A-SPAM: A Novel Asynchronous Semantic Padding-and-Matching Integrated Framework for Dynamic Loop Closure Detection

Qibin He, *Student Member, IEEE*, Yapeng Wang*, *Member, IEEE*, Yanming Chai, Qiyue Huang, Tiankui Zhang, *Senior Member, IEEE*, Sio-Kei Im, *Senior Member, IEEE*, Jie Zhang, *Member, IEEE*

***Abstract*—Loop closure detection in dynamic SLAM faces critical challenges when dynamic objects dominate camera views, degrading frame-to-frame methods reliant on static landmarks. We propose A-SPAM, an asynchronous framework that constructs spatiotemporal semantic graphs via semantic padding (entity tracking + rigid structure analysis) and validates loops via semantic matching (topology-feature hybrid correlation). Evaluated on TUM and BONN datasets, A-SPAM achieves at least 76.8% recall rate at 100% precision in dynamic environments, while maintaining a mean translational error of less than 0.07m across dynamic sequences under degraded odometry conditions. The proposed framework corrects erroneous trajectories and enhances robustness against odometry failures in dynamic environments.**

***Index Terms*—Dynamic SLAM, Loop detection, Asynchronous event, Scene reconstruction, Semantic graph, Semantic matching**

I. INTRODUCTION

SIMULTANEOUS Localization and Mapping (SLAM) systems have advanced significantly in capability and scope over the past decade. SLAM operates autonomously across environments, unlike infrastructure-dependent technologies (e.g., GPS, UWB [1]). Visual SLAM, widely adopted for robotic applications due to its cost-effectiveness, estimates poses relatively, leading to accumulated drift that grows unbounded without loop closures [2]. Robust loop closure detection is thus critical-without it, localization drifts irreversibly, compromising map consistency.

Loop closure detection in Visual SLAM is crucial for reducing accumulated errors by recognizing revisited locations. Upon revisiting a mapped area, the system must recognize the location and compute pose corrections to mitigate drift. While appearance-based approaches like Bag of Words (BoW) [3] efficiently detect loops in static scenes with overlapping features, they struggle in dynamic environments. Modern robotics increasingly operates in populated settings, where moving objects dominate the camera view, demanding robust dynamic loop closure solutions.

Loop closure detection in Visual SLAM demands extremely high precision, as false positives can cause catastrophic map corruption, while missed loops allow error accumulation [7]. Dynamic features exacerbate this challenge: inconsistent appearances of moving objects between visits render them unreliable for matching. Vision systems, unlike Lidar-based solutions, cannot have a wide perceptual domain (see Fig. 1). Minor camera orientation differences between loops can drastically alter perspectives, while dynamic objects further occlude static co-visible regions. These factors compound, leading to extreme scarcity of reliable overlapping features when camera viewpoints diverge and dynamics dominate the scene.

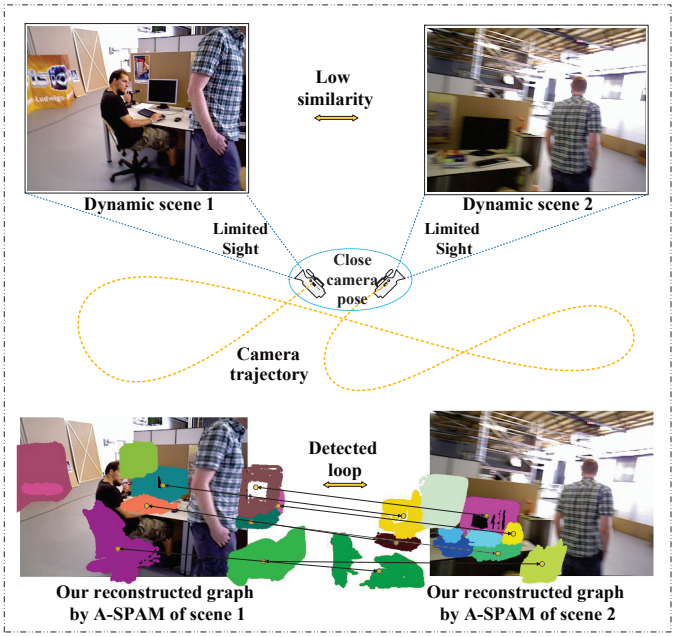

Fig. 1: Our loop detection by A-SPAM in a challenging situation for loop closure detection in previous studies [4]–[6]. Dynamic objects and limited camera sight reduce overlapping areas, feature matches, and image similarity, but our method overcomes these challenges.

In this work, we address the above challenges by leveraging static semantic entities and their relations for dynamic scene invariance. Using multi-object tracking, we extract entity trajectories and build a temporal semantic graph. For revisited locations, loop detection is achieved via semantic graph rigid structure and entity similarity from a Siamese Network [8].

Thus, we propose A-SPAM (Fig. 2), a novel dynamic loop closure detection framework using the above conceptions. The contributions of this article are as follows:

- A new dynamic loop closure detection framework named A-SPAM extracts static information from a temporal frame sequence and provides robust and sufficient loop closure detections in dynamic scenes.
- A novel padding algorithm that combines rigid structure constraints with entity trajectories to maintain topological consistency across dynamic occlusions.
- An optimized semantic matching approach that integrates graph topology and feature similarity metrics for robust loop validation.

The remainder of this paper is organized as follows: Section II reviews related work on dynamic SLAM and loop closure detection. Section III presents the theoretical foundations of our approach. Section IV details the implementation of the A-SPAM framework. Section V presents experimental results

Manuscript received: October 4, 2025 ; Accepted October 30, 2025.

