LEXI-SG: Monocular 3D Scene Graph Mapping with Room-Guided Feed-Forward Reconstruction

Christina Kassab*[1], Hyeonjae Gil*[2], Matías Mattamala[3], Ayoung Kim[2], Maurice Fallon[1]

***Abstract*— Scene graphs are becoming a standard representation for robot navigation, providing hierarchical geometric and semantic scene understanding. However, most scene graph mapping methods rely on depth cameras or LiDAR sensors. In this work, we present LEXI-SG, the first dense monocular visual mapping system for open-vocabulary 3D scene graphs using only RGB camera input. Our approach exploits the semantic priors of open-vocabulary foundation models to partition the scene into rooms, deferring feed-forward reconstruction to when each room is fully observed—enabling scalable dense mapping without sliding-window scale inconsistencies. We propose a room-based factor graph formulation to globally align room reconstructions while preserving local map consistency and naturally imposing the semantic scene graph hierarchy. Within each room, we further support open-vocabulary object segmentation and tracking. We validate LEXI-SG on indoor scenes from the Habitat-Matterport 3D and self-collected egocentric office sequences. We evaluate its performance against existing feed-forward SLAM methods, as well as established scene graphs baselines. We demonstrate improved trajectory estimation and dense reconstruction, as well as, competitive performance in open-vocabulary segmentation. LEXI-SG shows that accurate, scalable, open-vocabulary 3D scene graphs can be achieved from monocular RGB alone. Our project page and office sequences are available here.**

I. INTRODUCTION

3D scene graphs represent scenes sparsely and hierarchically—capturing both high-level semantic regions like rooms and individual objects. These representations offer several advantages for autonomous robotic systems: they support downstream tasks such as navigation [1], can operate in real-time [2], and can scale to large environments [3]. Recent work has further enhanced their generalizability by grounding open-vocabulary language models into scene graph representations [4], [5].

However, current methods typically require high-quality depth and pose estimates [2], [1], [4], which presents a practical challenge in real-world deployments where such inputs may be noisy or unavailable. Constructing and maintaining accurate scene representations in real-world scenarios requires robust localization and mapping. Two decades of SLAM research has provided a foundation for this task [6], [7], [8], with semantic SLAM systems extending this by leveraging scene semantics for pose estimation [9], [10]. These methods require careful calibration procedures along with complex front-end engineering to achieve high performance and robustness in real settings.

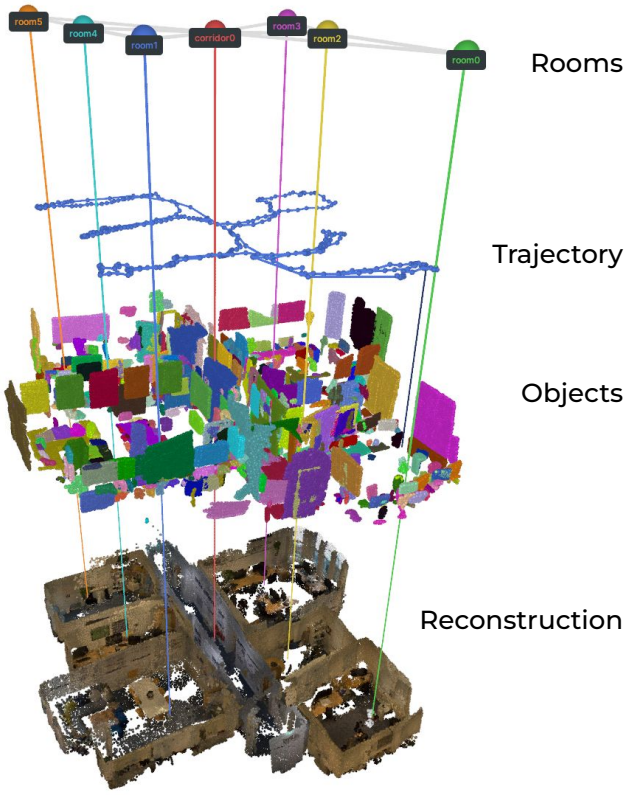

Fig. 1: **LEXI-SG is the first dense monocular mapping system to build open-vocabulary 3D scene graphs from RGB input alone.** We first partition the incoming image stream room by room. Within each room, we jointly estimate camera trajectories and dense geometry using feed-forward reconstruction models, amortizing expensive model queries while ensuring local scale consistency. The room graph is then expanded to a full 3D scene graph using open-vocabulary object segmentation.

Recently, feed-forward reconstruction models [11], [12], [13] have emerged as a new reconstruction paradigm. By leveraging large sets of training samples, these models can infer dense reconstruction and pose estimates from uncalibrated camera setups without additional sensing. While their reconstruction capabilities in smaller scenes (e.g., rooms) is impressive and more flexible than engineered pipelines, scalability is limited. Recent work has sought to integrate

* Indicates equal contribution

[1] Christina Kassab and Maurice Fallon are with the Department of Engineering Science, University of Oxford, UK. Email: {christina, mfallon}@robots.ox.ac.uk

[2] Hyeonjae Gil and Ayoung Kim are with the Department of Mechanical Engineering, Seoul National University, South Korea. Email: {h.gil, ayoungk}@snu.ac.kr

[3] Matías Mattamala is with the School of Informatics, University of Edinburgh, UK. Email: matias.mattamala@ed.ac.uk

