Change-Robust Online Topological Memory for Long-Term Relocalization and Semantic Navigation

Jiaming Wang[1*†] **Jizhuo Chen**[1*] **Diwen Liu**[2*] **Atharva Ghotavadekar**[1]

Jiaxuan Da[1] **Linh Kästner**[3] **Harold Soh**[1†]

[1]National University of Singapore [2]The Chinese University of Hong Kong
[3]Singapore Management University

Abstract

Long-term semantic navigation requires a robot to reuse past observations after appearance and scene change, but semantic memories are only useful if the robot can relocalize into the memory without corrupting it with false visual matches. We propose CROSS, a change-robust topological memory that introduces a *pre-commitment* localization layer between visual place recognition and map update. Instead of treating a retrieved keyframe as an immediate place association or loop-closure factor, CROSS lifts each RGB-D retrieval into a candidate global $\mathrm{SE}(3)$ pose mode using relative pose estimation. A bounded Gaussian-mixture filter then propagates competing continuous trajectory branches with odometry, rejects branches that are physically inconsistent, and promotes only persistent branches to loop closures. This moves ambiguity handling from discrete place IDs or post-hoc graph-factor rejection to continuous pose-space validation before map commitment. Across public long-term relocalization benchmarks and real quadruped object-navigation experiments, CROSS improves reuse of a single sparse RGB-D memory under illumination, seasonal, dynamic-scene, and object-level change. Project page and code: `https://jiaming.im/CROSS/`

1 Introduction

Robots operating over long time horizons need spatial memories that remain useful after illumination, seasonal, object-level, and dynamic-scene changes. Such memories should support semantic navigation tasks such as object search or language-goal navigation, but these tasks first require reliable spatial reuse: the robot must know where relevant observations were made, how to reach them, and where it is relative to them.

A common approach attaches object labels, language embeddings, or scene-graph structure to a SLAM-built metric map [7, 6, 13, 17, 44]. This is effective when the metric substrate remains valid, but long-term deployment exposes a brittle dependency: if relocalization fails under appearance change or stale map priors, the semantic layer becomes difficult to use.

We instead study a sparse topological memory for long-term semantic navigation. Each node stores a raw RGB-D observation and a pose distribution, while edges encode local navigability. Keeping raw observations is deliberately task-agnostic: the robot need not know during mapping which object categories or language goals will be queried later. Stored observations can be processed at query time by detectors [52, 29], VLMs [3], or other visual foundation models [34].

*Equal contribution.
†Corresponding authors: {jiaming, harold}@comp.nus.edu.sg

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

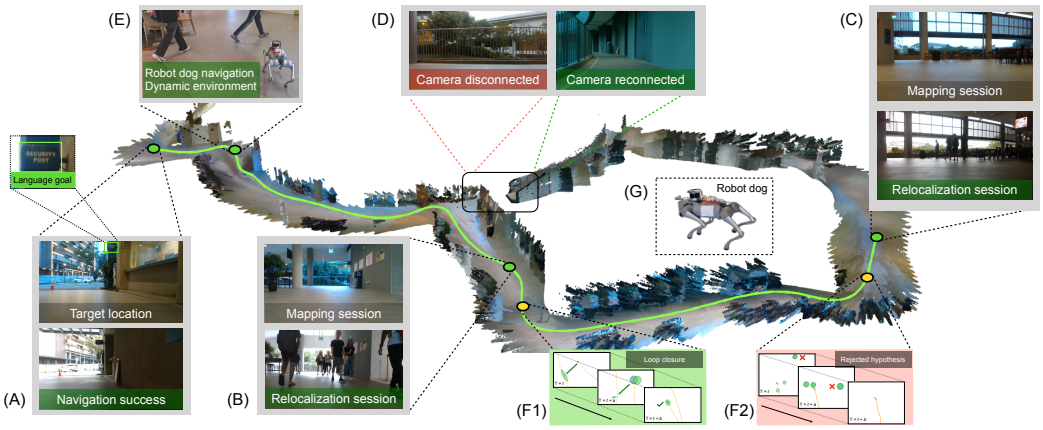


Figure 1: CROSS supports long-term semantic navigation with a sparse RGB-D topological memory under lighting variation, object rearrangement, dynamic pedestrians, and sensor degradation. Visual retrievals are lifted into continuous $\mathrm{SE}(3)$ hypotheses, temporally validated by odometry, and only persistent branches are promoted to loop closures or relocalization events.

The central localization challenge is deciding *when to trust* a visually plausible association. Modern visual place recognition (VPR) improves recall under appearance change [1, 4], yet high-scoring retrievals can still be false positives in repetitive or changed environments. Topological and topometric methods reduce such errors by filtering over place IDs, route positions, or image sequences [9, 18, 25, 43, 50, 49, 40]. These methods delay commitment, but usually over discrete states, discarding much of the relative-pose geometry needed to test physical compatibility with odometry. Metric SLAM back-ends preserve geometry, but candidate loop closures typically enter the graph as factors before robust mechanisms accept or reject them [39, 31, 20, 14]. This leaves a missing interface: visual associations should remain uncommitted, yet still be tested as continuous pose evidence.

CROSS (**C**hange-**R**obust **O**nline **S**patial memory **S**ystem) addresses this interface with a pre-commitment localization layer. VPR proposes candidate keyframes; RGB-D relative pose estimation lifts each candidate into a global $\mathrm{SE}(3)$ pose mode. A bounded Gaussian-mixture belief then propagates competing continuous trajectory branches with odometry, prunes motion-inconsistent branches, and promotes only persistent branches to loop closures, tracking recovery, or kidnapped-robot relocalization. The contribution is not any individual component in isolation, but the commitment boundary: ambiguous visual associations are validated in continuous pose space before they modify the topological memory.

This paper makes three contributions:

- An online topological memory that stores raw RGB-D keyframes as task-agnostic semantic anchors while maintaining pose uncertainty for navigation. Code is available at `https://github.com/jiaming-ai/CROSS`.
- A retrieval-to-pose hypothesis layer that converts visual matches into continuous $\mathrm{SE}(3)$ trajectory branches and validates them against odometry before map commitment.
- An evaluation on public long-term relocalization benchmarks and real-robot object-goal navigation, showing improved memory reuse under illumination, seasonal, dynamic-scene, and object-level change.

2 Related Work

Semantic maps and long-term robot memory. Many semantic navigation systems attach object, language, or scene-graph information to a metric map built by SLAM, including 2D semantic occupancy maps [7, 6, 24, 44], 3D scene graphs [16, 48, 13], and dense open-vocabulary feature maps [17, 53, 22, 51]. These representations provide rich semantic interfaces, but their spatial reuse depends on reliable relocalization in the underlying metric map. CROSS is complementary: it does not aim to be a richer semantic representation, but a change-robust spatial memory substrate. By storing raw RGB-D keyframes, it defers semantic interpretation to query time while focusing on whether past observations can be localized and reused after appearance or scene change.

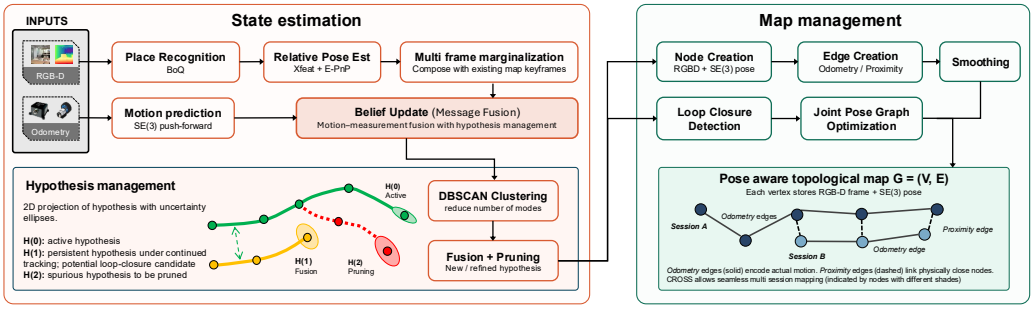


Figure 2: **System Overview.** Given an RGB-D frame and odometry, the online tracking module (orange) performs temporal hypothesis filtering in continuous $\mathrm{SE}(3)$. Motion updates are propagated via $\mathrm{SE}(3)$ push-forward, while measurement updates are constructed through VPR-based keyframe retrieval. Competing hypotheses are efficiently managed using Gaussian-mixture clustering, pruning, and fusion (see Section 4.1). The map-management module (blue) uses the estimated state from the tracking module to create nodes and edges, detect loop closures, and perform pose-graph optimization and smoothing, yielding a consistent pose-aware topological map (see Section 4.2).

Topological, topometric, and route-based localization. Appearance-based systems such as FAB-MAP and RTAB-Map perform probabilistic place recognition and online map growth in the space of visual observations [9, 18]. CAT-SLAM, SeqSLAM-style sequence matching, visual teach-and-repeat, and experience-based navigation add temporal or route consistency to improve long-term operation under changing appearance [25, 27, 43, 12, 8]. More recent topometric systems combine VPR with Bayesian filtering, odometry-conditioned transitions, and off-map states [50, 49, 40]. These methods are robust and scalable because their beliefs are tied to place IDs, route positions, or particles over a fixed topometric state space. CROSS uses the same insight—visual matches should be temporally validated—but changes the state being validated: visually retrieved keyframes are lifted into continuous $\mathrm{SE}(3)$ pose modes, so the filter can test full geometric compatibility with odometry before committing to a place association. Appendix E discusses these differences in more detail.

Multi-hypothesis localization and robust SLAM. Multi-modal pose beliefs have long been studied in Markov and Monte Carlo localization [11, 41], and modern multi-hypothesis SLAM back-ends represent ambiguous odometry, data association, or loop closures as multiple factor-graph solutions [14, 15]. Robust pose-graph methods such as switchable constraints, max-mixtures, and related loop-closure consistency checks downweight or reject erroneous loop closures after candidate factors have been proposed [39, 31, 20]. CROSS is not a replacement for these back-ends. Its contribution is a front-end, pre-commitment layer for visual topological memory: candidate visual associations remain as bounded continuous pose branches, are propagated and pruned by sequential motion consistency, and are promoted to loop-closure factors only after sustained support.

3 Problem Formulation

We aim to construct an environment representation that supports robust navigation in real-world settings subject to natural and often substantial change. Such a representation should satisfy three desiderata: (i) *VLM compatibility*, enabling semantic inference directly over stored observations; (ii) *change robustness*, allowing the robot to localize and reuse a single map despite significant appearance variation without remapping; and (iii) *local navigability*, providing sufficient spatial structure to support short-horizon pose estimation and goal-directed motion.

Guided by these requirements, we represent the environment as a sparse, pose-aware topological graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each node $v_i \in \mathcal{V}$ stores an RGBD keyframe z_i and an associated camera pose $c_i \in \mathrm{SE}(3)$. The robot pose at time t is represented by a random variable $x_t \in \mathrm{SE}(3)$, and the robot state is summarized as (x_t, n_t), where n_t indexes the current (or nearest) keyframe. Importantly, both x_t and $\{c_i\}$ are treated as random variables and are modeled using finite Gaussian mixtures on $\mathrm{SE}(3)$ (Section 4.1), which enables a compact representation of multi-modal pose uncertainty.

We use u_t to denote the odometry increment from x_{t-1} to x_t. Given observations $z_{1:t}$ and odometry inputs $u_{1:t}$, our goal is to maintain the joint posterior over the robot trajectory and the topological map, $p(x_{1:t}, \mathcal{G} \mid z_{1:t}, u_{1:t})$. The current node index n_t is inferred from a pose estimate $\hat{x}_t$ as $n_t =$

$\arg\min_{v_i \in \mathcal{V}} d(\hat{x}_t, \hat{c}_i)$, where $d(\cdot)$ denotes a pose distance on $\mathrm{SE}(3)$ and $\hat{c}_i$ denotes a representative estimate of the keyframe pose (e.g., the highest-weight mixture component).

This formulation is topological in what it stores, but continuous in what it tests. The graph contains sparse RGB-D keyframes rather than persistent landmarks, voxels, or fused semantic embeddings used in metric SLAM and metric-semantic maps [5, 19, 17, 13]. However, each visual association is evaluated as a constraint on the robot's continuous pose. Thus, the latent ambiguity is not only "which node is this?" but "which $\mathrm{SE}(3)$ trajectory branch makes this visual association physically consistent with the odometry?" This distinguishes CROSS from topological, topometric, and route-based localization methods whose beliefs are tied to place IDs or route states [9, 25, 12, 8, 49, 40], and from multi-hypothesis or robust pose-graph methods that reason about ambiguous associations after they have been introduced as candidate factors [14, 39, 31, 20].

4 Method: CROSS

In this section, we describe CROSS (overview in Fig. 2). At a high level, the system is organized into state estimation and map management. Direct inference over the joint trajectory–map posterior is intractable because trajectory and map estimates are coupled, so we use the conditional factorization

$$p(x_{1:t}, \mathcal{G} \mid z_{1:t}, u_{1:t}) = \underbrace{p(\mathcal{G} \mid x_{1:t}, z_{1:t}, u_{1:t})}_{\text{map management}} \cdot \underbrace{p(x_{1:t} \mid z_{1:t}, u_{1:t})}_{\text{state estimation}}. \tag{1}$$

Due to space constraints, we focus on the key ideas in each module and relegate details to Appendix D.

4.1 State Estimation: Approximate Inference and Hypothesis Management

To represent multi-modal pose uncertainty while keeping computation bounded, we model both the robot pose x_t and keyframe poses $\{c_i\}$ as finite Gaussian mixtures on $\mathrm{SE}(3)$. Since $\mathrm{SE}(3)$ is a nonlinear manifold, we define each Gaussian in the local tangent space $\mathfrak{se}(3)$ of its mean using the logarithm map, and reconstruct poses via the exponential map,

$$p(s_j) \approx \sum_{k=1}^{K} w_j^{(k)} \mathcal{N}_{\mathfrak{se}(3)}\Big(\log\big((\mu_j^{(k)})^{-1} s_j\big); \mathbf{0}, \Sigma_j^{(k)}\Big), \quad s_j \in \{x_t, c_i\}. \tag{2}$$

where each component specifies a local distribution around its mean on $\mathrm{SE}(3)$: $x_t^{(k)} = \mu_t^{(k)} \exp(\xi)$, $c_i^{(k)} = \mu_i^{(k)} \exp(\xi)$, $\xi \sim \mathcal{N}(0, \Sigma^{(k)})$. The residual form $\log\big((\mu)^{-1}(\cdot)\big)$ ensures that uncertainty is expressed in the tangent space where linearization is valid.

We perform inference over the current pose by forward message passing on the factor graph (Fig. 10 in Appendix) induced by the motion and measurement factors. Let $m_t^{\text{mot}}(x_t)$ and $m_t^{\text{meas}}(x_t)$ denote the motion and measurement messages arriving at node x_t. The belief is given by:

$$p(x_t \mid z_{1:t}, u_{1:t}, \mathcal{G}) \propto \underbrace{m_t^{\text{meas}}(x_t)}_{\text{measurement message}} \underbrace{m_t^{\text{mot}}(x_t)}_{\text{motion message}}. \tag{3}$$

Eq. (3) is evaluated in closed-form using the approximations below and is similar to the classic Gaussian-sum filter (GSF) [35, 2], but differs in how the measurement message is constructed.

