RoboHop: Segment-based Topological Map Representation for Open-World Visual Navigation

Sourav Garg[1], Krishan Rana[*2], Mehdi Hosseinzadeh[*1], Lachlan Mares[*1],
Niko Sünderhauf[2], Feras Dayoub[1], Ian Reid[1,3]

***Abstract*— Mapping is crucial for spatial reasoning, planning and robot navigation. Existing approaches range from metric, which require precise geometry-based optimization, to purely topological, where image-as-node based graphs lack explicit object-level reasoning and interconnectivity. In this paper, we propose a novel topological representation of an environment based on *image segments*, which are semantically meaningful and open-vocabulary queryable, conferring several advantages over previous works based on pixel-level features. Unlike 3D scene graphs, we create a purely topological graph with segments as nodes, where edges are formed by *a)* associating segment-level descriptors between pairs of consecutive images and *b)* connecting neighboring segments within an image using their pixel centroids. This unveils a continuous sense of a 'place', defined by inter-image persistence of segments along with their intra-image neighbours. It further enables us to represent and update segment-level descriptors through neighborhood aggregation using graph convolution layers, which improves robot localization based on segment-level retrieval. Using real-world data, we show how our proposed map representation can be used to *i)* generate navigation plans in the form of *hops over segments* and *ii)* search for target objects using natural language queries describing spatial relations of objects. Furthermore, we quantitatively analyze data association at the segment level, which underpins inter-image connectivity during mapping and segment-level localization when revisiting the same place. Finally, we show preliminary trials on segment-level 'hopping' based zero-shot real-world navigation. Supplementary details can be found on our project page: `oravus.github.io/RoboHop/`.**

I. INTRODUCTION

A map of an environment represents spatial understanding which an embodied agent can use to operate in that environment. This manifests in existing approaches in multiple ways, e.g., 3D metric maps used for precise operations [1], [2], implicit maps as a robot's memory [3], hierarchical 3DSGs based explicit memory [4], and topological maps with image-level connectivity for robot navigation [5]–[8]. Metric maps enable direct spatial reasoning, e.g., 6-DoF poses of a driverless vehicle, or measuring distances to or between physical entities in the environment. Even for purely topological representations, some spatial reasoning can be encoded through image-level connectivity, e.g., recent advances in bio-inspired topological navigation [5] and the follow-up work [6], [9], [10]. However, such topological representations discretized by images are limited in their semantic expressivity as the physical entities in the world are never explicitly represented or associated across images.

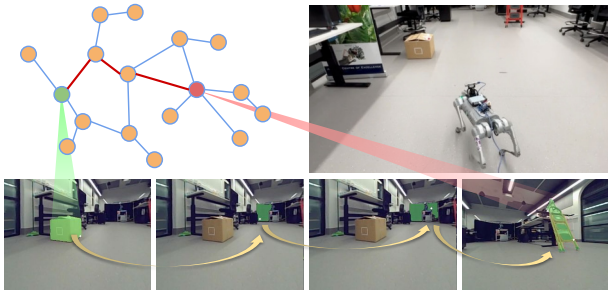

Fig. 1. We present a topological, segment-based map representation which can generate navigation plans from open-vocabulary queries in the form of 'hops' over segments to reach the target goal, without the need for a learned policy.

In this paper, we propose a novel topological representation of an environment based on *image segments*. Unlike the use of pixel-level features [11], the segments we use are semantically meaningful and open-vocabulary queryable. Our segments-based approach is enabled by recent advances in image segmentation, i.e., SAM [12] and vision-language coupling, i.e., CLIP [13]. We create a topological graph using image segments as nodes, with edges formed by *a)* associating segment-level descriptors between pairs of consecutive images and *b)* connecting neighboring segments within an image using their pixel centroids.

We show how our map representation can be used to create intra-image *hops* over inter-image *segment tracks* to generate navigation plans and actions, as shown in Figure 1. Unlike existing image-level topological navigation methods [5]–[7], the use of segments directly enables finer-grained plan generation for object-goal navigation. We additionally show how complex natural language instructions based on object relationships can be decomposed into relevant text strings using an LLM-based interface, which when combined with the spatial connectivity of our proposed map, enables semantic search of specific objects in the environment. Furthermore, we show how our proposed segment-level inter- and intra-image connectivity unveils a continuous sense of a 'place' [14], represented by a segment descriptor and its neighboring nodes. These segment descriptors are updated, enhanced and augmented with their neighbours via graph convolution. This rich descriptor enables accurate robot localization via segment-level retrieval.

[1] The University of Adelaide, Australia.
[2] Queensland University of Technology, Australia.
[3] Mohamed Bin Zayed University of Artificial Intelligence, UAE.
*Equal Contribution

In summary, the contributions of this paper are as follows:

- We introduce a novel topological representation of environments, utilizing *image segments* as nodes. This enables semantically rich and open-vocabulary queryable mapping.
- We establish a novel mechanism for intra- and inter-image connectivity based on segment-level descriptors and pixel centroids. We present quantitative analyses of the efficacy of the segment-level descriptor for data association during mapping and for localization.
- We develop a unique method for generating semantically interpretable, segment-level plans for navigation, leveraging text-based queries for defining object-level source and target nodes.
- We demonstrate the utility of our segment-level mapping, planning, and localization through preliminary trials of zero-shot real-world navigation, including LLM-based relational query decomposition.

