Semantic Mapping in Indoor Embodied AI – A Survey on Advances, Challenges, and Future Directions

Anonymous authors
Paper under double-blind review

Abstract

Intelligent embodied agents (e.g. robots) need to perform complex semantic tasks in unfamiliar environments. Among many skills that the agents need to possess, building and maintaining a semantic map of the environment is most crucial in long-horizon tasks. A semantic map captures information about the environment in a structured way, allowing the agent to reference it for advanced reasoning throughout the task. While existing surveys in embodied AI focus on general advancements or specific tasks like navigation and manipulation, this paper provides a comprehensive review of semantic map-building approaches in embodied AI, specifically for indoor navigation. We categorize these approaches based on their structural representation (*spatial* grids, *topological* graphs, dense *point-clouds* or *hybrid* maps) and the type of information they encode (*implicit* features or *explicit* environmental data). We also explore the strengths and limitations of the map building techniques, highlight current challenges, and propose future research directions. We identify that the field is moving towards developing open-vocabulary, queryable, task-agnostic map representations, while high memory demands and computational inefficiency still remaining to be open challenges. This survey aims to guide current and future researchers in advancing semantic mapping techniques for embodied AI systems.

1 Introduction

Over the past few years there has been a growing interest across computer vision, natural language and the robotics community in embodied AI – where we study how intelligent agents with an embodiment (e.g. robots, autonomous vehicles) learn to perform tasks through interaction with its environment (Deitke et al., 2022; Puig et al., 2023; Batra et al., 2020a). This paves the way towards creating service robots (Hawes et al., 2017; Khandelwal et al., 2017; Veloso et al., 2015) that can be safely deployed in familiar environments, co-existing with humans and performing various tasks autonomously. The key difference between embodied AI and robotics is that the former focuses on building robot intelligence by interacting with simulated physical world while abstracting most of the low-level control including noisy sensors and actuators. This allows the embodied AI researchers to focus on complex semantic challenges such as object search in unfamiliar environments based on natural language instructions, spatial reasoning, multi-agent systems, interactive object search and much more. Similar to robotics, this requires the agent to possess a blend of sensorimotor skills, understanding of the environment and decision-making abilities, thus enabling them to navigate, manipulate objects and perform complex tasks in the world. Imagine a robot is tasked with finding an object in an unseen environment – "a red and blue striped zebra toy in the nursery". This is a complex task which requires the robot to have multiple skills – visual perception to identify the toy (Matthias Minderer, 2023), natural language understanding to make sense of the given language instruction (Anderson et al., 2018b), navigation to move to the nursery (Yadav et al., 2022), maintaining a semantic map to remember where the zebra toy was if it has seen it already (Raychaudhuri et al., 2023), reasoning and planning to take actions in order to complete the task (Gordon et al., 2018), and so on.

Among these skills, progressively building and maintaining a map of the world has been found to be especially crucial when the robot is in unfamiliar environments such as in autonomous driving (Bao et al., 2023), search-and-rescue operations (Gautham et al., 2023), automated vacuum cleaning robots (Singh et al., 2023) and others. An accurate map of the environment allows the agent to make informed decisions, handle unexpected situations and perform a complex task autonomously without human supervision. The significance of mapping in robotics navigation methods is motivated by findings in cognitive science (Tolman, 1948; Trullier & Meyer, 2000) which shows that humans and animals create an internal representation of their surroundings in the form of 'cognitive maps' or 'mental maps' to aid spatial cognition and form navigation strategies. Moreover such cognitive maps allow them to remember and recall the location of objects and places in the environment. Following this, most robotics navigation models are map-based (Kostavelis & Gasteratos, 2015; Filliat & Meyer, 2003; Meyer & Filliat, 2003; Song et al., 2024) and can be broken down into three processes – (a) mapping or memorizing appropriate representation of the environment, (b) localization or determining the current position of the agent on the map, and (c) path planning or choosing a set of actions that lead to the goal, given the map and the current location. While path planning is dependent on the first two, mapping and localization are also dependent on each other. In other words, estimating the current agent position depends on the map and a map can be built once the agent knows where it is located. This problem has been studied traditionally in robotics as *simultaneous localization and mapping (SLAM)*, which is the problem of mapping an unknown environment and at the same time estimating a robot's pose within it. The robotics community has seen tremendous progress in SLAM approaches which do not have to rely on external reference systems like GPS and instead use the onboard sensors to map a complex environment. This is particularly useful in indoor spaces where GPS is not available. Localization and path planning are beyond the scope of this survey. We instead focus on building maps progressively and memorizing features of an unknown environment while the agent navigates.

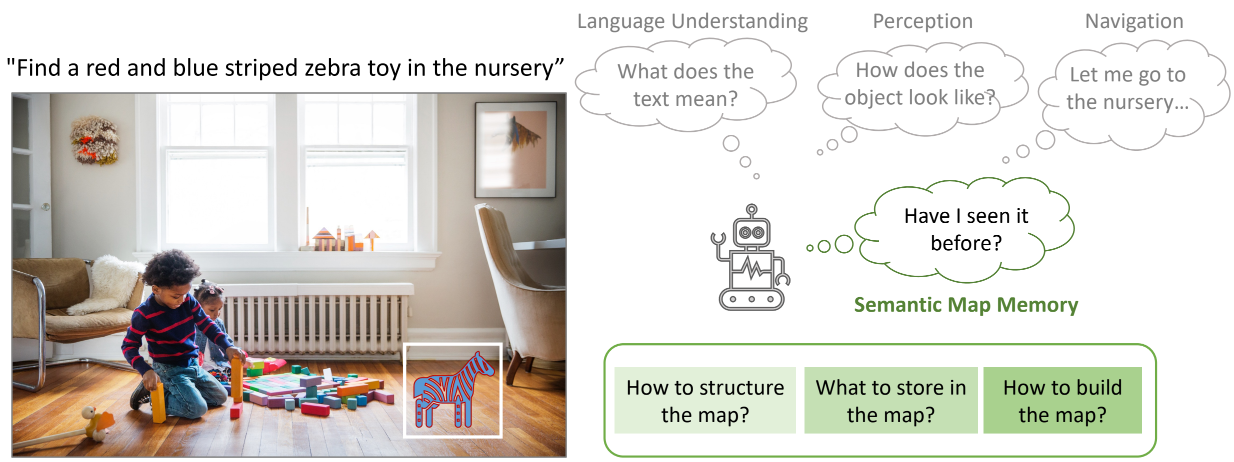

Figure 1: **Semantic mapping.** To perform a complex task in an indoor environment, the robotic agent must possess multiple skills of language understanding, visual perception, navigation, etc. Among these the most crucial is building and maintaining a semantic map of the environment so that it can come back to it while performing the task.

In addition to studying how to build intelligent agents, research in robotics has to consider various low-level aspects (low-level path planning and control, hardware sensors, robot hardware, etc.). In contrast, embodied AI research can focus more on high-level task planning by abstracting out the low-level details. This has led embodied AI researchers to explore map building techniques as part of the high-level task planning and address questions such as 'is mapping even necessary' (Partsey et al., 2022), 'what should be the structure of the map', 'what type of information to store inside the map', and 'which type of information is useful for what tasks'. There has also been a recent shift in focus to build general-purpose AI solutions by leveraging foundation models (Radford et al., 2021; Oquab et al., 2023; OpenAI, 2023) which has allowed the community to explore building general-purpose, open-vocabulary semantic maps independent of the downstream tasks. These open-vocabulary maps can be later queried using natural language (Gu et al., 2023; Peng et al., 2023; Chen et al., 2023a) or images.

Unlike existing surveys in embodied AI, which often focus on general task advancements (Pfeifer & Iida, 2004; Duan et al., 2022; Deitke et al., 2022), or specific sub-fields like visual navigation (Zhu et al., 2021; Zhang et al., 2022; Wu et al., 2024; Lin et al., 2024) or manipulation (Batra et al., 2020a; Zheng et al., 2024), this paper offers the first comprehensive review of semantic map-building techniques tailored to indoor environments in embodied AI. We organize existing approaches along two key axes – how the maps are *structured* and what types of information or *encoding* is stored inside them. By analyzing the advantages and limitations of various methods, we identify gaps in the current literature and propose future research directions to guide the community. This survey aims to unify disparate approaches to semantic mapping in embodied AI, shedding light on its foundational role in enabling intelligent behavior. Beyond merely summarizing existing work, we hope to inspire new research that pushes the boundaries of semantic map building in embodied agents. To this end, we begin with a background on embodied AI and SLAM in Sec. 2, before delving into key questions and challenges of semantic mapping in Sec. 3. We then explore map structures (Sec. 4) and encoding techniques (Sec. 5). Finally, we discuss applications and evaluation techniques in Sec. 6 and conclude with insights into future directions for this rapidly evolving field (Sec. 7). Note that semantic mapping in robotics (real world robots) is out of scope for this survey and we direct the readers to Thrun (2003); Kostavelis & Gasteratos (2015); Lluvia et al. (2021); Racinskis et al. (2023); Sousa et al. (2023) for a comprehensive reading. That said, in this survey we provide comparisons to similar mapping techniques from robotics, wherever applicable.

2 Background Reading

In this section we provide a brief overview of different types of embodied AI tasks (Sec. 2.1) and various approaches towards solving them, including end-to-end (Sec. 2.2) and modular (Sec. 2.3) approaches. As a full survey on Embodied AI tasks is beyond the scope of this survey, please see (Deitke et al., 2022) for a more detailed survey on the tasks and their current state of research. In Sec. 2.4, we discuss classical SLAM based techniques. This discussion aims to provide a background to the readers about the general advance in the embodied AI research and its connection to traditional SLAM-based techniques before we dive in to discuss semantic mapping.

2.1 Embodied AI tasks

Embodied AI tasks vary depending on the type interaction of an agent with its environment. Broadly, we can group embodied tasks into three groups – *Exploration* task (Chaplot et al., 2019) requires an agent to efficiently explore its environments; *Navigation* task (Wijmans et al., 2019; Batra et al., 2020b) requires the agent to take actions in order to move around the environment; *Manipulation* task (Szot et al., 2021; Weihs et al., 2021) requires the agent to perform interactive actions to change the state of other objects in the environment.

The taxonomy of tasks can be further broken down by the target specification provided to the agent, which impacts the information that need to be retained. For instance, for navigation, the following are commonly studied. In *Point-Goal Navigation* (Wijmans et al., 2019) (PointNav), the agent is given a target coordinate relative to its starting position, whereas in *Image-Goal Navigation* (ImageNav) it is given a target image (Chaplot et al., 2020b). In the *Object-Goal Navigation* (ObjectNav) task, the agent needs to navigate to any instance of an object category (Yadav et al., 2022). An extension to the ObjectNav task is the *Multi-Object Navigation* (MultiON) (Wani et al., 2020) task where the agent is required to navigate to multiple objects in a particular sequence. *Vision-and-Language Navigation* (VLN) (Anderson et al., 2018b) requires the agent to find the target as specified by a natural language instruction. In *Audio-Visual Navigation* task, the agent needs to navigate to an object emitting a particular sound in an indoor environment (Chen et al., 2020a; Gan et al., 2019). Depending on the type of task, it may be sufficient to store just the object category (e.g. for ObjectNav) in the map, or it is may be necessarily to retain more finer-grained information (e.g. VLN). In this survey, we will mainly focus on recent work on room-scale map building for navigation as these methods can be extended for maps for manipulation and used for exploration.

2.2 End-to-end approaches

The embodied AI community has seen a lot of progress in training task-specific end-to-end models with reinforcement learning (RL) that directly learns to predict discrete (Wani et al., 2020) or continuous

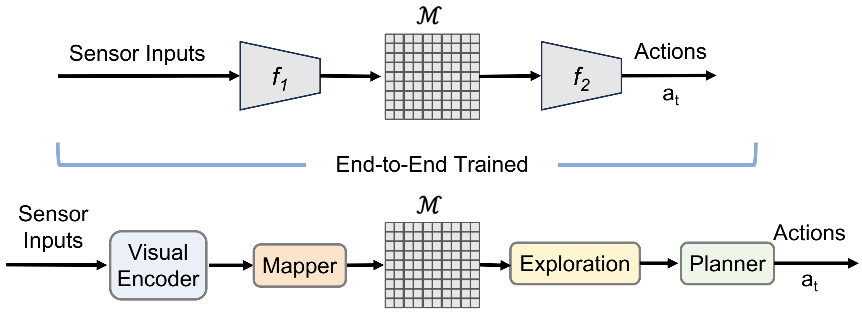

Figure 2: **End-to-end vs Modular.** (Top) End-to-end model is trained as a single pipeline which generates actions directly from sensory inputs. (Bottom) Modular pipeline consists of various sub-modules, each with a specific function so that they can be trained independently of the others.

actions (Kalapos et al., 2020) from visual observations (see Fig. 2). These methods may consist of an unstructured memory such as LSTM Dobrevski & Skočaj (2021). However, such representation lacks reasoning about 3D space and geometry and fail to perform well in long-horizon path planning. This has lead to the development of approaches that build an intermediate map representation. Such a map can be implemented using differentiable operations so as to facilitate end-to-end training. Gupta et al. (2017) shows that an egocentric map built this way is beneficial for both PointNav and ObjectNav tasks, whereas Henriques & Vedaldi (2018) learns a global map for the task of localization. While these methods are trained using supervised learning, Wani et al. (2020) use RL to learn to predict actions based on an intermediate global map of the environment to address the complex MultiON task. However, irrespective of the map representation or the mode of training, these approaches need to be retrained every time the task definition changes and even the basic skills required to perform a task need to be learned from scratch.

2.3 Modular approaches

Table 1: We show how prior works build map either as part of an end-to-end architecture or as a modular architecture consisting of four basic modules. For the modular approaches, we summarize the different module choices as well as what they store in their map.