Forward motion message. The motion factor is $\phi_t^{\text{mot}}(x_{t-1}, x_t, u_t) \triangleq p(x_t \mid x_{t-1}, u_t)$, and its forward message to x_t is the Bayes prediction $m_t^{\text{mot}}(x_t) = \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1})\, dx_{t-1}$. Let $\{w_{t-1}^{(k)}, \mu_{t-1}^{(k)}, \Sigma_{t-1}^{(k)}\}_{k=1}^{K}$ denote the weights, means, and covariances of the mixture at time $t-1$. We approximate the pushed-forward mixture at time t as $m_t^{\text{mot}}(x_t) \approx \sum_{k=1}^{K} w_{t-1}^{(k)} \mathcal{N}_{\mathfrak{se}(3)}\Big(\log\Big((\mu_{t|t-1}^{(k)})^{-1} x_t\Big);\ 0, \Sigma_{t|t-1}^{(k)}\Big)$, where each mixand is mapped to a new Gaussian with mean $\mu_{t|t-1}^{(k)} = \mu_{t-1}^{(k)}\, u_t$ and covariance $\Sigma_{t|t-1}^{(k)} \approx \mathrm{Ad}_{u_t^{-1}}\, \Sigma_{t-1}^{(k)}\, \mathrm{Ad}_{u_t^{-1}}^{\top} + Q_t$. Here, u_t is the odometry input, and Q_t is the process-noise covariance in $\mathfrak{se}(3)$. The mixture weights are preserved ($w_{t|t-1}^{(k)} = w_{t-1}^{(k)}$) because the motion kernel is probability-preserving (see Appendix D.2).

Measurement Message. A standard fixed-support Gaussian-sum filter refines only the pose modes already present in the prior (see Appendix D.1). Long-term relocalization, however, requires a measurement to introduce modes at arbitrary previously mapped locations. We therefore construct

a global measurement message whose components are born from retrieved keyframes rather than from the current dominant tracking branch. To overcome this limitation, we introduce a *global measurement* message. We begin with a local measurement model between the current RGB-D observation z_t and node i. The relative pose can be estimated using standard PnP methods [21, 33, 42]: $T_{i,t} = f_{\mathrm{PnP}}(z_t, z_i)$, where z_t and z_i denote the RGBD observations from the current step t and node i, respectively. In our implementation, $f_{\mathrm{PnP}}(\cdot)$ first extracts and matches features using XFeat [32] and LightGlue [23] to obtain 2D-3D correspondences. We adopt this combination primarily for its efficiency, as both methods provide fast and reliable feature processing. The resulting correspondences are then passed to an E-PnP [21] solver to estimate the relative pose.

Given this relative measurement, we obtain an estimate of the robot's global pose by composing the keyframe pose c_i with the relative transform, i.e., $x_t \approx c_i\, T_{i,t}$. The distribution of the composed pose is obtained by pushing forward each mixture component by $T_{i,t}$, analogous to the operation used in the motion factor. The resulting global pose distribution conditioned on keyframe i is therefore a Gaussian mixture over the K components of c_i:

$$p(x_t \mid v_i) \approx \sum_{k=1}^{K} w_i^{(k)}\, \mathcal{N}_{\mathfrak{se}(3)}\Big(\log\big((\mu_i^{(k)} T_{i,t})^{-1} x_t\big);\; 0, \Sigma_{i,t}^{(k)}\Big)\,, \tag{4}$$

where $\Sigma_{i,t}^{(k)} \approx \mathrm{Ad}_{T_{i,t}^{-1}}\, \Sigma_i^{(k)}\, \mathrm{Ad}_{T_{i,t}^{-1}}^{\top} + \Gamma_{i,t}^{\mathrm{pnp}}$. Here, $\Gamma_{i,t}^{\mathrm{pnp}}$ is the relative-pose covariance from PnP in $\mathfrak{se}(3)$; we approximate it using the PnP inlier support via a fixed isotropic base covariance scaled by the inlier count, $\Gamma_{i,t}^{\mathrm{pnp}} = \lambda I_6 / \max(n_{i,t}^{\mathrm{in}}, 1)$. Eq. (4) describes the pose likelihood conditioned on a specific keyframe i. However, multiple keyframes may be visually compatible with the current observation, making data association an inherent component of the measurement update. To model this uncertainty, we introduce a discrete latent variable $Y_t \in \{1, \ldots, N_t\}$ indicating which keyframe explains z_t. Marginalizing over this association yields a mixture-form measurement message: $\mathcal{H}_t(x_t) = \sum_{i=1}^{N_t} p(x_t \mid z_t, Y_t{=}i, \mathcal{G})\, p(Y_t{=}i \mid z_t, \mathcal{G})$, subject to $\sum_{i=1}^{N_t} p(Y_t{=}i \mid z_t, \mathcal{G}) = 1$. The association term $p(Y_t{=}i \mid z_t, \mathcal{G})$ is estimated using a compact proposal-weighting approximation. For each retrieved keyframe, we combine two cues: the VPR score, which captures appearance-level compatibility, and the PnP inlier count, which captures whether a geometrically consistent relative pose can be estimated. In our system, place recognition is performed by BoQ [1], while relative pose estimation is obtained via RANSAC E-PnP [21]. We compute an association weight $P(i) = \mathrm{softmax}\Big(\pi(i)\, \frac{\mathrm{inlier}(i)}{M}\Big)$, where $\pi(i)$ is the VPR retrieval score, $\mathrm{inlier}(i)$ denotes the PnP inlier count, and M is the maximum number of detected features used for normalization. Empirically, this appearance–geometry weighting provides a reliable approximation to $p(Y_t = i \mid z_t, \mathcal{G})$ for ranking candidate associations. The subsequent product with the motion message and delayed loop-closure test provide the main geometric and temporal validation of these proposals. Using $P(i)$, the global measurement message takes the mixture form:

$$\mathcal{H}_t(x_t) = \sum_{i=1}^{N_t} P(i) \sum_{k=1}^{K} w_i^{(k)}\, \mathcal{N}_{\mathfrak{se}(3)}\Big(\log\big((\mu_i^{(k)} T_{i,t})^{-1} x_t\big);\; 0, \Sigma_{i,t}^{(k)}\Big)\,. \tag{5}$$

In contrast to conventional GSFs, which apply per-component EKF-style measurement updates, this global measurement message $\mathcal{H}_t$ aggregates evidence over all retrieved keyframes and enables the filter to naturally handle tracking loss and loop closures, since the measurement update is not restricted to refining only the currently dominant mixands.

Hypothesis Management. One challenge that arises from the above message passing is that the direct product produces up to $N_t K^2$ mixture terms. Such quadratic growth quickly becomes impractical in long sequences. Rather than computationally-expensive EM-style reduction [10], we adopt an explicit hypothesis management strategy. Let $w_{\mathrm{mot}}^{(i)}$ and $w_{\mathrm{meas}}^{(j)}$ be the weights of the i-th motion and j-th measurement components, respectively. The product component (i, j) has weight $w_t^{(i,j)} \propto w_{\mathrm{mot}}^{(i)} w_{\mathrm{meas}}^{(j)} C_{ij}$, where C_{ij} is their Gaussian overlap. Most pairs have negligible overlap because they are far apart in $\mathrm{SE}(3)$. To control the growth in mixture size, we maintain a small set of physically meaningful hypotheses. Each surviving mixture component in $p(x_t \mid z_{1:t}, u_{1:t}, \mathcal{G})$ is interpreted as a hypothesis $h^{(l)}$ representing a distinct, dynamically consistent trajectory. We summarize $h_t^{(l)} = \{x_{1:t}^{(l)}, \Sigma_{1:t}^{(l)}, \bar{w}_t^{(l)}, \mathcal{E}_{\mathrm{vis}}^{(l)}\}$, where $x_{1:t}^{(l)}$ are the node poses along the trajectory, $\Sigma_{1:t}^{(l)}$ are their covariances, $\bar{w}_t^{(l)}$ is the normalized hypothesis weight, and $\mathcal{E}_{\mathrm{vis}}^{(l)}$ is the set of visual constraints (further described in the smoothing step in Section 4.2).

Rover (Campus)



OpenLORIS (Corridor)



Figure 3: Appearance change at the same physical locations for the Rover and OpenLORIS benchmarks.

In brief, we manage hypotheses in two stages (complete details in Appendix D.3):

1. **Measurement Clustering**, which clusters all measurement components on $\mathrm{SE}(3)$ into a small number of dominant modes using an $\mathrm{SE}(3)$-aware DBSCAN method;
2. **Fusion, Birth, and Pruning**, whereby we fuse these modes with the motion mixture, updating existing hypotheses and spawning new ones if needed. In addition, we prune hypotheses whose weights or overlaps are too small.

In practice, we do not duplicate node poses; each hypothesis $h^{(l)}$ is represented solely by its component index l at every node, and under the motion model this component evolves continuously in $\mathrm{SE}(3)$, so the index l is preserved over time (the l-th component of m_t^{mot} is the propagation of the l-th component at time $t-1$). New hypotheses are created only when the global measurement message proposes a mode that does not overlap with any existing motion component, at which point a new component index, and thus a new hypothesis, is branched out.

4.2 Topological Map Management

Given the state estimates, we maintain a topological map. This involves creating new nodes, adding edges, and performing pose-graph optimization (PGO) to smooth past trajectories.

Node Creation. At each timestep, we first perform visual place recognition (VPR) to retrieve keyframe candidates whose appearances match the current observation. We create a new node whenever the maximum similarity score falls below a threshold β, using the current RGB-D frame and its pose estimate (e.g., the robot visits a previously unseen location or when a known location exhibits a substantially different appearance). Since the current pose estimate is represented as a Gaussian mixture on $\mathrm{SE}(3)$, the newly added node inherits the same mixture structure, i.e., a set of mixture components $\{(\mu_{\text{new}}^{(k)}, \Sigma_{\text{new}}^{(k)}, w_{\text{new}}^{(k)})\}_{k=1}^{K}$, where each component is parameterized by a pose mean $\mu_{\text{new}}^{(k)} \in \mathrm{SE}(3)$ and diagonal covariance $\Sigma_{\text{new}}^{(k)}$ defined in the tangent space $\mathfrak{se}(3)$.

Edge Creation. Whenever a new keyframe is created, we add two types of edges:

- **Odometry edge:** connecting the new node to the previous node, induced by the physical motion between timesteps.
- **Proximity edge:** connecting the new node to nearby existing nodes within a spatial radius d. This helps ensure that multiple nodes representing the same physical place (e.g., under different appearances) remain connected in the topological graph. In principle, a learned traversability model [45] could be used to validate whether two nodes should be connected, but in practice we find that a simple distance threshold (e.g., $d = 0.5\,\mathrm{m}$) is reliable and lightweight.

Smoothing. While our representation is topological, we periodically refine node poses to support accurate temporal hypothesis filtering. This refinement is computationally inexpensive because the map contains no explicit point landmarks. We estimate node poses by optimizing the posterior, $p(\{c_i\} \mid \mathcal{E}_{\text{odo}}, \mathcal{E}_{\text{vis}}, z_{1:t}, u_{1:t})$, where $\mathcal{E}_{\text{odo}}$ is the set of odometry edges and $\mathcal{E}_{\text{vis}}$ is the set of visual constraints. Each edge encodes a noisy relative pose in $\mathrm{SE}(3)$; under a Gaussian noise model, the negative log-posterior is a sum of squared residuals in $\mathfrak{se}(3)$.

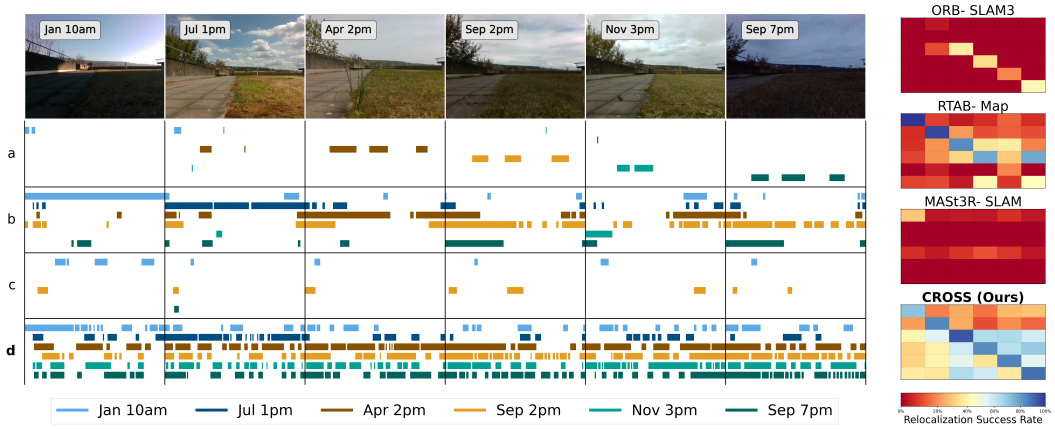


Figure 4: Multi-session relocalization results on the Rover [37] Campus scene. **Left:** Relocalization outcomes across different locations. Each row corresponds to a mapping trajectory (indicated by different colors), while columns show relocalization attempts at the same physical locations captured at different times or months, as illustrated in the top image. Empty space indicates relocalization failed at that specific location. The compared methods are: (a) ORB-SLAM3, (b) RTAB-Map, (c) MASt3R-SLAM, and (d) Ours. Most baseline methods struggle under significant lighting and appearance changes, whereas our approach consistently relocalizes despite substantial visual variation. **Right:** A compact summary, where each grid cell reports the relocalization success rate for a given mapping–testing sequence pair. Full results and additional analysis are provided in Section 5.1.

For each hypothesis $h^{(l)}$ we maintain $\mathcal{E}_{\text{vis}}^{(l)} = \{(i,j)\}$, where (i,j) denotes a visual constraint between node i and node j for hypothesis l. Under our hypothesis management policy, all visual constraints in $\mathcal{E}_{\text{vis}}^{(l)}$ implicitly connect component l at both endpoints. Conditioning on $h^{(l)}$ therefore fixes the component identities, and the continuous part of the problem reduces to a standard pose-graph optimization (PGO) over the node means. Thus, we minimize the negative log posterior $p(\{c_i^{(l)}\} \mid \mathcal{E}_{\text{odo}}, \mathcal{E}_{\text{vis}}^{(l)}, z_{1:t}, u_{1:t})$. Hypotheses whose optimized pose graphs are inconsistent with odometry or other constraints naturally receive lower posterior weight and are subsequently pruned.

Loop Closure. Loop closure detection in our system arises naturally from the temporal hypothesis filtering (THF) formulation: a revisit, whether a standard loop closure or a kidnapped-robot event, appears as an additional hypothesis whose trajectory becomes consistent with an older region of the map. After the THF step, we retain a small set of hypotheses $\{h_t^{(l)}\}_{l=0}^{L-1}$ with normalized weights $\{\bar{w}_t^{(l)}\}_{l=0}^{L-1}$, where $h_t^{(0)}$ is the currently tracked branch. For each alternative hypothesis $l > 0$, we compute a log posterior odds against the null hypothesis $h^{(0)}$: $\ell_t^{(l)} = \log \bar{w}_t^{(l)} / \bar{w}_t^{(0)}$. A positive $\ell_t^{(l)}$ indicates that, at time t, hypothesis $h^{(l)}$ is more likely than the no-loop hypothesis $h^{(0)}$. Rather than accepting a loop closure from a single frame, we accumulate evidence over a sliding window of length W by counting how many times $\ell_\tau^{(l)} > 0$ for $\tau \in \{t - W + 1, \ldots, t\}$. A loop closure for hypothesis $h^{(l)}$ is declared only when this count exceeds a threshold r. This suppresses spurious place-recognition outliers and ensures that only hypotheses with sustained support are promoted.

Once a loop closure between $h^{(0)}$ and $h^{(l)}$ is accepted, we perform a joint PGO analogous to the smoothing procedure, but over both hypotheses simultaneously. For each node index i in the temporal overlap of the two trajectories, we introduce a loop-closure residual $\log((\mu_i^{(0)})^{-1}\mu_i^{(l)})$, with a small covariance, enforcing consistency between the two pose estimates that correspond to the same physical location. We then construct an augmented PGO whose variables are the union of node poses $\{\mu_i^{(0)}\} \cup \{\mu_i^{(l)}\}$, and whose factors include: (i) the odometry and visual constraints from each hypothesis, and (ii) the identity loop-closure constraints $\mathcal{E}_{\text{lc}}^{(0,l)}$ linking corresponding nodes. Solving this joint optimization aligns the two branches and yields a single, globally consistent trajectory.