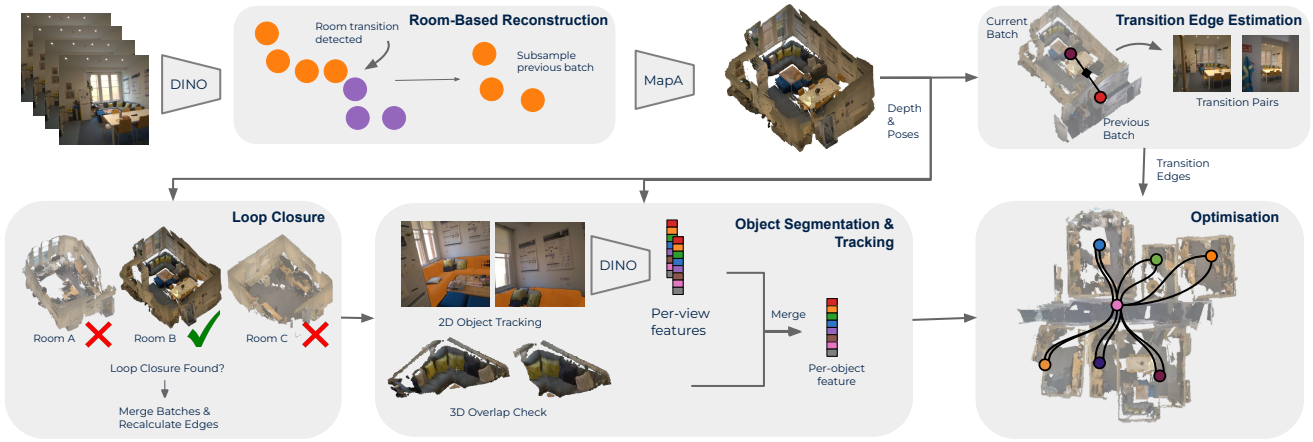

Fig. 2: **LEXI-SG System Overview.** RGB frames are segmented into rooms using DINO features. Upon detecting a room transition, the accumulated batch is passed through a feed-forward model (MapAnything—*MapA* in the figure) to produce per-frame depths and poses in a local room frame. Transition edges are estimated by feeding transition frame pairs through the same model. New rooms are checked for loop closures, with any match triggering a merge. Once finalized, rooms are processed by the object segmentation module. The pose graph is then globally optimized over Sim(3).

these models into minimal "plug-and-play" SLAM systems to reconstruct larger scenes [14], [15], [16]. Integrating feed-forward SLAM methods with semantic scene representations remains largely unexplored, particularly at scale.

In this work, we present LEXI-SG (Language EXtended Indoor Scene Graphs), a unified system for dense monocular mapping and semantic scene graph construction. Unlike current feed-forward SLAM systems, LEXI-SG defers reconstruction until each room is fully observed, amortizing model queries while improving both scale consistency and scene understanding. Our contributions are:

- LEXI-SG, the first dense monocular SLAM system that builds an open-vocabulary 3D scene graph from RGB alone, without depth, or ground-truth pose input.
- A vision-only room identification method that detects transitions using DINO.
- A room-based reconstruction strategy that defers feed-forward inference until a room is fully observed, avoiding sliding-window scale inconsistencies.
- A Sim(3) room-level factor graph that globally aligns per-room reconstructions while preserving local consistency and correcting monocular scale ambiguity.
- An open-vocabulary segmentation module that lifts 2D mask tracklets into the scene graph as 3D object nodes.
- Evaluations on SLAM and scene graph tasks on standard datasets and self-collected office recordings.

II. Related Work

A. 3D Scene Understanding using Foundation Models

Recent open-vocabulary 3D scene understanding methods have leveraged advances in visual foundation models such as CLIP [17], SAM [18] and DINO [19]. These representations can be broadly categorized into two types: *object-centric* [2], [1], [20], such as scene graphs, and *dense*, which provide per-pixel or per-point semantic representations [21], [22].

Object-centric methods typically extract segments using instance segmentation techniques, either in 2D using models such as SAM [2], [1], [20] or directly in 3D [4]. In the former case, segments are projected into 3D using depth images and temporally fused to create coherent scene representations. Spatial relationships between objects can be encoded as edges connecting object nodes, as demonstrated in ConceptGraphs [2]. Other works extend this with higher-level layers such as rooms in HOV-SG [1]. Clio [5] optimizes efficiency by generating task-specific, compact scene graphs [5].

In contrast dense methods assign semantic features to every 3D point, bypassing explicit object segmentation. OpenScene [21] and ConceptFusion [22] distill CLIP features into 3D point clouds, while LERF [23] embeds CLIP descriptors directly into a radiance field. Rayfronts [24] extends dense mapping beyond the depth range using semantic rays at map frontiers. While these methods offer fine-grained spatial resolution, they are often more computationally expensive compared to object-centric representations.

These methods typically rely on high-quality depth images which limits their use in real-world settings. LEXI-SG addresses this gap by constructing an open-vocabulary scene graph from monocular RGB input alone, leveraging feed-forward reconstruction in place of depth sensors and removing the dependency on ground truth poses.

B. Feed-Forward Monocular SLAM

Visual SLAM has been studied for more than two decades with many generations of engineered methods proposed for camera motion estimation and 3D scene reconstruction. These approaches typically rely on feature extraction and tracking [6] or the minimization of photometric error [25] to estimate camera motion and reconstruct 3D scene structure. While effective in well-conditioned settings, they require careful calibration and engineering, and remain brittle under challenging illumination, motion blur, or low-texture scenes.

Recent advances in feed-forward 3D reconstruction models, such as MASt3R [12], VGGT [11] and MapAnything [13], have demonstrated the ability to infer dense geometry and camera poses directly from multi-view inputs. This has motivated a growing body of research on SLAM systems that can operate without camera intrinsics yet can still generate dense scene representations. MASt3R-SLAM [14] couples MASt3R's two-view pointmap predictions with a tracking and global optimization back-end. Similarly, ViSTA-SLAM [16] predicts per-pair dense pointmaps and relative poses using a symmetric two-view association network. In contrast, VGGT-based approaches [15], [26] construct submaps in a rolling manner and use pose graph optimization to minimize scale and spatial drift between submaps.

While LEXI-SG also uses feed-forward reconstruction models, it does not use them on a sliding-window basis, but instead exploits semantic information across different levels to guide the reconstruction and pose estimation process. We demonstrate that using room-level semantics can improve the reconstruction results, refine the pose estimates, and enable scalable scene graph construction.

C. Semantic SLAM Systems

Semantic SLAM systems aim to tightly couple semantic and geometric information to jointly estimate camera poses and build scene representations. Early methods such as SLAM++ [10] and CubeSLAM [27] integrate object detections into the SLAM pipeline to jointly optimize camera poses and object landmarks. Inspired by these works, LEXI-SG extends the tight coupling of semantics and geometry to a monocular feed-forward setting. Room-level information is used to adaptively batch inputs for globally consistent reconstruction and to guide object segmentation and tracking, producing a full 3D scene graph without depth sensors or ground truth poses.