Methods	Task	End-to-End	Modular			
			Visual Encoder	**Map**	**Exploration**	**Planner**
ANS (Chaplot et al., 2019)	Exploration		ResNet18 (He et al., 2016)	occupancy + explored	learned policy	Fast Marching
NTS (Chaplot et al., 2020b)	ImageNav		ResNet18 (He et al., 2016)	topological map	learned policy	A*
SemExp (Chaplot et al., 2020a)	ObjectNav		MaskR-CNN (He et al., 2017)	occupancy + explored + semantic labels	learned policy	Fast Marching
ModLearn (Gervet et al., 2023)	ObjectNav		MaskR-CNN (He et al., 2017)	occupancy + explored + semantic labels	learned SemExp	Fast Marching
MOPA (Raychaudhuri et al., 2023)	MultiON		FasterRCNN (Ren et al., 2015)	semantic labels	Uniform Sampling Exploration	PointNav(Wijmans et al., 2019)
CMP (Gupta et al., 2017)	PointNav, ObjectNav	✓				
MapNet (Henriques & Vedaldi, 2018)	Localization	✓		–		
MultiON (Wani et al., 2020)	MultiON	✓				

Another line of work explores how to breakdown a complex task into a set of basic skills that the agent needs to acquire. Such skills can then be learned independently of each other so that they can be leveraged across various tasks without the need to be retrained from scratch. This has led to various works on modular pipelines (Chaplot et al., 2019; 2020a; Gervet et al., 2023; Raychaudhuri et al., 2023), where each module is responsible for a particular skill and the modules interact with each other to perform the entire task. In the modular approach, it is common to have a **visual encoder** that processes and encodes the visual information at each time step, a **mapper** that aggregates the encoded information into a map, a **exploration** module that determine what parts of the environment needs to be explored, and a **planner** that determines the low-level action to take. Fig. 2 shows the difference between the end-to-end approach and the module approach, and we elaborate on the modules and popular design choices below.

Visual Encoder. This module encodes agent observations to produce semantic visual features and predictions at every time step. Prior works have used visual features from pretrained backbones such as ResNet He et al. (2016) or ViT (Dosovitskiy et al., 2020), and often leveraging object detectors MaskRCNN He et al. (2017) or FasterRCNN Ren et al. (2015). As we will see in Sec. 5.2.2, with the development of large pretrained vision-language models and open-vocabulary detectors, pretrained models such as CLIP (Radford et al., 2021), LSeg (Li et al., 2022), DINO (Caron et al., 2021; Oquab et al., 2023), and others are increasingly popular as the basis for building large to open-vocabulary maps. The visual encoder used will determine the information captured in the features, and whether there are detected object instance bounding boxes or segmentations for integration into the mapping modules.

Mapper. The mapper is responsible for building a semantic map of the environment from the encoded image features and agent pose. To build a global map over time, the mapper typically aggregates the current map with the map from previous step (see Sec. 3.4 for details). In this paper, we survey how recent methods structure the map (Sec. 4) and what information can be encoded in it (Sec. 5). Tab. 1 summarizes the type of information stored in the map by various methods. This can be occupancy information, explored area or semantic labels of the detected objects.

Exploration. This module enables the agent to explore its environment efficiently to either ensure the map is complete (by maximizing the covered area) or selecting unvisited areas where the target is likely to be. Typically, the exploration module selects a point or region to explore given the obstacle map built by the mapper and the current agent pose. Agents can use simple heuristics-based methods such as sampling a point at uniform (Zhang et al., 2021; Raychaudhuri et al., 2023), systematically sampling four corners of a grid centered at the agent (Luo et al., 2022) or selecting a point from the unexplored frontier (Yamauchi, 1997). To decide which frontier point the agent should explore, various strategies are employed, such as selecting the nearest point to the agent (Gervet et al., 2023) or the most promising point based on semantic reasoning. In the semantic reasoning based exploration methods, the agents may select the highest text-image relevance score (Gadre et al., 2023; Yokoyama et al., 2023) from a pretrained large vision-language model such as BLIP-2 (Li et al., 2023), select the highest probabilistic output of the VLM directly (Ren et al., 2024), or leverage a LLM to extract common-sense knowledge (Zhou et al., 2023). Researchers have also used learned policies (Chaplot et al., 2019; 2020a), where the agents are generally trained with RL using rewards, such as coverage (Chen et al., 2019) or curiosity (Pathak et al., 2017; Mazzaglia et al., 2022). Although there is less hand-crafted rules in learning-based methods, they need millions of training steps and careful reward engineering.

Planner. Once a map is built, a low level path-planning module is used to plan a path to the goal location from the agent's current location. The path consists of low-level actions that can be executed by the agent to move to the goal. While this is implemented as a heuristics-based Fast Marching Method (Sethian, 1996) in most of the prior works, a recent approach MOPA by (Raychaudhuri et al., 2023) has used a learned PointNav policy trained offline with DD-PPO (Wijmans et al., 2019).

In Tab. 1, we compare common modular approaches and look at how they leverage various heuristics-based or learned approach for each of the modules. The advantages of a modular pipeline include its ability to leverage pretrained models from other tasks (Gervet et al., 2023; Raychaudhuri et al., 2023) and its ability to transfer from simulation to real-world robots better (Gervet et al., 2023).

2.4 Active SLAM

In this section, we first discuss a popular approach in classical robotics known as active SLAM, with a focus on *mapping* later. The key idea behind Active SLAM is that, instead of just passively collecting data and using it to construct a map, the robot makes decisions about where to move next in order to reduce uncertainty in the map and its estimated location. This is crucial for an autonomous robot to efficiently perform complex tasks in an unknown environment and involves three key problems (Makarenko et al., 2002; Fairfield, 2009) – mapping, localization, and planning (Fig. 3). In the context of classical robotics, we first define these three problems. *Mapping* involves collecting sensor data to create an environmental model, accounting for sensor noise and uncertainty. In traditional robotics, these methods address such uncertainties, while embodied

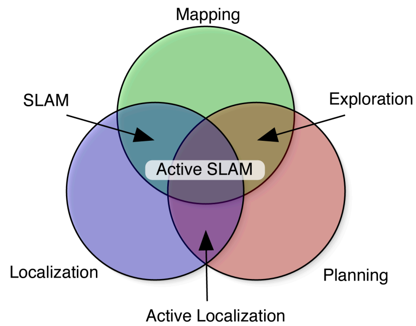

Figure 3: **Active SLAM.** At the core of classical mobile robotics lie three core tasks – mapping, localization, and planning. These are often interdependent on each other and overlap to form other tasks such as SLAM (localization and mapping), exploration (mapping and planning), active localization (planning and localization) and active SLAM (mapping, localization, and planning). *Figure reproduced from Fairfield (2009).*

AI simulators often assume ideal, noiseless sensors. *Localization* is the process of determining the robot's position within the map using sensor data, which can be noisy. *Planning* is deciding the robot's next actions, considering constraints like safety and uncertainties in the map, localization, and action outcomes. Note that path planning is a sub-task of planning, focusing on finding the safest and most efficient route to a goal.

These problems are interconnected, leading to tasks such as *Simultaneous Localization and Mapping (SLAM)*, where mapping and localization depend on each other, *classical exploration* where the robot takes actions to maximize environment coverage and build a complete map, *active localization* where the robot actively plans and refines its position by taking actions. At the intersection of all three is Active SLAM, where the robot actively reduces uncertainty by exploring, building a map, and localizing itself while refining both. This enables the robot to perform tasks more efficiently.

Next we look at the maps built during active SLAM. Traditionally, the map captured purely geometric information about an environment, by using measurements from LIDAR or depth sensors to build an obstacle map while moving around in an unseen environment. However, this fails to capture crucial semantic cues about the environment needed to perform complex tasks. This has led to the development of Semantic SLAM approaches which additionally detect and identify objects in the scene. Classical methods in Semantic SLAM rely on feature extraction (SIFT (Lowe, 2004), SURF (Bay et al., 2008), ORB (Rublee et al., 2011)) which is then referenced against a dictionary based on Bag-of-Visual Words (Peng & Li, 2016) to determine its closest similarity match. Chen et al. (2022) and Placed et al. (2023) survey the landscape of classical and semantic SLAM approaches in a greater detail. Modern semantic SLAM systems, however, use deep learning based approaches to capture semantic information from its environment. This survey focuses on the mapping approaches in embodied AI with noiseless sensors and perfect odometry, thus allowing us to focus on the built map representation and encoding without having to worry about dealing with noise.

3 Semantic map

Traditionally, maps built for navigation stored only obstacle information in a spatial grid-based structure. While these maps capture the geometry of the space and can help an agent to avoid obstacles, they are not sufficient for the demands of more complex embodied AI tasks. An enhanced map that goes beyond geometry to capture meaning and context in its environment aligns with how humans perceive and navigate its surroundings. We call these "semantic maps" which provide a richer and more nuanced understanding about the objects and places in the environment. These maps are indispensable for performing complex tasks such as navigating to a specific room (kitchen) (Narasimhan et al., 2020), rearranging objects (Trabucco et al., 2022) or performing a specific action on a specific object (sitting on a couch) (Peng et al., 2023). Semantic maps can also be structured in different ways that goes beyond just grids. In this section, we introduce necessary terminology and concepts used in the rest of the survey. We start by defining semantic maps

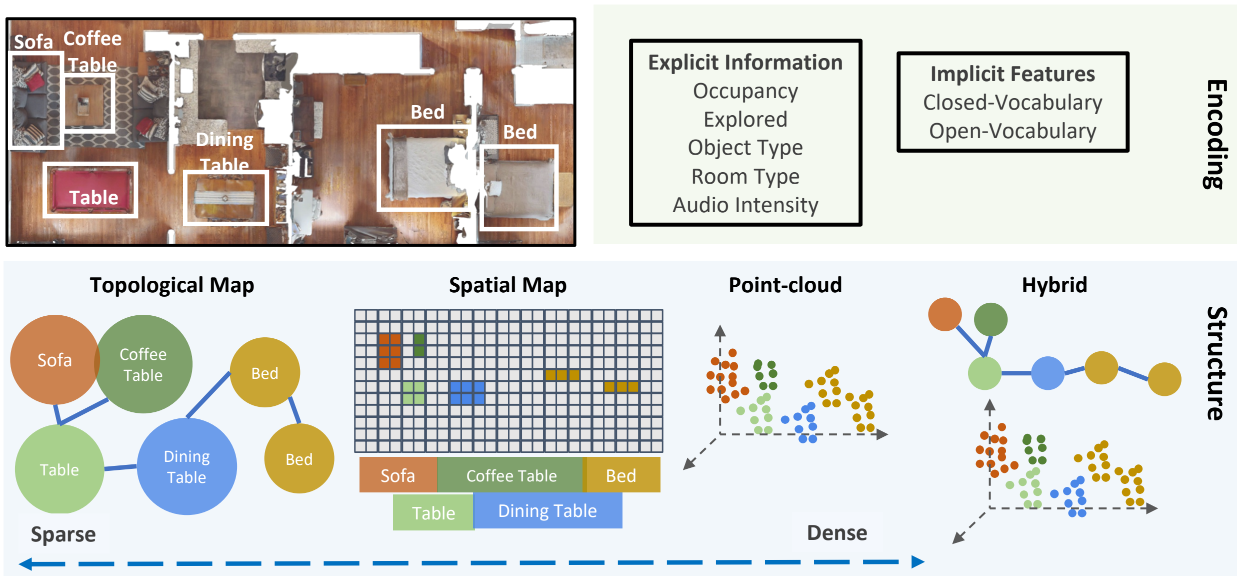

Figure 4: **Structure and encoding.** The map corresponding to a physical environment can be structured as a topological map (with nodes and edges), a spatial grid, a point-cloud or a hybrid map combining two or more of the others. These structures can store either *explicit* or *implicit* information corresponding to the observation made at that location. The figure shows examples of types of explicit information that can be stored at each cell or node. Implicit features are typically extracted and aggregated using vision encoders such as ResNet for close-set vocabulary, or large-vision language models such as CLIP Radford et al. (2021) for open-vocabulary semantics.

(Sec. 3.1), and how they can be structured (Sec. 3.2) and encoded (Sec. 3.3). Fig. 4 provides an overview of the different map structures and encodings, and Tab. 2 summarizes the structure and encoding used in prior works. We then describe the common techniques for building semantic maps (Sec. 3.4) and present various challenges that exist in those approaches (Sec. 3.5).

3.1 What are semantic maps?

A semantic map captures not just the physical space of the environment but also semantic information about the environment, such as names of identified objects and regions, key features and other attributes relevant to how the agent navigates or interacts with the environment. It can also store spatial and functional relationships between the objects and regions. For a robot or agent to build accurate semantic maps, the robot perceives the environment through sensors (camera, LiDAR, etc.) and uses cognition to classify the objects and regions it perceives (Ren et al., 2015; He et al., 2017; Liu et al., 2023b; Kirillov et al., 2023; Zhang et al., 2023b). As the robot takes actions and navigates around, it needs to store the information in a structured memory (semantic map) which can be retrieved as needed. Semantic maps enable the robot to reason about the environment so that it can efficiently interact with the environment in downstream tasks such as navigation, instruction-following and object manipulation. Suppose an agent is tasked with finding an apple and putting it in the refrigerator. Let's say that the agent has seen the refrigerator first and then the apple. After picking up the apple, it would be efficient for the agent to retrace its steps, if it has memorized the location where it had seen the refrigerator.

3.2 What is the structure of this map?

A semantic map can be structured as a *spatial grid map*, *topological map*, *point-cloud map* or a *hybrid map* (Fig. 4). A spatial grid map is a top-down grid where each grid cell represents an area in the physical environment. So if an object is at a certain location (X, Y, Z) in a 3D scene, the semantic map will contain information about that object at the corresponding grid cell (x, y), where x and y are the row and column

Table 2: **Semantic maps in indoor embodied AI.** We characterize works that use maps for indoor embodied AI by the type of structure they use (**Grid**, **Topological**, **Point Cloud**) and how information is encoded in the map (**Expl**icit vs **Impl**icit). *Explicit* encodings are pre-selected information such as occupancy $\boxed{\bullet}$, explored-area $\boxed{\text{x}}$, object category $\boxed{\triangle}$, visitation time $\boxed{t}$ and others. *Implicit* encodings are learned representations such as visual (V) or visual-and-language (VL) features. The use of VL features (typically from large pretrained models) enable building *open vocabulary* maps. Works that aggregate implicit features onto a grid map, but finally decode into explicit encodings are marked as 'Implicit to Explicit' in this table.