After optimization, we collapse the merged pair into a single hypothesis by retaining index 0 with updated poses and aggregated weight $\bar{w}_t^{(0)} \leftarrow \bar{w}_t^{(0)} + \bar{w}_t^{(l)}$, and removing $h^{(l)}$ from the mixture. In this way, loop closures and tracking losses are realized as hypothesis mergers driven by the same THF machinery, yielding a unified and robust treatment of revisits under strong perceptual aliasing.

5 Experimental Evaluation

The objective of our experiments is to test whether CROSS can localize and act using a previously built spatial memory despite realistic environmental change. We evaluate this under three representative condition shifts: (i) illumination changes across time of day, (ii) long-term appearance changes across months, and (iii) object-level changes caused by rearrangement or motion.

Additional experiments and uncertainty analyses in Appendix B evaluate perceptual aliasing, a controlled ablation of the commitment policy, sensitivity to the measurement gates, graded object rearrangement, relocalization on a second robot, failure cases, fast motion, occlusion, noisy odometry, runtime, and Wilson confidence intervals for the binary success-rate metrics, showing that CROSS remains stable under degraded sensing and runs in real time.

5.1 Appearance Change Across Time

Datasets. We evaluate *change-robustness* under appearance variation using two public datasets (Figure 3). The OpenLORIS [38] *Corridor* scene is an indoor dataset with repeated traversals captured at different times of day, exhibiting substantial illumination variation along an approximately 80 m corridor. The Rover [37] *Campus* scene is an outdoor dataset comprising repeated traversals collected across different months and times of day, covering an approximately 270 m route and capturing long-term appearance changes due to lighting, weather, and seasonal effects.

Methods. We compare our approach against representative and state-of-the-art SLAM systems, including ORB-SLAM3 (RGBD) [5], RTAB-Map [19], and MASt3R-SLAM [30]. In addition, we evaluate common topological localization baselines: Greedy Matching (GM), Sequence Matching (SM), Probabilistic Belief Update (PBU), and Appearance-based Mapping [18]. GM, SM, and PBU represent widely used strategies for online topological mapping and localization [36, 26, 40]. Several methods are not publicly available, so we reimplemented the topological baselines using the same VPR models and comparable hyperparameters to ensure a fair comparison (see Appendix A.1).

Setup and Metrics. We evaluate all methods under a map–query protocol. A single traversal is first designated as the *mapping sequence*, during which the system incrementally constructs its spatial representation online. Relocalization performance is then evaluated on a separate traversal recorded under different environmental conditions (the *testing sequence*). The testing sequence is partitioned into fixed-length subsequences of 200 frames, each treated as an online relocalization trial. This protocol results in 735 trials and 1,232 trials for OpenLORIS and Rover, respectively.

Table 1: Relocalization success rate (RS) under appearance change on indoor (OpenLORIS) and outdoor (Rover) benchmarks, without vs. with self mapping.

Method	cross-sequence			w. self mapping		
	RS↑ (OLS)	RS↑ (Rover)	RS↑ (All)	RS↑ (OLS)	RS↑ (Rover)	RS↑ (All)
SLAM Systems						
ORB3 [5]	0.032	0.005	0.013	0.003	0.048	0.040
RTAB [19]	0.292	0.158	0.198	0.438	0.275	0.336
MASt3R [30]	0.173	0.014	0.062	0.226	0.024	0.094
Topological Methods						
GM	0.080	0.064	0.069	0.483	0.263	0.345
SM	0.073	0.152	0.128	0.479	0.333	0.387
PBU	0.080	0.063	0.068	0.482	0.262	0.344
ABM [18]	0.231	0.020	0.090	**0.569**	0.229	0.356
CROSS (Ours)	**0.353**	**0.397**	**0.384**	0.478	**0.494**	**0.488**

We report the *relocalization success* (RS) rate as the primary evaluation metric. For methods that output a discrete topological estimate (e.g., node or place ID), a trial is considered successful if the predicted location lies within a distance threshold r_D of the ground-truth position. For metric SLAM systems that output a full pose estimate, success is defined as the estimated pose lying within the same threshold r_D of the ground-truth pose. We set $r_D = 2\,\mathrm{m}$ for indoor environments and $r_D = 5\,\mathrm{m}$ for outdoor environments. We use two evaluation settings: *with self-mapping*, where the mapping and testing sequences are identical, and *cross-sequence*, where relocalization is performed using maps built from different sequences.

Results. Overall, CROSS achieves the highest relocalization success rates across both indoor and outdoor benchmarks (Table 1), indicating improved robustness to appearance change compared to existing SLAM-based and topological methods. Wilson confidence intervals for all success rates are provided in Appendix B.1. Absolute performance remains limited under severe condition shifts: even for CROSS, success rates in cross-sequence settings are around 40%. Performance degrades substantially under large appearance gaps, such as day-to-night transitions or long-term seasonal changes (e.g., January to September). While non-perfect success rates are acceptable in practice due to repeated relocalization opportunities as the robot moves, higher success rates directly translate to greater tolerance to appearance change and faster recovery.

5.2 Object Navigation on a Real Quadruped Robot

To evaluate whether the spatial memory facilitates downstream semantic navigation, we deployed CROSS on a quadruped robot in a changing indoor environment. This experiment is not intended as a benchmark of open-vocabulary semantic reasoning; instead, it tests whether our stored visual memory can be reused for localization and object-goal navigation under conditions commonly encountered in real-world deployment.

Task and Setup. During an initial *mapping session*, the robot constructs the spatial memory. In a subsequent *query session*, the robot is initialized at a random location within the same environment and tasked with *navigating to a specified target object* (e.g., a bin, plant, or coke bottle) that was previously observed.

We evaluate three experimental settings that capture common sources of environmental change. *Lighting Change (LC)* considers only appearance variation induced by different times of day between the mapping and query sessions. *Object Rearrangement (OR)* evaluates robustness to object-level changes, where furniture and objects are moved while lighting conditions remain similar. *Combined Change (LC+OR)* includes both time-of-day variation and object rearrangement simultaneously. Figure 5 illustrates representative examples of the LC, OR, and LC+OR settings. For each setting, we conduct 10 independent trials with different scene configurations and target objects. A trial is considered successful if the robot reaches the specified target object within a distance of 1 m.

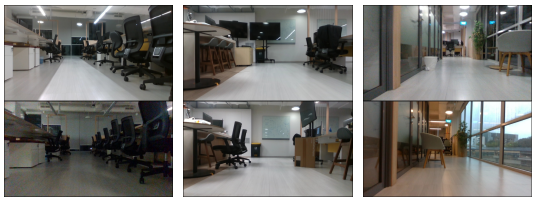


Figure 5: Example evaluation settings from the real quadruped-robot experiment, with each scene shown *before* (top) and *after* (bottom) the change. **Left:** Lighting Change (LC). **Middle:** Object Rearrangement (OR). **Right:** Combined Change (LC+OR).

Methods. We compare CROSS against the commonly used metric map + semantic layer **(M+S)** paradigm, which augments a metric SLAM map with an additional semantic layer, as adopted in prior work such as [7, 6, 13]. We implement two variants of this baseline, using ORB-SLAM3 [5] and RTAB-Map [19] for localization and mapping. Implementation details are provided in Appendices A.2 and D.4.

Results. Across all three settings, CROSS consistently achieves a higher task success rate than the baseline (Table 2), which demonstrates that the proposed representation can effectively support object-goal navigation under both appearance and object-level changes.

Table 2: Success rate (%) in real-robot object-goal navigation under environmental changes.

Method	LC	OR	LC+OR
M+S (ORB-SLAM3)	30.0	30.0	20.0
M+S (RTAB-Map)	40.0	60.0	30.0
CROSS (Ours)	**70.0**	**80.0**	**80.0**

We conducted a qualitative real-robot experiment to examine if CROSS remains reliable in the presence of dynamic agents (i.e., moving pedestrians) across different times of the day (Fig. 1). We performed a mapping session over a $\approx$170 m trajectory covering walkways and a university canteen. We then ran a query session when the same area was crowded with different lighting conditions, and commanded the robot to navigate to a language-specified target (security post). Despite substantial scene changes, the robot re-localized and completed the navigation task.

6 Conclusion and Future Work

We presented CROSS, a change-robust online topological spatial memory for long-term semantic navigation. CROSS stores raw RGB-D keyframes as task-agnostic visual observations and maintains a bounded Gaussian-mixture belief over continuous $\mathrm{SE}(3)$ poses. By constructing global visual observation modes and validating them through motion consistency over time, CROSS can recover from tracking loss, reject many perceptual aliases, and promote consistent branches to loop-closure constraints. Experiments on public indoor/outdoor benchmarks and real-robot object-goal navigation show improved reuse of a single memory under appearance and object-level change.

Limitations and Future Work. CROSS relies on reliable relative-pose estimates between the current RGB-D frame and retrieved keyframes. Our current implementation uses keypoint matching

and PnP, which can fail under extreme appearance shift, low overlap, motion blur, or noisy depth; the filtering formulation itself can also use learned relative-pose estimators when they become sufficiently accurate and efficient. A second limitation is semantic scope: this work uses stored observations and object-query annotations/detections to demonstrate semantic navigation, rather than evaluating full open-vocabulary spatial reasoning. Future work will improve relative-pose estimation under large visual change and evaluate richer language-conditioned spatial queries.

Acknowledgments and Disclosure of Funding

This research / project is supported by the National Research Foundation, Singapore, under its Thematic Competitive Research Programme 2025 (NRF-T-CRP-2025-0003) and partially supported by the Ministry of Education, Singapore, under the Academic Research Fund Tier 1 (T1 251RES2515). The authors declare no competing interests.

References

[1] Amar Ali-Bey, Brahim Chaib-draa, and Philippe Giguere. Boq: A place is worth a bag of learnable queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17794–17803, 2024.

[2] D. L. Alspach and H. W. Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, 1972. doi: 10.1109/TAC.1972.1100034.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] Gabriele Berton and Carlo Masone. Megaloc: One retrieval to place them all. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2861–2867, 2025.

[5] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

[6] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, Roozbeh Mottaghi, Jitendra Malik, and Devendra Singh Chaplot. GOAT: GO to any thing. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.073.

[7] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020.

[8] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 2013. doi: 10.1177/0278364913499193. URL `https://doi.org/10.1177/0278364913499193`.

[9] Mark Cummins and Paul Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011. doi: 10.1177/0278364910385483.

[10] M.A.T. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002. doi: 10.1109/34.990138.

[11] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI '99)*, pages 343–349, Orlando, FL, USA, July 1999. AAAI Press. URL `https://aaai.org/papers/050-aaai99-050-monte-carlo-localization-efficient-position-estimation-for-mobile-robots/`.

[12] Paul T. Furgale and Timothy D. Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560, 2010. doi: 10.1002/rob.20342. URL `https://doi.org/10.1002/rob.20342`.

[13] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Aditya Sen, Aditya Agarwal, Corban Rivera, William Knudson, Erik Sudderth, Oscar Beijbom, et al. ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[14] Ming Hsiao and Michael Kaess. MH-iSAM2: Multi-hypothesis iSAM using bayes tree and hypo-tree. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1274–1280, Montreal, QC, Canada, May 2019. IEEE. doi: 10.1109/ICRA.2019.8793854. URL `https://doi.org/10.1109/ICRA.2019.8793854`.

[15] Ming Hsiao, Joshua G. Mangelson, Sudharshan Suresh, Christian Debrunner, and Michael Kaess. ARAS: Ambiguity-aware robust active SLAM based on multi-hypothesis state and map estimations. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5037–5044, Las Vegas, NV, USA, October 2020. IEEE. doi: 10.1109/IROS45743.2020.9341384. URL `https://doi.org/10.1109/IROS45743.2020.9341384`.

[16] N. Hughes, Y. Chang, and L. Carlone. Hydra: A real-time spatial perception system for 3D scene graph construction and optimization. In *Robotics: Science and Systems (RSS)*, 2022.

[17] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. ConceptFusion: Open-set multimodal 3D mapping. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[18] Mathieu Labbe and Francois Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3):734–745, 2013.

[19] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of field robotics*, 36(2):416–446, 2019.

[20] Yasir Latif, César Cadena, and José Neira. Robust loop closing over time for pose graph SLAM. *The International Journal of Robotics Research*, 32(14):1611–1626, December 2013. doi: 10.1177/0278364913498910. URL `https://doi.org/10.1177/0278364913498910`.

[21] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81(2):155–166, 2009.

[22] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *European Conference on Computer Vision*, pages 163–179. Springer, 2024.

[23] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 17627–17638, 2023.

[24] Peiqi Liu, Yaswanth Orru, Jay Vakil, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. OK-Robot: What really matters in integrating open-knowledge models for robotics. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.091.

[25] Will Maddern, Michael Milford, and Gordon Wyeth. CAT-SLAM: Probabilistic localisation and mapping using a continuous appearance-based trajectory. *The International Journal of Robotics Research (IJRR)*, 31(4):429–451, 2012. doi: 10.1177/0278364912438273.

[26] Xiangyun Meng, Nathan Ratliff, Yu Xiang, and Dieter Fox. Scaling local control to large-scale topological navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 672–678. IEEE, 2020.

[27] Michael J Milford and Gordon F Wyeth. Mapping a suburb with a single camera using a biologically inspired slam system. *IEEE Transactions on Robotics*, 24(5):1038–1053, 2008.

[28] Michael J. Milford and Gordon F. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1643–1649, 2012. doi: 10.1109/ICRA.2012.6224623.

[29] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European conference on computer vision*, pages 728–755. Springer, 2022.

[30] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025.

[31] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *The International Journal of Robotics Research*, 32(7):826–840, June 2013. doi: 10.1177/0278364913479413. URL `https://doi.org/10.1177/0278364913479413`.

[32] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Xfeat: Accelerated features for lightweight image matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2682–2691, 2024.

[33] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *IEEE Transactions on pattern analysis and machine intelligence*, 21(8):774–780, 1999.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[35] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Radar Library. Artech House, Boston, London, 2004. ISBN 9781580536318.

[36] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations*, 2018.

[37] Fabian Schmidt, Julian Daubermann, Marcel Mitschke, Constantin Blessing, Stephan Meyer, Markus Enzweiler, and Abhinav Valada. Rover: A multi-season dataset for visual slam. *IEEE Transactions on Robotics*, 2025.

[38] Xuesong Shi, Dongjiang Li, Pengpeng Zhao, Qinbin Tian, Yuxin Tian, Qiwei Long, Chunhao Zhu, Jingwei Song, Fei Qiao, Le Song, Yangquan Guo, Zhigang Wang, Yimin Zhang, Baoxing Qin, Wei Yang, Fangshi Wang, Rosa H. M. Chan, and Qi She. Are we ready for service robots? the OpenLORIS-Scene datasets for lifelong SLAM. In *2020 International Conference on Robotics and Automation (ICRA)*, pages 3139–3145, 2020.

[39] Niko Sünderhauf and Peter Protzel. Switchable constraints for robust pose graph SLAM. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1879–1884, Vilamoura, Algarve, Portugal, October 2012. IEEE. doi: 10.1109/IROS.2012.6385590. URL `https://doi.org/10.1109/IROS.2012.6385590`.

[40] Lauri Suomela, Jussi Kalliola, Harry Edelman, and Joni-Kristian Kämäräinen. Placenav: Topological navigation through place recognition. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5205–5213. IEEE, 2024.

[41] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1–2):99–141, May 2001. doi: 10.1016/S0004-3702(01)00069-8. URL `https://doi.org/10.1016/S0004-3702(01)00069-8`.