This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers'comments. This work was funded by Macao Polytechnic University under the research project RP/FCA-02/2025

Qibin He, Yapeng Wang, Yanming Chai, Qiyue Huang, Sio-Kei Im are with Faculty of Applied Sciences, Macao Polytechnic University, Macao SAR, Tiankui Zhang is with School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, China

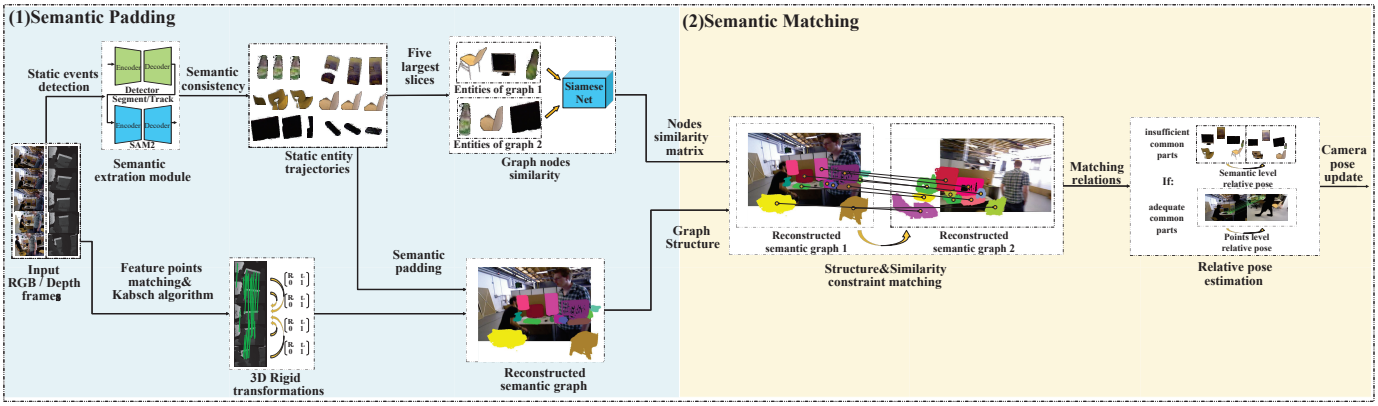

Fig. 2: Overview of the A-SPAM Framework: It comprises two key stages: (1). Semantic Padding: Sequential RGB frames are processed using GroundingDino [9] for semantic events detection and SAM2 [10] to extract and track static semantic entities for cross-frame association. Rigid transformations are computed from matched feature points and depth data to reconstruct the semantic graph. (2). Semantic Matching: Entity trajectories are analyzed with Siamese Network [8] to compute similarity metrics, validate loop closures, and optimize camera poses, ensuring trajectory consistency. The novelty lies in combining semantic padding and semantic matching for robust loop detection and trajectory optimization.

and comparative analysis. Section VI concludes the paper and discusses future work.

II. RELATED WORK

Traditional loop detection methods like Bag-of-Words (BoW) [3] [11] rely on static features but struggle with rotations and dynamic environments due to the limitations of a single frame [2] [12]. Sequence-based approaches address this by leveraging temporal consistency and enhancing robustness and reliability. This highlights the need for A-SPAM's hybrid approach, which integrates both strategies through the scene graph.

A. *Frame-based Dynamic Loop Detection*

Some approaches like DS-SLAM [5] and DynaSLAM [13] pioneered dynamic object handling by leveraging semantic segmentation to identify and exclude moving objects, ensuring static map reconstruction. DynaSLAM further improved robustness through image inpainting to recover occluded static features for loop closure. While effective, these methods rely on traditional Bag-of-Words (BoW) [3] for loop detection, limiting adaptability.

More advanced learning-based advances, such as PatchNetVLAD [14] (local-global descriptor fusion), have enhanced loop detection. SemanticLoop [6] further improved robustness by focusing on static semantic features. However, these frame-level methods fail when dynamic objects dominate, as sparse static features compromise loop closure reliability. Our approach overcomes this limitation by using different strategies.

B. *Sequence-based Dynamic Loop Detection*

Sequence-based approaches provide robust solutions for environmental variations in SLAM [15]. The seminal SeqSLAM [16] introduced sequence matching through image similarity matrices, demonstrating strong performance under appearance changes. Subsequent improvements include enhanced feature extraction and depth-augmented matching [17].

While sequence-based methods effectively handle appearance variations, they typically ignore valuable inter-frame relationships. This limitation is especially problematic in dynamic environments. Our approach overcomes this by tracking semantic entities across sequences, enabling robust loop detection despite sparse static features.

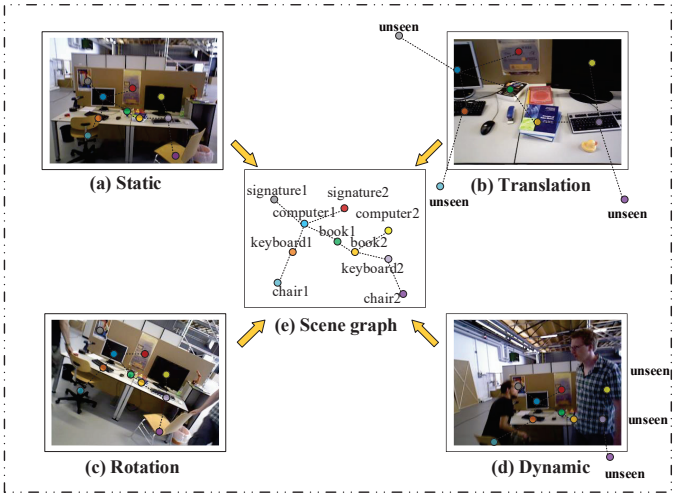

Fig. 3: The semantic scene graph (e) provides an invariant representation across different scenarios (a-d). It maintains alignment with world coordinates during rotations/translations and enables the prediction of occluded entities' positions through their relationships with visible static points.

C. *Scene Graph-based Dynamic Loop Detection*

A scene graph provides a structured representation of objects, their attributes, and inter-object relationships within a scene [18]. While recent works have advanced scene graph generation [19], dynamic environments require more than single-image analysis.

Compared to traditional approaches, scene graphs demonstrate superior capability for scene correspondence [20], [21] under significant viewpoint changes. Building on this strength, our framework bridges the gap between sequence-based and frame-level representations.

III. PROBLEM ANALYSIS

Dynamic scenes exhibit unique properties versus static ones [4], [5], [13]. A-SPAM analyzes: (1) rigid structures for static relationships and (2) asynchronous events handling dynamic occlusions, enabling robust loop detection.