III. Method

A. System Overview

A system overview of LEXI-SG is presented in Fig 2. The only input to the system is a stream of RGB images. The system maintains a room pose graph for optimization and incrementally populates a 3D scene graph with room and object nodes as output. The main modules of the system are: room-based reconstruction, object segmentation and tracking, loop closure, and global optimization.

B. Graph Structure

3D Scene Graph. The scene is represented as a hierarchical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with two layers of nodes and two types of edges. We denote nodes as $\mathcal{V} = \mathcal{V}_R \cup \mathcal{V}_O$. $\mathcal{V}_R$ is the set of room nodes and $\mathcal{V}_O$ is the set of object nodes.

Each room node $v_{r_i} \in \mathcal{V}_R$ has an associated local reference frame $\mathcal{F}_{r_i}$ with reference pose $\mathbf{T}_{r_i} \in \mathrm{Sim}(3)$, and a point cloud $\mathcal{P}_{r_i}$. Each object node $v_{o_i} \in \mathcal{V}_O$ has an associated frame o_i with pose $\mathbf{T}_{r_i o_i} \in \mathrm{Sim}(3)$ expressed *relative to its parent room frame* r_i, a point cloud $\mathcal{P}_{o_i}$, and a semantic feature vector $\mathbf{f}_{o_i}$. The edge set decomposes as $\mathcal{E} = \mathcal{E}_{RR} \cup \mathcal{E}_{RO}$. Room-to-room edges $e_{r_i r_j} \in \mathcal{E}_{RR}$ connect neighboring room nodes and encode the relative transformation $\mathbf{T}_{r_i r_j}$ between their local reference frames. Room-to-object edges $e_{r_i o_i} \in \mathcal{E}_{RO}$ connect each object node to its parent room node and encode the relative pose $\mathbf{T}_{r_i o_i}$ of the object frame in the room's local reference frame.

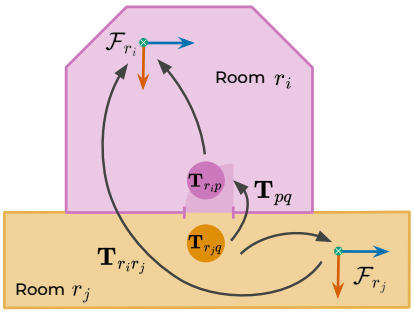

Fig. 3: **Transition edge estimation.** The relative transform $\mathbf{T}_{r_i r_j}$ between adjacent rooms is estimated by retrieving transition image pairs (p, q) and computing $\mathbf{T}_{pq}$ via a feed-forward reconstruction model.

Room Pose Graph. For global optimization we operate on a sub-graph of $\mathcal{G}$ obtained by restricting to the room layer,

$$\mathcal{G}_P = (\mathcal{V}_R, \mathcal{E}_{RR}) \subseteq \mathcal{G}, \tag{1}$$

which we refer to as the *room pose graph*. Its nodes are the room reference poses $\mathbf{T}_{r_i}$ and its edges are the room-to-room relative transforms $\mathbf{T}_{r_i r_j}$, instantiated either as transition edges between temporally adjacent rooms (Sec. III-D) or as loop closure edges between revisited rooms (Sec. III-E). The room pose graph is a Sim(3) pose graph and its optimization is described in Sec. III-G. Room-to-object edges $\mathcal{E}_{RO}$ encode parent–child containment in the hierarchy and are not optimized.

C. Room-Based Reconstruction

A common way to integrate feed-forward reconstruction models into a SLAM pipeline is to query them on a sliding window basis. However, this strategy produces locally inconsistent geometry across overlapping windows, leading to double-walled surfaces and scale drift—all of which degrades downstream object-level mapping. To avoid these limitations, we propose a room-aware strategy that defers reconstruction until a room has been fully observed, and then reconstructs it once from a single, curated batch of views. This results in a minimal but effective approach.

For each incoming frame, our system extracts DINO features $\mathbf{f}_t$, which are compared against text encodings of transition cues (e.g. 'doorways' and 'corridors') yielding a per-frame semantic label. Since these predictions can be noisy, room transitions are governed by a hysteresis mechanism: a running confidence score is accumulated across frames, and a room transition is only triggered when this score crosses a threshold. This requires there be a sustained signal to confirm a transition.

Upon detecting a room transition, the frames accumulated since the last transition are finalized as a batch. A small

