). They act as constraints in a pose graph optimization framework to reduce trajectory errors or jointly correct both global pose and map inconsistencies in the SLAM backend. Because the representation remains a rigid metric grid of occupied and free space, the impact on path planning is secondary, and loop closures mainly improve localization and mapping accuracy.

b) Image-Relative Topological Maps: In purely topological frameworks, loop closures typically establish a single edge between discrete image nodes (Fig. 3 (Right)). Path planning relies on proxy metrics, such as predicted temporal distances, which depend on the model's implicit ability to infer spatial relationships and scene overlap. This reliance on implicit learning makes the system highly susceptible to perceptual aliasing and broader failures in the underlying image-matching task. Furthermore, providing the controller with an entire discrete image as a sub-goal creates a significant information bottleneck. The controller lacks fine-grained geometric cues indicating which specific regions of the image

actually lead towards the target, often resulting in imprecise steering or navigation failures.

c) Pixel-Relative Topological Maps: By leveraging MASt3R's dense correspondences, we establish numerous zero-cost inter-frame edges between matched pixels of I_i and I_j. As visualized by the star edge in Fig. 3 (Middle), this effectively converts a standard image-level loop closure into a dense set of pixel-level loop closures. Rather than a single edge between two images, this injects fine-grained geometric shortcuts directly into the graph, enriching the pixel-level topology and altering the planning costmap.

C. Path Planning and Control Prediction

Exposing these dense geometric shortcuts to the controller effectively resolves the information bottleneck between the planner and the controller. Because every matched pixel acts as a localized sub-goal, the system gains a larger bandwidth to tolerate matching errors or noisy sub-goals. Unlike image-level methods restricted to discrete image-pair servoing, dense pixel matching exposes granular, pixel-level path costs directly to the controller.

This structural shift directly improves navigation behavior by providing the planner with precise shortcuts. As illustrated in Fig. 3 (Middle), navigating from a start (yellow) to a goal (magenta) node using a strictly sequential map forces the agent along an indirect path (red trajectory). However, introducing pixel-level loop closure edges between the detected pair (① and ②) creates direct pathways across the graph. This new connectivity propagates these shorter path distances throughout the graph, fundamentally altering the dense WayPixel costmaps. Consequently, the planner can bypass redundant sub-trajectories of the original teach-and-repeat route and compute a significantly shorter, direct path (green trajectory).

During the execution phase, the current query image is matched against this updated topological graph to produce a pixel-level query costmap. Because the dense costmap explicitly encodes these shortcuts, the controller can robustly predict waypoints along the shorter route, outperforming discrete image-level sub-goals. To ensure safe deployment around obstacles, these waypoint predictions are refined by a reactive Collision Avoidance via Repulsive Estimation (CARE) [31] module, paired with a history-based recovery mechanism that detects when the robot is stuck and reorients it towards open space.

IV. EXPERIMENTS

A. Experimental Setup

Dataset. We evaluate our approach on long ($\sim$100m on an average) indoor trajectories collected via offline teleoperation in 6 distinct HM3D [32] scenes. For each scene, we define three goal nodes and 3–5 start states per goal for evaluation, resulting in a total of 73 different start-to-goal navigation episodes. The trajectories are collected in an expansive manner, covering all navigable regions with diversity in camera orientations during revisits. As a result, each mapping sequence can be viewed as a composition of multiple interconnected teach-and-repeat sub-trajectories, as shown in Fig. 1.

From these trajectories, we construct topological graphs and corresponding WayPixel Costmaps as in Fig. 2 for planning and navigation evaluations.

Navigation. We evaluate start-to-goal navigation performance of PixelLoop against three configurations: image-topological loop closures (GNM), object-topological loop closures (ObjectReact [1]), and MASt3R-Nav (i.e., without loop closures). At each timestep, the agent localizes against a local sub-map and selects the best matching reference frame for control. A trial is considered successful if the agent reaches within 0.5 m of the goal node within 750 navigation steps. Collision avoidance using CARE is kept identical across all methods to ensure a fair comparison. Additionally, if the agent's displacement falls below 10 cm over a sliding window of 15 consecutive steps, an estimated-depth-guided in-place recovery rotation of 45° is triggered towards the direction with fewer obstacles in the top-down depth map.

Evaluation Metrics. We evaluate all methods using four standard embodied navigation metrics widely adopted in prior work [33], [34]. These include Success Rate (SR), Success weighted by Path Length (SPL, termed as SPL-A in our evaluations), SPL computed over successful episodes only (SPL-S), and Soft-SPL (SSPL), which assigns full credit to successful episodes and partial credit to failed episodes based on the agent's final proximity to the goal.