II. Related Work

Mapping: Mapping techniques fall into three main categories: 3D metric maps [1], [15]–[18], purely topological maps [5], [19], and hybrid maps which often combine semantics with 'topometric' information, e.g., 3D Scene Graphs [20]–[23]. 3D approaches like ORB-SLAM [15], LSD-SLAM [16], and PTAM [24] excel in accuracy but suffer from computational overhead and a lack of semantics, limiting their application in high-level task planning. Hybrid methods such as SLAM++ [25] and QuadricSLAM [26] attempt to address this by incorporating semantic information but remain computationally intensive. Purely topological methods like FAB-MAP [19] and SPTM [5] simplify the computational load by using graphs to represent places and paths but lack explicit object-level connectivity.

Navigation: Semantic and spatial reasoning is crucial for object-goal navigation [27], where a robot navigates toward a specified object represented through an image or a natural language instruction. Although some works have advocated for end-to-end learning through reinforcement [28]–[30] or imitation [31], [32], these approaches often necessitate large training datasets that are impractical in real-world scenarios. A less data-hungry alternative is to segregate the task into the classical three-step process: mapping, planning and then acting. Map-based strategies have exhibited superior modularity, scalability and interpretability, thus being suitable for real-world applications [33]. LM-Nav [6] and TGSM [34] build on SPTM [5] to create topological graph representations, coupled with image-based CLIP features or closed-set object detections associated with each location. These representations can then be used to generate sub-goals which a robot can navigate towards with an image-based, low-level control policy. Learning such policies requires both environment- and embodiment-specific training data, limiting the generality of the approach. More recent work in this direction is aimed at creating foundation models for navigation [35]. However, these topological maps with images-as-nodes lack explicit object-level reasoning, unless combined with 3D input [34], [36], [37]. In our work, we present a novel topological representation with 'segments-as-nodes', which provides the robot with *segment tracks of persistent entities*, where each node in the graph is connected to the next via segment matching across images. As segments disappear from parts of an image, other segments match to the next image allowing for a continuous *hopping* over a stream of nodes. Such a representation enables a robot to progress towards a goal by "segment servoing" sub-goals, which relaxes the need for embodiment specific and sample-inefficient learned policies. Moreover, unlike existing image-based servoing [38]–[46] and visual teach-and-repeat methods [47]–[55] for navigation, our map representation is purely topological *and* based on segments [12] which are semantically meaningful and open-vocabulary queryable.

III. RoboHop

Figure 2 illustrates our proposed pipeline for *RoboHop* and its key modules: mapping, localization, planning and open-vocabulary natural language querying, as detailed below.

A. Mapping

We define a map of an environment as a topological graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ and $\mathcal{E}$ represent the nodes and edges. For a given sequence of images $I^t \in I$, we first obtain image segmentation from a method such as SAM [12] – the zero-shot capability of these recent foundation models is important because we do not want to tie our topological representation to a closed-world of known/recognised objects. Furthermore, these methods naturally support the link to richer descriptors and language models.

For each segment in an image, we define a node n_i in $\mathcal{G}$ with attributes $(x_i, y_i, \mathbf{M}_i, \mathbf{h}_i^l)$. (x_i, y_i) represent the pixel centroid of the binary mask $\mathbf{M}_i$, $\mathbf{h}_i^0$ represents the l2-normalized segment descriptor obtained by aggregating pixel-level deep features (using DINO [56]) corresponding to $\mathbf{M}_i$, and $l \in [0, l_{max}]$ is the layer index for descriptor aggregation in the graph (as explained later). As a semantic preprocessing step, we also compute CLIP [13] descriptors for individual segments (similar to [57]) and exclude the segments with high (image-language) similarity to semantic labels for 'stuff' (i.e., *floor, ceiling, and wall*).

Edges: An edge e_{ij} is defined as either of the two edge types: a) *intra-image edges*, which are defined through the centroids of segments (x_i^t, y_i^t) within each image I^t using Delaunay Triangulation and b) *inter-image edges*, which are defined through segment-level data association, i.e., vector dot product $s_{ij}^{t,t+1} = \mathbf{h}_i^t \cdot \mathbf{h}_j^{t+1}$ between node descriptors of consecutive images (I^t, I^{t+1}) as follows:

$$\mathcal{E}^{t,t+1} = \{(n_i^t, n_j^{t+1}) \,|\, n_j^{t+1} = \underset{k}{\operatorname{argmax}}\, s_{ik}^{t,t+1} \wedge s_{ij}^{t,t+1} > \theta\} \tag{1}$$

where an edge between a pair of segment nodes (n_i^t, n_j^{t+1}) only exists if n_j^{t+1} is the closest match for n_i^t and their similarity is greater than a threshold θ. If no edge is found for a consecutive image pair, we retain a single edge with the highest similarity value. This ensures that our map is a