<table>
<tr><th></th><th colspan="3">Structure</th></tr>
<tr><th>Encoding</th><th>Grid</th><th>Topological</th><th>Point Cloud</th></tr>
<tr><td rowspan="2">Explicit (no semantics)</td><td>ANS (Chaplot et al., 2019) $\boxed{\bullet}\boxed{\text{x}}$</td><td></td><td></td></tr>
<tr><td colspan="2">Thrun et al. (1998) $\boxed{\bullet}$
Tomatis et al. (2001) $\boxed{\bullet}$</td><td></td></tr>
<tr><td rowspan="2">Explicit (semantics)</td><td>SemExp (Chaplot et al., 2020a) $\boxed{\bullet}\boxed{\text{x}}\boxed{\triangle}$
MOPA (Raychaudhuri et al., 2023)$\boxed{\triangle}$
GOAT-Bench (Khanna et al., 2024) $\boxed{\bullet}\boxed{\text{x}}\boxed{\triangle}$
MapNav (Zhang et al., 2025a) $\boxed{\bullet}\boxed{\text{x}}\boxed{\triangle}$</td><td></td><td></td></tr>
<tr><td colspan="2">BEVBert (An et al., 2023) $\boxed{\bullet}\boxed{\triangle}$
3D-DSG (Rosinol et al., 2020a) $\boxed{\bullet}\boxed{\triangle}$</td><td></td></tr>
<tr><td>Implicit to Explicit</td><td>SemanticMapNet (Cartillier et al., 2021) $\boxed{\triangle}$</td><td></td><td></td></tr>
<tr><td>Implicit (V)</td><td>CMP (Gupta et al., 2017)
MapNet (Henriques & Vedaldi, 2018)
MultiON (Wani et al., 2020)</td><td>SPTM (Savinov et al., 2018)
NTS (Chaplot et al., 2020b)
CMTP (Chen et al., 2021)
VGM (Kwon et al., 2021) $\boxed{t}$</td><td></td></tr>
<tr><td rowspan="3">Implicit (VL)</td><td rowspan="2">CoW (Gadre et al., 2023)
VLMap (Huang et al., 2023a)
NLMap (Chen et al., 2023a)
VLFM (Yokoyama et al., 2023)
InstructNav (Long et al., 2024)</td><td>RoboHop (Garg et al., 2024)</td><td>OpenScene (Peng et al., 2023)
CLIPFields (Shafiullah et al., 2023)
CLIP2Scene (Chen et al., 2023c)
3D aware ObjNav (Zhang et al., 2023a)</td></tr>
<tr><td colspan="2">ConceptGraphs (Gu et al., 2023)</td></tr>
<tr><td colspan="3">StructNav (Chen et al., 2023b)</td></tr>
</table>

numbers respectively such that there is a direct mapping from (X, Y, Z) to (x, y). For navigation, most spatial maps are 2D, such that a grid cell ignores the Z-axis (up direction) by aggregating the semantic information across the up axis. However they can also be 3D where the Z-axis is divided into discrete bins. On the other hand, a topological map is a graph-like structure where nodes represent objects or important landmarks in the scene and edges represent relationship (distance, spatial relation, etc.) between them. It is also possible to store semantic information on a point-cloud map, which can be viewed as a 3D map with varying density. In a point-cloud map, information is associated with each point (x, y, z) corresponding to 3D location (X, Y, Z) in the physical space. Unlike the voxel-grid, which is regularly spaced, points can be sampled at varying densities. Some works combine two or more of the above structures to form a hybrid map since each structure has its own advantages and limitations. We discuss each of these in detail in Sec. 4.

3.3 What is stored in this map?

The semantic map stores information about a particular 3D location (X, Y, Z) in the physical environment. This information can either be *explicit* or *implicit*. Explicit encodings have clear specific meanings assigned to each value. For instance, each cell (X, Y, Z) can store information about whether there are any obstacles at that position, whether that location has been explored by the agent, the category of the object present there and so on. On the other hand, an implicit encoding is a feature encoding capturing information derived from the sensory input (e.g. images) that the agent observes at that particular location (X, Y, Z). The features are typically extracted from pre-trained encoders. Depending on whether the feature encoder was pre-trained on a set of images from limited categories or a large internet-scale dataset of image and language data, the implicit encoding can be either *closed-vocabulary* or *open-vocabulary*. The term *closed-vocabulary* is used to indicate only a limited set of object categories is recognized, while in a *open-vocabulary* setting, the features extractors can theoretically identify 'any' object.

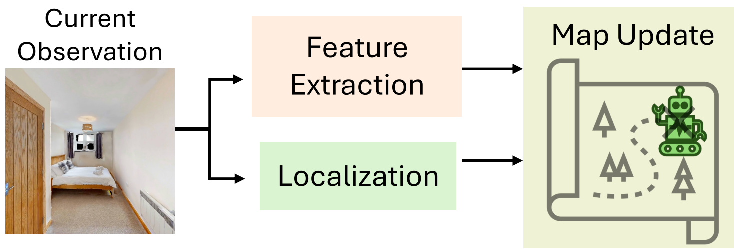

Figure 5: Map building involves *localization* (where the agent is on the map), *feature extraction* (extracting useful semantic information from the observations), and *map update* (building the map by aggregating the semantic information over time).

3.4 How is the map built?

Creating accurate and detailed semantic maps requires integrating data from various sources and sensors such as camera, LiDAR and depth sensors. More specifically, map building consists of having an agent navigate about a space, and accumulating observations O_t at time step t into the appropriate map structure m_t. To build an accurate map, the agent first needs to have an estimate of where it is (*localization*). Next it extracts semantic information from an observation $F(O_t)$ (*feature extraction*), and combines the features into a common map over time (*accumulation*) (refer to Fig. 5). While building a spatial grid map, an additional step is to project the features onto the map (*projection*). It is common to group the last three steps into *map building* and study it jointly with *localization* in Simultaneous Localization and Mapping (SLAM). We discuss SLAM methods briefly in Sec. 2.4.

Localization. Localization can be challenging due to noisy sensors and actuators. To simplify the problem, it is common in the embodied AI community to either assume perfect localization is given at each time step (Cartillier et al., 2021) or to localize the agent with respect to its starting position in an episode (Henriques & Vedaldi, 2018) assuming perfect actuation. The latter is more easily adapted to the real-world setting since it doesn't require the exact knowledge about the agent's pose. Instead the relative displacement of the agent with respect to its starting pose is enough to build the map eventually.

Feature extraction. Feature extraction is a crucial part of building a semantic map. Ideally these features should be representative of the objects present in the map. We discuss this topic at length in section Sec. 5.

Projection. An important step in building a spatial grid map is taking the 2D observations and project them into 3D. Typically, this relies on having depth information and known camera parameters in order to convert 2D pixel coordinates to 3D world coordinates. To project a particular pixel in the camera frame, first a ray is shot from the camera center through the image pixel (i, j) to the depth $d_{i,j}$ to get a 3D point in the camera coordinate frame. Next the camera coordinates are converted to the world coordinates (X, Y, Z). For a 2D spatial map, the 3D coordinate(X, Y, Z) is mapped to the grid cell indices x and y in the spatial map. The transformation for the standard pinhole camera with known camera pose (3D rotation R and 3D translation t) and intrinsics (K) can be written as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = d_{i,j} R^{-1} K^{-1} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} - t \tag{1}$$

and the orthographic projection can be written as,

$$\begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = P_v \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

where where P_v is known orthographic projection matrix to convert 3D world coordinates into 2D grid cell indices. If more than one point are projected to the same grid cell in the spatial map, they are accumulated into the cell using an function to aggregate the features or predictions.

Accumulation. During *map update* there are many ways to aggregate features or prediction into the map including 1) overwriting the map with the latest observations ($m_t = m_{t-1}$), 2) performing mathematical operations such as max ($m_t = \max(m_t, m_{t-1})$) or mean ($m_t = \text{mean}(m_t, m_{t-1})$), and 3) using a learned neural network. For learned aggregation functions, it is common to use a recurrent network (LSTM, GRU) ($m_t = \text{GRU}(m_t, m_{t-1})$).

During the process of building the map, there are several other important aspects to consider.

Egocentric vs allocentric. There are also choices in the reference frame used for map-building, to either maintain a map with an *egocentric* coordinate frame that is relative to the agent (e.g. +y coordinate to the front of the agent) or *allocentric* (e.g. world) coordinate frame.

Tracking visited areas. For the map to be complete, it is important for the agent to be able to determine whether it has already visited a location or not, and whether there are unexplored locations. For a specific embodied task, it may not always be necessary for the agent to built a complete map if the task can be accomplished.

View point selection. In the case of the embodied setting, the agent is also limited in the possible viewpoints it can observe, and must accumulate into the map, observations in a sequential manner. This is in contrast to the non-embodied setting, where there can be more freedom in selection of viewpoints, and observations can be first collected and then analyzed together.

Online vs offline map building. It is possible for the agent to build a map by exploring an environment first. After the map has been built, the agent can then start to perform the specific task. In this scenario, the agent builds the map and performs the task in two separate phases, a process known as *offline* method of map building. Although this method saves compute time during the actual task, there is the overhead of an extra exploration phase for the agent to familiarize itself with a new environment. This approach can be appropriate when the agent is expected to be reused in the same environment repeatedly. However, since the map is a static snapshot and if it is not updated during the task, there can be mismatch between the actual state of the environment vs what was precomputed. For instance, it might happen that the agent ends up at a location which has not been captured in the map. This might lead to the task failure. Moreover, in real-life applications where a robot is expected to perform a task in an unseen environment, such as search-and-rescue operations, it's not ideal for it to spend extra time exploring the environment first and then performing the task. In contrast a better way is to build or update the map during the task or *online* so as to keep it updated at all times.

Map building in real world. Maps built in simulation in embodied AI tasks are often noisy due to unrealistic assumptions that limit its usage in the real-world. Map building has been mostly studied in the community as a sub-module in conjunction to solving more complex high-level reasoning tasks. Researchers have thus tried to investigate what type of maps are useful for which tasks and decoupling the issue of noisy sensors from map building enables them to do exactly that. The most prominent of the assumptions is that of noiseless sensors. For example, sometimes the community assumes *perfect localization* (agent's current location and orientation) at all times during navigation, which is unrealistic in real-world. This is mainly because GPS and Compass sensors are generally noisy, whenever available. However, in most indoor spaces, GPS might not even be available. SLAM methods which work really well in real-world robots operate under the assumption that GPS is not available, and relies on the onboard sensors to estimate its location on the map. Another example of the noiseless sensor assumption is that of a perfect actuation, which means that when an agent initiates an action to move forward by 25cm, it will end up exactly at a location 25cm ahead of its current position. But real-world actuators are noisy and affected by varying friction on different surfaces, which results in significant drifts over time. SLAM systems are inherently capable of addressing such issues by operating under uncertainty in the robot's pose estimation. Loop Closure is a sub-algorithm of SLAM which identifies previously visited locations and then uses them to correct accumulated errors in pose estimation. In general SLAM systems build a more consistent and accurate map of the environment in a real-world setting than the current mapping techniques in embodied AI.

3.5 Considerations in map building

Despite the huge progress in map building in the embodied AI community, there are many challenging aspects that need continual research and improvement.

Real-time processing. When building the map online, the agent needs to continually process sensory data and update the map in real-time by accumulating semantic information. This is a computationally intensive task and hence requires efficient algorithms. Moreover, in dynamic environments such as autonomous driving, the algorithms need to have low latency.

Memory consumption and scalability. Creating and maintaining large-scale semantic maps require sufficient storage and may be difficult to scale. Such large maps may result from navigating a large environment such as outdoor cityscapes or navigating intricate indoor spaces for a long period of time or storing dense semantic features inside the map. Efficiently updating such large-scale maps is still a challenge that remains to be solved (Gu et al., 2023).

Noise and uncertainty. In real-world robots the data from sensors (cameras, IMU, etc.) might be noisy due to reflective surfaces, uneven gradient, etc. Such noisy data introduces uncertainty in the map built by the robot. However, in the embodied AI community, it is a common practice to assume noiseless sensors in simulated environments, which results in a noiseless map. When transferred directly from simulation to real-world, the map building methods might not work very well, unless special techniques to deal with noise and uncertainty are employed (Chaplot et al., 2020a; Georgakis et al., 2022b).

Dynamic environments. In many applications such as autonomous driving or robotic surgery, the environment is always changing because of moving objects, changing lighting conditions and evolving structures. The map building techniques need to be adaptable to these dynamic environmental changes and build accurate and consistent map.

Semantic understanding. Identifying objects in an environment accurately is crucial for creating useful semantic maps. This relies on advanced computer vision techniques which have seen a huge progress from identifying a fixed set of objects (He et al., 2017; Ren et al., 2015) to identifying 'any' object in an image (Zhou et al., 2022; Liu et al., 2023b). Moreover, contextual understanding and spatial relationships between objects is useful for effective decision-making in complex semantic tasks (Antol et al., 2015; Gordon et al., 2018).

Usability and reliability. Creating interactive interfaces for humans to query semantic maps is a new area many researchers are focusing on (Peng et al., 2023; Yamazaki et al., 2023). Such queryable maps have not yet been used in any of the downstream embodied AI tasks, but remains a promising direction to explore. Moreover, ensuring that the built maps are reliable and consistent induces trust in real-world safety-critical applications. This requires building accurate maps of the environment through rigorous validation and testing.

Standardization. Developing standardized frameworks for map building might help in collaboration and integration across different systems and platforms. Although the robotics community has standardized practices in map building (e.g. SLAM), the embodied AI community relies on different techniques (Gupta et al., 2017; Chaplot et al., 2020a) and may benefit from a common framework.