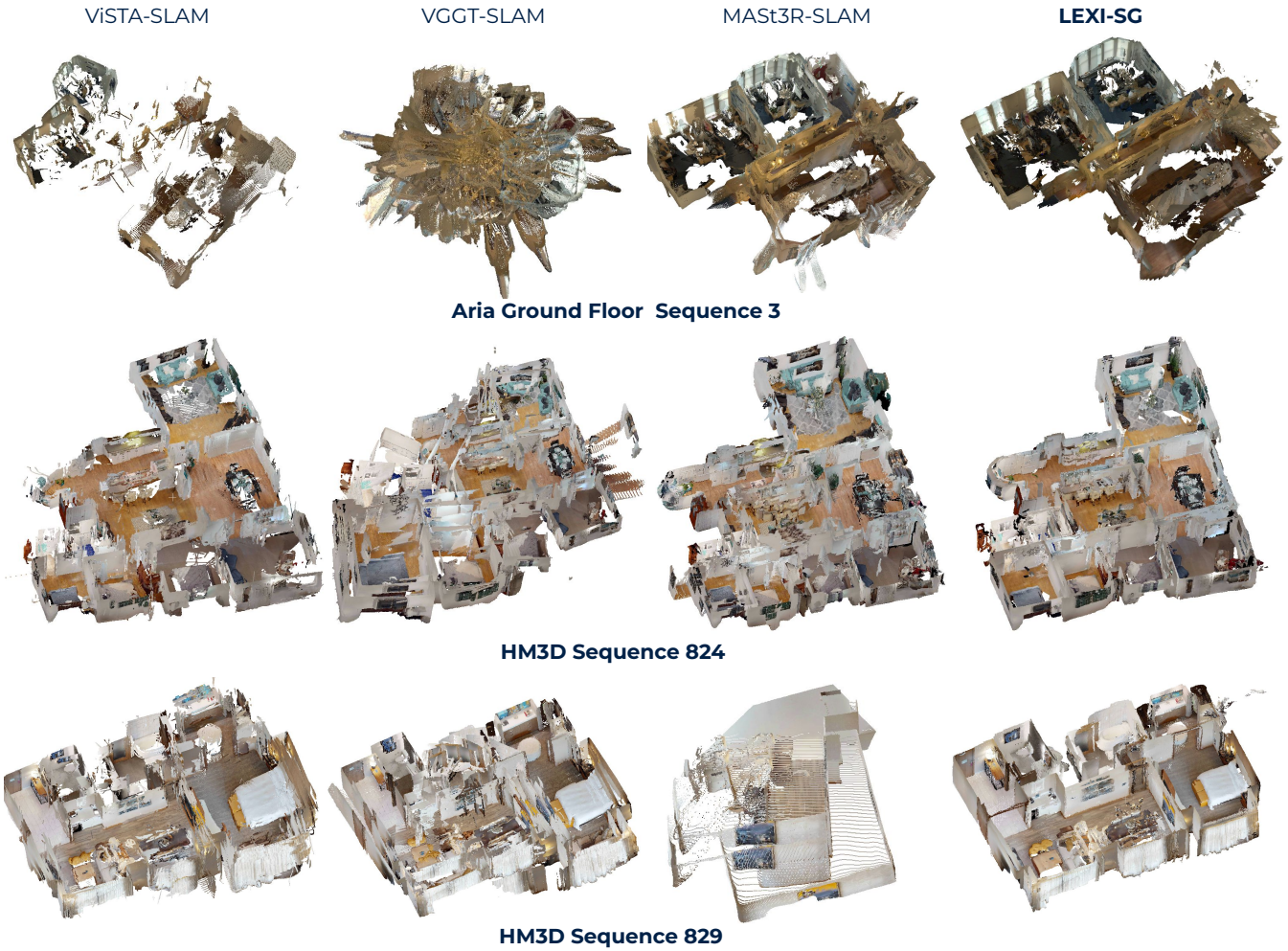

Fig. 4: **Qualitative reconstruction results on our office dataset AOD and HM3D.** LEXI-SG produces more accurate, globally consistent reconstructions with minimal double-walling and greater robustness across a range of indoor sequences.

number of frames from the end of each batch are carried forward as an overlap into the following segment, ensuring continuity across rooms. The finalized batch is subsampled and passed through a feed-forward reconstruction model, which converts the images into a dense 3D point cloud together with per-frame depth estimates and poses defined relative to the local reference frame $\mathbf{T}_{r_i}$, anchored at the first image of the batch. This reference frame is subsequently incorporated as a node in the room pose graph. In this way, each room is reconstructed exactly once from its full set of observations, naturally reducing double-walling and scale inconsistencies inherent in sliding-window approaches.

D. Transition Edge Estimation

To construct an edge between two adjacent room nodes v_i^r and v_j^r, we estimate the relative transformation $\mathbf{T}_{r_i r_j}$ between their respective local reference frames $\mathcal{F}_{r_i}$ and $\mathcal{F}_{r_j}$. As illustrated in Fig. 3, this is achieved by retrieving a set of transition image pairs. We assume that the boundary between two rooms is observed in the final frames of the preceding batch and the initial frames of the following batch.

For each transition pair (p, q), where p and q denote frame indices in $\mathcal{F}_{r_i}$ and $\mathcal{F}_{r_j}$ respectively, the two images are passed through the feed-forward reconstruction model to obtain their relative transform $\mathbf{T}_{pq}$. Here, $\mathbf{T}_{r_i p}$ denotes the pose of frame p expressed in $\mathcal{F}_{r_i}$, and $\mathbf{T}_{r_j q}$ denotes the pose of frame q expressed in $\mathcal{F}_{r_j}$. The transform between the two room reference frames is then recovered as:

$$\mathbf{T}_{r_i r_j} = \mathbf{T}_{r_i p} \cdot \mathbf{T}_{pq} \cdot \mathbf{T}_{r_j q}^{-1} \tag{2}$$

This process is repeated for all transition pairs, yielding a set of transform estimates $\{\mathbf{T}_{r_i r_j}\}$ which are stored as room-to-room edges $e_{r_i r_j} \in \mathcal{E}_{RR}$ in the room pose graph.

E. Loop Closure

As the agent traverses the environment, it may revisit previously observed rooms. To detect this, the loop closure module maintains a database of room nodes encountered so far, and checks each newly finalized room node against this database for potential matches.

For each new room node v_{r_i}, its image features are compared against those of every room in the database using cosine similarity. A candidate match v_{r_j} is confirmed if the number of image feature pairs with a similarity score above

TABLE I: **Chamfer distance evaluation (m) on self-collected scenes from an office environment.** X denotes rooms for which a valid reconstruction could not be recovered.

	Floor 1							Ground Floor			
	Room 0	Room 1	Room 2	Room3	Room 4	Room 5	Corridor	Room 0	Room 1	Room 2	Corridor
ViSTA-SLAM	0.255	X	0.365	X	0.153	0.193	0.416	X	X	X	X
VGGT-SLAM Sim(3)	0.215	X	X	X	X	X	X	X	X	X	X
MASt3R-SLAM	0.203	0.225	0.201	0.210	X	X	0.163	0.213	0.152	**0.130**	0.147
LEXI-SG	**0.114**	**0.147**	**0.136**	**0.170**	**0.128**	**0.148**	**0.085**	**0.126**	**0.095**	0.235	**0.093**

τ_s exceeds a threshold τ_r. Upon finding a candidate match, the image sets of the two nodes are merged to form a unified room batch. This merged batch is passed through the feed-forward reconstruction model to produce a new set of local poses and a unified room reconstruction, yielding a candidate node $\tilde{v}_r$ with local reference frame $\mathcal{F}_{\tilde{r}}$ that replaces both v_{r_i} and v_{r_j} in the room pose graph.