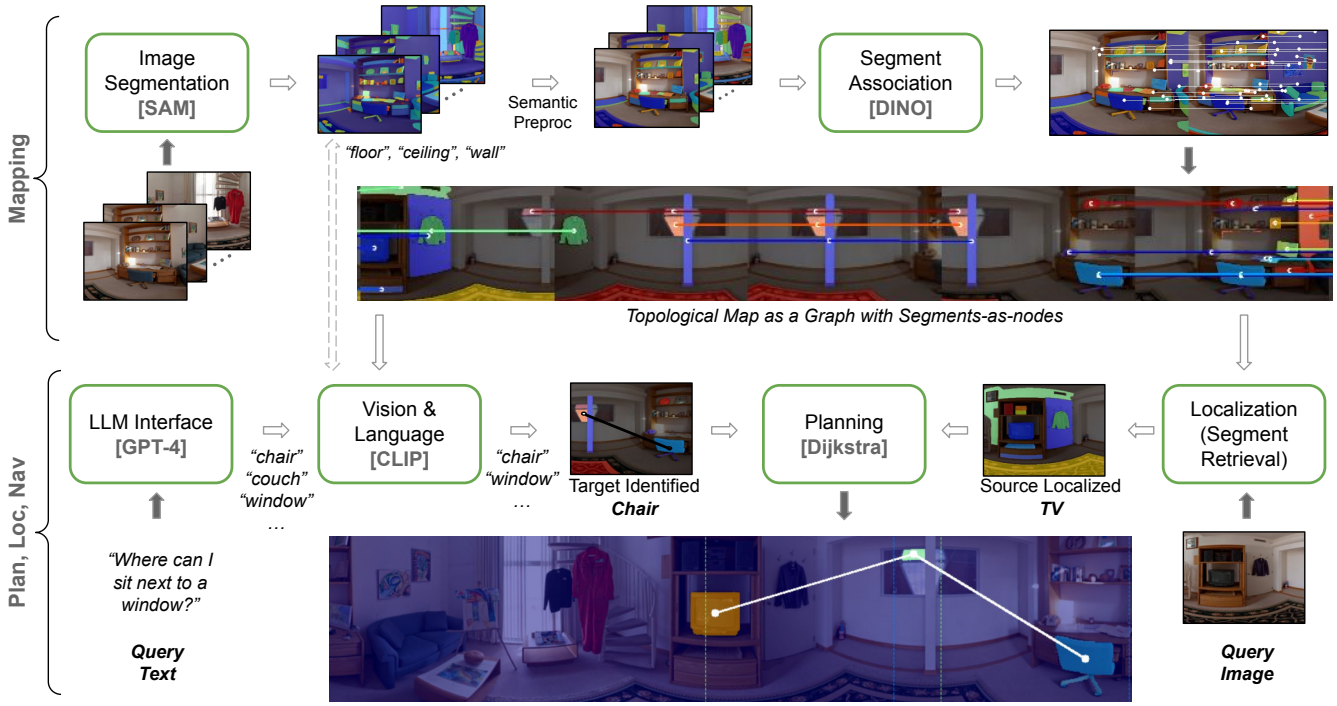

Fig. 2. Illustration of our overall pipeline from image segments to mapping, language querying, and planning.

connected graph. In this work, we have not defined inter-image edges between non-consecutive frames, which can be used to further enhance the map through loop closures.

Node Descriptor & Aggregation: The nodes in our map are based on segments which represent semantically meaningful entities in the environment. By defining a segment descriptor for each node based on robust features such as DINO [56] (e.g., see AnyLoc [58]), these segments can be considered as unique landmarks. Thus, from a 'place descriptor' and localization perspective, these segments do not necessarily need to be interpretable as "objects". However, a standalone image segment descriptor $\mathbf{h}_i$ might suffer from *perceptual aliasing* during the localizaton phase. To alleviate this, we add more *place* context to a node from its neighborhood by aggregating descriptors through multi-layered graph convolutions. This is achieved by simplifying the standard graph convolution network [59] to compute average node descriptors as below:

$$\mathbf{H}^{(l+1)} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{I} \tag{2}$$

where $\mathbf{H}$ is the node descriptor matrix (composed of $\mathbf{h}$), A is the adjacency matrix for $\mathcal{G}$, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, $\mathbf{I}$ is the identity matrix and $\tilde{\mathbf{D}}$ is the degree matrix for $\tilde{\mathbf{A}}$. Here, aggregation over successive layers influences a node descriptor through the neighbors of its neighbors, thus gradually expanding the 'place' context of any given node. After the mapping phase, we perform this aggregation on both the map and the query image, using $l_{max} = 2$.

B. Localization

In our proposed map with segments-as-nodes, we define localization at the node level through node retrieval. For each of the segment descriptors in the query image, we match it with all the segment nodes in the map and consider it localized if its similarity is greater than a threshold. Although more sophisticated retrieval methods are available, we found that the richness of the descriptor, together with a simple threshold, provided high-quality retrieval. These segment descriptors are informed by their neighbours (see Eq. 2), which improves their localization ability due to the added 'place' context.

C. Planning for Navigation

Through the interconnectivity of segments, we aim to obtain navigation plans from our map in the form of *segment tracks* with continuous *hopping* from one track to another, as these segments exit and enter the field of view.

1) Edge Weighting: Given the source and destination segment nodes in our proposed map, we generate a plan using Dijkstra algorithm, where the edge weights are set to 0 and 1 respectively for inter- and intra-image edges. This specific design choice is what encourages the *shortest path* search to always prefer edge connections *across* images. It leads to the emergence of *segment tracks* of *persistent entities* that a robot can use as navigation sub-goals, where continuous hopping across the sub-goals of the navigation plan leads to the final destination. We use these edge weights only for generating navigation plans, not for node descriptor aggregation.