4 Map structure

In this section we will look at various map structures that have been used in prior works (see Tab. 2). A semantic map can be structured in various ways: *spatial grid map*, *topological map*, *point-cloud map* or a *hybrid map*. Spatial grid maps are metric maps of the environment such that its dimensions align to that of the environment and they can be structured as either 2D or 3D grids. Topological maps, on the other hand, represent the environment through a set of landmarks represented as nodes and relation between adjacent landmarks represented as edges in the form of a graph. Point-cloud maps are the densest form of 3D maps

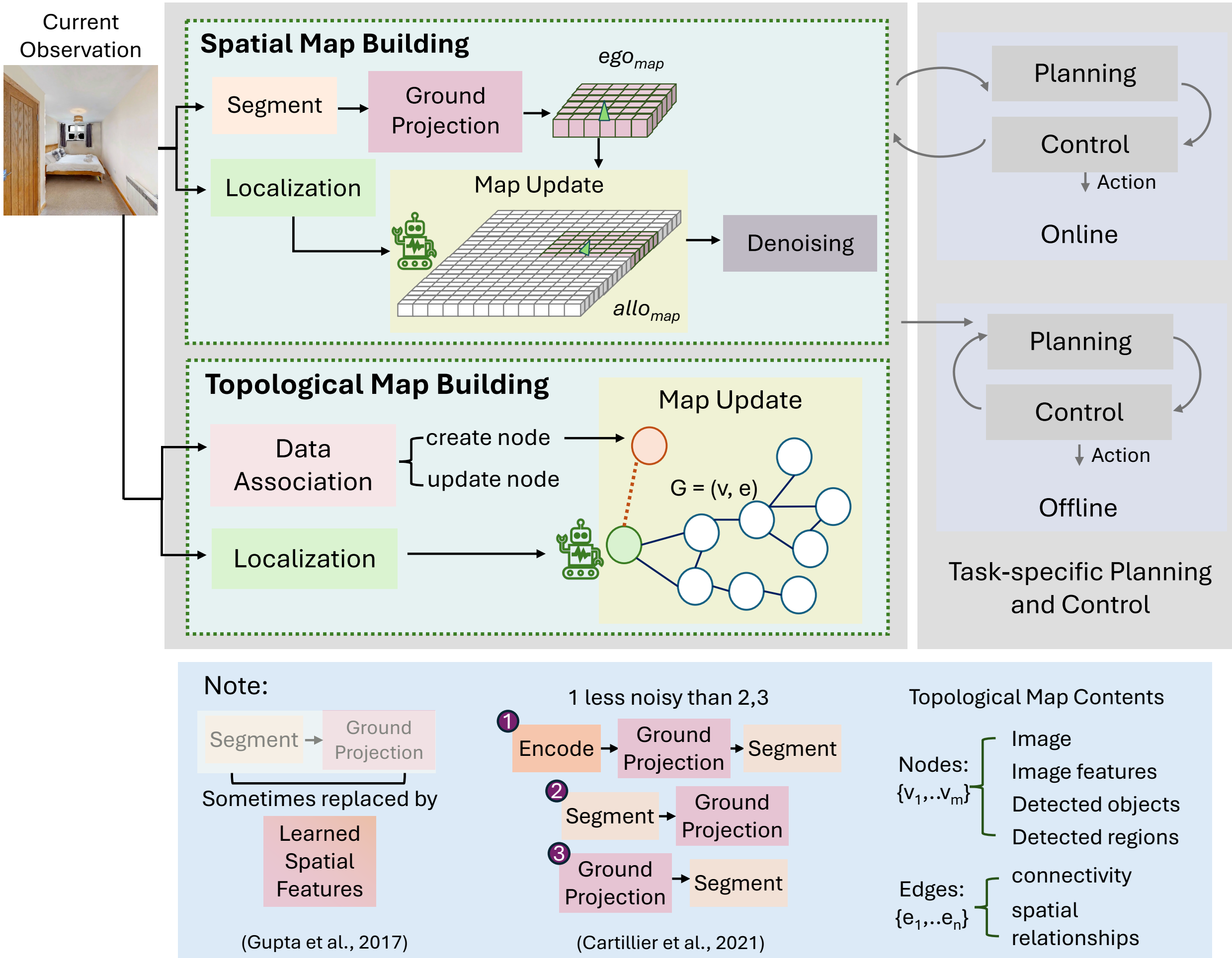

Figure 6: **Spatial map building.** A spatial map is a grid map of dimensions $(M \times N \times K)$ where M and N are spatial dimensions and K is the number of semantic channels. The common pipeline to build the map is to *segment* the input image, then *ground project* into an egocentric map ego_{map}, which is then registered to the allocentric map $allo_{map}$ via *map update* using the *localized* agent pose. A *denoising* step generally follows and the map is built online along with the task planning and control. While Gupta et al. (2017) learns a spatial representation without segmenting and ground projecting, Cartillier et al. (2021) observes that encoding followed by ground projecting and then segmenting reduces noise in the produced map.
Topological map building. A topological map is a graph-like structure, $G = (v, e)$ with nodes $(v = \{v_1, ..., v_m\})$ and edges $(e = \{e_1, ...e_n\})$, that can be built *Online* or *Offline* with or before task-specific planning and control respectively. Based on current observation, the agent performs *localization* on the graph and then performs matching (*Data Association*) with the current node, based on which either a new node is created or an existing node is updated. The nodes and edges may contain various types of semantic information thus enabling decision making in a task.

whose all three dimensions align to the 3D space such that each 3D point in the scene is captured in the map. Two or more of these types of maps are sometimes combined together to form Hybrid maps.

4.1 Spatial grid map

A spatial grid map m_t is a $(M \times N \times K)$ matrix where M and N are the spatial dimensions of the map and K denotes the number of channels to store semantic information at that location. It is a grid like structure where each cell has a width and a height which correspond to a certain area in the physical environment.

Current research on indoor embodied AI make use of environments from datasets such as Matterport3D (MP3D) (Chang et al., 2017) and Habitat-Matterport3D (HM3D) (Ramakrishnan et al., 2021) which are 3D reconstructions of real-world spaces. Compared to MP3D, HM3D contains around 10x more scenes with high visual fidelity and lesser reconstruction artifacts. These environments are typically for houses or office spaces with a total area of $< 1000\text{m}^2$. A grid map with each cell representing an area of $400 - 900\text{cm}^2$ is found to be good enough to represent such spaces (Wani et al., 2020; Raychaudhuri et al., 2023). At the start of each episode, the spatial map is initialized with a tensor of size $(M \times N \times K)$, and gradually built as the agent moves around the environment. These are often 2D top-down maps (Gupta et al., 2017; Henriques & Vedaldi, 2018; Narasimhan et al., 2020; Cartillier et al., 2021) with the first two dimensions corresponding to the spatial dimensions of the environment. However, some build 3D spatial maps (Chaplot et al., 2021) to capture the vertical dimension, in which case the map m_t is a 4D tensor $(M \times N \times P \times K)$.

A spatial map may be built in a number of ways depending on whether raw features are directly projected onto the map, or whether a semantic segmentation is used (see Fig. 6). One way is to learn an egocentric projection of the image features as in CMP (Gupta et al., 2017) which forms the egocentric map. In CMP, the egocentric observations are first encoded with a learned image encoder network such as ResNet (He et al., 2016)), and then the network learns to predict an egocentric projection of the image features, without explicit supervision on the map. Instead, the mapper is trained end-to-end along with the planner to predict actions. Egocentric maps, however, not only fail to capture the global structure of the environment, but 'forget' most of the past observations. Thus in long-horizon planning tasks, where the agent needs to 'remember' its past observations for efficiency, egocentric maps fall short.

To maintain an allocentric map, it is necessary to take the egocentric information at each time step and aggregate it into a global map. One way to achieve this is to first obtain egocentric projection of image features and then aggregate to a global allocentric map of the environment via a process known as *registration*. Registration allows the map to incorporate new observations on to specific grid cells. In case the grid cells are already occupied, the new observations are accumulated with the existing ones by employing an aggregation function. This aggregation function can be as simple as taking the latest or the average, but can also be a learned network (see Sec. 3.4). MapNet (Henriques & Vedaldi, 2018) builds an allocentric map by projecting the egocentric image features using depth observations and known camera intrinsics on a 2D top-down grid. This ground projection results in an egocentric projection on a spatial neighborhood around the camera. Next it performs registration by first obtaining a stack of egocentric maps rotated r times and then performing a dense matching with the allocentric map from the previous step to obtain the agent's current pose on the map. The dense matching is efficiently implemented with convolution operators. A LSTM then performs the aggregation of the current observations rotated by the current pose with the allocentric map from the previous step. While an LSTM is used in this work for aggregation, other functions and neural architectures can be used for aggregation as well (see Tab. 3 for a summary of aggregation methods used in different works).

Another way to build an allocentric spatial map is to first convert the image pixels to 3D coordinates in the camera space with known camera intrinsics. The camera coordinates are then converted to world coordinates with known camera pose. Finally the 3D world coordinates are voxelized and projected on a top-down 2D grid with a known projection matrix by summing over the height dimension. This approach is followed in Semantic MapNet (Cartillier et al., 2021) and MOPA (Raychaudhuri et al., 2023). Spatial maps built this way might be noisy due to noisy sensors and need an additional denoising step. Semantic MapNet uses a learned denoising network while MOPA employs a heuristic approach by selecting the centroid of a noisy cluster to obtain a clean map. Following on the last technique, Semantic MapNet (Cartillier et al., 2021) shows that first encoding the image, followed by projecting on to the ground plane and finally performing segmentation on the 2D map reduces noise in the spatial map thereby eliminating the need for an additional denoising step. Irrespective of the approach, it may happen that multiple image features are projected on to the same grid cell. In such cases, it is important to have a scheme for aggregating the features. Some common approaches to aggregation is to take the maximum (Henriques & Vedaldi, 2018; Wani et al., 2020; Cartillier et al., 2021), mean (Huang et al., 2023a), or the sum of the feature values (Chaplot et al., 2020a).

Summary. Spatial grid maps capture dense information about the environment. Such representations are useful for the agent to better reason about the spatial structure of the environment. However, the spatial

Table 3: **Spatial grid maps.** Prior works build spatial grid maps for various embodied AI tasks. Information is aggregated onto the map over time in many different ways such as learned recurrent networks and replacing with most recent information among others.

Method	Task	Environment	Dataset	Aggregation
CMP (2017)	PointNav, ObjectNav	Custom simulator	S3DIS (2016)	weighted mean
MapNet (2018)	Mapping	Doom (2016)	Active Vision Dataset (2017)	LSTM
ANS (2019)	Exploration	Habitat (2019)	Gibson (2018), MP3D (2017)	channel-wise max-pool
MultiON (2020)	MultiON	Habitat (2019)	MP3D (2017)	element-wise max-pool
SemExp (2020a)	ObjectNav	Habitat (2019)	Gibson (2018), MP3D (2017)	channel-wise max-pool
SemanticMapNet (2021)	ObjectNav, Visual QA	Habitat (2019)	MP3D (2017)	GRU
MOPA (2023)	MultiON	Habitat (2019)	HM3D (2021)	most recent
GOAT-Bench (2024)	Multimodal ObjectNav	Habitat (2019)	HM3D-Sem (2023)	most recent

maps need to be initialized with a certain width and height and as such is hard to scale if the environment size changes. Moreover, it consumes a lot of memory which might affect agent performance in the task.

4.2 Topological map

Compared to the high-precision grid maps, topological maps are graph-like structures with nodes connected to each other by edges. This essentially abstracts a large space into significant areas (nodes) where the agent can take decisions and connections or paths between them (edges) (Johnson, 2018). This enables parsing the environment into a local and a global structure such that the agent can plan locally in the small space represented as nodes while navigating the large space through graph search following the edges. This way of planning and navigating is inspired from how humans navigate in an unseen environment in that they identify and memorize significant landmarks and find paths to reach those landmarks (Janzen & Van Turennout, 2004; Foo et al., 2005; Chan et al., 2012; Epstein & Vass, 2014). Thus topological maps have been a popular choice in both traditional robotics research (Thrun & Montemerlo, 2006; Lorbach et al., 2014; Rosinol et al., 2020b; Campos et al., 2021) as well as in embodied AI research (Savinov et al., 2018; Chen et al., 2021; Chaplot et al., 2020b; Kwon et al., 2021; Gu et al., 2023; Mehan et al., 2024; Garg et al., 2024; An et al., 2024; Yang et al., 2024; Tang et al., 2025).

A key design decision during topological map building is what should be represented as nodes and what should be edges. Generally speaking, the nodes encode semantic information about locations in the environment such that the agent can make a decision whereas the edges store relationship or connection between the nodes. For indoor navigation, the landmarks for the nodes are typically objects in the environments. They can also be openings or intersections (Fredriksson et al., 2023), locations the agents has visited (Chaplot et al., 2020b), and other regions of interest (Kim et al., 2023; Shah et al., 2023; Garg et al., 2024). For navigation, two nodes are connected with an edge if it is possible to navigate from one node to another. Some methods also store spatial relationships between the nodes (Gu et al., 2023) in the edges to enable better reasoning.

One way to construct a topological map (see Fig. 6) is during an exploration phase previous to the actual task and then use the graph to plan a path to the node most similar to the target, for example, in Semi-Parametric Topological Memory (SPTM) (Savinov et al., 2018). During exploration the agent follows multiple random trajectories for each environment to form a node for every visited location and add an edge between the current node and the previous one to encode connectivity or reachability between them. A common post-processing step includes trimming out redundant nodes and edges to form a sparse graph (Chen et al., 2021). When the graph of one environment is collected from multiple random trajectories, it is also common to merge these graphs into one. However, a topological map generated this way in a pre-exploration phase is still sparse meaning that some observations in the environment might not have been captured by the graph. This affects the agent performance in the downstream task. Moreover, they need a pre-exploration phase which makes them unsuitable for unseen environments.

To mitigate this issue some works construct the topological map online while the agent is navigating during performing the task as is the case with Neural Topological SLAM (NTS) (Chaplot et al., 2020b). NTS consists of several modules – ‘Graph Update’ to update the topological map from observations, ‘Global Policy’

Table 4: **Topological maps.** Various works in indoor embodied AI build topological map either in an exploration phase or online while performing the task. The nodes often store learned features about the observation or temporal information (visitation timestep), while edges may store relative poses between a pair of nodes or types of edges.