[42] S Urban, J Leitloff, and S Hinz. Mlpnp–a real-time maximum likelihood solution to the perspective-n-point problem. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3: 131–138, 2016.

[43] Olga Vysotska and Cyrill Stachniss. Lazy data association for image sequences matching under substantial appearance changes. *IEEE Robotics and Automation Letters*, 1(1):213–220, January 2016. doi: 10.1109/LRA.2015.2512936. URL `https://doi.org/10.1109/LRA.2015.2512936`.

[44] Jiaming Wang and Harold Soh. Probable object location (polo) score estimation for efficient object goal navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5221–5227. IEEE, 2024.

[45] Jiaming Wang, Diwen Liu, Jizhuo Chen, Jiaxuan Da, Nuowen Qian, Minh Man Tram, and Harold Soh. Genie: A generalizable navigation system for in-the-wild environments. *IEEE Robotics and Automation Letters*, 2025.

[46] Jiaming Wang, Diwen Liu, Jizhuo Chen, and Harold Soh. Topo-bench: An open-source topological mapping evaluation framework with quantifiable perceptual aliasing. *arXiv preprint arXiv:2510.04100*, 2025.

[47] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[48] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.077.

[49] Ming Xu, Tobias Fischer, Niko Sünderhauf, and Michael Milford. Probabilistic appearance-invariant topometric localization with new place awareness. *IEEE Robotics and Automation Letters*, 6(4):6985–6992, October 2021. doi: 10.1109/LRA.2021.3096745. URL `https://doi.org/10.1109/LRA.2021.3096745`.

[50] Ming Xu, Niko Sünderhauf, and Michael Milford. Probabilistic visual place recognition for hierarchical localization. *IEEE Robotics and Automation Letters*, 6(2):311–318, April 2021. doi: 10.1109/LRA.2020.3040134. URL `https://doi.org/10.1109/LRA.2020.3040134`.

[51] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.

[52] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European conference on computer vision*, pages 350–368. Springer, 2022.

[53] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, and Hesheng Wang. Sni-slam: Semantic neural implicit slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21167–21177, 2024.

A Experiment Details

A.1 Topological Localization Baselines

This appendix describes the topological localization baselines used in our experiments. All methods operate on the same topological graph and use the same underlying VPR model [1] for computing visual similarity scores, differing only in how temporal information and motion constraints are incorporated.

A.1.1 Greedy Matching (GM)

The greedy matching baseline localizes by selecting the node with the highest similarity score to the current observation. If the maximum similarity exceeds a fixed threshold τ, the corresponding node is selected as the localization result; otherwise, the localization estimate remains unchanged. This baseline reflects a common retrieval-based relocalization strategy without temporal reasoning.

A.1.2 Sequence Matching (SM)

Instead of matching a single observation, sequence matching aggregates similarity scores over a short temporal window to improve robustness against perceptual aliasing and viewpoint variation. A candidate match between nodes (v_i, v_j) is accepted if the aggregated similarity over a window of size $2h{+}1$ satisfies

$$\mathrm{f}\big(\mathrm{sim}(z_{v_i-h}, z_{v_j-h}), \ldots, \mathrm{sim}(z_{v_i+h}, z_{v_j+h})\big) \geq \tau,$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes the visual similarity score and $\mathrm{f}(\cdot)$ is an aggregation function. In our implementation, we use the median of the similarity scores within the window, which provides robustness to outliers. This baseline is inspired by prior work on sequence-based place recognition [36, 26].

A.1.3 Probabilistic Belief Update (PBU)

The probabilistic belief update baseline maintains a discrete posterior belief $b_t(v) = P(v_t = v \mid z_{1:t})$ over the topological nodes $v \in \mathcal{V}$ at time t. Given the belief at the previous timestep, the state is first propagated via a motion model $P(v_t \mid v_{t-1})$ that constrains allowable transitions based on the graph topology:

$$P(v_t \mid v_{t-1}) \propto \begin{cases} \alpha & \text{if } \mathrm{dist}_G(v_t, v_{t-1}) \leq w_u, \\ \beta & \text{otherwise,} \end{cases}$$

where $\mathrm{dist}_G(\cdot,\cdot)$ denotes the hop distance in the graph, w_u is the maximum allowed movement per timestep, and α, β are constants controlling transition likelihoods.

The predicted belief is then updated using the retrieval similarity scores as the observation likelihood $P(z_t \mid v_t) \propto g(\mathrm{sim}(z_t, z_{v_t}))$, where $g(\cdot)$ maps similarity scores to likelihood values. The recursive update is formulated as:

$$b_t(v_t) = \eta \cdot P(z_t \mid v_t) \sum_{v_{t-1} \in \mathcal{V}} P(v_t \mid v_{t-1}) b_{t-1}(v_{t-1}),$$

where η is a normalization constant. This discrete Bayesian filtering suppresses spurious matches that violate motion constraints and improves robustness to perceptual aliasing, following prior topological localization approaches [40].

A.2 Real robot experiment implementation

We design the real robot experiments to test relocalization and object goal navigation, under controlled combinations of appearance change and object rearrangement. To prevent differences in environment coverage, the spatial-semantic representation for all methods is constructed from data collected along an identical mapping trajectory in each scene. The navigation policy performs local obstacle avoidance and relative waypoint tracking, and does not consider the downstream impact of navigation actions on the state estimation. This setup allows us to attribute success or failure rates to the robustness of the underlying representation, rather than to planning or navigation strategies. We augment the default map representation of ORB-SLAM3 [5] to include object memory by associating the recorded RGB observations to computed map poses via timestamp alignment. RTAB-MAP in its default

Table 3: Localization performance under perceptual aliasing on *Topo-Bench*. Results are reported for: *Ambiguous + Positive* (A+P), *Positive Only* (P.O.), *Ambiguous Only* (A.O.) and *Balanced Localization Accuracy* (BLA), defined as the geometric mean over the three regimes.

Method	A+P	P.O.	A.O.	BLA
RTAB-Map [19]	0.059	**0.433**	**1**	0.301
ORB-SLAM3 [5]	0.059	0.183	**1**	0.227
ABM [18]	0.02	0.328	**1**	0.201
OpenFabmap	0	0.112	0.732	0.0748
Rat-Slam	0.02	0.097	0.443	0.103
GM	0.078	0.302	0.959	0.288
SM	0.118	0.187	0.959	0.281
PBU	0.078	0.302	0.959	0.288
CROSS (Ours)	**0.275**	0.336	0.99	**0.452**

implementation provides the associated RGB image for each map pose. Consequently, all methods considered for evaluation support semantic queries, which we use for the object navigation task. We retain the default tunable parameter values for all methods, to maintain fairness in comparison and avoid method specific tuning.

B Additional Experiments, Results, and Analysis

B.1 Statistical Uncertainty

All main success metrics are binary trial outcomes. A relocalization trial either succeeds or fails under the distance threshold defined in Section 5.1, and a real-robot object-navigation trial either reaches the target object within $1\,\mathrm{m}$, as defined in Section 5.2, or fails. We therefore report two-sided 95% Wilson score confidence intervals for Bernoulli proportions. For a success rate $\hat{p} = k/n$, the interval is

$$\frac{\hat{p} + \frac{z^2}{2n}}{1 + \frac{z^2}{n}} \pm \frac{z}{1 + \frac{z^2}{n}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}, \qquad z = 1.96.$$

This interval does not assume normally distributed errors and remains well-defined for small success counts. For the public relocalization benchmarks, we use the trial counts in Section 5.1: $n = 735$ for OpenLORIS and $n = 1232$ for Rover. Since Table 1 reports rates rounded to three decimals, the intervals below are computed by rounding each reported proportion to the nearest integer count. For the real-robot study, each setting contains $n = 10$ independent trials.

Table 4: Approximate 95% Wilson confidence intervals for relocalization success rates in Table 1. Values are percentages. OLS denotes OpenLORIS.

Method	CROSS OLS	CROSS Rover	Self OLS	Self Rover
ORB-SLAM3	3.2 [2.2, 4.7]	0.5 [0.2, 1.1]	0.3 [0.1, 1.0]	4.8 [3.7, 6.1]
RTAB-Map	29.2 [26.1, 32.6]	15.8 [13.9, 18.0]	43.8 [40.2, 47.4]	27.5 [25.1, 30.1]
MASt3R-SLAM	17.3 [14.8, 20.2]	1.4 [0.9, 2.2]	22.6 [19.7, 25.7]	2.4 [1.7, 3.5]
GM	8.1 [6.3, 10.3]	6.4 [5.2, 7.9]	48.3 [44.7, 51.9]	26.3 [23.9, 28.8]
SM	7.2 [5.6, 9.3]	15.2 [13.3, 17.3]	47.9 [44.4, 51.5]	33.3 [30.8, 36.0]
PBU	8.1 [6.3, 10.3]	6.3 [5.1, 7.8]	48.2 [44.6, 51.8]	26.2 [23.8, 28.7]
ABM	23.1 [20.2, 26.3]	2.0 [1.4, 3.0]	56.9 [53.3, 60.4]	22.9 [20.6, 25.3]
CROSS	**35.3** [**32.0**, **38.8**]	**39.7** [**37.0**, **42.5**]	47.8 [44.2, 51.4]	**49.4** [**46.6**, **52.2**]

The intervals support the main conclusion that CROSS improves cross-sequence relocalization under appearance change. The separation is strongest on Rover, where the interval for CROSS is well separated from all baselines. On OpenLORIS cross-sequence relocalization, CROSS has the highest point estimate, but its interval overlaps with RTAB-Map; we therefore interpret this result as improved average performance rather than definitive per-trial dominance. In the self-mapping setting, ABM has the highest OpenLORIS point estimate, while CROSS has the highest Rover point estimate and the highest aggregate rate in Table 1.

Table 5: 95% Wilson confidence intervals for real-robot object-goal navigation success rates in Table 2. Values are percentages; each setting has $n = 10$ trials.

Method	LC	OR	LC+OR	All settings
M+S (ORB-SLAM3)	30.0 [10.8, 60.3]	30.0 [10.8, 60.3]	20.0 [5.7, 51.0]	26.7 [14.2, 44.4]
M+S (RTAB-Map)	40.0 [16.8, 68.7]	60.0 [31.3, 83.2]	30.0 [10.8, 60.3]	43.3 [27.4, 60.8]
CROSS	**70.0 [39.7, 89.2]**	**80.0 [49.0, 94.3]**	**80.0 [49.0, 94.3]**	**76.7 [59.1, 88.2]**



(1) (2) (3) (4) (5) (6)

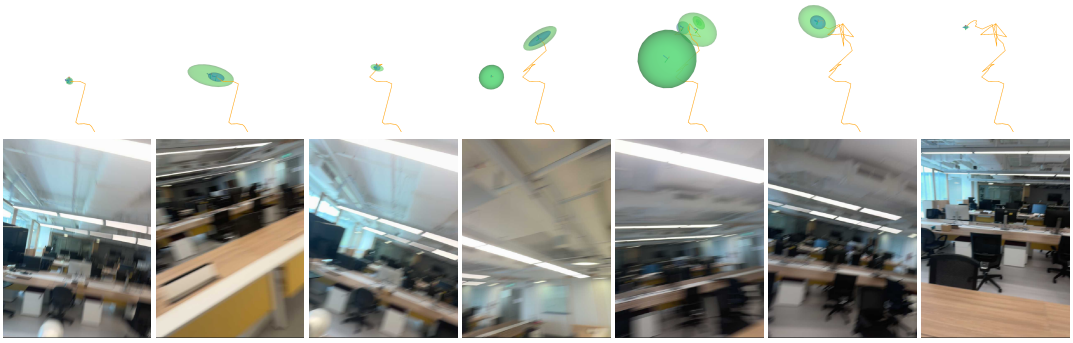
Figure 6: Occlusion sequence across six timestamps. Top row shows the online mapping and belief evolution of our system: the yellow line denotes the estimated trajectory; blue ellipses represent the propagated motion message, and green ellipses represent the measurement message. The lengths of the principal axes indicate the estimated uncertainty. Bottom row shows the corresponding RGB observations. At timestamp (1), with no occlusion, uncertainty is low. At (2) and (3), occlusion prevents reliable visual observations, causing the motion message to continue propagating using local odometry while the measurement message remains static. At (4), the two messages overlap, indicating successful retrieval and alignment with the stored map. At (5), occlusion occurs again and uncertainty increases as the motion message propagates. Finally, at (6), once the occlusion disappears, uncertainty rapidly decreases and the system localizes with high confidence.

The real-robot intervals are wide because each condition contains only ten trials. Consequently, we treat the real-robot experiment as evidence that the representation transfers to embodied object-goal navigation under realistic changes, while avoiding strong claims of fine-grained statistical separation between all methods and all conditions.

B.2 Topo-Bench Evaluation

We evaluate robustness to perceptual aliasing using *Topo-Bench* [46], which categorizes perceptual aliasing scenarios into three regimes: *Ambiguous + Positive* (A+P), *Positive Only* (P.O.), and *Ambiguous Only* (A.O.). A+P represents revisits to previously mapped regions where strong distractor matches closely resemble the true location; P.O. corresponds to revisits without highly similar distractors; and A.O. captures novel regions that spuriously resemble known locations. To quantify balanced performance across these regimes, Topo-Bench reports *Balanced Localization Accuracy* (BLA), defined as the geometric mean of accuracies on A+P, P.O., and A.O., thereby penalizing methods with imbalanced behavior. Further details regarding these scenarios and metrics can be found in [46].

Quantitative results are summarized in Table 3. All methods exhibit low accuracy in the challenging A+P regime, but our approach achieves the highest A+P performance, demonstrating the effectiveness of the proposed THF mechanism in resolving aliasing under appearance changes. At the same time, it maintains performance comparable to prior SOTA methods on P.O. and A.O., resulting in the best overall BLA. This indicates stronger and more balanced robustness across diverse aliasing scenarios.



(1) (2) (3) (4) (5) (6) (7)

Figure 7: Fast-motion sequence across seven timestamps. Top row shows the online mapping trajectory of our system: yellow lines indicate the estimated trajectory; blue ellipses denote the Gaussian mixture of the measurement message, while cyan ellipses denote the Gaussian mixture of the propagated motion message. The principal axis lengths of each ellipse correspond to the estimated uncertainty (variance) along that direction. Bottom row shows the corresponding RGB observations. At timestamp (1), uncertainty is low. At (2), observation uncertainty increases significantly due to severe motion blur. At (3), the system maintains tracking with high uncertainty despite residual blur. At (4) and (5), observation uncertainty becomes very large again due to heavy blur, introducing spurious hypotheses. At (6) and (7), the system rapidly eliminates these spurious hypotheses via temporal hypothesis filtering once visual observations return to normal.



SNR=0.2 SNR=0.5 SNR=1

Figure 8: Noisy-odometry experiment under different signal-to-noise ratio (SNR) settings. Each column shows the full trajectory for the same sequence with increasing levels of odometry noise injected via right-multiplicative perturbations in $\mathrm{SE}(3)$. The yellow line denotes the trajectory estimated by our system, the green line shows the reference trajectory provided by the SLAM system, and the blue line corresponds to the corrupted odometry used for motion propagation. As the SNR decreases, the odometry becomes increasingly distorted, as evidenced by the highly corrupted blue trajectories. Notably, even at SNR$= 0.2$, which corresponds to very large noise relative to the true odometry measurements, our estimated trajectory remains close to the reference, demonstrating robustness to severe odometric corruption.

B.3 Robustness to Odometry Noise, Occlusion, and Fast Motion

We found our system is robust to fast motion and camera occlusion: the states are propagated when image matching is not possible due to the camera blur or occlusion, but the global measurement message using VPR model can quickly retrieve the correct frame once it is available.

Figure 7 and Figure 6 provide qualitative demonstrations of how the proposed system performs when fast motion and occlusion occur.