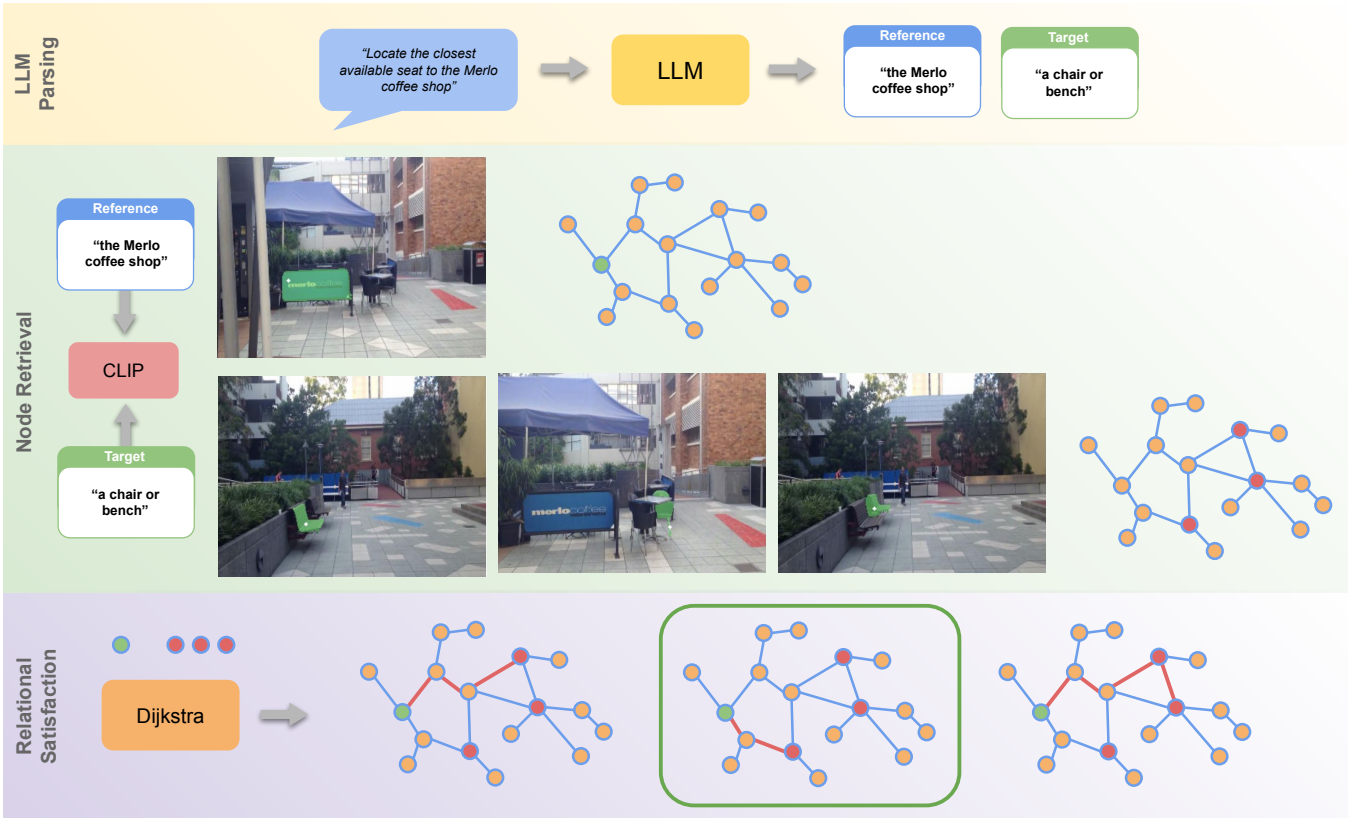

Fig. 3. *Target Object Search Based on Relational Natural Language Queries*: The LLM parses a relational query into a *reference* and *target* node textual description suitable for CLIP to process into language feature vectors. We then retrieve top-3 candidate *target* and *reference* nodes from the map by respectively matching the CLIP language feature vector with the CLIP vision feature vector of each node. Within the topological graph of our map, Dijkstra's algorithm finally selects the object goal for navigation based on the shortest path between the candidate *target* and *reference* nodes.

2) Planning Strategy: There exist many different methods [5], [6], [9], [10], [36] for local motion control that operate on the pair of current observation and sub-goal to generate actions. Since the exact form of input to such controllers, as well as the exact end-task specifications can potentially vary [9], [27], [60], [61], we define two variants of segment-level plan generation depending on how the intra-image edges are connected. The default mode is to use Delaunay Triangulation (as described in Section III-A), which we refer to as *Intra-DT* for planning purposes. With intra-image edge weights as 1, this mode will only ever traverse multiple intra-image neighboring segments when it is able to reach a node that has long inter-image tracks, thus saving the overall path cost. This type of planning can be directly useful for 'smooth' robot control as there are no intra-image 'long hops'. We also consider an alternative mode of planning, dubbed *Intra-All*, where we create a complete subgraph using all the segments *within* a single image, thus allowing long intra-image hops. This mode of planning can be useful when there is a large number of objects in a single image (e.g., a shelf full of items) which will otherwise incur a high cost for moving from one corner of the image to another. In Section IV-B, we show how these different planning strategies lead to variations in the choice of persistent segment tracks.

D. Querying the Map with Open Vocabulary

We demonstrate one potential use case of our map representation for object-goal navigation based on object-level *relational* queries. We associate each node in our map with a CLIP descriptor of the corresponding image segment, thereby offering an interface for open-vocabulary, natural language queries entailing vague and complex task instructions. More importantly, we introduce an algorithm (see Figure 3) that enables path plans generated from complex *relational* queries such as *"locate the closest available seat to the Merlo's coffee shop"* which exploits the map's ability to capture both intra- and inter-image spatial relationships not present in existing methods. The key here is to identify the *target* ("chairs or benches") and the *reference* (to that target, i.e., "the Merlo coffee shop") nodes in the scene based on the relational query. We do this by utilising an LLM appropriately prompted to parse the query and identify textual descriptions of these nodes-of-interest. This does not require the LLM to be aware of the map. Across all experiments in the work, we leverage `GPT-4` as the underlying LLM. The parsed text descriptions of *reference* and *target* are processed into language feature vectors by CLIP's text encoder. We then retrieve top-3 candidate *target* and *reference* nodes from the

TABLE I

ACCURACY OF SEGMENT-LEVEL OBJECT RECOGNITION.