Method	Map Building Phase	Node values	Node Feature Encoder	Edge values
SPTM (2018)	Pre-Exploration	Image features	ResNet18	✗
NTS (2020b)	Online	Image features	ResNet18	Relative pose in polar coordinates
CMTP (2021)	Pre-Exploration	Image features	ResNet152	Relative pose in discrete polar coordinates (8 directions, 3 distances (0-2m,2-5m,>5m)
VGM (2021)	Online	Image features, visitation timestep	ResNet18	✗
TSGM (2023)	Online	ImageNode stores image features; ObjectNode stores features for detected objects	pretrained image and object encoders	✗
RoboHop (2024)	Online	Image features for each image segment	CLIP (2021), DINOv2 (2023)	edge types denoting inter- and intra-image connectivity

to sample subgoals on the map and 'Local Policy' which outputs discrete navigation actions to reach the subgoal. The 'Graph Update' method gradually updates the nodes and edges in the graph from the current observations and agent poses. It first attempts to localize the agent on the graph from the previous timestep. If the agent gets localized in an existing node, it adds an edge between that node and the node from the last timestep. It also stores the relative pose between the two nodes represented as (r, θ) where r is the relative distance between the nodes and θ is the relative direction. If the agent is unable to be localized, a new node is added to the graph.

Another important aspect in the topological map creation is how to determine if two observations are similar to each other, in which case the two are mapped to the same node. If they are not similar, two different nodes exist for the two observations. This requires the map building methods to compare RGB images. The goal here is to classify two images as similar if (1) they are exactly the same or (2) there is a slight change in direction or distance between the two. Traditionally this is the problem of *data association* in SLAM-based systems, where an incoming observation could be matched to multiple landmarks (nodes) and either the best match is selected or a new node is created to mark a new landmark (Bowman et al., 2017; Dellaert et al., 2000). In embodied AI some works use a pretrained classifier network to implement this. The network is trained to classify whether two images are from the same area. NTS uses MLP trained with a cross-entropy loss in a supervised manner to predict whether are similar. This however needs annotated pairs of training data. Cross-Modal Transformer Planner (CMTP) (Chen et al., 2021), on the other hand, uses an oracle 'Reachability Estimator' to first obtain the geodesic distance between the two underlying locations based on the traversibility of the 3D mesh. If the distance is below a threshold, it maps them to the same node. Visual Graph Memory (VGM) (Kwon et al., 2021) also uses a pretrained network to determine if two images are similar. But they learn an unsupervised representation of the observations which are then projected onto an embedding space. The idea is to have the embeddings of observations coming from nearby areas clustered together because they are likely to have similar appearances. The training data in this case consist of randomly sampled observations from the training environments, thus eliminating the need for manual annotations. Kim et al. (2023) on the other hand use semantic similarity score obtained from a pretrained network (Li et al., 2021) between two images to determine whether they are the same nodes. A similar approach is taken in LM-Nav (Shah et al., 2023) where they use CLIP to calculate the cosine similarity between image features. Tab. 4 compares different methods that build topological maps.

Summary. In summary, topological maps are convenient to build and maintain due to their concise and condensed representation when compared to spatial maps. They are memory-efficient and can easily be scaled as the environment size increases by simply adding more nodes to the graph. However, they capture only certain landmarks in the environment and as such lack dense global information. This might lead to overlooking visual cues in a cluttered indoor scene that could be helpful for the agent to carry out spatial reasoning.

4.3 Point-cloud map

It is also possible to accumulate semantic information directly on to the 3D geometry of a scene, either triangle meshes or point clouds (which are increasingly popular as they are easier to work with than triangle meshes). For instance, given a point cloud, we can associate semantic information with each point in the scene. While not traditionally considered *maps*, we can view these representations as a type of semantic map as they associate semantics with spatial information. Often, a neural network is used to obtain the features associated with each point given the position (x,y,z) as input. Such neural functions that transform coordinates to real-valued vectors are also known as neural fields. These neural fields can produce dense 3D semantic map where each 3D point in the scene is captured and represented in the map.

Although traditionally such representations are mostly built for 3D scene understanding tasks, there has been increasing interest in using them for embodied AI. In scene understanding tasks, the goal is to perform various semantic inferences for each 3D point (Peng et al., 2023; Xu et al., 2024). In other words, given a point-cloud of a scene, the goal is to perform semantic segmentation, affordance estimation, room type classification, 3D object search and so on. Representations learned for such tasks are also often useful downstream for embodied AI tasks.

There are typically two strategies to train neural fields 1) use distillation (Peng et al., 2023; Kerr et al., 2023; Taioli et al., 2023; Qin et al., 2024; Guo et al., 2024; Qiu et al., 2024) to provide features that are similar to a pretrained 2D backbone, such as CLIP (Radford et al., 2021), 2) use of differentiable renderer to match the rendered semantics in addition to color (Zhi et al., 2021; Vora et al., 2021).

OpenScene (Peng et al., 2023), for example, predicts dense 3D features so that they are co-embedded with the corresponding text and the image in the CLIP embedding space. This allows the association of each 3D point in the scene with semantic information such that the scene can be queried using text to infer physical properties, affordances, etc. CLIP2Scene (Chen et al., 2023c) also uses CLIP to perform a 3D point cloud segmentation on outdoor scenes for application in autonomous driving. However, because the optimization happens per scene in many of these methods, they are typically expensive and not suitable for real-time use in embodied applications. A representative work that uses neural field maps for embodied AI include CLIP-Fields (Shafiullah et al., 2023). Zhang et al. (2023a), on the other hand, builds a 3D semantic scene representation based on an online point cloud-based construction algorithm (Zhang et al., 2020) made efficient by using a tree-based dynamic data structure. This method is very memory efficient and a lot faster when applied to the ObjectNav task. Similarly, Lei et al. (2024) shows that a 3D representation based on Gaussian Splatting is able to achieve state-of-the-art performance on the ImageNav task.

Summary. Use of such dense map remain relatively unexplored in the context of embodied AI tasks and can be a promising a future direction. However, using such a dense map could lead to significant memory consumption and computational inefficiency during querying. Moreover, in a typical indoor scene, most of the 3D space is empty and might not be useful in reasoning about the environment.

4.4 Hybrid map

So far we have seen how prior works structure maps as either a spatial grid or a landmark based scene graph or a more dense point-cloud. However there is a more recent effort on combining these different structures into a single map representation. This helps to capture information at various granularity and perform different types of reasoning on the environment.

The combination of metric information from grid-based maps together with a topological map is also known as a *topometric* map (Thrun et al., 1998; Tomatis et al., 2001; Blanco et al., 2008; Konolige et al., 2011; Ko et al., 2013; An et al., 2023). Thrun et al. (1998) proposes a single statistical mapping algorithm that first constructs a coarse topological map and uses it to construct a fine-grained grid map. They show that the topological map solves a global alignment problem by correcting large odometry errors, while the grid map solves a local alignment problem by producing high-resolution maps. Tomatis et al. (2001), on the other hand, proposes a compact environmental model where corners and hallways are represented by a topological map and rooms are represented by a grid map, both of which are connected in a single representation. When the robot is moving in hallways, it creates and updates the global topological map, and as soon as it enters a room, it creates a new local metric map[1]. They argue that the robot will only need to be precise inside rooms (e.g. manipulating objects, etc.) which justifies the need for the fine-grained precise metric maps, while the topological map is used to simply maintain global consistency in indistinguishable spaces such as long hallways and transitioning between significant places. BEVBert (An et al., 2023) is a more recent method that constructs hybrid maps offline and then learn a multimodal map representation to perform better spatial reasoning in the complex task of language-guided navigation.

There are also hybrid maps that combines grids, point-clouds, and topological maps. StructNav (Chen et al., 2023b) builds such a hybrid map where the spatial grid stores occupancy information, a scene graph stores landmarks with their connectivity, and a 3D semantic point cloud where each 3D point in the environment has a semantic label associated with it. Thus the spatial grid allows for obstacle avoidance and low-level path planning, the scene graph allows high-level reasoning about the relationship among the landmarks, and the 3D point cloud allows for a more dense semantic and spatial matching.

It is also possible to build maps that capture the semantic hierarchy of scenes at various levels of abstraction that allows for various levels of reasoning. Armeni et al. (2019) represent a static scene at multiple levels of hierarchy (buildings, rooms and objects), where entities are represented as nodes in a hierarchical graph and connected by edges representing coordinate frame transformations. Tang et al. (2025) (OpenIn) build a similar hierarchical scene graph to track objects in dynamically changing indoor environments. Rosinol et al. (2020a) (Dynamic Scene Graphs) too build layered scene graphs to track moving agents and objects in addition to building a dense 3D metric spatial map by modeling spatio-temporal relations between objects and agents. Hughes et al. (2024) demonstrate that hierarchical scene representations scale better than flat representations in a large environments and thereafter introduces a system called Hydra that incrementally builds 3D scene graphs from sensor data in real-time. Fischer et al. (2024) introduces a multi-level scene graph representation of large-scale dynamic urban environments from a set of images captured from moving vehicles and proposes a new view synthesis benchmark for urban driving scenarios.

Summary. Different types of map structures have their own strengths and limitations. While the coarse topological map is useful to represent significant landmarks in an environment, fine-grained metric maps are useful to represent its precise geometry and point-cloud map is useful to represent an even more dense 3D geometry of objects in the scene. Therefore it's crucial to combine two or more of these structures in order to create better representations of the environment, preferably at varying levels of semantic abstraction. While hybrid maps have been explored in robotics, they still remain under-explored in embodied AI. However, weaknesses of these maps need to be considered carefully before combining them. For example, topological and point-cloud map scale better than a grid map with larger environments, and point-cloud map needs the most and topological map needs the least memory to be stored.

5 Map encoding

In this section we will discuss different ways information is encoded and stored into semantic maps. Irrespective of how the map is structured, the map encoding, *i.e.* the values stored in the map can be either *explicit* or *implicit*. An explicit map encoding is one where the type of information stored is clearly known. An implicit encoding, on the other hand, uses a feature embedding that may not be directly interpretable. We now summarize various works that explore these two types of encodings (Fig. 7).

[1]They differentiate hallways and rooms by using laser sensor such that thin long open spaces are considered as hallways whereas other open spaces are considered as rooms.

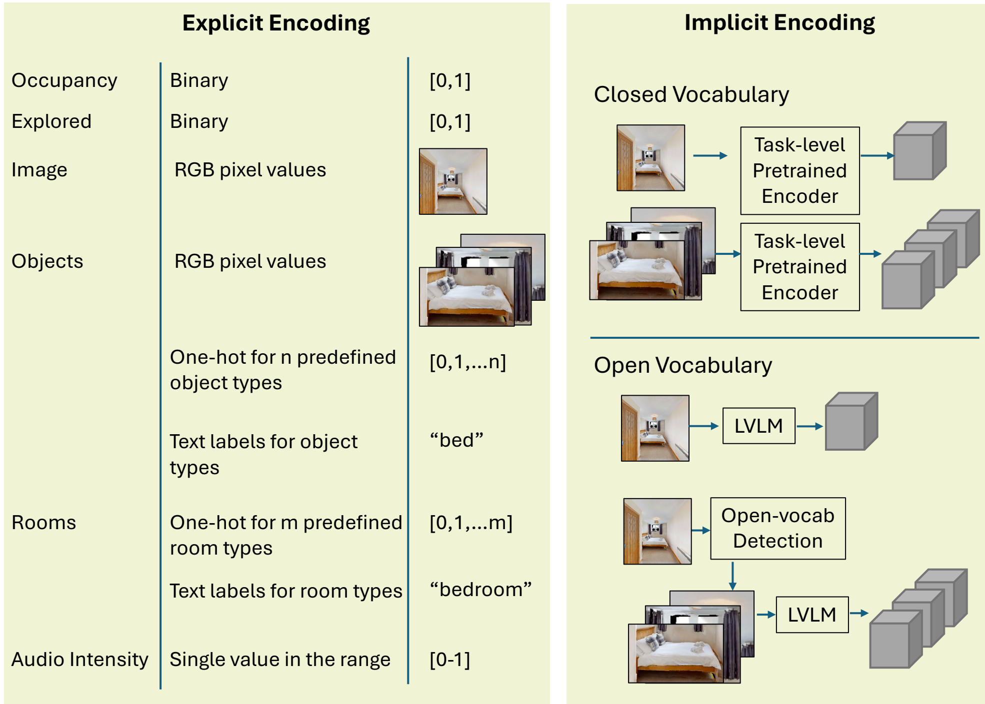

Figure 7: **Map Encoding** refers to the values stored in the map and can be either *explicit* or *implicit*, depending on whether the information is hand-selected or a learned feature representation of the observation.

5.1 Explicit encoding

Many prior works store explicit information in the map. Prior works have found that almost every embodied AI task benefits from the information about obstacles present in the environment and as such maintaining a spatial map that has this *occupancy* information helps the agent learn to avoid obstacles on its path. In such cases, the spatial map stores a binary occupancy value of 1 or 0 depending on whether the corresponding location in the physical environment is 'occupied' by objects or not.

In an exploration task, however, the agent needs to maximize the explored area in an environment while being efficient, in which cases, the information of whether a location has already been explored encourages the agent to explore unexplored areas more. Active Neural SLAM (Chaplot et al., 2019) stores the *explored* (binary) information in addition to the occupancy information. Active Neural SLAM consists of several modules connected together to perform the task of Exploration to maximize the explored area. The 'Neural SLAM' module takes as inputs the visual observations and agent pose and outputs a top-down egocentric spatial map by learning to predict occupancy and explored information using a binary cross-entropy loss. All the modules are jointly trained with different losses. VLMNav (Goetting et al., 2024) similarly stores *explored* information in a top-down voxel map to demonstrate its generalizability to downstream navigation tasks.