A. *Rigid Scenario Structure*

A scene is typically represented as a set of rigid 3D points [22]. As the camera moves, static point relationships remain

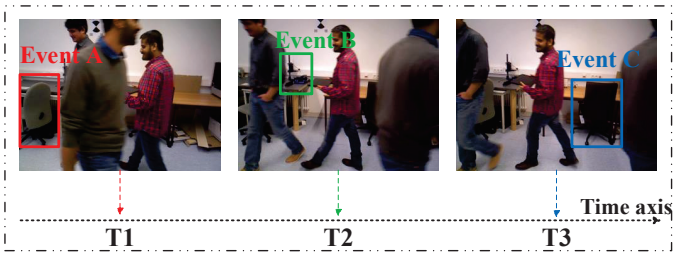

Fig. 4: Asynchronous static semantic events occur at varying times. While one entity may be visible, others could be obscured. In dynamic environments, collecting sufficient useful information simultaneously is challenging.

fixed in world coordinates. Similarly, higher-level structures like semantic entities [6] maintain these invariant relationships. However, the sheer number of points complicates structured representations at the point level. Since semantic entities are significantly fewer than 3D points, scenes can be efficiently represented with structured semantic graphs [21]. We thus define the semantic graph G:

$$G = (V, \omega). \tag{1}$$

In this definition of G, V is the graph's vertex defined as follows:

$$V = \{v_i\}_{i=1}^{n}, \quad v_i = (\underbrace{p_i \in \mathbb{R}^3}_{\text{positions}}, \underbrace{f_i \in \mathbb{R}^d}_{\text{visual features}}). \tag{2}$$

And ω_{ij} is the edge between the vertex v_i and vertex v_j of distance d_{ij} and direction u_{ij} defined as follows:

$$\omega_{ij} = \langle v_j, v_j \rangle \quad \begin{cases} d_{ij} = \|p_j - p_i\|_2 & (distance) \\ u_{ij} = \frac{p_j - p_i}{d_{ij}} & (direction) \end{cases} \tag{3}$$

Fig. 3 illustrates various situations for the same location. These edges in the graph inherently constrain rigid transformations across all nodes. When nodes between two graphs are non-identical, such rigid structures significantly restrict their common substructures. Understanding these connections enables the prediction of unseen semantic entities when the sight is limited or dynamics obscure the scene.

B. Asynchronous Static Semantic Events

Scenes can be represented as static semantic entity graphs [23], detectable when dynamic objects are absent. While these graphs should remain consistent across revisit events [6], limited camera visibility and dynamic occlusions may prevent entity detection, potentially compromising loop closure.

Dynamic objects are inherently in motion [4], so the occlusions they cause are not permanent-otherwise, such entities (e.g., stationary vehicles) would be static and serve as landmarks [23]. In a semantic graph $G = (V, E)$, each node V is detected asynchronously over time: at any moment t, it is either detected or not. Different nodes may be observed at different times due to dynamic occlusions.

Despite intermittent visibility, static semantic entities remain detectable in dynamic scenes (Fig. 4). While each may eventually be detected, simultaneous observation is unlikely. The key challenge is to temporally aggregate static information to overcome these occlusions and achieve consistent semantic understanding.

IV. METHODOLOGY

Our A-SPAM framework (see Fig. 2) addresses dynamic scene challenges (Sec. III) through semantic padding and matching, enabling robust static feature tracking and precise loop detection for SLAM systems.

A. Semantic Padding

*1) **Cross-Frame Semantic Consistency**:* While static semantic events occur continuously, only select events create new entities. Multiple events may correspond to the same entity across time, making event-entity matching critical. SAM2 [10] segments entities across videos when prompted (via mask/box), while GroundingDINO [9] provides open-vocabulary prompts. As discussed in III-B, detection events are asynchronous-identical entities may be prompted from different frames, causing duplicate trajectories without proper event correlation. Our tracking-based algorithm resolves this through event-event matching.

Algorithm 1 Cross-Frame Semantic Consistency Algorithm

Require: Static semantic events S_i for each frame F_i, $i = 0, 1, 2, \ldots, N$
Ensure: $\mathcal{T} = \{T_0, ..., T_n\}$(set of semantic entity trajectories)
1: Initialize the target tracker T with all events from the first frame F_0
2: Track initial events across the entire sequence and update $\mathcal{T}$
3: **for** $i = 1$ to N **do**
4: Detect if there are new events in the current frame F_i
5: **if** new events are detected **then**
6: Add the new events to the target tracker T
7: Track new events across the entire sequence and update $\mathcal{T}$
8: **end if**
9: **end for**
10: Return the list of trajectories $\mathcal{T}$

In our framework, a semantic trajectory is defined as the temporal sequence of an object represented by segmentation masks in the camera plane. (Fig. 5) Algorithm 1 first detects static landmarks using GroundingDINO to create event set S_i for each frame F_i. Events in frame F_0 generate the initial semantic trajectory set $\mathcal{T}$. Since $\mathcal{T}$ may miss entities that appear later, we compute the IoU between subsequent detections and masks in $\mathcal{T}$. When the IoU falls below the threshold i, SAM2 generates new trajectories to expand $\mathcal{T}$. This iterates until all events map uniquely to $\mathcal{T}$, eliminating redundancy while maintaining cross-frame consistency, where each trajectory's segments represent identical semantic entities. This algorithm also enhances the stability of the framework, because detecting an object entity in a single frame is less likely than detecting it at least once in a whole frame sequence.

*2) **Semantic Graph Reconstruction**:* After collecting the trajectories of different semantic entities, we must generate the semantic graph from them. Due to the impacts of dynamics, some frames may lack the direct locations of entities. Thus, we need to predict occluded entities through the rigid structure of the scene. For any static point in the scene, there is a rigid transformation between its two different camera coordinates with different camera poses.

$$P_2 = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} P_1, \tag{4}$$

where, R is the 3×3 rotation matrix, t is the 3×1 translation vector, P_1 and P_2 are the homogeneous coordinates of the points in the two camera coordinate systems.

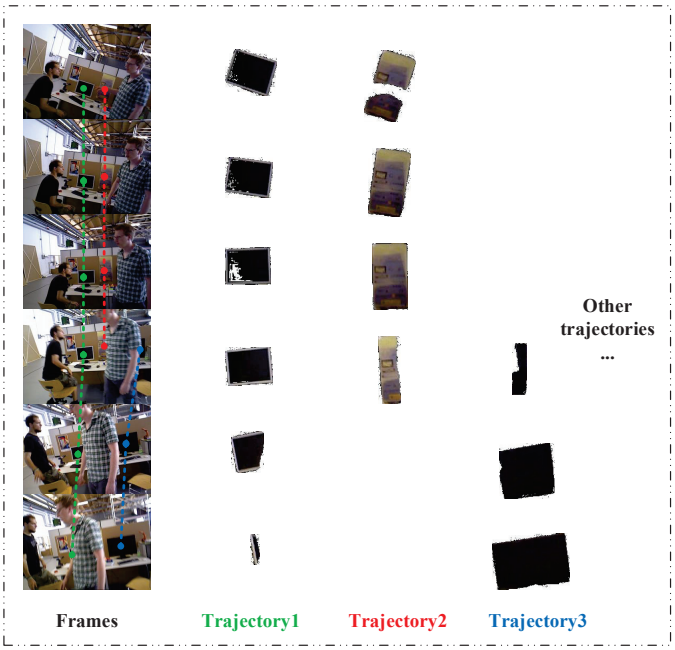

Fig. 5: The trajectories of various semantic entities can occupy different positions along the time axis. A consistency algorithm allows these cross-frame segmented masks to be unified and treated as a single entity rather than separate events.