Ground Truth (GT)-Covisibility Loop Closures. These are computed offline using ground-truth depth and camera poses. For each frame pair (i, j) outside each other's mapping window (i.e., not already connected in the mapping graph), pixels in frame i are backprojected to 3D and reprojected into frame j using the known relative transform. A pair is declared a loop if the number of mutually consistent 3D points exceeds a predefined minimum (1000 points). Consistency is defined by the projected 3D location in j agreeing with the ground-truth depth at that pixel within a small threshold (1 cm). We compare these against configurations without loop closures (None) and with our detected loop closures (SeqVLAD + UFM) in Tables I and II.

B. Evaluating Planning Costmaps

We evaluate the cost-quality of our planning costmaps by comparing them against those produced without pixel-level loop closures. Specifically, we compute the Mean Absolute Error (MAE) between binary masks formed by selecting pixels corresponding to the minimum $k\%$ ($k \in \{5, 15, 30, 50\}$) of the predicted costmap and the ground-truth costmap. We use two forms of ground truth, as explained below:

a) Geodesic- / Navigation Mesh-Based Costmaps: Using Habitat-Sim's [35] navigation mesh, we assign each pixel a cost equal to the geodesic distance between its closest navigable 3D point and the goal node. Concretely, each pixel is backprojected to its 3D world position using ground-truth depth and snapped to the nearest navigable point on the scene's navigation mesh. The geodesic distance from

TABLE I: Predicted cost MAE against two ground-truth references. Scores are computed over pixels within the minimum $k\%$ predicted cost and averaged across all mapping images. We observe lower MAE scores using loop closures detected using SeqVLAD + UFM than without. **Bold: best overall. Underline: best non-GT** (lower is better).

Loop Source	Geodesic / Navigation Mesh-Based Costmaps								Ground Truth Pixel Correspondences							
	Mean MAE ↓				Median MAE ↓				Mean MAE ↓				Median MAE ↓			
	5%	15%	30%	50%	5%	15%	30%	50%	5%	15%	30%	50%	5%	15%	30%	50%
GT-Covisibility	0.088	**0.226**	**0.361**	**0.408**	**0.099**	**0.259**	**0.369**	0.385	**0.073**	**0.174**	**0.262**	**0.299**	**0.081**	**0.174**	**0.246**	**0.269**
None	0.090	0.239	0.387	0.441	0.100	0.279	0.415	0.427	0.078	0.192	0.296	0.337	0.088	0.200	0.285	0.309
SeqVLAD + UFM	**0.087**	**0.226**	**0.361**	0.409	**0.099**	0.260	0.370	**0.382**	**0.073**	0.177	0.270	0.308	0.082	0.178	0.256	0.279

that snapped 3D point to the goal is then queried using Habitat-Sim's shortest-path solver and stored as the pixel's cost. These costs approximate true geodesic distances to the goal at each spatial location. MAE computed against this reference quantifies how closely the predicted costmaps approximate true geodesic distances.

b) Ground Truth Pixel Correspondences: We generate ground-truth pixel correspondences by registering all pixels in global 3D space and subsampling matches (common 3D points in an image-pair) within an image using farthest-point sampling, while maintaining the same density as MASt3R. These matches and ground-truth depth are then provided to our topological map construction pipeline, producing ground truth WayPixel costmaps, which are now unaffected by matching or depth estimation errors. MAE against this reference highlights path planning errors under perfect image matching but without the assumption of a globally-registered 3D map.

Table I reports the mean and median MAE scores aggregated across all mapping images and scenes. We compare costmaps generated without loop closures, with GT-covisibility loop closures, and with loop closures detected using our pipeline (SeqVLAD + UFM). We observe consistent reductions in MAE with the inclusion of loop closures across both the ground-truth references, indicating that pixel-level loop closures produce costmaps that more closely approximate ideal planning costs. Furthermore, the performance gap between loop-closure and no-loop configurations increases with larger values of k, suggesting that loop closures improve the global structure of the cost distribution. Notably, costmaps generated with loop pairs detected using SeqVLAD + UFM closely match those obtained using GT-covisibility loop closures.

C. Navigation Comparisons

We benchmark PixelLoop's loop closures against image-level and object-level loop closure strategies under identical experimental settings. Specifically, we evaluate start-to-goal navigation performance by benchmarking PixelLoop against GNM [2] (image-topological) and ObjectReact [1] (object-topological), both with and without loop closures. This evaluation uses MASt3R matches and estimated depth for PixelLoop, while employing the reported state-of-the-art configurations for the other benchmarks.