However, in more complex tasks such as ObjectNav, which requires the agent to navigate to a particular semantic category in its environment, storing occupancy and explored information alone might not be sufficient. In such cases, it is important for the agent to identify the semantic category of the object. Goal-oriented Semantic Exploration (SemExp) by Chaplot et al. (2020a) additionally stores the semantic class labels of the objects that the agent identifies through its visual observations. SemExp uses a MaskRCNN (He et al., 2017) on the RGB observations to predict the semantic categories of the objects and then project these on the map using depth observations. It aggregates the occupancy and explored information using element-wise max pooling whereas the semantic categories are overwritten by the latest prediction. A denoising network is then used to get the final map. This work demonstrates that using this semantic map to predict a long-term goal helps the agent find the goal object category more efficiently. The mapping module is trained using supervised learning with cross-entropy loss on the predicted semantic map. Similarly, a semantic categories map also helps a more complex longer-horizon task of MultiON. MOPA (Raychaudhuri et al., 2023) shows that maintaining a memory with objects that the agent observes while moving around is crucial to perform this task efficiently. However, a map built by segmenting the image first and then projecting may result in 'label splattering' *i.e.* noisy category labels splattered across multiple grid cells in a spatial map. This arises mostly due to noisy depth observations and might negatively affect agent performance in a task. Semantic MapNet (Cartillier et al., 2021) finds that projecting encoded features on to a map and then segmenting produces a more noise-free map. They show that this map can be then be applied to two different tasks effectively. However this method requires an additional exploration phase where the agent effectively explores the environment to build the map first and then use that map in the downstream task. A recent work GOAT (Chang et al., 2023; Khanna et al., 2024) shows that having an object instance map helps to navigate to a goal specified by either language, image or a category label. They achieve this by storing raw images and later using CLIP (Khandelwal et al., 2022) for image-to-language matching and SuperGlue (Sarlin et al., 2020) for image-to-image matching. MapNav (Zhang et al., 2025a) builds an annotated semantic map by storing text labels of the segmented objects and shows that this helps a VLM to ground objects better.

While a map with semantic categories help in object navigation, an acoustic map storing audio intensity is found to be useful in the Audio-Visual Navigation task (Chen et al., 2020b). Here the map is aggregated by averaging the intensity. In the more complex Interactive Question Answering task, Gordon et al. (2018) find that storing object detection probabilities in a spatial map helps agent performance. They use a GRU recurrent memory to aggregate the current map with the previous one.

While the above works build explicit spatial maps, some works also build explicit topological maps. Kwon et al. (2021) store the visitation timestep in the graph node in order to encode a temporal relation between visited locations. This information is then replaced by the latest visitation timestep while aggregating the map.

Summary. The advantage of explicit map encoding is its interpretability and the fact that it allows investigating the type of information that is beneficial for various downstream tasks. However, the type of semantic information to be stored is a design choice based on the task. Moreover, the above approaches require a predefined set of categories to be mentioned beforehand to the mapper. This restricts the map to store only a limited number of object categories.

5.2 Implicit encoding

Implicit maps store latent features in a semantic map. While most of the prior works use extracted features from a vision model pre-trained on a closed-vocabulary set of object categories, recent methods use features extracted from a pre-trained large vision-language model thus producing flexible open-vocabulary queryable maps. It is also possible to store features that are not necessarily queryable with language, but captures the visual information at that location. One example is RNR-Map (Kwon et al., 2023) which uses a grid map with latent codes that corresponds to a neural field that can be used to render possible views at that location.

5.2.1 Closed-vocabulary encoding

These features can be learned from scratch during training. For example, (Wani et al., 2020) learn image features using CNN blocks and use them to build a global spatial map of the environment. This is trained end-to-end to predict actions in the MultiON task. On the other hand, the features may also be extracted from a vision model such as ResNet, pre-trained on the ImageNet (Deng et al., 2009) data to encode RGB images. For example, (Gupta et al., 2017) introduces Cognitive Mapper and Planner (CMP) that uses a pretrained ResNet-50 model to encode the egocentric RGB images and then projecting on the map using a differentiable mapper module. In CMP, the learned map encoding is not explicitly supervised but learned in conjunction with a differentiable planner. This enables the mapper to learn to store information that is most useful for the planner to perform the task efficiently. Also the map is accumulated over time meaning that the map from one navigation step is integrated into the next using a differentiable warp. Similarly MapNet (Henriques & Vedaldi, 2018) also uses a pretrained ResNet-50 model to extract image features but they build an allocentric global map of the environment instead of an egocentric map. They do so by first ground projecting the features and then registering these into a global allocentric map, updating the map at every navigation step. This model is learned end-to-end on the task of localization and trained with a series of RGB-D observations and the corresponding ground-truth positions and orientations.

While the above methods build a spatial map, other methods use a similar approach to store implicit features in the nodes of a topological map. Each node stores encoded features from the observations at a particular location in the environment. Here too, using a pretrained ResNet encoder to extract RGB image features is a popular choice among prior works (Chaplot et al., 2020b; Chen et al., 2021).

Summary. The advantage of using a pretrained ResNet model over learning from scratch is that it has already been trained to encode useful features and is thus sample efficient. However, using a pre-trained ResNet is limited by the number of object categories it was trained on.

Table 5: **Open-vocabulary maps.** Various works build open-vocabulary semantic map using trained as well as off-the-shelf pretrained models in both simulation and real-world robots. These methods use either heuristics-based or LLM-based planner to perform the downstream taskss.

Method	Environment	Training	VL Encoder	Task Planner	Aggregation
OneMap (2024)	Habitat, robot	✗	SED	A*	weighted sum with uncertainty-based weights
CoW (2023)	Habitat, RoboTHOR	✓	CLIP	A*	highest similarity score
VLMap (2023a)	Habitat, robot	✗	LSeg	A*	mean
NLMap (2023a)	robot	✗	CLIP, ViLD	LLM-based	multi-view fusion
ConceptGraphs (2023)	AI2Thor, robot	✗	CLIP, DINO	GPT-4	highest similarity score
VoxPoser (2023b)	Sapien, robot	✗	OWL-ViT	GPT-4	-
VLFM (2023)	Habitat, robot	✗	BLIP-2	pretrained PointNav	highest similarity score
CLIP-Fields (2023)	Habitat, robot	✓	CLIP	SLAM	weighted mean

5.2.2 Open-vocabulary encoding

The limitation of the closed-vocabulary encoding can be mitigated by extracting features from a Large Vision-Language Model (LVLM) that was jointly trained on a vast amount of internet data of images and their text captions, such as CLIP (Radford et al., 2021). This allows the map to store information about 'any' object in the environment and eventually be queried via an open-vocabulary text query not limited to a predefined set of object categories. For example, an agent may be asked to 'find a *red and blue striped zebra toy* in the children's room'. It is likely that the agent has not seen a 'red and blue striped zebra toy' during training but can leverage a LVLM to reason about its prior knowledge about zebras, colors and rooms in general.

Moreover, recently large language models (LLMs), such as GPT-4 (OpenAI, 2023), have been shown to be able to perform complex task planning. Thus open-vocabulary maps built using LVLM along with LLM-based planners have led to a recent line of works in embodied agents (Tab. 5, Fig. 8).

A pretrained CLIP model can be used to compute similarity scores between an input image and a natural language description with the highest score corresponding to the most likely image-text match. CLIP has been successfully used in map building where the map stores the similarity scores between each image that the agent observes and the language instruction describing an object. Popularly, these maps are called *value maps*. CoW (Gadre et al., 2023) shows that such a 2D value map can be successfully applied in the downstream task of language-driven ObjectNav in a zero-shot manner without any re-training. The planner in this method plans a path to the object when the stored similarity score exceeds a certain threshold. VLFM (Yokoyama et al., 2023) follows a similar strategy to perform ObjectNav task by using the BLIP-2 (Li et al., 2023) 2D value map to semantically explore the environment. InstructNav (Long et al., 2024) extends this idea to enhance semantic value maps with multi-sourced value maps encoding actions, landmarks, and navigation history, thus improving generic instruction following. VoxPoser (Huang et al., 2023b), on the other hand, builds a similar 3D value map to efficiently perform table-top robot manipulation. Such map encoding is quite powerful compared to the previous encodings that stores the semantic labels for a predefined set of objects. However, this particular map still lacks the ability to perform spatial reasoning because it performs similarity matching to the entire input image ignoring the semantic information about individual objects in the scene.

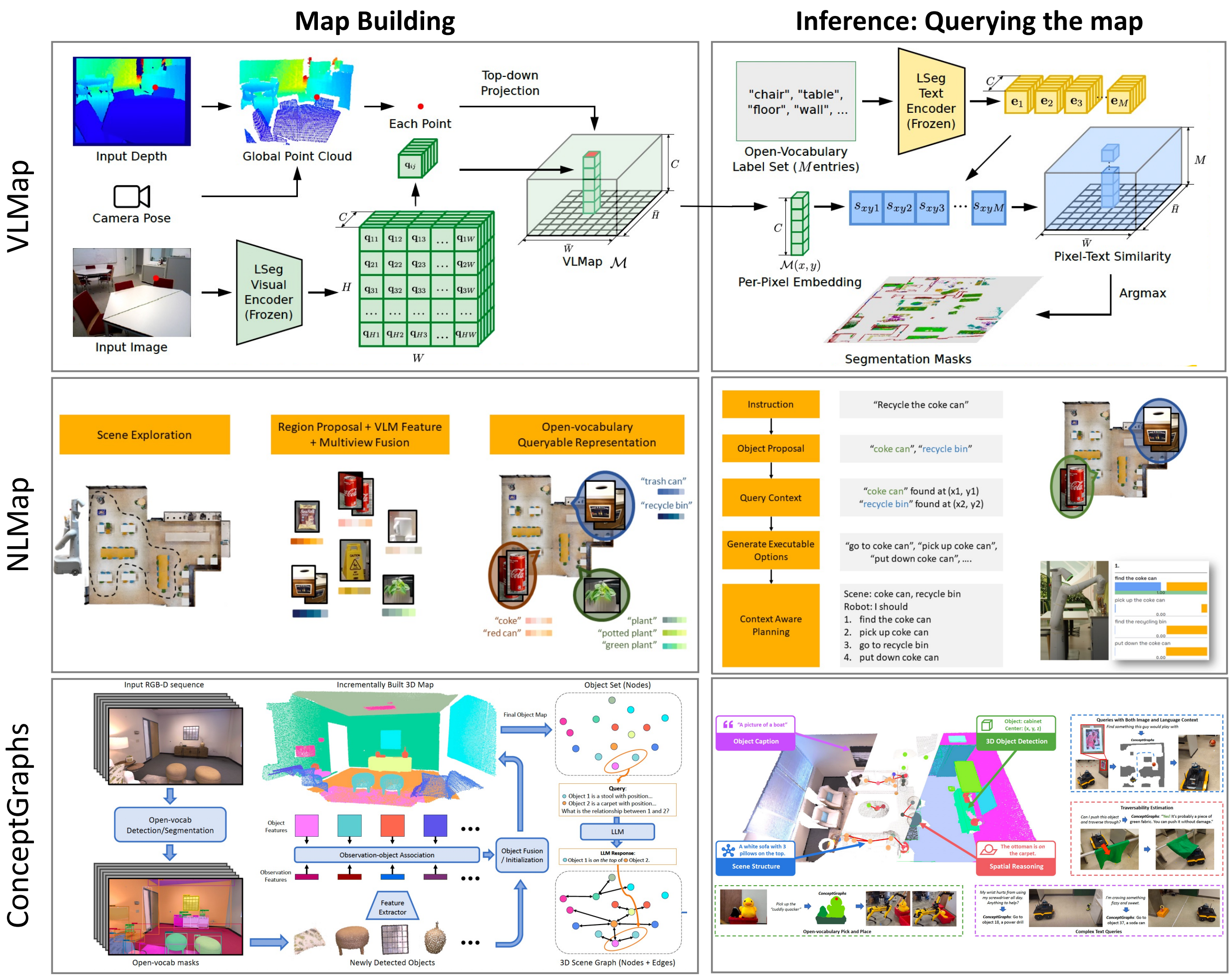

Figure 8: **Open-Vocabulary map building.** There has been a growing interest to build flexible open-vocabulary maps which can be built once and then used in various downstream tasks during inference. VLMap (Huang et al., 2023a) and NLMap (Chen et al., 2023a) structure their maps as spatial maps while ConceptGraphs (Gu et al., 2023) build a topological map. *(Figures reproduced from paper.)*

To mitigate this issue, researchers need to develop methods that identify where objects (Zhang et al., 2025b) are located in the image, and then extract features for those objects. One way to achieve this is to store feature embeddings for each pixel in the input image. This can be done using LSeg (Li et al., 2022) which outputs pixel-level embeddings given an image as is done in VLMaps (Huang et al., 2023a). The embeddings are then projected into a 2D spatial map using depth observations. When multiple points are projected onto the same grid cell on the map, VLMaps aggregates by averaging the features. During inference, they extract object names from the language query and calculate pixel-text similarity scores on the map to retrieve the objects of interest. However, these pixel-level embeddings are very dense and can be redundant since not all pixels in an image contain objects. Second, some information might be lost while averaging the pixel-level embeddings. A third issue is that it does not encode object-level semantics, thus ignoring information about spatial relationships. OneMap (Busch et al., 2024) stores patch-level features extracted from SED (Xie et al.,

2024) with a hierarchical encoder-based backbone which has shown to capture spatial information better than the transformer-based architecture. This alleviates the issues of using pixel-based features.

However, a better way is to first identify all objects present in an image, which can be done by using a class-agnostic region proposal network to propose regions of interest (objects) in an image. This approach is used in NLMap (Chen et al., 2023a) which uses ViLD (Gu et al., 2021) as its class agnostic region proposal network. For each region of interest, they extract image embeddings using an ensemble of CLIP and ViLD. These features are then stored in a 3D spatial map along with the 3D location and estimated size of the object. They show that this spatial map representation can be applied to any downstream task by performing a natural language based query. This is achieved by extracting object names from the query and using them to select the object from the map with the highest similarity scores.