	CLIP [13]	DINO [56]
Object Instance Recognition	35.11%	56.43%
Object Category Recognition	62.87%	79.04%

map by respectively matching the CLIP language feature vector with the CLIP vision feature vector of each node. Within the topological graph of our map, Dijkstra's algorithm finally selects the object goal for navigation based on the shortest path between the candidate *target* and *reference* nodes.

IV. EXPERIMENTS AND RESULTS

This section details our experimental design and results, aimed at validating the proposed topological map representation by answering the following key questions[1]:

- How effective is our segment-level data association in both indoor and outdoor environments?
- Can our method accurately localize a robot using segment-level topological information?
- How can we effectively generate segment-level plans for *hopping* based navigation?

A. Segment-Level Data Association

As the quality of segment-level data association lies at the heart of the robustness and integrity of our mapping, as well as for the plans made within these maps, we conduct experiments to evaluate the efficacy of the data association component of our pipeline. Our method is simple but backed by rich descriptors based on local and broader contextual information. We consider two kinds of experiments on real-world data, which are outlined in more detail below. In the first set of experiments, the ground truth segments and instances are available indoors, such as GibsonEnv [62], This availability allows us to perform quantitative evaluation of segment-level association. However, in the second set of experiments, the lack of similar ground truth data outdoors means that we must resort to evaluating a downstream task – localisation – to assess its performance based on our segment correspondences.

1) Object Instance and Category Recognition: In this experiment, to demonstrate the efficacy of our segment-level association, we make use of ground truth detections and segmentation of instances in an indoor environment: GibsonEnv [62]. In particular, we show here examples from the house `Klickitat` as it is representative of the diverse range of environments in the dataset. To align with the standard input requirements of SAM, and to "simulate" a forward-facing camera, we extract perspective images with a field-of-view of 120 degrees from the real-world GibsonEnv panoramas and treat these as the raw images. Next, we obtain class-agnostic SAM segments from each image and assign these segments to their corresponding ground truth object instances in each image using Intersection over Union (IoU), with a minimum threshold of 0.2. To ensure data quality, we consistently exclude segments with sizes comprising less than 0.2% of the overall image. Finally, for this experiment, we have a total of 544 distinct views (SAM segments) of 68 unique objects from 18 diverse categories. We assess the quality of descriptors (such as DINO [56] and CLIP [13]) for segment-level association by evaluating the (top-1) accuracy of our descriptor matching with the correct object. As explained in Section III-A, the matches are selected based on the nearest neighbour criterion over descriptors.

Fig. 4. Object Instance Recognition in GibsonEnv [62]: The columns show segment masks (in green) for the query, DINO match, and CLIP match respectively. Symbols (✓/×) adjacent to images indicate success or failure in association. The final row illustrates category-level recognition success despite both methods failing at the instance level (multiple chairs in close proximity).

Table I shows a comparative analysis of different descriptors for object instance and category recognition from diverse viewpoints. It is apparent that DINO achieves better results than CLIP in this context, which can be attributed to differences in how they are supervised and their training objectives. While CLIP performs reasonably well in predicting categories, DINO features exhibit greater distinctiveness in both instance-level and category-level recognition. In Figure 4, we show some of the object instance and category recognition outcomes, featuring both successful and unsuccessful cases.

2) Segment-level Topological Localization: Since segment- or object instance-level ground truth associations are not always available, we also conduct experiments to measure the quality of both our map and the localization ability through a segment-level topological localization task. For this purpose, we use a popular visual place recognition dataset, GPCampus [63], which comprises three traverses of a University Campus: two day and one night time. We only use its Day Left and Day Right traverse as the reference map and query set respectively. We coarsely evaluate segment-level association by first tagging both the query segment and its matched segment to their respective image indices, and then using these associated images to compute

[1]Additional implementation details for image preprocessing and models (i.e., SAM [12], DINO [56]), and CLIP [13]) are in the supplementary.

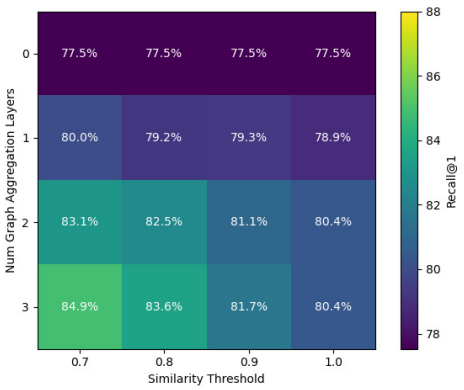

Fig. 5. Node-level localization (GPCampus Dataset) across varying number of graph convolutional layers (y-axis) and incremental inclusion of inter-image edges based on a similarity threshold (x-axis).

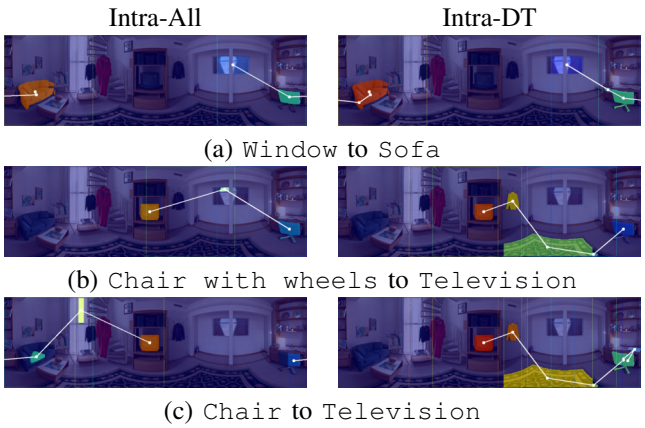

Fig. 6. Segment-level plans using text queries for source and target, showing shortest paths for panoramic 'pure rotation'.