TABLE II: Navigation performance across 73 start-to-goal tasks. SPL-S denotes SPL over successful episodes only, SPL-A denotes SPL over all episodes. We clearly observe higher metrics for PixelLoop over other benchmarks with loops generated using SeqVLAD + UFM. Moreover, we observe a higher relative gain in performance on using pixel-level loop closures (as opposed to without), over image-level or object-level loop closures. **Bold: best overall. Underline: best non-GT per category** (higher is better).

Loop Source	SR ↑	SPL-S ↑	SPL-A ↑	SSPL ↑
PixelLoop (Ours)				
GT-Covisibility	54.79	**91.70**	50.25	58.39
None	35.62	80.45	28.65	40.70
SeqVLAD + UFM	**68.49**	91.15	**62.43**	**73.97**
GNM [2] (HM3D)				
GT-Covisibility	24.66	70.50	17.38	18.40
None	20.55	64.80	13.32	15.95
SeqVLAD + UFM	27.40	76.97	21.09	22.44
GNM-Adapted (HM3D)				
GT-Covisibility	32.88	83.56	27.47	28.01
None	17.81	75.28	13.41	16.95
SeqVLAD + UFM	32.88	83.97	27.61	28.56
ObjectReact [1]				
GT-Covisibility	63.01	65.21	41.09	65.91
None	58.90	63.21	37.23	38.77
SeqVLAD + UFM	67.12	69.37	46.82	68.35

We evaluate two GNM variants each using an HM3D-pretrained controller checkpoint provided by [1]. a) *GNM*, the publicly-released version, selects sub-goals directly at runtime based on its predicted temporal distances, without any graph-level path planning. b) *GNM-Adapted*, which is our modification to GNM, augments this controller with a sequential image sub-graph akin to teach-and-repeat, achieved through Dijkstra-based shortest path computed over image-level edges weighted by GNM-predicted temporal distances; during execution, candidate sub-goals are restricted to a sliding window along this teach-and-repeat sub-graph, while localization continues to rely on GNM-predicted distances. For the ObjectReact baseline, we use their state-of-the-art reported configuration.

Our method outperforms image-level loop closures (GNM and GNM-Adapted) and object-level loop closures (Objec-

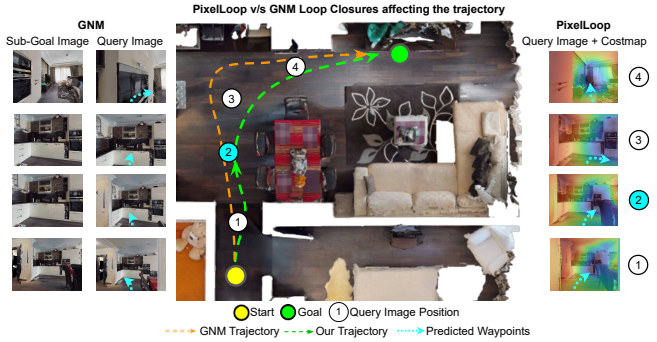

Fig. 4: **GNM v/s PixelLoop.** Both methods detect the same loop at point ②. GNM treats the matched image as a holistic sub-goal, resulting in imprecise steering and trajectory drift. PixelLoop instead uses pixel-level correspondences at the loop closure to localize regions in the matched image that lead towards the goal, producing smoother and more direct trajectories.

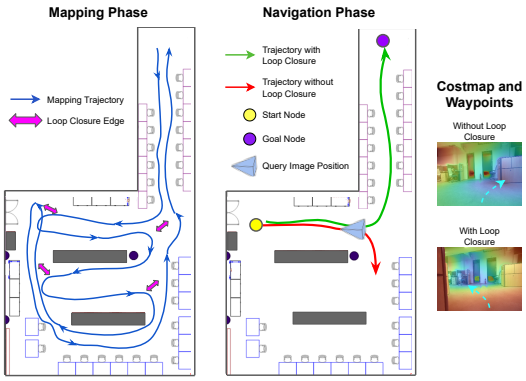

Fig. 5: **Real World Demonstration.** We observe spatially accurate costmaps, controller waypoints and hence a shorter navigation trajectory in the presence of pixel-level loop closures as opposed to without.

tReact) across all evaluation metrics presented in Table II. Incorporating pixel-level loop closures yields substantial improvements over configurations without loop closures, including a ∼34 point absolute increase in SPL-A (28.65 → 62.43). This gain is significantly larger than the improvement obtained by adding image-level loop closures to GNM-Adapted itself (14.20 points; 13.41 → 27.61). Similar comparisons can be seen against ObjectReact.