Our system relies on the odometry information for high frequency updates. To test how our system performs when the odometry is noisy, we record trajectories with RGBD and pose information using the ARKit API on iPhone. We then add noise to the computed delta pose between two consecutive frames as the noisy odometry input as follows:

Given an odometry pose $T \in \mathrm{SE}(3)$, we apply a right-multiplicative perturbation

$$\tilde{T} = T \exp(\xi), \qquad \xi = \begin{bmatrix} \delta\rho \\ \delta\phi \end{bmatrix} \in \mathbb{R}^6,$$

where $\exp(\cdot)$ is the Lie exponential. The translational and rotational noises are sampled as

$$\delta\rho \sim \mathcal{N}(0, \sigma_t^2 I_3), \qquad \delta\phi \sim \mathcal{N}(0, \sigma_r^2 I_3),$$

with units of meters and radians, respectively.

To obtain a scale-aware and interpretable noise level, we set

$$\sigma_t = \frac{\|t\|}{\mathrm{snr}\sqrt{3}}, \qquad \sigma_r = \frac{\|\log(R)\|}{\mathrm{snr}\sqrt{3}},$$

where $T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$ and $\log(R) \in \mathfrak{so}(3)$. Since $\mathbb{E}\|\delta\rho\|^2 = 3\sigma_t^2$ and $\mathbb{E}\|\delta\phi\|^2 = 3\sigma_r^2$, the expected noise magnitudes scale as $\|t\|/\mathrm{snr}$ and $\|\log(R)\|/\mathrm{snr}$, respectively, yielding an SNR-like control over pose perturbations.

Figure 8 provides a qualitative evaluation of our system under different odometry noise levels. As the signal-to-noise ratio (SNR) decreases, the injected perturbations induce progressively larger drift in the motion propagation. Despite increasingly noisy odometry, the system remains stable and is able to recover and maintain a coherent trajectory by leveraging visual observations and temporal hypothesis filtering, demonstrating robustness to substantial odometric corruption.

B.4 Ablation Study

Our ablations are designed to diagnose the two components most directly tied to the central claim: (i) representing competing explanations as continuous $\mathrm{SE}(3)$ pose modes rather than discrete place IDs, and (ii) constructing global measurement modes using geometrically verified relative pose estimates.

Continuous versus discrete hypothesis testing. PBU is the closest controlled baseline for testing the state representation. It uses the same topological graph, the same BoQ retrieval scores, the same query protocol, and the same high-precision calibration procedure, but its belief is defined over discrete node IDs rather than continuous poses. Thus, the PBU–CROSS comparison isolates the additional value of retaining continuous pose evidence during temporal hypothesis filtering. As shown in Table 1, CROSS improves aggregate cross-sequence relocalization success from 0.068 to 0.384 over PBU, with the largest gain on Rover (0.063 to 0.397). This indicates that many visually plausible matches cannot be rejected by discrete temporal consistency alone; maintaining hypotheses in $\mathrm{SE}(3)$ allows odometry to test whether a retrieved place is physically consistent before promoting it.

Beyond VPR-only temporal aggregation. GM, SM, and PBU share the same VPR backend where applicable and differ primarily in how temporal evidence is accumulated: single-frame retrieval, sequence-level aggregation, and discrete Bayesian filtering. ABM provides an additional appearance-based mapping baseline. The lower cross-sequence success rates of these methods indicate that CROSS's improvement is not due solely to the BoQ descriptor or to temporal smoothing of retrieval scores. Instead, the gain comes from coupling VPR proposals with relative-pose verification and delayed branch validation in continuous pose space.

Relative-pose estimator. Another key design choice is the use of a classical PnP-based pipeline for relative pose estimation between frames. As an alternative, we evaluated a learned variant by replacing the XFeat/LightGlue + PnP module with VGGT [47]. In our relocalization setting, this variant achieved zero success rate. We observed two practical failure modes: VGGT predicts poses with arbitrary scale across independent inferences, which prevents reliable global pose composition, and its average runtime is 490 ms per forward pass on an RTX4000 GPU, compared with 16 ms for the PnP-based relative-pose module. This ablation does not imply that learned relative-pose estimation is unsuitable in principle; rather, it shows that the current system benefits from a metric RGB-D PnP estimate whose scale is fixed and whose inlier support provides an explicit quality signal for measurement weighting.

Commitment policy with identical inputs. To isolate the pre-commitment layer, we first map an office environment during the daytime and then evaluate the system on ten nighttime query trajectories that revisit the same area under substantial changes in illumination and object layout. Immediate, Discrete, and CROSS use the same map, BoQ retrievals, PnP relative-pose estimates, odometry, query trajectories, and PGO backend; the only difference is the commitment policy. *Immediate* commits the first eligible top visual/PnP association. *Discrete* filters a belief over map-node IDs using visual likelihoods and graph-hop motion. *CROSS* propagates continuous pose branches with metric odometry and requires sustained evidence before graph modification. We evaluate the first applied relocalization commitment. It is correct when the selected map location is within $1\,\mathrm{m}$ of the true query location; a wrong or absent commitment is a failure.

Table 6: First applied relocalization commitment in ten nighttime query trajectories. All policies use the same map, retrievals, relative-pose estimates, odometry, and PGO backend.

Commitment policy	Correct	Wrong	No commitment	Success
Immediate	5/10	5/10	0/10	50%
Discrete belief	6/10	4/10	0/10	60%
CROSS (Ours)	**8/10**	**0/10**	2/10	**80%**

The failure patterns differ (Table 6). Immediate accepts the first visually and geometrically plausible association and makes five wrong attachments. Discrete improves to six correct commitments, but graph-hop consistency is only a proxy for motion: uneven keyframe spacing and different mapping and query speeds can reinforce a wrong region, which likely caused its four wrong attachments. CROSS tests continuous-pose compatibility with metric odometry; all eight of its applied commitments are correct, while its two failures are conservative no-commits under insufficient coherent support. It therefore improves both success (80% vs. 60% and 50%) and association safety (no wrong commitment vs. four and five). A no-commit leaves the graph unchanged, whereas a wrong commitment can corrupt it. We do not report a robust-PGO-only variant: our PGO sees only factors accepted upstream, so enabling its robust kernel never exposes raw ambiguities, and a fair baseline would require a separate path that inserts and manages raw candidate factors.

B.5 Sensitivity to the Measurement Gates

Two visual-measurement parameters determine which VPR/PnP associations enter the measurement message (5): the VPR score threshold and the minimum number of PnP-RANSAC inliers. To test the sensitivity to them, we first construct a daytime map of an office environment and then perform 10 relocalization trials from randomly sampled initial poses. Throughout the study, we keep the retrieval model, relative-pose estimator, odometry, PGO backend, and commitment rule fixed, so that performance changes can be attributed to the two parameters. We vary one parameter at a time by -50% to $+50\%$ around its nominal value. Success means that the first applied map commitment is within $1\,\mathrm{m}$ of the query's true location; a wrong or absent commitment is counted as a failure.

Table 7: Relocalization success (out of ten trials) when one measurement gate is varied around its nominal value.

Relative change	VPR threshold	Success	Min. PnP inliers	Success
-50%	0.15	9/10	5	8/10
-40%	0.18	9/10	6	8/10
-30%	0.21	9/10	7	8/10
-20%	0.24	9/10	8	8/10
-10%	0.27	8/10	9	8/10
Nominal	0.30	8/10	10	8/10
$+10\%$	0.33	8/10	11	8/10
$+20\%$	0.36	8/10	12	8/10
$+30\%$	0.39	8/10	13	8/10
$+40\%$	0.42	7/10	14	8/10
$+50\%$	0.45	7/10	15	8/10

The PnP result is invariant throughout the tested range: changing the minimum support from 5 to 15 inliers leaves success at 8/10 (Table 7). VPR is also stable around the operating point: every setting

from -10% through $+30\%$ gives the nominal 8/10, while relaxing the gate by 20–50% improves the result to 9/10. Only aggressive tightening at $+40\%$ and $+50\%$ reduces success to 7/10. Overall, CROSS shows low sensitivity to both parameters within the tested $\pm 50\%$ range, with success between 7/10 and 9/10 in all configurations.

B.6 Graded Object Rearrangement

To test how CROSS performs as object rearrangement becomes more severe, we run a controlled stress test on one office route. We construct the map once in the Base configuration and reuse it without modification. We then collect ten true-revisit trajectories in each rearranged scene. The levels are cumulative: Middle retains the Easy changes, and Hard retains both the Easy and Middle changes (Table 8). The scene definitions follow the recorded object interventions; approximate distances describe each individual translation, while rotations, additions and removals, and opening actions are listed separately. We judge the first map commitment correct when it lies within $1\,\mathrm{m}$ of the query's true location; a wrong or absent commitment is a failure. Time and distance are measured at the first correct commitment and summarized over successful trajectories only.

Table 8: Scene changes in the graded rearrangement test, relative to the preceding scene.

Scene	**Changes relative to the preceding scene**
Base (mapping)	Original office arrangement; no objects are moved.
Easy	Rotate one cardboard box and add a pile of boxes; move the camera stand by ∼1 m, whiteboard by ∼0.5 m, stools by ∼1 m, a small cabinet by ∼1 m, and a jacket by ∼3 m.
Middle	Keep all Easy changes; additionally move two beanbags by ∼2 m each, three chairs by ∼2 m each, three cardboard items by ∼6 m each, two posters by ∼2 m each, the trash bin by ∼10 m, the round table by ∼4 m, and one monitor by ∼1 m; open one curtain.
Hard	Keep all Middle changes; additionally move the backdrop by ∼3 m, two chairs by ∼4 m each, both beanbags again by ∼6 m each, two monitors by ∼2 m each, two small cabinets by ∼1 m each, the fire extinguisher by ∼3 m, and one stool by ∼2 m; rotate the short table, remove another stool, and open two upper cabinet doors.

Table 9: Relocalization under graded object rearrangement (ten true-revisit trajectories per scene). Time and distance to the first correct commitment are summarized over successful trajectories.

Scene	Correct	Wrong	No commitment	Time / distance to correct commitment
Easy	10/10	0	0	median 3.176 s / 0.373 m (range 2.104–3.489 s / 0.310–0.436 m)
Middle	5/10	0	5	median 6.981 s / 2.525 m
Hard	1/10	0	9	2.304 s / 0.335 m for the single success

The trend is straightforward (Table 9). In Easy, CROSS relocalizes in every trial shortly after starting, despite the box additions and several small-object movements. In Middle, the additional changes include relocating the table and the trash bin by approximately 4 m and 10 m, together with multiple chairs, beanbags, boxes, posters, and a monitor. The five successful trajectories require more observations before committing, which raises the median time by $2.2\times$ and the distance by $6.8\times$ relative to Easy. The other five trajectories retrieve the relevant area and form plausible relative-pose measurements, but those measurements are not sufficiently consistent over time, so CROSS does not commit. In Hard, we further move the backdrop, beanbags, chairs, monitors, cabinets, extinguisher, and stool, while also rotating or removing furniture and opening cabinet doors. Only one trajectory accumulates enough consistent evidence to relocalize; the other nine end without a commitment. None of the 16 commitments made across the three scenes is attached to a wrong location. Thus, relocalization coverage decreases monotonically from 100% to 50% to 10%, while the failures under severe rearrangement are conservative no-commits rather than incorrect map updates.

B.7 Relocalization on a Stretch 3 Robot

To further evaluate robustness on a different robot embodiment, we conducted an experiment on a Stretch 3 robot. We first constructed a daytime map of an office environment and then performed 10 relocalization trials from randomly sampled initial poses, comparing CROSS against ORB-SLAM3 [5] and RTAB-Map [19] based systems. All components of CROSS, including the retrieval model, relative-pose estimator, odometry, PGO backend, and commitment rule, were kept fixed during evaluation. Table 10 reports the results with 95% Wilson confidence intervals (Appendix B.1).

Table 10: Relocalization on a Stretch 3 robot (ten trials per system), with 95% Wilson confidence intervals and failure types.

System	Success (95% Wilson CI)	Relocalization failed	Localization lost	Timeout
ORB-SLAM3	2/10, 20% [5.7, 51.0]	5	3	0
RTAB-Map	7/10, 70% [39.7, 89.2]	2	0	1
CROSS (Ours)	**9/10, 90% [59.6, 98.2]**	0	0	1

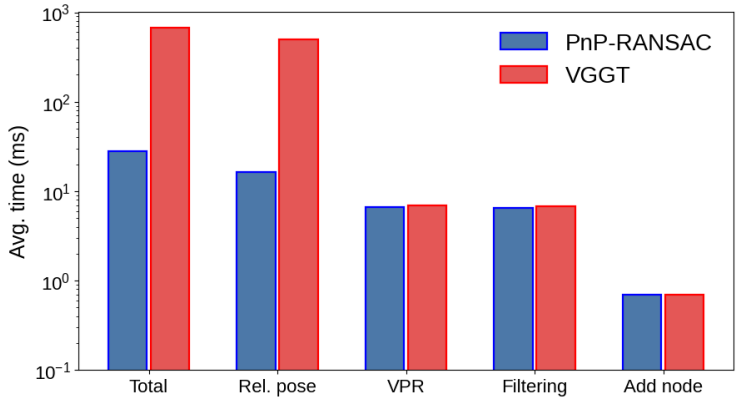


Figure 9: Runtime breakdown of the mapping pipeline per step, comparing relative pose estimation via **PnP-RANSAC** versus **VGGT** [47]. Bars report the *total* average step time and its main components: relative pose estimation (Rel. pose), visual place recognition retrieval (VPR), and temporal hypothesis filtering. The log-scale y-axis highlights the large disparity in pose-estimation cost.

CROSS succeeds in 9 of 10 trials, and its only failure is a timeout. This study shows that CROSS remains effective beyond the original quadruped platform. As in the quadruped study, ten trials give wide confidence intervals, and the interval of CROSS overlaps that of RTAB-Map.

B.8 Failure Analysis of the Real-Robot Trials

The quadruped study has seven unsuccessful CROSS trials among its 30 runs (7/10, 8/10, and 8/10 successes in the three settings), and the Stretch 3 study has one unsuccessful trial among ten. In all eight cases, VPR retrieved candidates from the relevant map region; the failure occurred in the subsequent 2D-to-3D correspondence and relative-pose stage. We observed three recurring causes: (i) extensive object motion removes or displaces much of the mapped geometry, leaving too few mutually consistent 2D-to-3D correspondences; (ii) transparent or reflective surfaces, especially windows, produce unreliable depth and therefore incorrect or missing 3D back-projections; and (iii) extremely dark observations degrade feature detection and matching, preventing stable correspondences from being formed. Thus, appearance retrieval can identify the relevant place while the subsequent correspondence and relative-pose stage lacks sufficient geometric evidence for localization. CROSS uses the motion message and temporal continuous-$\mathrm{SE}(3)$ consistency to verify this geometric evidence before map commitment. Consequently, these front-end failures produce no localization and no commitment, rather than a wrong commitment to an incorrect map region: when the relative pose cannot be supported reliably, CROSS preserves the map instead of accepting the VPR candidate alone.

B.9 Runtime Analysis

Figure 9 presents a runtime breakdown of our system measured on an RTX 4000 GPU workstation, detailing the computation time of each major module. The complete pipeline requires approximately 28 ms per step, enabling real-time operation at over 30 Hz.

In practical deployments with more limited computational resources, the full pipeline need not be executed at such a high frequency. Local odometry can propagate the robot state at high rate between

global updates, while the observation module can construct observations at a lower frequency without degrading overall performance.