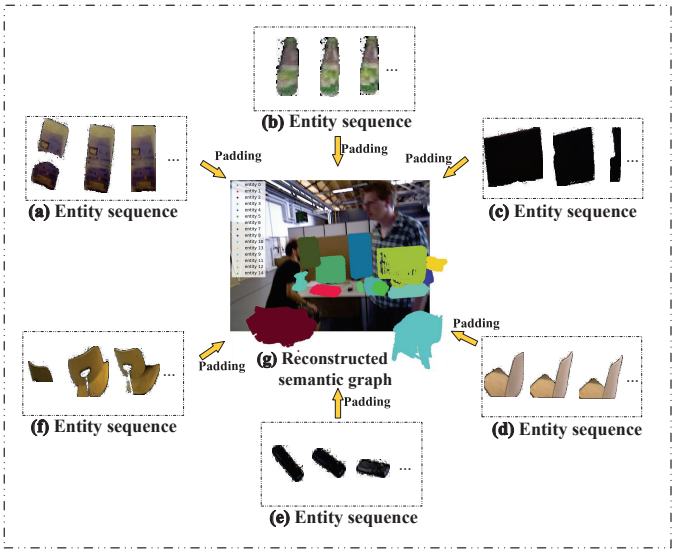

Fig. 6: Reconstructed semantic graph generated through reprojection of associated 3D points across frames. (a-e) Entity trajectories; (g) semantic graph with spatiotemporal consistency. The graph encodes semantic entities and relationships, enabling a structured representation of dynamic scenes, even though some entities are not visible.

To improve computational efficiency, we formulate keyframe selection as a set cover problem. Each frame is treated as a set of semantic events, and the entire trajectory serves as the universal set of frames. Our method employs both a forward greedy algorithm-which iteratively selects the frame covering the most uncovered semantic events-and a backward redundancy removal step-which starts with all frames and prunes those whose removal preserves full coverage. This two-stage process yields two sets of frames as keyframes, which contain all semantic entities with extra redundancy for semantic graph generation. Then, dynamic objects are removed (cf. DS-SLAM [5]), and feature correspondences between frames are established via Superpoint [24] and Superglue [25]. Matched points with depth data are fed to the Kabsch algorithm [26] to compute rigid transformations, which align semantic entity trajectories into a shared coordinate frame. This generates a semantically consistent 3D point set S for scene reconstruction.

To reduce the computational cost, we then use a minimum enclosing sphere to represent the semantic entity defined as follows:

$$\min_{c,r} r \quad \text{s.t.} \quad \|p_i - c\|_2 \leq r, \ \forall p_i \in S, \ r \geq 0, \tag{5}$$

where r is the radius and c is the center. This sphere will cover all points of the entity, and the center of this sphere is subsequently used as the entity's location for loop detection.

Fig. 6 demonstrates asynchronous semantic padding: entity trajectories vary temporally due to dynamics but maintain rigid spatial relationships with static landmarks. By preserving topological consistency while aggregating multi-frame observations, the method overcomes single-view limitations. This process filters dynamic interference to construct a complete, query-ready semantic graph.

B. *Semantic Matching*

*1) **Rigid Topology Constraint Semantic Matching**:* Defined by equation 3 and maintained during padding, the semantic graph's rigid topology allows cross-pose graph matching to be formulated as a registration task, setting it apart from image-based approaches. This geometric consistency requires prior computation of semantic entity similarities to establish correspondences. In this section, the entity is the graph node defined in the equation 2. Transformations between semantic entities' positions are based on the sphere center c defined in equation 5, and visual features are latently used in the Siamese Network for similarity.

Coarse rigid matching: We employ a Siamese Network [8] to generate a similarity matrix for various semantic entities. To enhance the stability of image features, select the five largest segmented slices from the trajectory to represent each entity. These slices are then input into the Siamese Network, where visual features are extracted and used in hidden layers for similarity calculation. Then, we exchange the input sequence from (e_i, e_j) to (e_j, e_i) and calculate the average similarity to obtain a balanced metric. Then, the average metric among the five slices is selected as the final similarity. If the similarity score exceeds a predefined threshold s, the pair is added to the sampling set as a candidate. Subsequently, we apply the Random Sample Consensus (RANSAC) method to identify the coarse rigid transformation within the sampling set by the following equation:

$$R, t = \min_{R,t} \sum_{i=1}^{3} \|Rp_i + t - q_i\|^2. \tag{6}$$

In each iteration, we randomly select three semantic pairs (q_i, p_i) from the sampling set and calculate the rotation and translation using Equation 6. For two entities, if the distance is less than d and the similarity is greater than s, we consider the entity an inlier. We then use the transformation (R, t) with the maximum number of inliers as the coarse matching result.

Fine semantic matching: The coarse transformation is derived from only three pairs, which may result in imbalanced errors across all entities. To achieve a more precise match, we propose an energy function for each semantic entity e_j to enhance matching accuracy. The energy L is defined as follows:

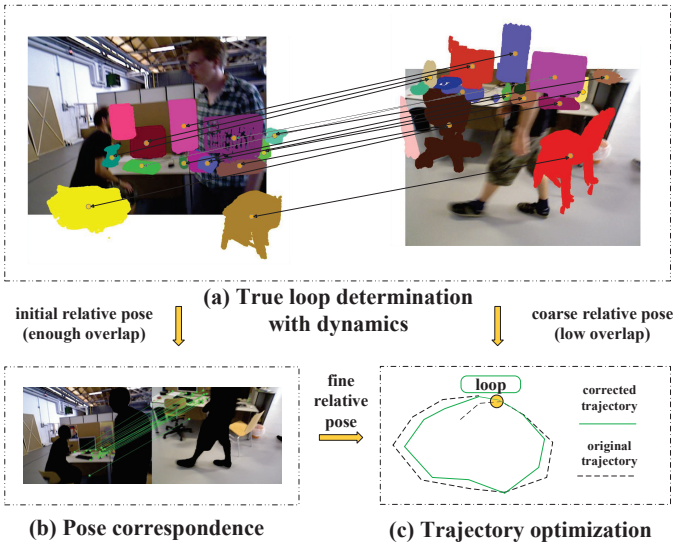

Fig. 7: Loop detection and correction. Once the semantic entities identify a true loop, we use feature points for more accurate pose estimation.