Before finalizing the merge, the geometric consistency of the candidate node with its neighbors must be verified. For each room node v_{r_k} that shares a room-to-room edge with either v_{r_i} or v_{r_j}, a new relative transformation $\mathbf{T}_{\tilde{r}r_k}$ is estimated via pairwise boundary frame matching, as described in Sec. III-D. The merge is only accepted if valid transformations can be estimated for *all* affected edges.

If the merge is accepted, the two original nodes are removed from the room pose graph and replaced by the candidate node $\tilde{v}_r$. All room-to-room edges $\mathcal{E}_{RR}$ previously incident to v_{r_i} or v_{r_j} are deleted and replaced by the newly verified loop closure edges connecting $\tilde{v}_r$ to its neighbors. The candidate node is then added to the database, and the original entries are removed. If no match is found, or if edge verification fails, the new node is added to the database unchanged and the graph remains unmodified.

F. Open-Vocabulary Object Segmentation and Tracking

Once a room has been finalized and any loop closures resolved, its image frames (drawn from the room's existing batch) are passed to the object segmentation and tracking module, which populates the room with child object nodes to complete the scene graph hierarchy.

Given RGB-D sequences for each room processed with a feed-forward reconstruction model, per-frame object masks are generated using Recognize Anything [28] and Grounding Dino [29]. We use a SAM2-based [18] tracking module to propagate each seed mask to adjacent frames and form object-centric tracklets across views. This propagation step turns an originally single-view signal into a multi-view signal, which provides a stronger support for subsequent 3D object reconstruction. It also makes cross-view mask association possible directly in image space through mask-level overlap, not solely relying on the 3D point cloud overlap. The propagated and merged mask tracklets are lifted into object-level point clouds. Each object point cloud becomes an object node $v_{o_i} \in \mathcal{V}_O$ with pose $\mathbf{T}_{r_i o_i}$ expressed in the parent room frame and attached via a room-to-object edge $e_{r_i o_i} \in \mathcal{E}_{RO}$ completing the scene graph hierarchy.

TABLE II: **Absolute trajectory error (ATE (m)) on selected scenes from Habitat-Matterport 3D.** X denotes failed sequences or sequences with more than 2m of error.

	824	829	843	847	873	877	890
ViSTA-SLAM	0.351	0.309	1.154	1.884	X	0.576	X
VGGT-SLAM SL(4)	X	X	X	X	X	X	X
VGGT-SLAM Sim(3)	1.153	0.849	1.777	1.431	X	0.872	0.613
VGGT-SLAM 2	0.703	0.611	**0.620**	X	**0.477**	0.955	**0.553**
MASt3R-SLAM	0.693	X	1.521	0.717	X	X	1.695
LEXI-SG	**0.343**	**0.143**	0.628	**0.461**	X	**0.554**	0.712

TABLE III: **Absolute trajectory error (ATE (m)) from the Aria office environment.** X denotes failed sequences or sequences with more than 2m of error.

	Floor 1			Ground Floor			
	Seq 1	Seq 2	Seq 3	Seq 1	Seq 2	Seq 3	avg
ViSTA-SLAM	0.840	1.038	1.405	0.668	0.291	1.651	0.982
VGGT-SLAM SL(4)	1.194	X	X	0.745	0.804	X	-
VGGT-SLAM Sim(3)	1.320	0.751	0.469	0.787	0.503	X	-
VGGT-SLAM 2	1.037	0.586	X	0.548	0.485	X	-
MASt3R-SLAM	0.302	X	0.294	**0.241**	**0.139**	0.296	-
LEXI-SG	**0.166**	**0.277**	**0.277**	0.647	0.201	**0.262**	**0.305**

G. Global Optimization

We globally optimize the room pose graph $\mathcal{G}_P = (\mathcal{V}_R, \mathcal{E}_{RR})$ over the $\mathrm{Sim}(3)$ Lie group to minimize inconsistencies introduced by accumulated drift and noisy pairwise estimates, allowing the optimizer to correct for the scale ambiguity inherent in monocular reconstruction. Only the room reference poses $\mathbf{T}_{r_i}$ are optimized; object poses $\mathbf{T}_{r_i o_i}$, stored in room-local frames, update implicitly, so the full 3D scene graph remains globally consistent. The resulting factor graph is solved using the Levenberg–Marquardt optimizer.

IV. Experiments

A. Experimental Setup

We validated LEXI-SG across four standard SLAM and scene graph tasks: Pose Estimation (Task 1), Dense Reconstruction (Task 2), Room Segmentation (Task 3), and Open-Vocabulary Object Segmentation (Task 4).

All experiments are run on a workstation with an NVIDIA RTX 4090 GPU. We provide the technical details of the LEXI-SG configuration, baselines, and datasets as follows.

LEXI-SG. We use MapAnything [13] as the feed-forward reconstruction model throughout, with a batch size of 60 to ensure sufficient frame coverage across larger rooms and corridors. MapAnything was selected over models such as VGGT and DepthAnything3 despite its marginally lower

TABLE IV: **Room segmentation evaluation.** Precision and recall are calculated based on the metric provided by Hydra [31]. Hydra, HOV-SG and LEXI-SG GT use ground truth poses and RGB-D images in this experiment.

		824	829	843	873	877	890	Avg
Precision	LEXI-SG	0.52	0.58	0.77	X	0.54	0.62	-
	LEXI-SG GT	0.56	0.75	0.77	0.78	0.59	0.81	0.71
	HOV-SG	0.81	0.86	0.88	0.95	0.74	0.94	0.86
	Hydra	0.79	0.88	0.87	0.96	0.81	0.95	**0.88**
Recall	LEXI-SG	0.85	0.98	0.69	X	0.86	0.74	-
	LEXI-SG GT	0.96	0.91	0.74	0.85	0.78	0.93	**0.86**
	HOV-SG	0.80	0.88	0.87	0.67	0.92	0.87	0.84
	Hydra	0.78	0.85	0.78	0.80	0.88	0.62	0.79

pose estimation accuracy because our experiments show that it is more robust in long corridors. However, LEXI-SG could be easily adapted to other similar models.