While ObjectReact achieves a competitive success rate (67.12 vs. 68.49), its SPL-A remains substantially lower (46.82 vs. 62.43), indicating that object-level loop closures enable recovery from incorrect trajectories but lack precise metric grounding in relative 3D geometry, often resulting in longer paths. This indicates that the performance gains stem not merely from introducing loop closures, but from the relative-geometry grounding enabled by pixel-level correspondences.

Fig. 4 provides a qualitative comparison between image-level loop closures (GNM) and our pixel-level loop closures during start-to-goal navigation. Although both methods detect the same loop at point ②, their downstream control behavior differs significantly. GNM treats the matched image as a holistic sub-goal and predicts motion towards the entire view, without reasoning about which regions within the image actually lead towards the goal. Consequently, control signals are weakly specified, leading to imprecise steering and observable trajectory drift.

In contrast, PixelLoop constructs dense pixel correspondences between the current view and the loop image. While not grounded in a global reference frame or metric map, these correspondences encode relative 3D geometric relationships between matched scene points. This induces a locally consistent relative-geometry representation, allowing the controller to infer waypoint targets that are spatially anchored towards the goal within the current view rather than directed towards the sub-goal image as a whole.

Importantly, these relative geometric relations propagate not only across loop closures but also along the toplogically connected map images, forming a cascading chain toward the goal node. This provides a structured control signal that consistently points toward the goal across viewpoints, a property unavailable to image-level loop closures, which lack pixel-level geometric grounding. As a result, PixelLoop produces smoother and more direct trajectories, demonstrating that loop detection alone is insufficient unless translated into relative-geometry grounded control signals.

Performance using our loop detection pipeline surpasses even the GT-covisibility loop closure configuration by a ∼24% relative improvement in SPL-A. This improvement stems from pruning loop pairs with limited covisibility, which often share only small overlapping regions while the remaining image content contains visually similar yet geometrically inconsistent structures (e.g., doors on opposite sides of a wall), leading to incorrect matches.

D. Real-World Demonstration

TABLE III: Real-world navigation performance across 11 start-to-goal runs in 3 large indoor environments using a P3DX mobile robot with a RealSense camera.

Loop Source	SR ↑	SPL ↑	SSPL ↑
None	36.4	33.12	58.19
PixelLoop	**81.8**	**78.82**	**87.15**

We further validate our approach in real-world indoor environments using a Pioneer 3-DX (P3DX) mobile robot equipped with an Intel RealSense camera. We evaluate PixelLoop's navigation performance with and without pixel-level loop closures, measuring Success Rate (SR), SPL, and SSPL.

The results in Table III and Fig. 5 demonstrate that the benefits of pixel-level loop closures extend beyond simulation and generalize to real-world deployments. PixelLoop runs at ∼4.8 Hz during online navigation on a single NVIDIA RTX A4000 GPU, requiring ∼4.6 GB of VRAM.

V. CONCLUSION

In this work, we introduced **PixelLoop**, demonstrating that the key to robust topological navigation lies not merely in

detecting loop closures, but in executing them with dense geometric fidelity. By operating directly in pixel space, our framework overcomes the information bottlenecks that severely limit discrete image-relative planners. We showed that stitching disjoint trajectories via zero-cost pixel correspondences fundamentally alters the underlying planning geometry, allowing us to generate continuous, fine-grained costmaps.

Our empirical evaluations confirm that this structural shift empowers a local neural controller to reliably discover and exploit novel physical shortcuts. By achieving state-of-the-art performance in arbitrary start-to-goal navigation without the overhead of global metric SLAM, we validate the superiority of dense topological scaling. Crucially, successful real-world mobile robot deployments highlight the robustness and practical viability of our approach. Moving forward, we plan to extend this pixel-relative foundation by integrating semantic object-level abstractions, enabling the system to better adapt to dynamic environmental changes during lifelong navigation deployments.

REFERENCES

[1] S. Garg, D. Craggs, V. Bhat, L. Mares, S. Podgorski, M. Krishna, F. Dayoub, and I. Reid, "Objectreact: Learning object-relative control for visual navigation," in *Conference on Robot Learning*. PMLR, 2025.

[2] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "Gnm: A general navigation model to drive any robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7226–7233.

[3] S. Garg, K. Rana, M. Hosseinzadeh, L. Mares, N. Sünderhauf, F. Dayoub, and I. Reid, "Robohop: Segment-based topological map representation for open-world visual navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4090–4097.

[4] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 63–70.

[5] V. Garg, R. Jayanti, K. Pandya, S. Chittawar, S. Tourani, M. H. Khan, S. Garg, and M. Krishna, "Mast3r-nav: Waypixel navigation in relative 3d maps," in *International Conference on Robotics and Automation (ICRA)*, 2026.