$$L(e_j, R, t) = \sum_{\substack{i=1 \\ d(e^t, e_i) < \theta}}^{n} d(e^t, e_i) \cdot s(e^t, e_i), \tag{7}$$

where e^t is the transformed entity defined as follows:

$$e^t = R \cdot e + t. \tag{8}$$

In Equation 7, $d(e^t, e_i)$ represents the Euclidean distance between the transformed entity e^t and e_i. $s(e^t, e_i)$ denotes the similarity between the transformed entity e^t and entity e_i from the Siamese Network with exchanged input sequence. The parameter θ is the distance threshold, ensuring that only nearby entities are considered in the energy calculation. To find the rotation matrix R and translation vector t that minimize the total energy L, we solve the following optimization problem:

$$\min_{R,t} \quad \sum_{j=1}^{m} L(e_j, R, t). \tag{9}$$

Then, the coarse rigid transformation initializes parameters. Rigid topology enforces a shared SE(3) transformation across entities, minimizing optimization parameters. After refinement, entities with inter-distance d and similarity s are candidate matches. Entity-pair energies generate a cost matrix C, defined as:

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}, \tag{10}$$

where in the cost matrix, cost c_{ij} is defined as:

$$c_{ij} = d(e_j^t, e_i) \cdot s(e_j^t, e_i). \tag{11}$$

Then, we apply the Hungarian algorithm [27] to determine the final matching relation.

*2) **Loop Detection and Pose Correction**:* After the previous steps, we can determine the matching relations between reconstructed semantic graphs from different camera poses. To ultimately identify true loops, we define a semantic-level co-visibility metric, μ, for loop detection as follows:

$$\mu = \frac{\frac{1}{2}\sum_{i=1}^{n} V_m}{\frac{1}{2}\sum_{i=1}^{n} V_m + \sum_{i=1}^{n} V_u}. \tag{12}$$

In this definition, the entities in a semantic graph can be divided into matched entities and unmatched entities. For a matched entity E_m, we use voxels to fill 3D points of this entity to measure its volume V_m; then, the same method is used to get the volume V_u of an unmatched entity E_u. By computing the overlap between the semantic graphs' occupancy, this metric reflects the co-visibility of the two scenarios. For any two frames with μ greater than the threshold c, we consider them as a true loop.

While object-level relative poses remain inaccurate, we hierarchically combine semantic and geometric alignment: semantic rigid transformations initialize pose estimates under low overlap, while dynamically-filtered feature matches refine poses when sufficient shared features exist (Fig. 7). Both constraints are integrated into the pose graph for global trajectory optimization, adaptively leveraging semantic coherence or geometric precision based on scene overlap.

TABLE I: Parameter settings for key modules.

Modules	Parameter	Value
GroundingDINO	Prompts	book chair bottle screen mouse signature keyboard
	Threshold	0.35
SuperPoint	Threshold	0.15
SuperGlue	Threshold	0.25
SemanticPadding	Keyframe interval	20
	Padding number	50
	IoU threshold i	0.35
SemanticMatching	Similarity threshold s	0.8
	Distance threshold θ	0.05m
	Co-visibility threshold c	0.7

V. EXPERIMENTAL RESULTS

A. *Experimental Settings*

We select ORB-SLAM3 [22] as our pose graph generator since the ORB-SLAM series is widely adopted in the field-as exemplified by DynaSLAM's [13] use of ORB-SLAM2 [30]. ORB-SLAM3 [22] offers improved accuracy and robustness over earlier versions, and its stable implementation provides a reliable base for evaluating our method.

We evaluate our approach on the TUM RGB-D [28] and BONN RGB-D [29] datasets, widely used benchmarks for dynamic SLAM that challenge visual odometry in complex environments.

(1) TUM RGB-D: Includes indoor sequences with static and dynamic elements, often with moving people causing occlusion and disturbance. It includes ground truth trajectories for evaluation.

(2) BONN RGB-D: Contains highly dynamic scenes with frequent human motion and occlusion, designed to evaluate SLAM robustness. It is suitable for evaluating trajectory improvement in challenging real-world settings.

Parameters optimized for TUM/BONN: detection threshold 0.35, feature matching thresholds 0.15/0.25, semantic

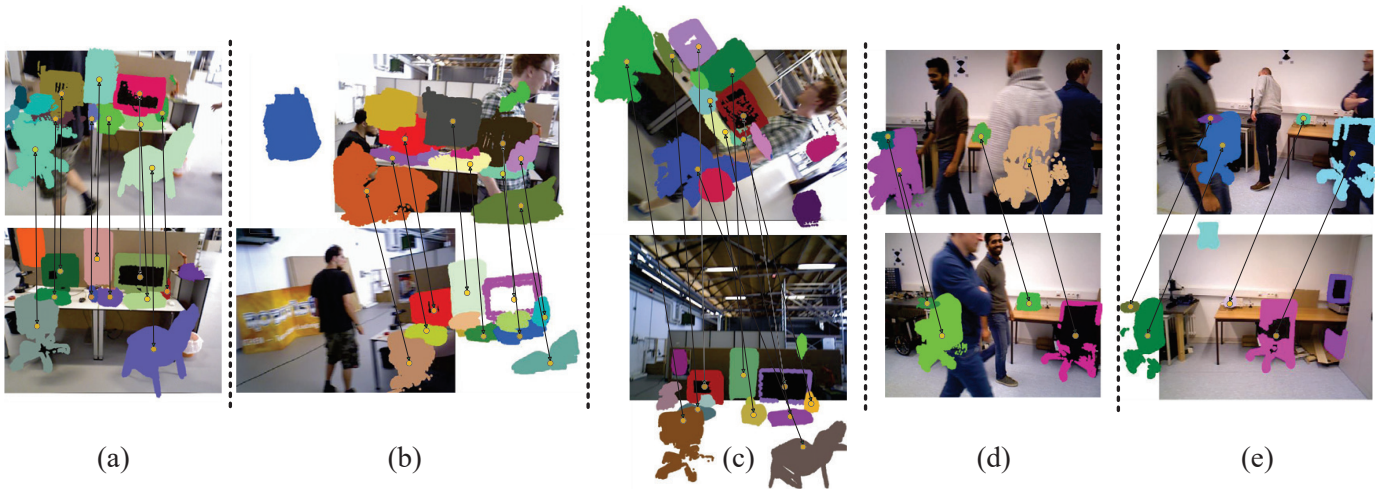

Fig. 8: Qualitative results on TUM [28] and BONN [29] datasets demonstrate our method's robustness: (a), (b), and (c) from the TUM dataset depict dynamic disturbances, varying orientations, and rotations. (d) and (e) from the BONN dataset illustrate scenarios with minimal co-vision. With semantic padding and matching, our approach effectively handles these dynamic and sight-limited challenges, achieving reliable loop detection.