Baselines. For Camera Pose Estimation (Task 1) and Dense Reconstruction (Task 2) we compared against MASt3R-SLAM [14], the Sim(3) and SL(4) variants of VGGT-SLAM [15], VGGT-SLAM2 [30] and ViSTA-SLAM [16].

For Room Segmentation (Task 3) we compared against HOV-SG [1] and Hydra [31]. For Open-Vocabulary Object Segmentation (Task 4) we compared against comparable object-centric methods such as ConceptGraphs [2].

Datasets. For pose estimation and reconstruction (Tasks 1 and 2), we evaluate on multi-room datasets rather than single-room benchmarks such as TUM RGB-D [32], since a single room constitutes one batch and would evaluate the feed-forward model alone rather than our full system.

Therefore, we evaluate on sequences from Habitat-Matterport 3D (HM3D) [33], a large-scale dataset of multi-room, multi-floor indoor home environments. Furthermore, we used self-collected sequences from a two-floor office environment recorded using Meta's Project Aria glasses (Gen 1) which we call the *Aria Office Dataset* (AOD). AOD contains six sequences in total, three per floor, each spanning between 2 and 6 rooms and including loop closures. We treat the output poses from the Meta's multi-camera visual-inertial SLAM system as ground-truth. Although atypical, this system has been shown to achieve sub-centimetre accuracy in indoor settings [34], providing reliable ground truth. We use only the rectified forward-facing RGB camera with an FoV of 110° from the glasses in our experiments.

The semantic evaluations (Tasks 3 and 4) were performed using the protocol defined in OpenLex3D [35].

B. Task 1 – Camera Pose Estimation

Results for HM3D and AOD sequences are presented in Tabs. II and III respectively. LEXI-SG achieves the lowest average trajectory error across the datasets. On HM3D, VGGT-SLAM2 achieves the next best performance, though all monocular methods struggle on the larger-scale sequences. The SL(4) variant of VGGT-SLAM regularly diverges on HM3D, with Sim(3) proving more stable. Our room-based image batching approach yields improvements on these sequences, demonstrating how monocular feed-forward SLAM can be scaled to large environments.

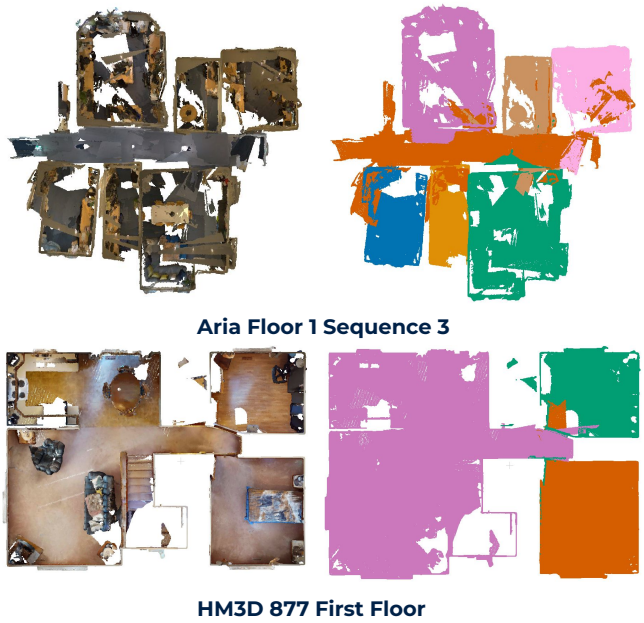

Fig. 5: **Qualitative room segmentation results on the AOD and HM3D sequences.** Our approach reliably delineates room boundaries by detecting transitional structures (such as doorways and corridors) from RGB input alone—without relying on a depth sensor or geometric priors.

On the AOD sequences, MASt3R-SLAM achieves the second best performance. The performance of LEXI-SG on ground floor sequences 1 and 2 is reduced by some room segmentation errors, where portions of a large meeting room are misclassified as corridor. As shown in Fig. 4, improved pose estimation translates directly to more globally consistent reconstructions, with minimal double-walling and reduced overlap between adjacent rooms.

We attribute these gains to our room-based reconstruction: deferring feed-forward inference until a room is fully traversed gives each batch maximal co-visibility, improving per-room reconstruction quality and reducing the drift that accumulates when chaining many sliding-window batches.

C. Task 2 – Dense Reconstruction

We segment the reconstructed scenes from each method into individual rooms and evaluate the resulting geometry against ground-truth terrestrial laser scans using the AOD sequences. We use the same set of baselines listed above.

To evaluate reconstruction accuracy we present chamfer distance results in Tab. I. We calculate the chamfer distances in the same manner as MASt3R-SLAM [14]. We denote with an X any rooms for which a valid reconstruction could not be recovered. On average LEXI-SG achieves the lowest reconstruction error, with MASt3R-SLAM achieving the next best results. ViSTA-SLAM and VGGT-SLAM produced reconstructions with overlapping rooms, though a small number of rooms could still be extracted and evaluated. VGGT-SLAM2 was excluded, as it either produced overlapping rooms or failed to converge.

Fig 4 shows qualitative comparisons with baseline methods. Our room-based batching approach achieves greater

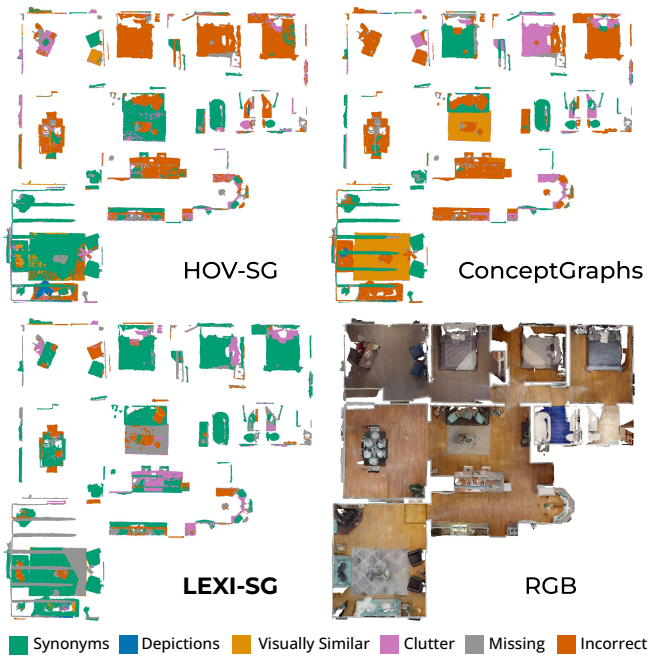

Fig. 6: **Visualization of OpenLex3D Benchmark on 00824 sequence of HM3D dataset using ground-truth poses and depth.** Our object segmentation module shows better performance in the synonyms category (green) and fewer incorrect labels compared to other baselines.

local as well as global geometric consistency. These improvements stem from LEXI-SG reconstructing each room in a single pass. In contrast, sliding-window approaches must fuse many partial reconstructions from overlap windows, blurring fine geometric detail.