Recall@1 based on a localization radius of 5 frames.

In Figure 5, we show how the recognition performance at the segment-level improves both with an increasing number of graph convolution layers and incremental inclusion of inter-image edges. The former only considers segments from within an image while the latter resembles sequential descriptor-type place recognition [64].

B. Planning

We show qualitative results of our full pipeline using two complementary datasets. a) *PanoContext-Living*, which refers to one of the living room panoramic images (`2cfc836333`) from the original PanoContext dataset [65], [66]. We split this pano image uniformly along the horizontal axis to create multiple frames, with a horizontal wraparound. Thus, this dataset represents a pure rotation-based robot traversal. We explicitly compute data association between the last and the first frame to close the loop. b) *GPCampus-DayLeft* [63], which is a forward-moving robot traverse using a front-facing camera.

For both these datasets, we first construct the segment-level map, then query the resultant graph with text to identify source and target node based on CLIP similarity, and then finally generate a plan between these pairs of nodes.

1) PanoContext-Living: Figure 6 shows multiple plans using a variety of text queries for both types of planning strategies: Intra-All and Intra-DT. Each of the selected segments and their connectivity based on the shortest path is shown, with path edges wrapped around the pano image. The subsampled frames from the pano are shown as dashed boxes in color corresponding to the segment belonging to that frame.

Intra-All: For Intra-All planning on this pure rotation setting, the inferred shortest path can be coarsely related to the horizontal offset (allowing wraparound) between the pixel centroids of the source and the target segment. In Figure 6(a) (Intra-All), for text queries `Window` (source) and `Sofa` (target), the shortest path is correctly found from the wraparound frames via `Chair`. In examples (b) and (c), we extract paths to `Television` from `Chair with wheels` and `Chair`. Indicating imperfections of the SAM+CLIP combination, `Chair` finds the best match with one of its partial visual observation, in contrast to `Chair with wheels` which matches correctly with the full chair. Nevertheless, both the paths in (b) and (c) are practically similar in terms of the number of yaw steps needed to reach the target.

Intra-DT: For the Intra-DT plans, in all the cases, paths span multiple objects (more than the Intra-All), inducing a smoother transition from source to target. In examples (b) and (c), the paths are composed of the carpet nodes – this consistent choice is justified from an almost 'omnipresence' of carpet throughout the scene, as it had not been filtered out in our preprocessing of common segments. Thus, in both the cases, intra-image hops try to land on to the carpet node to reach the target with the least inferred cost.

2) GPCampus-DayLeft: In Figure 7, we show the segment-level plan for the forward-moving robot traverse, with `Z block` and `Dustbin` as the source and target text queries. Here, we only show the planned segments close to the source node, please refer to the supplementary video for the full plan visualization. The first two columns correspond to the Intra-DT and Intra-All planning, and the last column corresponds to a naive baseline where an inter-image edge for each of the segments is included without any similarity thresholding (see Eq. 1). This implies that during planning there always exists a 0 cost inter-image edge for all the segments, thus never needing to traverse an intra-image edge. In the Intra-DT column, the first 4 frames (rows) show an intra-image traversal to reach the `door` which has a persistent track over multiple frames. In the Intra-All column, it can be observed that a single intra-image hop directly leads to a persistent track of a `closet`. In the DA-All column, the paths are formed based on rapid hopping, as soon as the current tracked object goes out of the field-of-view, regardless of any persistent segment tracks.

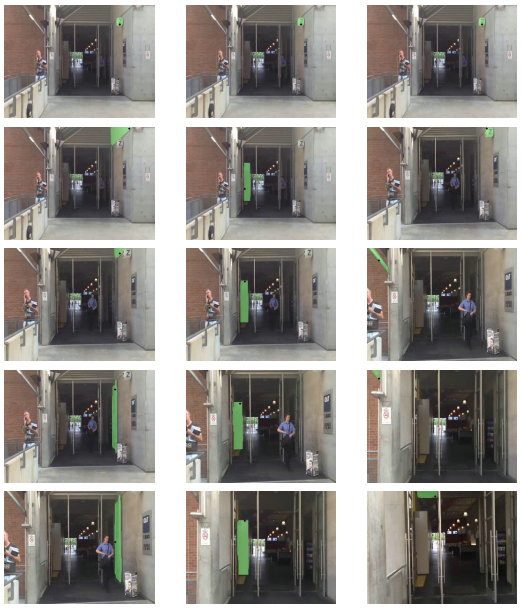

Fig. 7. Variations in segment-level navigation plans (one per column) depending on how the edges are defined and weighted for path search.

C. Navigation

We conducted preliminary trials of real-world zero-shot robot navigation using segment-level mapping and planning. We initialize the robot pose such that the first reference map node (sub-goal) of the plan is in its field of view. This reference node's segment descriptor is matched with all the segment descriptors in the current robot observation. The similarity value of the best match determines whether the robot is in the 'lost state' (i.e., unable to localize with respect to the reference node, thus explore randomly) or 'track state'. For the latter case, we use the horizontal pixel offset of the best match from the image center, and the segment size ratio between the match and the reference to determine a 'hop state'. This state implies that the robot has successfully tracked and reached to the reference sub-goal, and can hop on to the next node in the plan and repeat the process until it reaches the last node in the plan. We use PID controller to convert the horizontal pixel offset into yaw velocity, while the forward translation is always fixed to a small velocity. Our trials (please see supplementary video and a snapshot in Figure 1) show that our proposed representation, powered by the foundation models SAM and DINO, enables embodiment-agnostic control strategies for zero-shot goal-directed navigation without needing to train data-hungry task-specific policies.