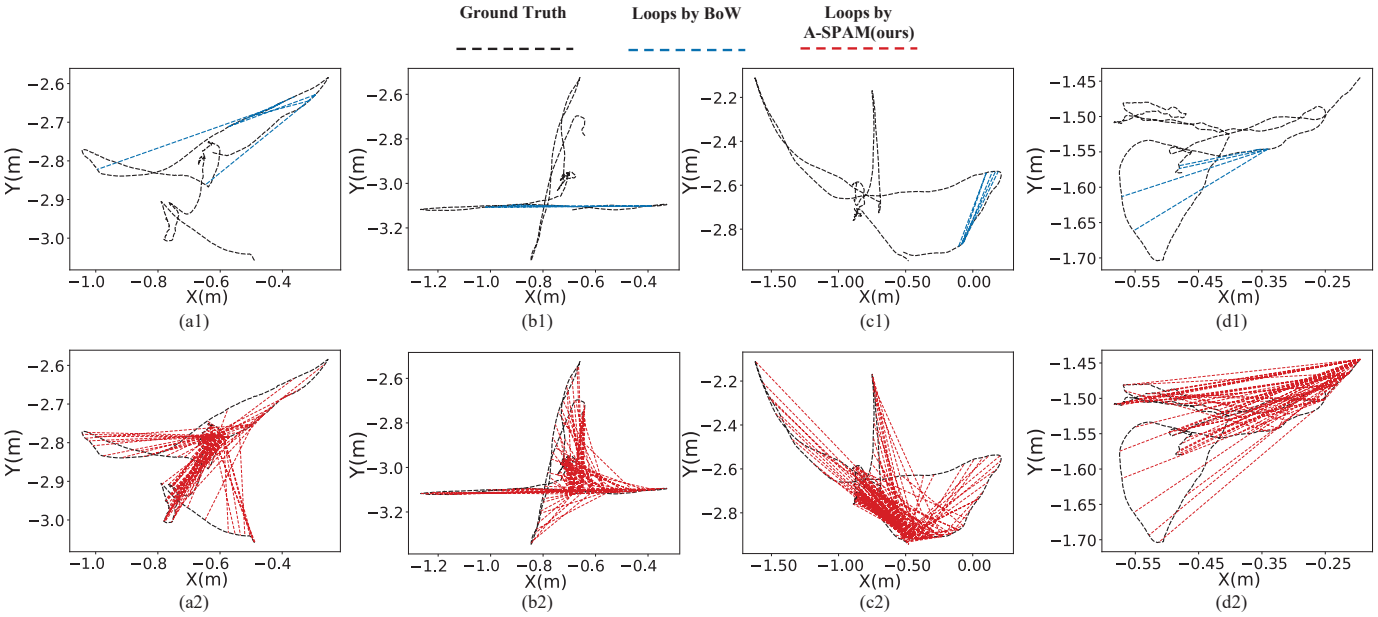

Fig. 9: Quantitative comparisons among BoW (used in different SLAM [5], [13], [22]) and A-SPAM(ours) in Dynamic Scenes: Overcoming Sparsity via Semantic Graph Reconstruction. (a) walking_rpy, (b) walking_xyz, (c) walking_halfsphere, (d) bonn_crowd. While traditional BoW methods yield sparse loop closures, especially challenged in dynamic scenes, our A-SPAM reconstructs semantic graphs to overcome scene dynamics robustly. This enables denser detection of potential loop closures (including some requiring verification, as shown), providing richer constraints for pose graph optimization compared to the sparse output of BoW.

padding every 20 keyframes, loop closure with similarity $s \geq 0.8$, co-visibility $c \geq 0.7$, and distance $\theta \leq 0.05$ m.

B. *Loop Detection Ability*

Our method is evaluated on the f3_walking_rpy and rgbd_bonn_crowd sequences to assess loop detection in dynamic environments (Fig. 8). Given their short trajectories and closely-spaced camera poses, these sequences are less suitable for long-term loop closure. Thus, we also include the f3_long_office_household dataset to evaluate the ability to distinguish between distinct locations.

*1) **Loop Detection Recall**:* We evaluate using both recall@100% precision (perfect reliability) and recall@95% precision (balanced utility). For loop detection in f3_walking_rpy and rgbd_bonn_crowd, our method samples one loop candidate every 20 frames and filters dynamic objects, achieving higher recall than other methods (Table II, Fig. 9). On the static long_office dataset, performance is slightly below MESA due to our balance between structural preservation and computational efficiency.

*2) **Ablation Study**:* Ablation studies decompose the framework into core components-semantic graph construction (entity detection and structural matching)-and supplementary components: semantic padding and similarity-enhanced matching. Table III shows baseline components perform weakly alone. Node similarities significantly boost recall in long_office due to repetitive structures, while padding stabilizes semantic graphs in dynamic sequences (walking_rpy, bonn_crowd). Together, they enhance structural discrimination and mitigate dynamic interference.