D. Task 3 – Room Segmentation

We evaluate on HM3D sequences following the evaluation protocol of Hydra [31], and compare against HOV-SG and Hydra. These methods use ground truth poses and depth images for their evaluations.

We present quantitative and qualitative room segmentation results in Tab. IV and Fig. 5, respectively. While relying only on RGB image, LEXI-SG achieves comparable recall, albeit with lower precision. This precision gap can be attributed to our room transition-based approach: by detecting room boundaries through doorways and corridors, our method does not fully segment open-plan layouts present in the HM3D sequences, as illustrated in the bottom example of Fig. 5.

We also evaluate LEXI-SG using the ground truth poses and the depth images provided with HM3D. In this setting, both precision and recall improve. LEXI-SG GT surpasses HOV-SG and Hydra in recall, while precision increases but remains comparatively lower, again owing to the limitations in our method for room segmentation in open-plan spaces. Nevertheless, the results demonstrate that reasonably accurate room segmentation is achievable from monocular RGB alone, without access to ground-truth poses or depth.

TABLE V: **Semantic Segmentation Results on the OpenLex3D Benchmark (using ground-truth poses and depth).** *S* is the FREQ at synonyms, *D* is depictions, *VS* is visually similar, *C* is clutter, *M* is missing and *I* is incorrect.

Data	Method	$S\uparrow$	$D\downarrow$	$VS\downarrow$	$C\downarrow$	$M\downarrow$	$I\downarrow$
Replica	ConceptGraphs [2]	0.41	0.01	0.11	0.24	**0.02**	0.22
	HOV-SG [1]	**0.45**	**0.00**	**0.05**	0.27	0.07	0.16
	OpenMask3D [4]	0.43	0.01	0.07	0.29	0.10	**0.10**
	LEXI-SG	0.42	**0.00**	0.07	**0.19**	0.16	0.16
ScanNet++	ConceptGraphs [2]	0.26	0.02	0.05	**0.10**	0.13	0.44
	HOV-SG [1]	**0.40**	0.02	0.04	0.16	**0.08**	0.30
	OpenMask3D [4]	0.27	**0.01**	**0.03**	0.29	0.13	0.27
	LEXI-SG	**0.40**	**0.01**	**0.04**	0.19	0.12	**0.24**
HM3D	ConceptGraphs [2]	0.27	0.02	**0.03**	**0.12**	**0.08**	0.47
	HOV-SG [1]	0.33	0.02	0.04	0.18	**0.08**	0.36
	OpenMask3D [4]	0.31	**0.01**	**0.03**	0.13	0.26	**0.26**
	LEXI-SG	**0.41**	**0.01**	**0.03**	0.14	0.16	**0.26**

E. Task 4 – Open-Vocabulary Semantic Segmentation

We evaluate our object segmentation module on the OpenLex3D benchmark [35], using its semantic segmentation metric. Results are reported in Tab. V. Following the same protocol as used by ConceptGraphs, HOV-SG, and OpenMask3D, we use the provided ground truth poses and depth images for each segment. This allows us to isolate the semantic segmentation performance, ensuring that the evaluation is not affected by potential misalignments between predicted and ground-truth geometry. A perfect score corresponds to a similarity (S) of 1.0, and 0.0 across all remaining categories.

On the Replica and ScanNet++ datasets, LEXI-SG achieves performance comparable to HOV-SG, and achieves the best synonym score on HM3D. These results suggest that tracking segments in 2D is more favourable than 3D merging strategies based on IoU and cosine similarity thresholds, as employed by ConceptGraphs and OpenMask3D. The strong performance of HOV-SG suggests that DBSCAN feature clustering yields more discriminative embeddings than averaging; however, in our approach we opt for averaging as it avoids additional computational overhead.

F. Ablations

Tables VI and VII report ATE ablations for the main components of LEXI-SG. Room-based reconstruction nearly halves ATE on AOD over a sliding-window baseline, with a smaller but consistent improvement on HM3D, confirming that deferring feed-forward inference until a room is fully observed reduces drift from chaining overlapping windows.

TABLE VI: **Ablation results reporting average ATE (m) on AOD and HM3D (excluding seq 877).** Room-based reconstruction improves performance on both datasets, with the loop closure module yielding further improvements.

Room-Based Reconstruction	Loop Closure	AOD	HM3D
✗	✗	0.812	0.626
✓	✗	0.427	0.518
✓	✓	**0.305**	**0.474**

TABLE VII: **Ablation on batch size showing average ATE (m) on AOD and HM3D.** A batch size of 60 is best on both datasets; AOD is relatively insensitive to batch size while smaller batches degrade HM3D performance.

Batch Size	AOD	HM3D
30	0.467	1.074
60	**0.305**	**0.474**
90	0.401	0.594

TABLE VIII: **Runtime Summary for AOD seq 3 (4248 frames).** The system runs at **12.92 FPS** without object segmentation and **1.56 FPS** with.

Stage	Time (s)	Share (%)
Feature extraction	35.63	1.31
Model inference	126.69	4.65
PCD building	3.87	0.14
Transition pairs	34.83	1.28
Loop closure (semantic)	31.27	1.15
Loop closure (IoU)	96.54	3.55
Object segmentation	2394.12	87.9
Optimisation	0.01	0.00
Total w/o objects	328.84	
Total w/ objects	2722.96	100.0

Loop closures further reduce error, with a smaller effect on HM3D due to fewer revisited rooms

Tab. VII reports ATE across varying batch sizes. A batch size of 60 performs best: AOD is relatively insensitive to this choice, while smaller batches degrade HM3D performance as they fail to capture sufficient coverage of larger rooms within a single pass. A batch of 60 also allows the retention of enough frames to support reliable object tracking.