While NLMap builds a 3D spatial map, ConceptGraphs (Gu et al., 2023) build a topological map following a similar approach. It retrieves the objects of interest from the input image by using the class-agnostic 2D segmentation model, Segment-Anything (SAM) (Kirillov et al., 2023) to obtain candidate masks. These objects then form the nodes of the topological map. The image features for each object can then be extracted by CLIP and DINO (Oquab et al., 2023) to be stored in the corresponding node. Additional information about the objects can also be stored in each node. For example, ConceptGraphs store the point cloud of the proposed masks and a caption for each object as obtained by using LLaVA (Liu et al., 2023a) and GPT-4 (OpenAI, 2023) along with a point cloud obtained by projecting the object mask proposed by SAM into 3D space. After the nodes are formed, edges between the nodes can be constructed depending on whether the objects are spatially related. In ConceptGraphs, spatial relation between two objects is determined by whether the point clouds of the respective objects have a geometric similarity or overlap. In other words, when a certain proportion of points in point cloud of one object lie within a distance threshold of that of the second object, the objects can be said to be spatially related to each other and an edge is constructed between the corresponding nodes. The edges additionally store a spatial relationship description obtained from LLM. When a newly detected object is found to be similar to a node, it is updated with averaged object features, union of point clouds and the latest object caption. In case it is not similar, a new node is added to the graph. ConceptGraphs show that a single map representation built in this manner can be successfully and effectively used in object grounding, robot navigation and robot manipulation tasks.

Summary. The advantage of an open-vocabulary map encoding is that it can be built once and then transferred to several different downstream tasks. It can be queried using an open-vocabulary text effectively and is highly interpretable. However one current limitation is that the computational costs of using large foundation models can be significant.

6 Map Evaluation

In the embodied AI literature, very little focus has been given to evaluating the built map. Instead the focus has been to evaluate the agent performance on the downstream tasks through various metrics. However, it is crucial to evaluate the semantic maps in terms of *accuracy*, *completeness*, *consistency* and *robustness* in addition to its *utility* on the downstream tasks. We next discuss these evaluation metrics in detail.

Utility. Most works build semantic maps as an intermediate step while performing downstream tasks, such as navigation, exploration, manipulation and so on. In such cases, the map is exploited by the task planner to plan a path and generate low-level actions in order to complete the task. Moreover, most of these works focus on a single task and hence it suffices for them to evaluate the task performance directly without caring about how well the map representation is. Gupta et al. (2017); Kim et al. (2023) build map for navigation tasks, while Chaplot et al. (2019) builds map to efficiently explore the environment. Although the open-vocabulary map in Gu et al. (2023) is pre-built once and used in multiple downstream tasks (navigation, manipulation, object segmentation etc.), they still evaluate the task performance and not the map itself. The navigation tasks are evaluated on Success and SPL (Anderson et al., 2018a), while the exploration tasks are evaluated on the coverage area (Cov) and the percentage of area explored (% Cov) (Chaplot et al., 2019).

Accuracy. Map accuracy refers to how accurately the map captures semantic information when compared against the ground truth. However obtaining the ground truth map can be challenging in most cases. Cartillier et al. (2021) and Georgakis et al. (2022b) evaluate the accuracy of the built 2D semantic map using semantic segmentation metrics such as pixel-wise labeling accuracy, pixel-based F1 score, Intersection-over-Union (IoU) score and a contour-based average boundary F1 score. While evaluating accuracy for semantic maps that store a fixed set of object categories is straightforward, it could be tricky to evaluate semantic maps that store implicit features. For example, OpenScene (Peng et al., 2023) performs semantic segmentation on their implicit map for a fixed set of categories and then evaluate it using accuracy and IoU. ConceptGraphs (Gu et al., 2023), on the other hand, evaluate their open-vocabulary topological map by employing human evaluators on Amazon Mechanical Turk (AMT) to report the scene graph accuracy.

Completeness. Map completeness measures how completely a generated map represents an environment, encompassing both geometric and semantic coverage. Geometric coverage refers to the fraction of environment that has been mapped whereas semantic coverage refers to the completeness of the semantic information captured in the map. Completeness is particularly critical for tasks like search and rescue, where a thorough understanding of the surroundings is essential. Moreover, complete geometric and semantic mapping support better decision-making by reducing dependence on incomplete or inaccurate data. The extent of map completeness depends on how thoroughly the robot explores the environment during the downstream task and is closely tied to the 'stopping criteria' – a method that determines when to end the exploration process. In embodied AI, exploration typically ends when the task is deemed complete or when a predefined time budget is reached. However, reaching this time budget does not always guarantee that the map is fully complete, making the development of reliable stopping criteria an ongoing challenge in robotics as well as embodied AI research (Placed & Castellanos, 2022; Luperto et al., 2024). Geometric coverage metric has been reported in prior works that performs the task of exploration (Chaplot et al., 2019) in terms of the fraction of the environment explored by the robot. However, a lot of works that tackle other embodied AI tasks fail to report this metric even if exploration is a crucial part of their task (Gervet et al., 2023; Raychaudhuri et al., 2023; Yokoyama et al., 2023). Semantic Coverage on the other hand relies on the presence of a detailed ground-truth map with semantic information and could be hard to obtain, as discussed in the previous paragraph 'Accuracy'. Although some works (Cartillier et al., 2021; Georgakis et al., 2022b) report semantic accuracy, none report semantic coverage of the built map.

Consistency. *Geometric consistency* of a map refers to how accurately the spatial structure (distances, angles, and relative positions of structures/objects) of the map represents the physical layout of the environment. Accurate geometry ensures safe and efficient path planning and obstacle avoidance. This is crucial in classical robotics for identifying loop closure, a popular technique to recognize previously visited locations in SLAM, which refines the map and reduces drift. However, in embodied AI systems, there is no drift due to absence of sensory and actuator noise and hence the generated maps are mostly consistent with the environment. Hence prior embodied AI works do not report geometric consistency metrics for the built map. That said, ensuring geometric consistency will be crucial in future where dynamic moving objects may disrupt the map's structural fidelity. It can be measured by reporting the Root Mean Square Error (RMSE) from the ground-truth geometry. On the other hand, the *semantic consistency* of a map refers to the alignment between the semantic information of structures/objects and their physical locations in the environment. Semantic consistency is particularly crucial to remain consistent over time despite changes in perspective, lighting, or environment dynamics when the robot moves around the environment. This could be measured using a temporal accuracy metric that measures how accuracy of the semantic information changes over time. However, none of the prior works measure semantic consistency and may be considered in future research.

Robustness. Evaluating robustness in semantic maps is mostly crucial for assessing their reliability in unpredictable or dynamic environments. A robust map exhibits low uncertainty and high confidence in its semantic information, allowing the robot to adapt to errors, sensor noise, and environmental changes. Since recent approaches use pretrained models, measuring the confidence of the model's predictions could be beneficial to assess map robustness, with higher confidence indicating a more robust map. Alternatively, model uncertainty can be measured by assessing the variance in model predictions, reflecting epistemic uncertainty in the model, where low uncertainty will indicate a robust map. While some studies (Georgakis et al., 2022a; Raychaudhuri et al., 2024) incorporate uncertainty into task planning, it has not been used as a formal metric in previous research, presenting an area for future exploration.

In summary, there is considerable scope for future research to develop better metrics that evaluate the built semantic map and not just the downstream task performance. This can further be improved by developing a standardized evaluation framework for semantic maps.

7 Future research direction

In this section we highlight the current challenges in semantic map building and outline potential directions for future research. Although semantic map building has advanced significantly over the past decades, several challenges and opportunities for improvement remain. The field is evolving toward creating maps that are flexible, general-purpose, open-vocabulary and queryable, enabling the same map representation to support a wide range of downstream tasks. This shift aims to make maps more versatile and suitable for complex, multi-task robotic systems. Moreover, to enable efficient reasoning about the spatial and semantic structure of the environment, the focus is also on developing dense, scalable, and memory-efficient maps. Such maps should maintain high resolution and detailed spatial information while being computationally efficient and consistent across dynamic and large-scale environments. Achieving this balance is critical for applications that require real-time processing or operate in resource-constrained settings. Furthermore, the emphasis has largely been on evaluating agent performance in downstream tasks using various metrics, with limited attention given to assessing the quality of the built maps. It is however essential to evaluate semantic maps beyond their utility for specific tasks, focusing on metrics such as accuracy, completeness, consistency, and robustness to ensure they are reliable and effective for broader applications. Next, we provide an in-depth discussion of potential future directions that we believe are most critical for advancing research in semantic mapping.

7.1 General-purpose maps

Creating general-purpose semantic maps in robotics and embodied AI is crucial for enabling robots to perform a wide variety of tasks in diverse environments with minimal reconfiguration. The idea is to design a general-purpose semantic map that serves as a single comprehensive representation of the environment, combining spatial geometry and semantic information. This eliminates the need for task-specific maps, making it easier to reuse the same map for different tasks such as navigation, object manipulation, and scene understanding. To enable this, the maps need to be open-vocabulary that allow the robots to understand and integrate previously unseen objects using natural language descriptions. This capability broadens the scope of the downstream tasks the robot can perform, especially in unstructured or novel environments. Such semantic maps provide an opportunity to thoroughly evaluate their ability to handle textual queries involving complex spatial and semantic reasoning. Despite the recent progress towards achieving this, it still remains an open research problem. Open-vocabulary maps are currently limited by the pretrained class-agnostic object detectors used in building such maps. For example, these detectors often struggle with detecting small, thin or obscure objects, thus limiting the semantic maps that rely on them. Moreover, open-vocabulary object detectors that incorporate unseen classes through textual descriptions are still not perfect and lack robustness. Thus improving open-vocabulary object detectors that can recognize new objects without extensive retraining could improve the quality of semantic maps that rely on them and hence present a future research avenue. Another challenge that general-purpose map building face is that they are computationally expensive and memory-intensive due to the rich semantic and geometric data stored in them. Balancing map detail with resource efficiency is a challenging task and impacts the ability of robots to process large areas or continuously update the map in real-time without overwhelming computational resources. This presents a future research direction worth pursuing.

7.2 Dense yet efficient maps

Following our discussion on various map structures, we find it crucial that the semantic map be able to capture dense visual cues to allow for complex spatial reasoning among objects. For example, beyond addressing straightforward queries like ‘Where is the table?’, the semantic maps can be assessed on more intricate spatial reasoning queries such as ‘Can you retrieve my phone from my desk beside my laptop?’. To perform such reasoning, the semantic map needs to capture fine-grained detail about the spatial arrangement of the objects (phone and laptop on the table) in the scene. While at one end of the spectrum topological maps are too sparse failing to capture the dense semantics of a scene, at the other end the point-cloud based maps are too dense capturing redundant empty space information. Although spatial top-down 2D maps exist somewhere in the middle, they fail to capture the semantic information in the third dimension (height). This is crucial where the navigation is in 3D space, for example in drones. Hence there is still a need of a dense-enough map to capture the 3D space in its entirety at the same time intelligently ignoring empty space information. Additionally, a dense map will consume more memory and will be difficult to update. So a dense map representation which is still scalable, memory and computation efficient is a research direction worthwhile to pursue.

7.3 Dynamic maps

Current mapping techniques in indoor environments assume that the objects present in the environment are static and only the agent is moving. Although this assumption is reasonable in an indoor environment, it is unrealistic in an outdoor setting in the presence of moving vehicles and people. This entails investigating how well current map building approaches capture moving objects effectively in a dynamic environment and focus on building efficient dynamic maps. Building such maps involves continuous tracking and updating of objects in real-time, as they may move unpredictably. Sensor fusion, which integrates data from multiple sensors like LiDAR and cameras, is often used to detect and track these objects. However, real-time updates can be computationally intensive, particularly in high-traffic areas. Thus, efficiently storing and representing dynamic data in a way that is both memory and computationally scalable remains a key challenge and an ongoing area of research. Moreover, the dynamic nature of these maps complicates their use in downstream tasks. For instance, a robot may need to navigate around a moving pedestrian or vehicle, which requires understanding the object's trajectory and predicting its future movements to avoid collisions. Efficiently integrating this dynamic data into decision-making processes is a significant challenge in autonomous navigation.

7.4 Hybrid map structure

A spatial map is able to capture the geometry of a 3D space, which helps to reason about complex spatial relations among objects and areas. A topological map, on the other hand, lacks such geometric spatial understanding but is able to explicitly capture semantic relationships (edges) among objects (nodes). Since both structures have different merits, there have been research to explore a 'hybrid' map structure that leverages the geometric accuracy of spatial maps with the semantic and relational power of topological maps, providing a more comprehensive and efficient tool for complex reasoning tasks. For example, in a large-scale outdoor environment, a robot could use the topological map for long-range navigation to reach one building from another, while switching to a spatial map for close-range navigation to avoid obstacles or interact with objects in a room. A hybrid approach can also help balance the computational load. Topological maps are less resource-intensive and can provide high-level guidance, while spatial maps can be used for precise actions in local regions of interest, reducing the need for continuous, high-cost processing across the entire environment. Moreover, maintaining a separate level of hierarchy to track dynamic objects can also reduce computation load of frequent real-time updates. Although several approaches to hybrid mapping have been proposed in recent years, one of the main challenges in creating hybrid maps is effectively integrating both spatial and topological representations without sacrificing the quality of either. Moreover, combining the two representations requires the robot to make intelligent decisions about when to transition from one to another. Hence significant research is still needed in optimizing hybrid map building, ensuring scalability for large-scale, real-time applications and intelligent algorithms to transition between the different maps.

7.5 Devising evaluation metrics

As we discuss in Sec. 6, the evaluating semantic maps in embodied AI research has received limited attention compared to assessing agent performance in downstream tasks. However, we believe that advancing the field requires a stronger emphasis on map evaluation using metrics such as accuracy, completeness, consistency, and robustness. Regardless of the downstream task, maps should be assessed on how well they capture semantic information (accuracy), their geometric and semantic coverage (completeness), their spatial and semantic reliability in dynamic environments (consistency), and their confidence and ability to handle uncertainty and noise (robustness). Establishing standardized evaluation metrics and frameworks for semantic maps remains a critical challenge with substantial opportunities for future research.