[6] N. Savinov, A. Dosovitskiy, and V. Koltun, "Semi-parametric topological memory for navigation," *arXiv preprint arXiv:1803.00653*, 2018.

[7] O. Kwon, N. Kim, Y. Choi, H. Yoo, J. Park, and S. Oh, "Visual graph memory with unsupervised representation for visual navigation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 890–15 899.

[8] X. Meng, N. Ratliff, Y. Xiang, and D. Fox, "Scaling local control to large-scale topological navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[9] M. Hahn, D. S. Chaplot, S. Tulsiani, M. Mukadam, J. M. Rehg, and A. Gupta, "No rl, no simulation: Learning to navigate without navigating," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 661–26 673, 2021.

[10] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[11] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*, September 2014.

[12] D. Shah, B. Eysenbach, G. Kahn, N. Rhinehart, and S. Levine, "Ving: Learning open-world navigation with visual goals," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 215–13 222.

[13] D. Shah, B. Eysenbach, N. Rhinehart, and S. Levine, "Rapid exploration for open-world navigation with latent goal models," in *Conference on Robot Learning*. PMLR, 2022, pp. 674–684.

[14] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of field robotics*, vol. 27, no. 5, 2010.

[15] S. Šegvić, A. Remazeilles, A. Diosi, and F. Chaumette, "A mapping and localization framework for scalable appearance-based navigation," *Computer Vision and Image Understanding*, vol. 113, no. 2, 2009.

[16] D. Shah and S. Levine, "ViKiNG: Vision-Based Kilometer-Scale Navigation with Geographic Hints," in *Proceedings of Robotics: Science and Systems*, 2022.

[17] D. Shah, B. Osinski, B. Ichter, and S. Levine, "LM-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *6th Annual Conference on Robot Learning*, 2022.

[18] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.

[19] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE transactions on robotics and automation*, vol. 18, no. 4, pp. 534–549, 2002.

[20] Q. Gu, A. Kuwajerwala, S. Morin, K. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, C. Gan, C. de Melo, J. Tenenbaum, A. Torralba, F. Shkurti, and L. Paull, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *arXiv*, 2023.

[21] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3d scene graph: A structure for unified semantics, 3d space, and camera," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5664–5673.

[22] Y. Wu, Y. Wu, A. Tamar, S. Russell, G. Gkioxari, and Y. Tian, "Bayesian relational memory for semantic visual navigation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2769–2779.

[23] H. Du, X. Yu, and L. Zheng, "Learning object relation graph and tentative policy for visual navigation," in *European Conference on Computer Vision*. Springer, 2020, pp. 19–34.

[24] W. Cai, S. Huang, G. Cheng, Y. Long, P. Gao, C. Sun, and H. Dong, "Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5228–5234.

[25] S. Podgorski, S. Garg, M. Hosseinzadeh, L. Mares, F. Dayoub, and I. Reid, "Tango: Traversablility-aware navigation with local metric control for topological goals," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

[26] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," in *European conference on computer vision*. Springer, 2024, pp. 71–91.

[27] B. P. Duisterhof, L. Zust, P. Weinzaepfel, V. Leroy, Y. Cabon, and J. Revaud, "Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion," in *2025 International Conference on 3D Vision (3DV)*. IEEE, 2025, pp. 1–10.

[28] D. Maggio and L. Carlone, "Vggt-slam 2.0: Real-time dense feed-forward scene reconstruction," *arXiv preprint arXiv:2601.19887*, 2026.

[29] Y. Zhang, N. Keetha, C. Lyu, B. Jhamb, Y. Chen, Y. Qiu, J. Karhade, S. Jha, Y. Hu, D. Ramanan, S. Scherer, and W. Wang, "Ufm: A simple path towards unified dense correspondence with flow," 2026. [Online]. Available: https://arxiv.org/abs/2506.09278

[30] R. Mereu, G. Trivigno, G. Berton, C. Masone, and B. Caputo, "Learning sequential descriptors for sequence-based visual place recognition," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, 2022.

[31] J. Kim, J. Sim, W. Kim, K. Sycara, and C. Nam, "Care: Enhancing safety of visual navigation through collision avoidance via repulsive estimation," *arXiv preprint arXiv:2506.03834*, 2025.

[32] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," 2021. [Online]. Available: https://arxiv.org/abs/2109.08238

[33] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On evaluation of embodied navigation agents," 2018.

[34] S. Datta, O. Maksymets, J. Hoffman, S. Lee, D. Batra, and D. Parikh, "Integrating egocentric localization for more realistic point-goal navigation agents," 2020.

[35] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A platform for embodied ai research," 2019.