B.10 Compute Resources

Offline experiments were run on two local workstations; no cloud TPU/large-scale cluster resources were used. The main GPU workstation used an AMD Ryzen 9 9900X (12 physical cores, 24 threads), 59.4 GB system memory, an NVIDIA RTX 4000 Ada Generation GPU with 19.6 GB VRAM, a 937 GB NVMe system SSD, and a 4 TB NVMe data SSD. RTAB-Map was run CPU-only on a separate AMD Ryzen Threadripper 9960X workstation with 48 logical threads and 128 GB system memory. The software environment used Linux 6.11.0, CUDA 12.6, Python 3.11.13, PyTorch 2.7.1+cu126, torchvision 0.22.1+cu126, OpenCV 4.11.0, NumPy 1.26.4, and pandas 2.3.1.

For CROSS, the online mapping/localization pipeline runs in approximately 28 ms per RGB-D frame on an RTX 4000 GPU workstation, including visual place recognition, relative pose estimation, and temporal hypothesis filtering (Fig. 9). The PnP-based relative pose module takes approximately 16 ms per frame; the VGGT ablation takes approximately 490 ms per forward pass on the same GPU, and was therefore not used in the main real-time system.

The real robot object navigation experiment is conducted on a Unitree Go2 EDU quadruped robot platform, equipped with an Intel Realsense D435i camera. The robot's onboard computer is connected to an off-board HP Omen Transcend 14 Laptop (Intel Core Ultra 7 155H processor, 16 GB RAM, NVIDIA GeForce RTX 4050 GPU) over WiFi. The camera data is streamed to the off-board laptop for updating the CROSS map and pose estimate in real time. This off-board setup is used to enable faster compute cycles as well as a longer runtime for the robot; as against an on-board processing unit which would either draw power from the robot's battery or an external mounted battery, which would in turn add payload weight decreasing the robot's runtime.

Table 11 summarizes the compute needed to reproduce each reported experiment. Wall-clock times include mapping, query-time localization, baseline evaluation, metric computation, and logging, but exclude dataset download time. Storage includes the raw datasets, extracted features, intermediate logs, and output result files. The table separates the OpenLORIS/Rover CROSS benchmark runs, the corresponding SLAM/topological baseline runs, the Topo-Bench evaluation, and the robot/ablation experiments. The full research project did not require substantially more compute than the experiments reported here; exploratory development and failed runs required approximately 100 additional hours.

C Broader Impacts

CROSS is intended to improve long-term robot navigation by enabling robots to reuse spatial memories under illumination, seasonal, object-level, and dynamic-scene changes. Potential positive impacts include more reliable service, inspection, delivery, and assistive robots, especially in environments where frequent remapping is costly or impractical. The sparse topological representation may also reduce storage and computation relative to dense metric-semantic maps, which can make long-term semantic navigation more accessible on resource-constrained robotic platforms.

The same capabilities also introduce potential risks. A system that stores RGB-D observations for later semantic querying may capture people, private spaces, or sensitive objects, raising privacy concerns if deployed without consent, retention limits, or access control. More robust relocalization and navigation could also be misused for unauthorized surveillance, tracking, or physical intrusion. Finally, navigation failures caused by incorrect relocalization, stale observations, biased semantic detections, or outdated object memories could cause a robot to navigate to an unintended location or act on scene information that no longer reflects the current environment.

Several mitigation measures are appropriate for deployment. Stored visual memories should be minimized to task-relevant environments, protected with access controls, and deleted or anonymized when no longer needed. Deployments should obtain appropriate consent in private or semi-private spaces and should avoid using the system for identifying, tracking, or inferring sensitive attributes of people. Safety-critical applications should include conservative navigation policies, obstacle avoidance independent of the global memory, confidence thresholds for relocalization, and human override or monitoring when operating near people. The present work is an enabling navigation

Table 11: Compute resources required to reproduce the experiments. **Offline experiments** (OpenLORIS, Rover, Topo-Bench) were conducted on an RTX 4000 workstation (Ryzen 9 9900X, 60GB RAM). **Real-robot experiments** utilized a Unitree Go2 EDU robot equipped with a Realsense D435i camera, with the computations on an offboard HP Omen Transcend 14 laptop (Intel Core 7 155H, RTX 4050 6GB VRAM). Real-robot compute numbers also include the compute required to run the object search algorithm to find the target frame for object-goal navigation, which contributes to VRAM usage for the other baseline methods.

Method	Time / step	Total wall-clock time	Peak RAM/VRAM
OpenLORIS Corridor + Rover Campus (Total: 735 + 1232 query trials)			
CROSS	28 ms/frame	≈139 hours	≈35/2 GB
RTAB-Map	36 ms/frame	≈170 hours	≈110/0 GB
ORB-SLAM3	14 ms/frame	≈10 hours	≈6/0 GB
MASt3R-SLAM	74 ms/frame	≈52 hours	≈32/18 GB
ABM	17 ms/frame	≈85 hours	≈28/0 GB
GM / SM / PBU	19 ms/frame	≈90 hours	≈5/1 GB
Topo-Bench perceptual-aliasing (Total: 513 query trials)			
CROSS	28 ms/frame	≈20 hours	≈36/2 GB
RTAB-Map	36 ms/frame	≈26 hours	≈110/0 GB
ORB-SLAM3	14 ms/frame	≈2 hours	≈6/0 GB
ABM	17 ms/frame	≈12 hours	≈28/0 GB
OpenFabmap	130 ms/frame	≈90 hours	≈38/0 GB
Rat-Slam	85 ms/frame	≈61 hours	≈29/0 GB
GM / SM / PBU	19 ms/frame	≈14 hours	≈5/1 GB
Real-robot object navigation (3 settings × 10 trials)			
CROSS	120 ms/frame	≈1 hour	≈3.5/4.5 GB
Metric-Semantic (RTAB-MAP)	90 ms/frame	≈1 hour	≈1.3/4.5 GB
Metric-Semantic (ORBSLAM)	34 ms/frame	≈1 hour	≈2.1/4.5 GB

method and does not release a high-risk generative model or scraped dataset, but practical use should still follow local privacy, robotics safety, and data-governance requirements.

D Additional Method Details

D.1 Classical Gaussian–sum Filtering (GSF)

This appendix briefly summarizes the classical Gaussian–sum filter (GSF). We first present the exact linear–Gaussian case, then the standard nonlinear extension via per-component local Gaussianization (EKF/UKF/CKF). All derivations are written in a local Euclidean chart; on Lie groups (e.g., $\mathrm{SE}(3)$) the same updates are applied in a consistent tangent chart.

D.1.1 Preliminaries and Notation

Let $x_t \in \mathbb{R}^n$ be the (locally Euclidean) state with motion and measurement models

$$\begin{aligned} x_t &= f_t(x_{t-1}, u_{t-1}) + q_t, \qquad q_t \sim \mathcal{N}(0, Q_t), \\ z_t &= h_t(x_t) + r_t, \qquad\quad r_t \sim \mathcal{N}(0, R_t), \end{aligned} \tag{6}$$

where $Q_t, R_t \succ 0$. The GSF represents the filtering density as a finite mixture

$$p(x_{t-1} \mid z_{1:t-1}, u_{1:t-2}) = \sum_{k=1}^{K_{t-1}} w_{t-1}^{(k)} \, \mathcal{N}\big(x_{t-1}; \mu_{t-1}^{(k)}, \Sigma_{t-1}^{(k)}\big),$$

with $w_{t-1}^{(k)} \geq 0$ and $\sum_k w_{t-1}^{(k)} = 1$. On a Lie group $\mathcal{X}$, one works in a fixed local chart $\phi : \mathcal{X} \to \mathbb{R}^n$ (e.g., left-/right-invariant log), performs the Euclidean update on $\xi_t = \phi(x_t)$, and maps back via the exponential.

D.1.2 Exact Linear–Gaussian GSF

Assume linear–Gaussian models:

$$x_t = F_t x_{t-1} + B_t u_{t-1} + q_t, \qquad z_t = H_t x_t + r_t.$$

Prediction (per component). For each $k = 1, \dots, K_{t-1}$,

$$\begin{aligned} \mu_{t|t-1}^{(k)} &= F_t \mu_{t-1}^{(k)} + B_t u_{t-1}, \\ \Sigma_{t|t-1}^{(k)} &= F_t \Sigma_{t-1}^{(k)} F_t^\top + Q_t, \\ w_{t|t-1}^{(k)} &= w_{t-1}^{(k)}. \end{aligned} \tag{7}$$

Thus $p(x_t \mid z_{1:t-1}, u_{1:t-1}) = \sum_k w_{t|t-1}^{(k)} \mathcal{N}(x_t; \mu_{t|t-1}^{(k)}, \Sigma_{t|t-1}^{(k)})$.

Update (per component Kalman update). Define, for each component,

$$\begin{aligned} y_t^{(k)} &= z_t - H_t \mu_{t|t-1}^{(k)}, \\ S_t^{(k)} &= H_t \Sigma_{t|t-1}^{(k)} H_t^\top + R_t, \\ K_t^{(k)} &= \Sigma_{t|t-1}^{(k)} H_t^\top \big(S_t^{(k)}\big)^{-1}. \end{aligned} \tag{8}$$

Then

$$\begin{aligned} \mu_t^{(k)} &= \mu_{t|t-1}^{(k)} + K_t^{(k)}\, y_t^{(k)}, \\ \Sigma_t^{(k)} &= \big(I - K_t^{(k)} H_t\big) \Sigma_{t|t-1}^{(k)}. \end{aligned} \tag{9}$$

Weights update by the marginal likelihood (Kalman evidence):

$$\tilde{w}_t^{(k)} = w_{t|t-1}^{(k)}\, \mathcal{N}\big(y_t^{(k)}; 0, S_t^{(k)}\big), \qquad w_t^{(k)} = \frac{\tilde{w}_t^{(k)}}{\sum_j \tilde{w}_t^{(j)}}. \tag{10}$$

The posterior remains a mixture $p(x_t \mid z_{1:t}) = \sum_k w_t^{(k)} \mathcal{N}(x_t; \mu_t^{(k)}, \Sigma_t^{(k)})$.

D.1.3 Nonlinear GSF via Local Gaussianization

For nonlinear (6), GSF applies a local Gaussian filter to each component.

EKF-style (per component). Linearize around the current component mean:

$$\begin{aligned} f_t(x, u) &\approx f_t(\mu_{t-1}^{(k)}, u) + F_t^{(k)}(x - \mu_{t-1}^{(k)}), \\ h_t(x) &\approx h_t(\mu_{t|t-1}^{(k)}) + H_t^{(k)}(x - \mu_{t|t-1}^{(k)}), \end{aligned}$$

where $F_t^{(k)}, H_t^{(k)}$ are Jacobians. Then apply (7)–(10) with (F_t, H_t) replaced by $(F_t^{(k)}, H_t^{(k)})$ and with the corresponding affine terms.

UKF/CKF (per component). Alternatively, propagate sigma points/quadrature points per component to obtain $(\mu_{t|t-1}^{(k)}, \Sigma_{t|t-1}^{(k)})$ and predicted measurement moments, then use (8)–(10).

D.1.4 Mixture Identities

Let $\mathcal{N}_i(x) = \mathcal{N}(x; m_i, S_i)$ for $i \in \{1, 2\}$.

Product of Gaussians.

$$\mathcal{N}_1(x)\mathcal{N}_2(x) = \mathcal{N}(m_1; m_2, S_1{+}S_2)\, \mathcal{N}(x; m, S), \tag{11}$$

where $S = (S_1^{-1} + S_2^{-1})^{-1}$ and $m = S(S_1^{-1} m_1 + S_2^{-1} m_2)$.

Innovation evidence. For predicted (μ^-, Σ^-) and measurement $z = Hx + r$, $r \sim \mathcal{N}(0, R)$, the innovation $y = z - H\mu^-$ satisfies $y \sim \mathcal{N}(0, S)$ with $S = H\Sigma^- H^\top + R$, yielding the evidence term in (10).

D.1.5 Mixture Growth Control

To prevent unbounded mixture growth, GSF typically uses:

Pruning. Remove components with $w_t^{(k)} < \varepsilon$.

Reduction / merging. Iteratively merge nearby components (e.g., using a KL-based criterion) until $K_t \le K_{\max}$. Merging two components with weights a, b by moment matching gives

$$\mu = \frac{a\mu_a + b\mu_b}{a+b}, \tag{12}$$

$$\Sigma = \frac{a\big(\Sigma_a + (\mu_a - \mu)(\mu_a - \mu)^\top\big) + b\big(\Sigma_b + (\mu_b - \mu)(\mu_b - \mu)^\top\big)}{a+b}.$$

Splitting (optional). If a component violates a consistency/nonlinearity test, split it into a small set whose moments match the parent, and distribute the parent weight across the children.

D.1.6 Mixture–Mixture Update (Optional)

If the measurement factor is approximated by a mixture $Q_t(x) = \sum_{c=1}^{C_t} \pi_t^{(c)} \mathcal{N}(x; \nu_t^{(c)}, \Lambda_t^{(c)})$, then the update is a mixture–mixture product:

$$p(x_t \mid z_{1:t}) \propto p^-(x_t)\, Q_t(x_t),$$

with

$$p^-(x_t) = \sum_k w_{t|t-1}^{(k)} \mathcal{N}\big(x_t; \mu_{t|t-1}^{(k)}, \Sigma_{t|t-1}^{(k)}\big),$$

and

$$p(x_t \mid z_{1:t}) = \sum_{k=1}^{K_{t-1}} \sum_{c=1}^{C_t} \tilde{w}_{k,c}\, \mathcal{N}(x_t; m_{k,c}, S_{k,c}). \tag{13}$$

Here $(m_{k,c}, S_{k,c})$ follow (11) and

$$\tilde{w}_{k,c} \propto w_{t|t-1}^{(k)}\, \pi_t^{(c)}\, \mathcal{N}\Big(\mu_{t|t-1}^{(k)} - \nu_t^{(c)};\, 0,\, \Sigma_{t|t-1}^{(k)} + \Lambda_t^{(c)}\Big),$$

followed by normalization. In practice, gating and sparsification reduce the $K \times C_t$ expansion.

D.1.7 Manifold Adaptation (Lie Groups)

On a Lie group $\mathcal{X}$ (e.g., $\mathrm{SE}(3)$), represent each mixture component as a Gaussian in a consistent tangent chart $\phi(\cdot)$, apply the Euclidean GSF updates to $\xi_t = \phi(x_t)$, and reconstruct means via $\exp(\cdot)$. During prediction, covariances are transported through group composition using the appropriate adjoint (first-order), yielding the manifold GSF expressions used in the main text.

D.1.8 One-Step GSF Summary

Given $\{w_{t-1}^{(k)}, \mu_{t-1}^{(k)}, \Sigma_{t-1}^{(k)}\}_{k=1}^{K_{t-1}}$:

1. **Predict:** propagate each component (linear (7), or EKF/UKF/CKF per component).
2. **Update:** apply (8)–(10) (or (13) for mixture likelihoods).
3. **Control:** prune/reduce (and optionally split) to enforce $K_t \le K_{\max}$.

On manifolds, perform all steps in the chosen chart and transport covariances via the adjoint during prediction.

D.2 Motion Message

We derive the GSF prediction for a single Gaussian mixand under a right-invariant $\mathrm{SE}(3)$ motion model, working in the right-invariant tangent chart $\varepsilon = \log(\mu^{-1}X) \in \mathfrak{se}(3)$.

D.2.1 Setup

Let $X_{t-1} \in \mathrm{SE}(3)$ be distributed as $X_{t-1} = \mu \exp(\varepsilon)$ with $\varepsilon \sim \mathcal{N}(0, \Sigma) \subset \mathfrak{se}(3)$. The (right-invariant) stochastic motion model is

$$X_t = X_{t-1}\,\Delta T_t\,\exp(\nu_t), \qquad \nu_t \sim \mathcal{N}(0, Q_t) \subset \mathfrak{se}(3),$$

with ν_t independent of ε.