V. Limitations

While our approach exhibits notable strengths in segment-level topological mapping and planning for spatial reasoning and navigation, it also has several limitations worth discussing. *a)* The efficacy of our approach is strongly tied to the quality of segment-level data association. We observed failures in navigation trials due to mismatches caused by repetitive structures, e.g., windows. *b)* Our method in its current form cannot deal with dynamic changes in the environment. *c)* Although we conducted comparative analyses to select the most robust descriptors, such as DINO, our approach's performance is still bounded by the quality and limitations of these descriptors. For instance, DINO's instance-level recognition superiority might not hold in all contexts. *d)* Considering 'things' vs 'stuff', despite the convenience of semantic preprocessing enabled by the combination of SAM and CLIP to remove 'stuff', some segments from ground or walls can still persist. *e)* For our robot experiments, we relied on *offboard* compute for inference of large models (SAM, DINO and CLIP). *f)* Finally, we note that handling relational queries through LLMs is prone to failures in cases where metric information is necessary to deem two objects being next to each other.

VI. Conclusion and Future Work

This paper presented a novel topological map representation centred on *image segments*, which serve as semantically-rich, open-vocabulary queryable nodes within a topological graph. The method uses an integrated strategy involving segment-level data association and segment-level planning for object-goal navigation. Our preliminary trials on segment-level *hopping* based navigation indicate that powerful foundation models like SAM (for segmentation) and DINO (for data association) can enable zero-shot navigation without requiring 3D maps, image poses or a learnt policy.

There are several promising directions for future work. One avenue involves incorporating visual servoing-based navigation to provide real-time visual feedback, which could improve the system's navigation capabilities and robustness. Furthermore, while our current approach predominantly relies on topological mapping, integrating *local* node- and edge-level metric information can introduce a higher degree of granularity and precision, thereby enhancing the system's navigation capabilities. Finally, semantically labelling each node could facilitate the construction of 3D scene graph representations suitable for higher-level task planning [67].

References

[1] K. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, J. Tenenbaum, C. de Melo, M. Krishna, L. Paull, F. Shkurti, and A. Torralba, "Conceptfusion: Open-set multimodal 3d mapping," in *RSS*, 2023.

[2] P.-E. Sarlin, M. Dusmanu, J. L. Schönberger, P. Speciale, L. Gruber, V. Larsson, O. Miksik, and M. Pollefeys, "Lamar: Benchmarking localization and mapping for augmented reality," in *European Conference on Computer Vision*. Springer, 2022, pp. 686–704.

[3] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *Conference on Robot Learning*. PMLR, 2023, pp. 2226–2240.

[4] Z. Ravichandran, L. Peng, N. Hughes, J. D. Griffith, and L. Carlone, "Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9272–9279.

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

[6] D. Shah, B. Osinski, B. Ichter, and S. Levine, "LM-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=UW5A3SweAH

[7] K. Chen, J. P. de Vicente, G. Sepulveda, F. Xia, A. Soto, M. Vazquez, and S. Savarese, "A behavioral approach to visual navigation with graph localization networks," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.

[8] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological slam for visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 875–12 884.

[9] Y. Li and J. Košecka, "Learning view and target invariant visual servoing for navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 658–664.

[10] X. Meng, N. Ratliff, Y. Xiang, and D. Fox, "Scaling local control to large-scale topological navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 672–678.

[11] E. Johns and G.-Z. Yang, "Global localization in a dense continuous topological map," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1032–1037.

[12] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[14] S. Garg, T. Fischer, and M. Milford, "Where is your place, visual place recognition?" in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*. International Joint Conferences on Artificial Intelligence, 2021, pp. 4416–4425.

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

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

[17] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[18] R. Dube, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "Segmap: Segment-based mapping and localization using data-driven descriptors," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 339–355, 2020.

[19] M. J. Cummins and P. M. Newman, "Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 3–10.

[20] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From slam to spatial perception with 3d dynamic scene graphs," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1510–1546, 2021.

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

[22] U.-H. Kim, J.-M. Park, T.-J. Song, and J.-H. Kim, "3-D scene graph: A sparse and semantic representation of physical environments for intelligent agents," *IEEE transactions on cybernetics*, vol. 50, no. 12, pp. 4921–4933, 2019.

[23] P. Gay, J. Stuart, and A. Del Bue, "Visual graphs from motion (vgfm): Scene understanding with object geometry reasoning," in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer, 2019, pp. 330–346.

[24] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234.

[25] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1352–1359.

[26] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE Robotics and Automation Letters*, 2019.

[27] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[28] T. Chen, S. Gupta, and A. Gupta, "Learning exploration policies for navigation," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/pdf?id=SyMWn05F7

[29] A. Wahid, A. Stone, K. Chen, B. Ichter, and A. Toshev, "Learning object-conditioned exploration using distributed soft actor critic," in *Conference on Robot Learning*. PMLR, 2021, pp. 1684–1695.