Table VIII breaks down the per-stage runtime of the pipeline for an AOD sequence on a NVIDIA RTX PRO 6000, showing that object segmentation dominates the total cost at 87.9%, followed by MapAnything inference at 4.65%. The full pipeline runs at 12.92 FPS without object segmentation and at 1.56 FPS with object segmentation.

V. CONCLUSION

We presented LEXI-SG, a monocular SLAM system that tightly couples feed-forward reconstruction with semantic scene graph mapping, achieving globally consistent reconstructions and competitive semantic understanding from RGB input alone. Evaluations across pose estimation, dense reconstruction, room segmentation, and open-vocabulary object segmentation demonstrate state-of-the-art performance among monocular feed-forward SLAM methods. The two primary limitations of the system are the reduced room segmentation precision in open-plan spaces, where the absence of doorways or corridors prevents transition detection, and a pose accuracy ceiling imposed by the feed-forward model, since keyframes are not individually optimized in order to preserve local batch consistency. Future work could address the former by incorporating geometric or appearance-based cues to handle open-plan layouts.

ACKNOWLEDGMENT

The work at the University of Oxford was supported by a Royal Society University Research Fellowship (Fallon, Kassab), and the work at Seoul National University (Kim, Gil) is supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(No. RS-2024-00461409).

REFERENCES

[1] A. Werby *et al.*, "Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation," *Robot.: Sci. Syst.*, 2024.

[2] Q. Gu *et al.*, "ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.

[3] A. Rosinol *et al.*, "Kimera: from SLAM to Spatial Perception with 3D Dynamic Scene Graphs," *Int. J. Robot. Res.*, 2021.

[4] A. Takmaz *et al.*, "OpenMask3D: Open-Vocabulary 3D Instance Segmentation," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

[5] D. Maggio *et al.*, "Clio: Real-time Task-Driven Open-Set 3D Scene Graphs," *IEEE Robot. Autom. Lett.*, vol. 9, no. 10, pp. 8921–8928, 2024.

[6] C. Campos *et al.*, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.

[7] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.

[8] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, 2007.

[9] J. McCormac *et al.*, "Fusion++: Volumetric Object-Level SLAM," in *Intl. Conf. 3D Vision (3DV)*, 2018, pp. 32–41.

[10] R. F. Salas-Moreno *et al.*, "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013.

[11] J. Wang *et al.*, "VGGT: Visual Geometry Grounded Transformer," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025.

[12] V. Leroy, Y. Cabon, and J. Revaud, "Grounding Image Matching in 3D with MASt3R," in *Eur. Conf. Comput. Vis. (ECCV)*, 2024.

[13] N. Keetha *et al.*, "MapAnything: Universal Feed-Forward Metric 3D Reconstruction," in *Intl. Conf. 3D Vision (3DV)*. IEEE, 2026.

[14] R. Murai, E. Dexheimer, and A. J. Davison, "MASt3R-SLAM: Real-Time Dense SLAM with 3D Reconstruction Priors," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025.

[15] D. Maggio, H. Lim, and L. Carlone, "VGGT-SLAM: Dense RGB SLAM Optimized on the SL (4) Manifold," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 39, 2025.

[16] G. Zhang, S. Qian, X. Wang, and D. Cremers, "ViSTA-SLAM: Visual SLAM with Symmetric Two-view Association," *CoRR*, vol. abs/2509.01584, 2025.

[17] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," *Int. Conf. Mach. Learn. (ICML)*, 2021.

[18] N. Ravi *et al.*, "SAM 2: Segment anything in images and videos," in *Intl. Conf. on Learning Representations (ICLR)*, 2025.

[19] O. Siméoni *et al.*, "Dinov3," *CoRR*, vol. abs/2508.10104, 2025.

[20] S. Lu *et al.*, "Ovir-3d: Open-vocabulary 3D Instance Retrieval Without Training on 3D Data," in *Conf. on Robot Learning (CoRL)*. PMLR, 2023, pp. 1610–1620.

[21] S. Peng *et al.*, "OpenScene: 3D Scene Understanding with Open Vocabularies," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[22] K. Jatavallabhula *et al.*, "ConceptFusion: Open-set Multimodal 3D Mapping," in *Robot.: Sci. Syst.*, 2023.

[23] J. Kerr *et al.*, "LERF: Language Embedded Radiance Fields," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 19 672–19 682.

[24] O. Alama *et al.*, "RayFronts: Open-Set Semantic Ray Frontiers for Online Scene Understanding and Exploration," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2025.

[25] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Eur. Conf. Comput. Vis. (ECCV)*, 2014.

[26] K. Deng *et al.*, "VGGT-Long: Chunk it, Loop it, Align it - Pushing VGGT's Limits on Kilometer-scale Long RGB Sequences," *CoRR*, vol. abs/2507.16443, 2025.

[27] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, 2019.

[28] Y. Zhang *et al.*, "Recognize Anything: A Strong Image Tagging Model," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 1724–1732.

[29] S. Liu *et al.*, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," in *Eur. Conf. Comput. Vis. (ECCV)*, 2024, pp. 38–55.

[30] D. Maggio and L. Carlone, "VGGT-SLAM 2.0: Real-time Dense Feed-forward Scene Reconstruction," *CoRR*, vol. abs/2601.19887, 2026.

[31] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization," in *Robot.: Sci. Syst.*, 2022.

[32] J. Sturm *et al.*, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2012.

[33] S. K. Ramakrishnan *et al.*, "Habitat-Matterport 3D Dataset (HM3D): 1000 Large-scale 3D Environments for Embodied AI," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021.

[34] A. Krishnan *et al.*, "Benchmarking Egocentric Visual-Inertial SLAM at City Scale," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2025.

[35] C. Kassab *et al.*, "OpenLex3D: A New Evaluation Benchmark for Open-Vocabulary 3D Scene Representations," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2025.