D.2.2 Prediction Kernel

For a single mixand, the predicted density is

$$\bar{p}(x_t) = \int p(x_t \mid x_{t-1})\,\mathcal{N}_{\mathfrak{se}(3)}\Big(\log(\mu^{-1}x_{t-1});\,0,\Sigma\Big)\,dx_{t-1}.$$

D.2.3 First-Order Pushforward

Write $X_{t-1} = \mu\exp(\varepsilon)$. Using the group adjoint and BCH,

$$\mu\exp(\varepsilon)\,\Delta T_t = \mu\Delta T_t\;\exp\Big(\mathrm{Ad}_{\Delta T_t^{-1}}\varepsilon + \mathcal{O}(\|\varepsilon\|^2)\Big).$$

Post-multiplying by $\exp(\nu_t)$ and applying BCH again yields

$$\mu\Delta T_t\;\exp\big(\mathrm{Ad}_{\Delta T_t^{-1}}\varepsilon\big)\;\exp(\nu_t) = \mu\Delta T_t\;\exp\Big(\mathrm{Ad}_{\Delta T_t^{-1}}\varepsilon + \nu_t + \mathcal{O}(\|\varepsilon\|^2 + \|\nu_t\|^2)\Big).$$

Neglecting higher-order terms, the updated error in the right-invariant chart at the predicted mean $\mu^- = \mu\Delta T_t$ is

$$\varepsilon^- \;\approx\; \mathrm{Ad}_{\Delta T_t^{-1}}\varepsilon \;+\; \nu_t,$$

and therefore

$$\varepsilon^- \sim \mathcal{N}\Big(0,\;\mathrm{Ad}_{\Delta T_t^{-1}}\Sigma\mathrm{Ad}_{\Delta T_t^{-1}}^{\top} + Q_t\Big).$$

Equivalently, the pushforward of a single Gaussian mixand is (to first order)

$$\bar{p}(x_t) \approx \mathcal{N}_{\mathfrak{se}(3)}\Big(\log\big((\mu\Delta T_t)^{-1}x_t\big);\;0,\;\Sigma^-\Big),$$

with

$$\Sigma^- = \mathrm{Ad}_{\Delta T_t^{-1}}\Sigma\mathrm{Ad}_{\Delta T_t^{-1}}^{\top} + Q_t.$$

D.2.4 Mixtures and Weights

Since $\int p(x_t \mid x_{t-1})\,dx_t = 1$, prediction preserves mixture weights: if $p(x) = \sum_k w_k p_k(x)$ then $\bar{p}(x) = \sum_k w_k \bar{p}_k(x)$.

D.2.5 Small-Increment Approximation

If $\Delta T_t = \exp(\xi_t)$ with $\|\xi_t\| \ll 1$, then

$$\mathrm{Ad}_{\Delta T_t^{-1}} = I - \mathrm{ad}(\xi_t) + \mathcal{O}(\|\xi_t\|^2),$$

and the transported covariance expands as

$$\mathrm{Ad}_{\Delta T_t^{-1}}\Sigma\mathrm{Ad}_{\Delta T_t^{-1}}^{\top} = \Sigma - \mathrm{ad}(\xi_t)\Sigma - \Sigma\,\mathrm{ad}(\xi_t)^{\top} + \mathcal{O}(\|\xi_t\|^2\|\Sigma\|).$$

At high update rates (small $\|\xi_t\|$) and when covariances are maintained in the updated right-invariant chart, a common conservative approximation is

$$\Sigma^- \approx \Sigma + Q_t,$$

which we use in our implementation in practice.

D.3 Hypothesis Management

In the following, we provide additional details on our hypothesis management strategy. In particular, we detail how we cluster measurements and then fuse/prune/birth hypotheses.

Measurement Clustering. The global measurement message (5) contains $N_t K$ Gaussian components. Many of these differ only by small perturbations around the same physical location, induced by feature noise and PnP variability. We therefore first reduce this redundancy by clustering the component means on $\mathrm{SE}(3)$.

SE(3)-aware DBSCAN. Let μ_a and μ_b denote two component means on $\mathrm{SE}(3)$. We define their Lie-algebra displacement

$$\delta_{ab} = \log\!\big(\mu_a^{-1}\mu_b\big) \in \mathfrak{se}(3) \simeq \mathbb{R}^6,$$

and a group-aware distance

$$d_{\text{dbscan}}(\mu_a, \mu_b) = \| \, W \, \delta_{ab} \, \|_2 \, ,$$

where W is a diagonal weighting matrix that balances translation (meters) and rotation (radians). We run DBSCAN using this distance metric, obtaining clusters $\mathcal{C}_c$ of component indices j.

Cluster mean and covariance. Each measurement component j is parameterized by a mean μ_j and covariance Σ_j in its own tangent chart. After forming a cluster $\mathcal{C}_c$, we compute the Riemannian (Fréchet) mean μ_c on $\mathrm{SE}(3)$ as

$$\mu_c = \arg\min_{T \in \mathrm{SE}(3)} \sum_{j \in \mathcal{C}_c} \alpha_j \left\| \log\!\big(T^{-1}\mu_j\big) \right\|^2 ,$$

where α_j are the mixture weights of the original measurement components in Eq. (5).

To aggregate uncertainty in a common tangent space, we transport each component covariance to the tangent at μ_c:

$$\Sigma_j' = \mathrm{Ad}_{\mu_c^{-1}\mu_j} \, \Sigma_j \, \mathrm{Ad}_{\mu_c^{-1}\mu_j}^{\top},$$

and define

$$\xi_j = \log\!\big(\mu_c^{-1}\mu_j\big) \in \mathfrak{se}(3).$$

The cluster covariance must capture both (i) the *within-component* uncertainty and (ii) the *between-components* spread of means. The total covariance is therefore

$$\Sigma_c = \frac{1}{\beta_c} \sum_{j \in \mathcal{C}_c} \alpha_j \big(\Sigma_j' + \xi_j \xi_j^{\top}\big), \qquad \beta_c = \sum_{j \in \mathcal{C}_c} \alpha_j .$$

The merged measurement message can then be written as

$$m_t^{\text{meas}}(x) \;\approx\; \sum_{c=1}^{C_t} \bar{w}_c \, \mathcal{N}_{\mathfrak{se}(3)}\big(\log\!\big(\mu_c^{-1}x\big);\; 0, \Sigma_c\big) \, , \tag{14}$$

where $C_t \ll N_t K$ and $\bar{w}_c = \beta_c / \sum_{c'} \beta_{c'}$. In practice, we retain only the top $K{=}5$ highest-weight components whose normalized weight exceeds a small threshold (e.g., 10^{-3}), and discard the remainder to maintain computational efficiency.

Fusion and Pruning. We next compute the product of the motion message m_t^{mot} and the clustered measurement message (14). Each existing hypothesis (a mixand of m_t^{mot}) is fused with at most one measurement cluster, since only clusters within the reachable region implied by the motion model have non-negligible overlap.

Let $(\mu_{\text{mot}}^{(i)}, \Sigma_{\text{mot}}^{(i)}, w_{\text{mot}}^{(i)})$ be a motion component and $(\mu_c, \Sigma_c, \bar{w}_c)$ a measurement cluster. Define the relative displacement $\delta_{ic} = \log\!\big((\mu_{\text{mot}}^{(i)})^{-1}\mu_c\big) \in \mathfrak{se}(3)$, $\Sigma_c' = \mathrm{Ad}_{\Delta_{ic}^{-1}} \Sigma_c \, \mathrm{Ad}_{\Delta_{ic}^{-1}}^{\top}$, where $\Delta_{ic} = (\mu_{\text{mot}}^{(i)})^{-1}\mu_c$, so that Σ_c' is expressed in the tangent at $\mu_{\text{mot}}^{(i)}$.

The fused covariance and mean (in the tangent at $\mu_{\text{mot}}^{(i)}$) are

$$\Sigma_{i,c}^{-1} = \big(\Sigma_{\text{mot}}^{(i)}\big)^{-1} + \big(\Sigma_c'\big)^{-1}, \tag{15}$$

$$\mu_{i,c} = \mu_{\text{mot}}^{(i)} \exp\Big(\Sigma_{i,c} \left(\Sigma_c'\right)^{-1} \delta_{ic}\Big). \tag{16}$$

The fused weight is proportional to the product weight times the overlap between the two Gaussians,

$$w_{i,c} \;\propto\; w_{\text{mot}}^{(i)}\, \bar{w}_c\, \mathcal{N}\Big(\delta_{ic};\, 0,\, \Sigma_{\text{mot}}^{(i)} + \Sigma_c'\Big). \tag{17}$$

We prune fused components whose weights $w_{i,c}$ fall below a small threshold (e.g. 10^{-3}), indicating that the motion and measurement components are too far apart to represent a physically consistent location.

Birth of new hypotheses. Some measurement clusters may have negligible overlap with all motion components (e.g., loop closure or kidnapped robot). We model this via a "restart" switch $b_t \in \{0, 1\}$ with prior $p(b_t{=}1) = \epsilon$:

$$p(x_t \mid x_{t-1}, u_t, b_t) = \begin{cases} p_{\text{mot}}(x_t \mid x_{t-1}, u_t), & b_t = 0, \\ p_{\text{birth}}(x_t \mid B_t), & b_t = 1, \end{cases}$$

where $B_t \in \mathcal{C}_t^{\text{new}}$ indexes the measurement cluster that initializes the new hypothesis. The birth distribution is a single Gaussian

$$p_{\text{birth}}(x_t \mid B_t{=}c) = \mathcal{N}_{\mathfrak{se}(3)}\big(\log(\mu_c^{-1} x_t);\, 0, \Sigma_c\big),$$

with $p(B_t{=}c \mid b_t{=}1) \propto \bar{w}_c$.

Marginalizing b_t yields a mixture transition kernel

$$\tilde{p}(x_t \mid x_{t-1}, u_t) = (1 - \epsilon)\, p_{\text{mot}}(x_t \mid x_{t-1}, u_t) + \epsilon\, p_{\text{birth}}(x_t). \tag{18}$$

Hypothesis capacity. We use the same hypothesis capacity, $K{=}5$, for every dataset and every experiment (OpenLORIS, Rover, Topo-Bench, and all robot studies), with no dataset-specific tuning. VPR/PnP proposals are ranked by probability and support and clustered into distinct pose modes, and the filter tracks at most five pose branches. Capacity affects the result only if the correct mode ranks below the top five. After retrieval, geometric verification, and continuous-pose clustering, we expect more than five simultaneously persistent plausible aliases to be rare in real-world environments, so K functions as a conservative compute bound rather than a dataset-tuned operating point. The current system does not infer K online. Consequently, in a highly adversarial environment containing more than five persistent, geometrically valid aliases, the correct branch could be pruned. A natural adaptive extension would increase the capacity, subject to a computational budget, when the discarded proposal mass remains high or slot replacements occur repeatedly; we leave adaptive capacity selection to future work.

D.4 Semantic Queries and Navigation

CROSS separates semantic goal selection from spatial-memory reuse. Each map node stores its raw RGB-D observation together with its pose distribution. At query time, a replaceable detector or VLM scores these stored observations for a text or object query and returns one or more target keyframes. CROSS then (i) relocalizes the live robot in the same memory and (ii) returns a pose-aware topological route to the selected target keyframe. The implementation therefore exposes semantic-memory search and graph planning as separate interfaces. The robot study uses an object query over stored observations; CROSS itself does not introduce a new language parser, object detector, or relational-reasoning model. Its role is to keep the spatial substrate reusable after appearance and scene change, so that a downstream query model can still access and navigate to the relevant observation. Storing raw RGB-D observations preserves visual information for future semantic queries, instead of committing to a fixed semantic vocabulary, and allows the semantic component to be replaced independently of the spatial memory.

For navigation, CROSS uses the sparse pose-aware graph as its persistent global representation and does not require a persistent occupancy grid. Consecutive nodes are linked by odometry edges corresponding to past traversals, and spatially nearby nodes are linked by short proximity edges ($d{=}0.5\,\text{m}$ in our implementation). Once semantic search selects a target keyframe, A* or Dijkstra search returns a sequence of keyframe waypoints from the robot's current relocalized node to that

target. A shared local controller then tracks the relative waypoints and performs short-horizon obstacle avoidance from the current RGB-D sensing. Such a controller may internally use temporary local obstacle information, but this is distinct from the persistent memory of CROSS and is held fixed across the compared systems. Table 12 summarizes this division of labor: CROSS does not address general motion planning, and it does not contribute the low-level controller.

Table 12: Division of labor between the CROSS memory, graph search, and the shared local controller.

Component	Role
CROSS memory	Persistent RGB-D keyframes, poses, traversed and proximity edges
Graph search	Global waypoint sequence to the selected keyframe
Shared local controller	Relative-waypoint tracking and local collision avoidance

E Relation to Topometric Filtering and Route-Consistency Methods

We group the closest prior work into three lines.

Sequence and route consistency. SeqSLAM [28] aligns query and reference image sequences, and lazy data association [43] searches an online data-association graph under seasonal change. Both exploit order and velocity consistency along a reference sequence, without a relative-pose measurement for each retrieved image.

Route-constrained metric and multi-experience systems. CAT-SLAM [25] is a close precursor: its Rao–Blackwellized particle filter combines sequential appearance with local metric motion along a prior continuous trajectory and models new places. Visual teach and repeat [12] repeats a taught route using linked stereo submaps, and experience-based navigation [8] maintains several route experiences across appearance conditions. These methods use metric information and handle long-term change, but remain tied to taught trajectories or stored experiences.

Probabilistic topometric and navigation filters. New-place-aware topometric localization [49] adds full 3-DoF odometry and an off-map state, and PlaceNav [40] uses VPR with discrete Bayesian filtering for stable subgoal selection.

The distinction of CROSS is the pre-commitment interface for an online-growing graph. CROSS converts each RGB-D retrieval into a measured relative pose and then into a global $\mathrm{SE}(3)$ Gaussian mode; competing trajectories evolve with metric odometry outside the graph, and only persistent support creates a loop-closure or relocalization factor. Unlike sequence and route filters, it is not limited to route-order consistency; unlike fixed-map localizers, it explicitly withholds ambiguous associations to protect map modification. Off-map states, route priors, and multi-experience memory are complementary to CROSS.

F Existing Assets, Licenses, and Terms of Use

This appendix documents the external assets used in the experiments and implementation, addressing the NeurIPS checklist item on licenses for existing assets. We include the datasets and benchmark assets, baseline systems, pretrained model/code assets, and reimplemented baselines used in Sections 5 and B.2. The license information reflects the public project pages and repositories checked on May 5, 2026. We exclude papers cited only as related work. All listed assets are credited through the citations in the main paper or appendix. We used the assets for academic research evaluation, did not redistribute third-party datasets, model checkpoints, or baseline code as part of this paper, and respected non-commercial, share-alike, and copyleft terms by using affected assets only as external evaluation baselines or ablations.

Table 13: External datasets and benchmark assets used in the paper. "Terms recorded" summarizes the license or official terms of use that we found from the asset's public distribution page.

Asset	Use in this paper	Terms recorded	Compliance note
OpenLORIS-Scene, Corridor split [38]	Indoor relocalization benchmark	Official page describes OpenLORIS as an open dataset and requests citation; we did not find a separate named data license on the official page. The accompanying OpenLORIS-Scene tools repository is under the MIT license.	Used only for benchmark evaluation. We cite the dataset paper and do not redistribute the dataset. Project page: OpenLORIS-Scene; tools: openloris-scene-tools.
ROVER, Campus scene [37]	Outdoor multi-season relocalization benchmark	MIT license on the public dataset distribution page.	Used only for benchmark evaluation. We cite the dataset paper and do not redistribute the dataset. Project page: ROVER; dataset page: Hugging Face.
Topo-Bench [46]	Perceptual-aliasing benchmark in Appendix B.2	Public benchmark paper states that datasets, baselines, and evaluation tools are open-sourced. The public paper distribution is under CC BY-NC-SA 4.0; any repository-specific license should govern repository files if different.	Used only for research evaluation. We cite the benchmark paper, do not redistribute benchmark assets, and follow the benchmark's non-commercial/share-alike terms where applicable.
Real-robot data collected for this work	Object-goal navigation, the office experiments in Appendix B, and qualitative deployment experiments	New data collected by the authors; not an external asset and not released with this paper.	No third-party data license applies. Privacy and deployment considerations are discussed in Appendix C.