[30] J. Bruce, N. Sünderhauf, P. Deepmind, London, R. Deepmind, and M. Milford, *Learning Deployable Navigation Policies at Kilometer Scale from a Single Traversal*. [Online]. Available: http://proceedings.mlr.press/v87/bruce18a/bruce18a.pdf

[31] Y. Lee, A. Szot, S.-H. Sun, and J. J. Lim, "Generalizable imitation learning from observation via inferring goal proximity," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=lp9foO8AFoD

[32] R. Ramrakhya, D. Batra, E. Wijmans, and A. Das, "Pirlnav: Pretraining with imitation and rl finetuning for objectnav," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2023. [Online]. Available: http://dx.doi.org/10.1109/CVPR52729.2023.01716

[33] J. Chen, G. Li, S. Kumar, B. Ghanem, and F. Yu, "How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers," 2023.

[34] N. Kim, O. Kwon, H. Yoo, Y. Choi, J. Park, and S. Oh, "Topological Semantic Graph Memory for Image Goal Navigation," in *CoRL*, 2022.

[35] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A large-scale, multi-task visual navigation backbone with cross-robot generalization," in *7th Annual Conference on Robot Learning*, 2023.

[36] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

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

[38] S. Feng, Z. Wu, Y. Zhao, and P. A. Vela, "Trajectory servoing: Image-based trajectory tracking using slam." *CoRR*, 2021.

[39] S. R. Bista, P. R. Giordano, and F. Chaumette, "Appearance-based indoor navigation by ibvs using line segments," *IEEE robotics and automation letters*, vol. 1, no. 1, pp. 423–430, 2016.

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

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

[42] A. Cherubini, F. Chaumette, and G. Oriolo, "Visual servoing for path reaching with nonholonomic robots," *Robotica*, vol. 29, no. 7, pp. 1037–1048, 2011.

[43] A. Ahmadi, L. Nardi, N. Chebrolu, and C. Stachniss, "Visual servoing-based navigation for monitoring row-crop fields," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4920–4926.

[44] A. Remazeilles, F. Chaumette, and P. Gros, "3d navigation based on a visual memory," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 2719–2725.

[45] A. Diosi, S. Segvic, A. Remazeilles, and F. Chaumette, "Experimental evaluation of autonomous driving based on visual memory and image-based visual servoing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 870–883, 2011.

[46] G. Blanc, Y. Mezouar, and P. Martinet, "Indoor navigation of a wheeled mobile robot along visual routes," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 3354–3359.

[47] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of field robotics*, vol. 27, no. 5, pp. 534–560, 2010.

[48] S. Šegvić, A. Remazeilles, A. Diosi, and F. Chaumette, "A mapping and localization framework for scalable appearance-based navigation,"

Computer Vision and Image Understanding, vol. 113, no. 2, pp. 172–187, 2009.

[49] A. M. Zhang and L. Kleeman, "Robust appearance based visual route following for navigation in large-scale outdoor environments," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 331–356, 2009.

[50] D. Dall'Osto, T. Fischer, and M. Milford, "Fast and robust bio-inspired teach and repeat navigation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 500–507.

[51] M. Mattamala, N. Chebrolu, and M. Fallon, "An efficient locally reactive controller for safe navigation in visual teach and repeat missions," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2353–2360, 2022.

[52] T. Krajník, F. Majer, L. Halodová, and T. Vintr, "Navigation without localisation: reliable teach and repeat based on the convergence theorem," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1657–1664.

[53] L. Halodová, E. Dvořráková, F. Majer, T. Vintr, O. M. Mozos, F. Dayoub, and T. Krajník, "Predictive and adaptive maps for long-term visual navigation in changing environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7033–7039.

[54] T. Do, L. C. Carrillo-Arce, and S. I. Roumeliotis, "High-speed autonomous quadrotor navigation through visual and inertial paths," *The International Journal of Robotics Research*, vol. 38, no. 4, pp. 486–504, 2019.

[55] T. Krajník, P. Cristóforis, K. Kusumam, P. Neubert, and T. Duckett, "Image features for visual teach-and-repeat navigation in changing environments," *Robotics and Autonomous Systems*, vol. 88, pp. 127–141, 2017.

[56] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[57] A. Maalouf, N. Jadhav, K. M. Jatavallabhula, M. Chahine, D. M. Vogt, R. J. Wood, A. Torralba, and D. Rus, "Follow anything: Open-set detection, tracking, and following in real-time," *arXiv preprint arXiv:2308.05737*, 2023.

[58] N. Keetha, A. Mishra, J. Karhade, K. M. Jatavallabhula, S. Scherer, M. Krishna, and S. Garg, "Anyloc: Towards universal visual place recognition," *arXiv preprint arXiv:2308.00688*, 2023.

[59] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[60] J. Wasserman, K. Yadav, G. Chowdhary, A. Gupta, and U. Jain, "Last-mile embodied visual navigation," in *Conference on Robot Learning*. PMLR, 2023, pp. 666–678.

[61] H. Wang, W. Liang, L. V. Gool, and W. Wang, "Towards versatile embodied navigation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 858–36 874, 2022.

[62] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: real-world perception for embodied agents," in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[63] A. Glover, "Day and night, left and right," Mar. 2014. [Online]. Available: https://doi.org/10.5281/zenodo.4590133

[64] S. Garg and M. Milford, "Seqnet: Learning descriptors for sequence-based hierarchical place recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4305–4312, 2021.

[65] Y. Zhang, S. Song, P. Tan, and J. Xiao, "Panocontext: A whole-room 3d context model for panoramic scene understanding," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*. Springer, 2014, pp. 668–686.

[66] C. Zou, A. Colburn, Q. Shan, and D. Hoiem, "Layoutnet: Reconstructing the 3d room layout from a single rgb image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2051–2059.

[67] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable task planning," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=wMpOMO0Ss7a