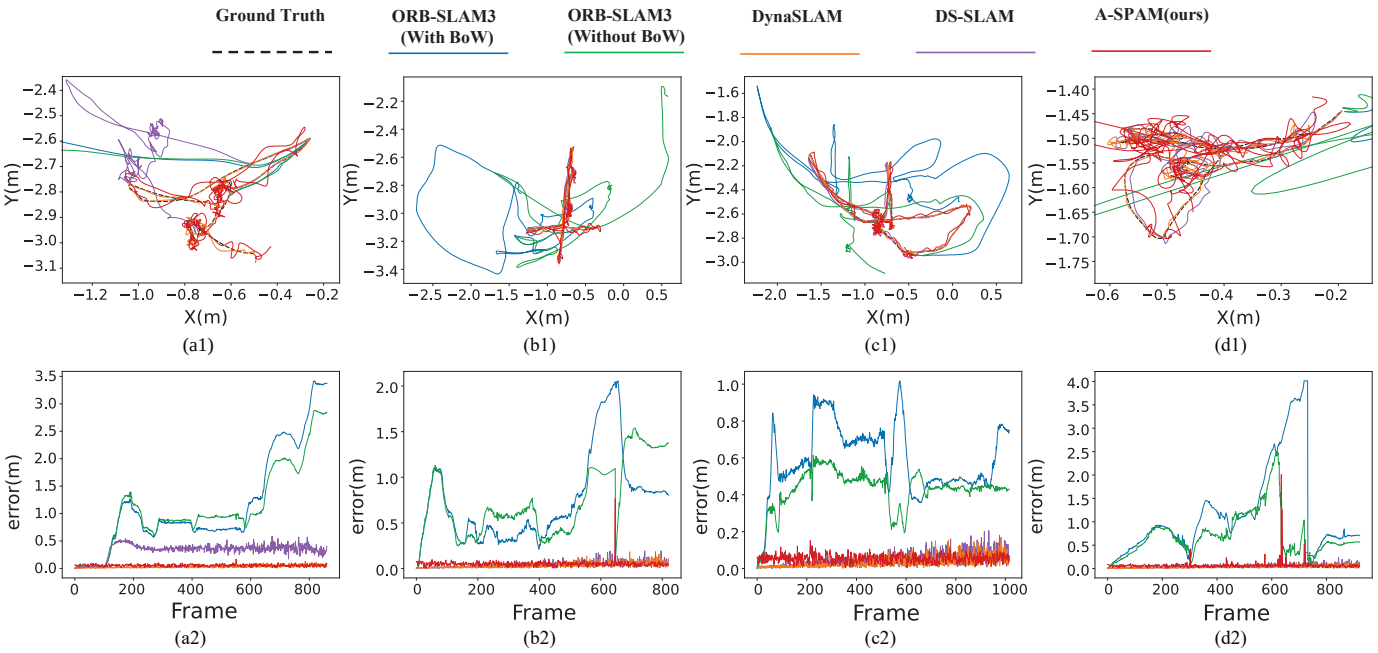

Fig. 10: Qualitative comparisons of optimized trajectories (top) and translational errors (bottom) for sequences: (a) walking_rpy, (b) walking_xyz, (c) walking_halfsphere, (d) bonn_crowd. Our method reduces dynamic-induced errors during odometry failures, outperforming ORB-SLAM3 (severe drift) and BoW-based detection. Despite poor front-end performance, our framework aligns trajectories closer to ground truth, demonstrating dynamic robustness and preventing SLAM system collapse.

TABLE II: Quantitative comparison of maximum recall at 100% / 95% precision. Unit: %. Note that our approach has the highest recall in dynamic environments.

Methods	**walking_rpy**	**bonn_crowd**	**long_office**
DELG [31]	37.8 / 38.2	36.9 / 39.4	89.1 / 90.7
MESA [19]	31.7 / 33.1	34.1 / 34.9	**91.6 / 94.9**
PatchNetVLAD [14]	29.4 / 30.9	19.7 / 22.1	85.8 88.3
CNNretrieval [32]	21.9 / 22.3	26.3 / 27.1	71.1 76.4
SP +SG [24], [25]	12.4 / 12.9	24.1 / 25.7	86.2 / 92.0
Ours (A-SPAM)	**78.1 / 82.5**	**83.9 / 88.4**	88.1 / 90.2

TABLE III: Ablation study results of maximum recall at 100% / 95% precision (Unit: %). Those modules improve our performance in diverse situations.

Modules	**walking_rpy**	**bonn_crowd**	**long_office**
A-SPAM (base)	24.7 / 25.2	21.3 / 21.8	13.9 / 14.2
+ similarities	39.6 / 41.9	23.1 / 23.7	69.8 / 74.6
+ padding	70.8 / 73.2	64.7 / 67.9	18.2 / 19.6
+ both	**78.1 / 82.5**	**83.9 / 88.4**	**88.1 / 90.2**

TABLE IV: Quantitative comparison of average relative pose error of detected loops (Unit: TE(m)/RE(°)). Sparse co-visibility in extreme conditions affects our relative pose accuracy.

Methods	**walking_rpy**	**bonn_crowd**	**long_office**
DELG [31]	0.04 / 2.8	0.04 / 2.6	0.03 / 2.1
MESA [19]	0.04 / 2.6	0.03 / 2.3	0.03 / 1.9
PatchNetVLAD [14]	0.06 / 3.7	0.07 / 2.9	0.03 / 2.4
SP+SG [24], [25]	0.04 / 4.7	0.04 / 4.3	0.02 / 3.5
Ours (Semantic)	0.10 / 7.1	0.08 / 6.3	0.07 / 5.4
Ours (Point)	0.05 / 5.1	0.04 / 4.8	0.02 / 3.8

TABLE V: Quantitative comparison of average pose trajectory error (Unit: TE(m)/RE(°), improvements with ORB-SLAM3's best performance(↓%)).

Methods	**walking_rpy**	**walking_xyz**	**bonn_crowd**	**walking_half**
DynaSLAM [13]	0.04/1.9	0.02/0.74	0.03/1.4	0.02/0.81
DS-SLAM [5]	0.44/2.9	0.02/0.87	0.06/2.1	0.03/0.96
ORB3 (BoW) [22]	1.2/17	0.74/14	1.2/56	0.59/13
ORB3 ~~(BoW)~~ [22]	1.1/16	0.74/14	0.71/41	0.42/8.7
Ours (A-SPAM)	0.06/5.4 ↓95/66	0.05/4.7 ↓93/66	0.07/9.2 ↓90/78	0.04/4.2 ↓90/52

C. *Trajectory Optimization Ability*

Our loop detection performance relies on relative pose accuracy and loop quantity. To evaluate this, we refine ORB-SLAM3's trajectory by processing sequences in chronological order to reduce early drift. For each detected loop, we compute the relative pose, update and fix the camera pose, and add a pose-graph edge before optimization, incrementally improving trajectory accuracy.

1) ***Relative Pose Accuracy****:* Our method achieves comparable accuracy to specialized approaches in static scenes (0.05m error), though errors increase in dynamic environments (up to 0.10m). This stems from a deliberate trade-off: our semantic-based method maintains detection capability under high dynamics (Table II) at the cost of some pose precision, whereas conventional methods often discard such challenging frames entirely.