G Additional Figures and Plots

Table 14: External baseline systems, model/code assets, and their license terms.

Asset	Use in this paper	License / terms recorded	Compliance note
ORB-SLAM3 [5]	SLAM baseline; also used inside the metric+semantic baseline	GPL-3.0.	Used as a separate external baseline for academic evaluation; we do not incorporate or redistribute ORB-SLAM3 code inside CROSS. Repository: ORB-SLAM3.
RTAB-Map [19]	SLAM/topological baseline; also used inside the metric+semantic baseline	BSD license when OpenCV is built without nonfree modules; research-only if built with OpenCV nonfree/SURF.	Used as a separate evaluation baseline. We use the permissive/default configuration where applicable and avoid redistributing RTAB-Map binaries or databases. Project page: RTAB-Map.
MASt3R-SLAM [30]	SLAM baseline	Creative Commons Attribution-NonCommercial-ShareAlike 4.0 (CC BY-NC-SA 4.0).	Used only as a non-commercial research evaluation baseline; not redistributed as part of CROSS. Repository: MASt3R-SLAM.
OpenFABMAP / FAB-MAP [9]	Classical baseline in the Topo-Bench comparison	GNU GPL v3 or later in the public openFABMAP repository.	Used only as an external benchmark baseline. We do not incorporate its code into CROSS or redistribute modified versions. Repository: openFABMAP.
RatSLAM / OpenRatSLAM [27]	Classical baseline in the Topo-Bench comparison	GPL-3.0 in the public RatSLAM/OpenRatSLAM repository.	Used only as an external benchmark baseline. We do not incorporate its code into CROSS or redistribute modified versions. Repository: ratslam.
BoQ / Bag-of-Queries [1]	Visual place recognition backend used by CROSS and reimplemented topological baselines	MIT license.	Used as an off-the-shelf VPR model/code asset with attribution. Repository: Bag-of-Queries.
XFeat [32]	Local feature extraction for relative pose estimation	Apache-2.0 license.	Used as an off-the-shelf feature extractor with attribution. Repository: accelerated_features.
LightGlue [23]	Local feature matching for relative pose estimation	Apache-2.0 license for LightGlue code and pretrained weights; the repository notes that SuperPoint has a different restrictive license.	Used with attribution as a matching module. We do not rely on SuperPoint-specific weights unless separately licensed. Repository: LightGlue.
VGGT [47]	Learned relative-pose-estimation ablation	Repository license permits commercial use for the code after the July 2025 license update, but only the VGGT-1B-Commercial checkpoint is licensed for commercial usage; the original checkpoint remains non-commercial.	Used only for a research ablation and not in the final system. We do not redistribute checkpoints. Repository: VGGT.

Table 15: Baselines that we reimplemented rather than importing as third-party code assets.

Baseline	Asset/license status	Compliance note
Greedy Matching (GM)	In-house implementation; no third-party code asset imported.	Based on the retrieval-based localization protocol described in Appendix A.1; the common VPR backend is BoQ, whose MIT license is listed in Table 14.
Sequence Matching (SM) [36, 26]	In-house implementation; no third-party code asset imported.	Original sequence/topological-localization papers are cited. The implementation shares only the experimental protocol and VPR backend described in Appendix A.1.
Probabilistic Belief Update (PBU) [40]	In-house implementation; no third-party code asset imported.	Original probabilistic/topometric-localization paper is cited. The implementation shares only the experimental protocol and VPR backend described in Appendix A.1.
Appearance-Based Mapping (ABM) [18]	Implemented in the RTAB-Map [19] package.	Original paper is cited.

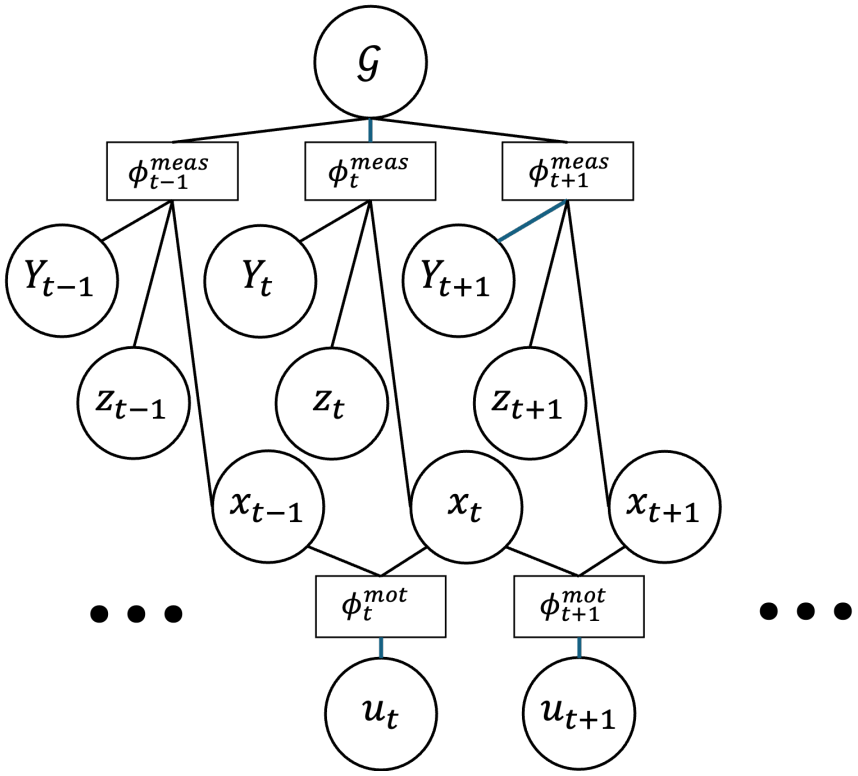


Figure 10: Factor-graph representation of our Gaussian mixture filtering model. ϕ_t^{mot} and ϕ_t^{meas} are the motion and measurement factors. x_t is the current pose, u_t the odometry input, and z_t the RGB-D observation. Y_t is a latent association variable identifying which keyframe explains z_t, and $\mathcal{G}$ is the set of stored keyframes.

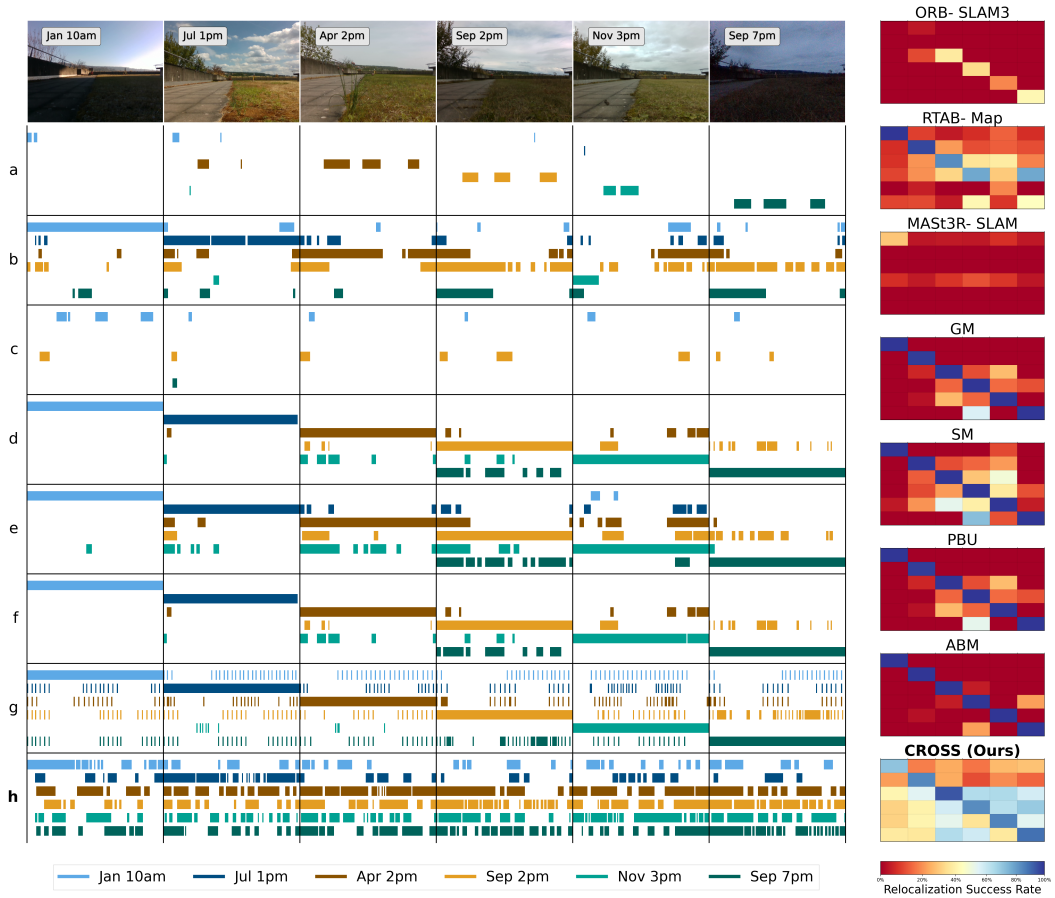


Figure 11: Multi-session relocalization results on the Rover [37] Campus scene. **Left:** Relocalization outcomes across different locations. Each row corresponds to a mapping trajectory (indicated by different colors), while columns show relocalization attempts at the same physical locations captured at different times or months, as illustrated in the top image. Empty space indicates relocalization failed at that specific location. The compared methods are: (a) ORB-SLAM3 [5], (b) RTAB-Map [19], (c) MASt3R-SLAM [30], (d) Greedy Matching (GM), (e) Sequence Matching (SM), (f) Probabilistic Belief Update (PBU), (g) ABM [18], and (h) Ours. Most baseline methods struggle under significant lighting and appearance changes, whereas our approach consistently relocalizes despite substantial visual variation. **Right:** A compact summary view, where each grid cell reports the relocalization success rate for a given mapping–testing sequence pair. Additional analysis is provided in Section 5.1.



Figure 12: Multi-session relocalization results on the OpenLORIS [38] Corridor scene. **Left:** Relocalization outcomes across different locations. Each row corresponds to a mapping trajectory (indicated by different colors), while columns show relocalization attempts at the same physical locations captured at different times or months, as illustrated in the top image. Empty space indicates relocalization failed at that specific location. Note that the last two sequences (20:54 and 20:56) are shaded gray to indicate a lack of spatial overlap. The compared methods are: (a) ORB-SLAM3 [5], (b) RTAB-Map [19], (c) MASt3R-SLAM [30], (d) Greedy Matching (GM), (e) Sequence Matching (SM), (f) Probabilistic Belief Update (PBU), (g) ABM [18], and (h) Ours. Most baseline methods struggle under significant lighting and appearance changes, whereas our approach consistently relocalizes despite substantial visual variation. **Right:** A compact summary view, where each grid cell reports the relocalization success rate for a given mapping–testing sequence pair. Additional analysis is provided in Section 5.1.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions: a pose-aware RGB-D topological spatial memory, bounded Gaussian-mixture filtering on $\mathrm{SE}(3)$, and evaluation on public long-term relocalization benchmarks and real-robot object-goal navigation. These claims are supported by the method and experiments in Sections 4.1, 4.2, and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in Section 6, including reliance on reliable relative-pose estimates, failure modes under extreme appearance shift, low overlap, motion blur, or noisy depth, and the limited semantic scope of the evaluation. Runtime and scalability are further discussed in Appendix B.9.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions.
   - The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made.
   - The authors should reflect on the factors that influence the performance of the approach.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not claim formal theoretical results or theorems. Mathematical derivations and approximations for Gaussian-sum filtering, motion propagation, and hypothesis management are provided as method details in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material.
- Any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper, regardless of whether the code and data are provided or not?

   Answer: [Yes]

   Justification: The paper specifies the datasets, map–query protocol, subsequence length, success thresholds, trial counts, baselines, topological baseline definitions, real-robot setup, and implementation details in Section 5 and Appendices A.1–A.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Reproducibility can be accomplished in various ways depending on the contribution.
   - Releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions.
   - NeurIPS does not require releasing code, but submissions should provide a reasonable avenue for reproducibility.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is released at `https://github.com/jiaming-ai/CROSS` (linked from the project page `https://jiaming.im/CROSS/`) with installation instructions and a getting-started guide on public data. OpenLORIS, Rover, and Topo-Bench are public datasets, and the protocols are described in Section 5 and Appendix A.1. The real-robot recordings are not released.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer.
   - Papers cannot be rejected simply for not including code, unless this is central to the contribution.
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - The authors should provide instructions on data access and preparation.
   - The authors should provide scripts to reproduce all experimental results for the proposed method and baselines.
   - At submission time, to preserve anonymity, the authors should release anonymized versions if applicable.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies the public datasets, map–query evaluation protocol, relocalization thresholds, number of trials, baseline methods, real-robot settings, and success criteria in Section 5. Additional baseline and implementation details are provided in Appendices A.1 and A.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Appendix B.1 reports two-sided 95% Wilson score confidence intervals for the main binary success-rate experiments, including the trial counts, interval formula, assumptions, and interpretation for both benchmark relocalization and real-robot object-navigation results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated.
- The method for calculating the error bars should be explained.
- The assumptions made should be given.
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- For asymmetric distributions, the authors should be careful not to show symmetric error bars that would yield results that are out of range.
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix B.10 reports the hardware and software environment, local workstation specifications, per-frame runtime, per-run compute, total reported CPU/GPU-hours, storage/RAM requirements, and additional exploratory compute for the reported experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers, CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each individual experimental run as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: The work evaluates robot localization and navigation using public datasets and controlled robot experiments, and does not report crowdsourcing, human-subject studies, scraped personal datasets, or high-risk model releases. The authors have reviewed and complied with the NeurIPS Code of Ethics, including applicable privacy and safety considerations for real-world robot data collection.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix C discusses positive impacts for long-term robot navigation, privacy and misuse risks from stored RGB-D memories and robust relocalization, possible harms from incorrect navigation, and deployment mitigations.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses, fairness considerations, privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications.
    - The authors should consider harms that could arise when the technology is used as intended and functioning correctly, harms from incorrect results, and harms from misuse.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

    Answer: [N/A]

    Justification: The paper does not release a high-risk pretrained language model, image generator, scraped dataset, or similarly high-risk asset.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards.
    - Datasets that have been scraped from the Internet could pose safety risks.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets used in the paper properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Appendix F explicitly documents the external datasets, benchmarks, baseline systems, pretrained model/code assets, and reimplemented baselines used in the paper. It records the license or terms of use found for each asset, provides URLs where applicable, credits the original creators through citations, and states how the reported usage respects redistribution, non-commercial, share-alike, and copyleft terms.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license should be included for each asset.
- For scraped data from a particular source, the copyright and terms of service should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new asset is the CROSS code (`https://github.com/jiaming-ai/CROSS`), released under the MIT license with installation instructions, usage documentation, and the licenses of its third-party components. No new dataset is released.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets if applicable.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or recruited human-subject studies. Moving pedestrians appear only as incidental dynamic agents in qualitative real-robot experiments, with no participant instructions, interventions, or compensation.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether IRB approvals or equivalent review were obtained?

Answer: [N/A]

Justification: The paper does not report crowdsourcing or human-subject experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval or equivalent may be required for human-subjects research.
- We expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

Answer: [N/A]

Justification: LLMs are not used as an important, original, or non-standard component of the core method.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to the NeurIPS LLM policy in the handbook for what should or should not be described.