2) ***Pose Trajectory Optimization Accuracy****:* Our framework shows stronger robustness in dynamic scenes than ORB-SLAM3, which suffers from trajectory errors under dynamic

TABLE VI: Computational efficiency analysis: Memory (GB) and time (s) per frame. Devices: CPU(i7-12800), GPU(3080ti)

Modules	Memory	Time	Components (Time)
Semantic Padding	GPU: 3.9 CPU: 2.4	1.3	Semantic Extraction:0.9 Graph Construction: 0.4
Semantic Matching	GPU: 0.2 CPU: 0.4	0.6	Similarity Calc.: 0.4 Geom. Verify: 0.2

interference (Fig. 10 a2–d2). Using frequent loop closures and semantic-rich sequences (e.g., walking_rpy), our method attains trajectory accuracy competitive with dynamic odometry approaches, despite persistent rotational drift from weak odometry and sparse constraints. While dynamic SLAM requires more than loop closure, our semantic-aware detection meaningfully reduces dynamic effects and matches specialized odometry performance.

D. *Computational Efficiency Analysis*

This section analyzes the computational efficiency of our framework as an odometry-independent backend, including time and memory costs. As summarized in Table VI, total latency averages 1.4s per frame, primarily consisting of semantic extraction (0.9s, 3.9GB GPU memory) and graph construction (0.4s). Lightweight matching and geometric verification contribute minimally, at 0.4s and 0.2s, respectively. The Semantic Padding module is identified as the most computationally intensive component.

VI. CONCLUSION

We propose A-SPAM, a semantic graph-based loop detection framework for dynamic environments. By tracking static semantic entities and constructing their trajectories, we build robust scene representations. Our matching leverages rigid topology and node similarity to efficiently identify loop candidates. Compared to existing methods, A-SPAM demonstrates superior loop detection in dynamic scenes, though with slightly reduced pose accuracy from padding. This capability significantly reduces accumulated errors and provides resilience against odometry failures in dynamic settings. Future robots will increasingly operate in dynamic environments like hospitals and supermarkets, where loop detection is particularly challenging. While our method supports long-term operation in such settings, limitations remain: dependency on detectable semantic entities and reduced pose accuracy from entity-center calculations. We will improve these issues in future work.

REFERENCES

[1] C. Wu, Z. Gong, B. Tao, K. Tan, Z. Gu, and Z.-P. Yin, "Rf-slam: Uhf-rfid based simultaneous tags mapping and robot localization algorithm for smart warehouse position service," IEEE Transactions on Industrial Informatics, vol. 19, no. 12, pp. 11 765–11 775, 2023.

[2] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 11, pp. 19 929–19 953, 2022.

[3] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," IEEE Transactions on Robotics, vol. 28, no. 5, pp. 1188–1197, 2012.

[4] H. Yin, S. Li, Y. Tao, J. Guo, and B. Huang, "Dynam-SLAM: An accurate, robust stereo visual-inertial SLAM method in dynamic environments," IEEE Transactions on Robotics, vol. 39, no. 1, pp. 289–308, 2023.

[5] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in International Conference on Intelligent Robots and Systems, 2018.

[6] H. Chen, G. Zhang, and Y. Ye, "Semantic loop closure detection with instance-level inconsistency removal in dynamic industrial scenes," IEEE Transactions on Industrial Informatics, vol. 17, no. 3, pp. 2030–2040, 2021.

[7] H. Osman, N. Darwish, and A. Bayoumi, "LoopNet: Where to focus? detecting loop closures in dynamic scenes," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 2031–2038, 2022, doi10.1109/LRA.2022.3142901.

[8] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in International Conference on Pattern Recognition, 2016, pp. 378–383.

[9] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," 2023.

[10] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. K. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. B. Girshick, P. Doll'ar, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," ArXiv, vol. abs/2408.00714, 2024.

[11] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "Probabilistic appearance-based place recognition through bag of tracked words," IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1737–1744, 2019, doi10.1109/LRA.2019.2897151.

[12] ——, "Assigning visual words to places for loop closure detection," in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 5979–5985.

[13] B. Bescos, J. M. Facil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 4076–4083, 2018, doi10.1109/LRA.2018.2860039.

[14] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-NetVLAD: Multi-scale fusion of locally-global descriptors for place recognition," in Conference on Computer Vision and Pattern Recognition, 2021, pp. 14 136–14 147.

[15] K. A. Tsintotas, L. Bampis, A. Gasteratos, and FIET, "Tracking-DOSeqSLAM: A dynamic sequence-based visual place recognition paradigm," IET Comput. Vis., vol. 15, no. 4, pp. 258–273, 2021.

[16] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in International Conference on Robotics and Automation, 2012, pp. 1643–1649.

[17] K. A. Tsintotas, L. Bampis, S. Rallis, and A. Gasteratos, "SeqSLAM with bag of visual words for appearance based loop closure detection," in Advances in Service and Industrial Robotics, 2019, pp. 580–587.

[18] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, "A comprehensive survey of scene graphs: Generation and application," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 1, pp. 1–26, 2023.

[19] C. Zhang, S. Stepputtis, J. Campbell, K. Sycara, and Y. Xie, "Hiker-sgg: Hierarchical knowledge enhanced robust scene graph generation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2024, pp. 28 233–28 243.

[20] J. Yu and S. Shen, "Semanticloop: Loop closure with 3d semantic graph matching," IEEE Robotics and Automation Letters, vol. 8, no. 2, pp. 568–575, 2023, doi10.1109/LRA.2022.3229228.

[21] H. Yue, V. Lehtola, H. Wu, G. Vosselman, J. Li, and C. Liu, "Recognition of indoor scenes using 3-d scene graphs," IEEE Transactions on Geoscience and Remote Sensing, vol. 62, pp. 1–16, 2024.

[22] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874–1890, 2021.

[23] N. Wang, X. Chen, C. Shi, Z. Zheng, H. Yu, and H. Lu, "SGLC: Semantic graph-guided coarse-fine-refine full loop closing for LiDAR SLAM," 2024.

[24] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 337–33 712.

[25] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in Conference on Computer Vision and Pattern Recognition, 2020, pp. 4937–4946.

[26] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 4, pp. 376–380, 1991.

[27] J. Munkres, "Algorithms for the assignment and transportation problems," Journal of the Society for Industrial and Applied Mathematics, vol. 5, no. 1, pp. 32–38, 1957.

[28] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in International Conference on Intelligent Robots and Systems, 2012, pp. 573–580.

[29] E. Palazzolo, J. Behley, P. Lottes, P. Giguere, and C. Stachniss, "ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals," in International Conference on Intelligent Robots and Systems, 2019, pp. 7855–7862.

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

[31] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for image search," in European Conference on Computer Vision, 2020, pp. 726–743.

[32] F. Radenovic, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 7, pp. 1655–1668, 2019.