8 Conclusion

In this survey, we explore various approaches to semantic map building in the Embodied AI literature, focusing on indoor environments. Existing works primarily employ spatial maps to capture geometric layouts or topological maps to model landmark-based relationships. While many robotics studies have explored hybrid maps that combine spatial and landmark information, this approach remains under-explored in embodied AI research. It presents a promising avenue for future work, particularly in leveraging such maps to enhance performance on complex spatial reasoning tasks. We also discuss how dense point-cloud maps, created by associating semantic information with point clouds or triangle meshes, offer potential for embodied AI tasks but remain under-explored. While promising for spatial reasoning, their high memory demands, computational inefficiency, and the presence of mostly empty 3D space in indoor environments pose significant challenges. With recent advances in large foundation models, the focus has shifted towards building open-vocabulary, queryable map representations that are task-agnostic. Despite these advances, learning to map an indoor 3D scene in simulation rely on unrealistic assumptions such as perfect localization and noiseless sensors, leading to a significant sim-to-real gap. Moreover, these approaches assume static environments, limiting their applicability in dynamic outdoor settings like autonomous driving.

Building on the insights from this survey, we highlight the limitations in existing semantic map-building methods and explore promising directions for future research. We aim for this discussion to not only summarize the current state of the field but also inspire the research community to advance it further.

References

Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Košecká, and Alexander C Berg. A dataset for developing and benchmarking active vision. In *IEEE Int. Conf. on Robotics and Automation*, pp. 1378–1385. IEEE, 2017. 14

Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. BEVBert: Multimodal map pre-training for language-guided navigation, 2023. URL https://arxiv.org/abs/2212.04385. 8, 17

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. ETPNav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE TPAMI*, 2024. 14

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018a. 22

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pp. 3674–3683, 2018b. 1, 3

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015. 11

Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1534–1543, 2016. 14

Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3D scene graph: A structure for unified semantics, 3D space, and camera. In *ICCV*, pp. 5664–5673, 2019. 17

Zhibin Bao, Sabir Hossain, Haoxiang Lang, and Xianke Lin. A review of high-definition map creation methods for autonomous driving. *Engineering Applications of Artificial Intelligence*, 122:106125, 2023. 1

Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020a. 1, 3

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020b. 3

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008. 6

Jose-Luis Blanco, Juan-Antonio Fernández-Madrigal, and Javier Gonzalez. Toward a unified bayesian approach to hybrid metric–topological slam. *IEEE Transactions on Robotics*, 24(2):259–270, 2008. 17

Sean L Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J Pappas. Probabilistic data association for semantic SLAM. In *ICRA*, pp. 1722–1729. IEEE, 2017. 15

Finn Lukas Busch, Timon Homberger, Jesús Ortega-Peimbert, Quantao Yang, and Olov Andersson. One map to find them all: Real-time open-vocabulary mapping for zero-shot multi-object navigation, 2024. URL https://arxiv.org/abs/2409.11764. 20, 21

Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 14

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021. 5

Vincent Cartillier, Zhile Ren, Neha Jain, Stefan Lee, Irfan Essa, and Dhruv Batra. Semantic MapNet: Building allocentric semantic maps and representations from egocentric views. In *AAAI*, 2021. 8, 9, 12, 13, 14, 19, 22, 23

Edgar Chan, Oliver Baumann, Mark Bellgrove, and Jason Mattingley. From objects to landmarks: The function of visual location information in spatial navigation. *Frontiers in Psychology*, 3:304, 08 2012. doi: 10.3389/fpsyg.2012.00304. 14

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *Intl. Conf. on 3D Comput. Vis.*, 2017. 13, 14

Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, et al. GOAT: Go to any thing. *arXiv preprint arXiv:2311.06430*, 2023. 19

Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural SLAM. In *ICLR*, 2019. 3, 4, 5, 8, 14, 18, 22, 23

Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *NeurIPS*, volume 33, pp. 4247–4258, 2020a. 4, 5, 8, 11, 13, 14, 18

Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological SLAM for visual navigation. In *CVPR*, 2020b. 3, 4, 8, 14, 15, 20

Devendra Singh Chaplot, Murtaza Dalal, Saurabh Gupta, Jitendra Malik, and Russ R Salakhutdinov. SEAL: Self-supervised embodied active learning using exploration and 3D consistency. *NIPS*, 34:13086–13098, 2021. 13

Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. In *ICRA*, pp. 11509–11522. IEEE, 2023a. 2, 8, 20, 21, 22

Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. SoundSpaces: Audio-visual navigation in 3D environments. In *ECCV*, pp. 17–36. Springer, 2020a. 3

Changan Chen, Sagnik Majumder, Ziad Al-Halah, Ruohan Gao, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Learning to set waypoints for audio-visual navigation. In *ICLR*, 2020b. 19

Junting Chen, Guohao Li, Suryansh Kumar, Bernard Ghanem, and Fisher Yu. How to not train your dragon: Training-free embodied object navigation with semantic frontiers. In *RSS*, 2023b. 8, 17

Kevin Chen, Junshen K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *CVPR*, pp. 11276–11286, 2021. 8, 14, 15, 20

Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. CLIP2Scene: Towards label-efficient 3D scene understanding by CLIP. In *CVPR*, pp. 7020–7030, 2023c. 8, 16

Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *ICLR*, 2019. 5

Weifeng Chen, Guangtao Shang, Aihong Ji, Chengjun Zhou, Xiyang Wang, Chonghui Xu, Zhenxiong Li, and Kai Hu. An overview on visual SLAM: From tradition to semantic. *Remote Sensing*, 14(13):3010, 2022. 6

Matt Deitke, Dhruv Batra, Yonatan Bisk, Tommaso Campari, Angel X Chang, Devendra Singh Chaplot, Changan Chen, Claudia Pérez D'Arpino, Kiana Ehsani, Ali Farhadi, et al. Retrospectives on the embodied AI workshop. *arXiv preprint arXiv:2210.06849*, 2022. 1, 2, 3

Frank Dellaert, Steven Seitz, Sebastian Thrun, and Charles Thorpe. Feature correspondence: A markov chain monte carlo approach. *Advances in Neural Information Processing Systems*, 13, 2000. 15

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009. 19

Matej Dobrevski and Danijel Skočaj. Deep reinforcement learning for map-less goal-driven robot navigation. *International Journal of Advanced Robotic Systems*, 18(1):1729881421992621, 2021. 4

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied AI: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2): 230–244, 2022. 2

Russell A Epstein and Lindsay K Vass. Neural systems for landmark-based wayfinding in humans. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1635):20120533, 2014. 14

Nathaniel Fairfield. Localization, mapping, and planning in 3D environments. *Ph. D. dissertation*, 2009. 5, 6

David Filliat and Jean-Arcady Meyer. Map-based navigation in mobile robots:: I. a review of localization strategies. *Cognitive systems research*, 4(4):243–282, 2003. 2

Tobias Fischer, Lorenzo Porzi, Samuel Rota Bulo, Marc Pollefeys, and Peter Kontschieder. Multi-level neural scene graphs for dynamic urban environments. In *CVPR*, pp. 21125–21135, 2024. 17

Patrick Foo, William H Warren, Andrew Duchon, and Michael J Tarr. Do humans integrate routes into a cognitive map? map-versus landmark-based navigation of novel shortcuts. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(2):195, 2005. 14

Scott Fredriksson, Akshit Saradagi, and George Nikolakopoulos. Semantic and topological mapping using intersection identification. *IFAC-PapersOnLine*, 56(2):9251–9256, 2023. 14

Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *CVPR*, pp. 23171–23181, 2023. 5, 8, 20

Chuang Gan, Yiwei Zhang, Jiajun Wu, Boqing Gong, and Joshua B Tenenbaum. Look, Listen, and Act: Towards Audio-Visual Embodied Navigation. *arXiv preprint arXiv:1912.11684*, 2019. 3

Sourav Garg, Krishan Rana, Mehdi Hosseinzadeh, Lachlan Mares, Niko Sünderhauf, Feras Dayoub, and Ian Reid. Robohop: Segment-based topological map representation for open-world visual navigation. In *IEEE Int. Conf. on Robotics and Automation*, 2024. 8, 14, 15

JS Gautham, Akash Sharma, Samiappan Dhanalakshmi, and Kumar Ramamoorthy. 3D scene reconstruction and mapping with real time human detection for search and rescue robotics. In *AIP Conference Proceedings*, volume 2427. AIP Publishing, 2023. 1

Georgios Georgakis, Bernadette Bucher, Anton Arapin, Karl Schmeckpeper, Nikolai Matni, and Kostas Daniilidis. Uncertainty-driven planner for exploration and navigation. In *International Conference on Robotics and Automation (ICRA)*, pp. 11295–11302. IEEE, 2022a. 23

Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. Learning to map for active semantic goal navigation. In *ICLR*, 2022b. 11, 23

Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. Navigating to objects in the real world. *Science Robotics*, 8(79):eadf6991, 2023. 4, 5, 23

Dylan Goetting, Himanshu Gaurav Singh, and Antonio Loquercio. End-to-End Navigation with Vision Language Models: Transforming Spatial Reasoning into Question-Answering. *arXiv preprint arXiv:2411.05755*, 2024. 18

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. IQA: Visual Question Answering in interactive environments. In *CVPR*, pp. 4089–4098, 2018. 1, 11, 19

Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv*, 2023. 2, 8, 11, 14, 20, 21, 22, 23

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2021. 22

Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic Gaussians: Open-Vocabulary Scene Understanding with 3D Gaussian Splatting. *arXiv preprint arXiv:2403.15624*, 2024. 16

Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *CVPR*, pp. 2616–2625, 2017. 4, 8, 11, 12, 13, 14, 19, 22

Nick Hawes, Christopher Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrova, Jay Young, Jeremy Wyatt, Denise Hebesberger, Tobias Kortner, et al. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine*, 24(3):146–156, 2017. 1

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016. 4, 5, 13

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, pp. 2961–2969, 2017. 4, 5, 7, 11, 18

Joao F Henriques and Andrea Vedaldi. MapNet: An allocentric spatial memory for mapping environments. In *CVPR*, pp. 8476–8484, 2018. 4, 8, 9, 13, 14, 19

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *ICRA*, pp. 10608–10615. IEEE, 2023a. 8, 13, 20, 21

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3D value maps for robotic manipulation with language models. In *CoRL*, 2023b. 20, 21

Nathan Hughes, Yun Chang, Siyi Hu, Rajat Talak, Rumaia Abdulhai, Jared Strader, and Luca Carlone. Foundations of spatial perception for robotics: Hierarchical representations and real-time systems. *The International Journal of Robotics Research*, pp. 02783649241229725, 2024. 17

Gabriele Janzen and Miranda Van Turennout. Selective neural representation of objects relevant for navigation. *Nature neuroscience*, 7(6):673–677, 2004. 14

Collin Eugene Johnson. *Topological Mapping and Navigation in Real-World Environments*. PhD thesis, University of Michigan, 2018. 14

András Kalapos, Csaba Gór, Róbert Moni, and István Harmati. Sim-to-real reinforcement learning applied to end-to-end vehicle control. In *International Symposium on Measurement and Control in Robotics (ISMCR)*, pp. 1–6. IEEE, 2020. 4

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LeRF: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19729–19739, 2023. 16

Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: CLIP embeddings for embodied AI. *CVPR*, 2022. 19

Piyush Khandelwal, Shiqi Zhang, Jivko Sinapov, Matteo Leonetti, Jesse Thomason, Fangkai Yang, Ilaria Gori, Maxwell Svetlik, Priyanka Khante, Vladimir Lifschitz, et al. Bwibots: A platform for bridging the gap between ai and human–robot interaction research. *The International Journal of Robotics Research*, 36 (5-7):635–659, 2017. 1

Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. Goat-bench: A benchmark for multi-modal lifelong navigation. In *CVPR*, pp. 16373–16383, 2024. 8, 14, 19

Nuri Kim, Obin Kwon, Hwiyeon Yoo, Yunho Choi, Jeongho Park, and Songhwai Oh. Topological semantic graph memory for image-goal navigation. In *CoRL*, pp. 393–402. PMLR, 2023. 14, 15, 22

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment Anything. *arXiv preprint arXiv:2304.02643*, 2023. 7, 22

Dong Wook Ko, Chuho Yi, and Il Hong Suh. Semantic mapping and navigation: A Bayesian approach. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2630–2636. IEEE, 2013. 17

Kurt Konolige, Eitan Marder-Eppstein, and Bhaskara Marthi. Navigation in hybrid metric-topological maps. In *ICRA*, pp. 3041–3047. IEEE, 2011. 17

Ioannis Kostavelis and Antonios Gasteratos. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 66:86–103, 2015. 2, 3

Obin Kwon, Nuri Kim, Yunho Choi, Hwiyeon Yoo, Jeongho Park, and Songhwai Oh. Visual graph memory with unsupervised representation for visual navigation. In *ICCV*, pp. 15890–15899, 2021. 8, 14, 15, 19

Obin Kwon, Jeongho Park, and Songhwai Oh. Renderable neural radiance map for visual navigation. In *CVPR*, pp. 9099–9108, 2023. 19

Xiaohan Lei, Min Wang, Wengang Zhou, and Houqiang Li. GaussNav: Gaussian Splatting for Visual Navigation, 2024. URL https://arxiv.org/abs/2403.11625. 16

Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. URL https://openreview.net/forum?id=RriDjddCLN. 5, 21

Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KmykpuSrjcq. 15

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 5, 20

Ming-Yi Lin, Ou-Wen Lee, and Chih-Ying Lu. Embodied AI with Large Language Models: A Survey and New HRI Framework. In *2024 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 978–983. IEEE, 2024. 3

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. *arXiv preprint arXiv:2304.08485*, 2023a. 22

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023b. 7, 11

Iker Lluvia, Elena Lazkano, and Ander Ansuategi. Active mapping and robot exploration: A survey. *Sensors*, 21(7):2445, 2021. 3

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. InstructNav: Zero-shot System for Generic Instruction Navigation in Unexplored Environment. In *CoRL*, 2024. 8, 20

Malte Lorbach, Sebastian Höfer, and Oliver Brock. Prior-assisted propagation of spatial information for object search. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2904–2909. IEEE, 2014. 14

David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 6

Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A strong baseline for indoor object navigation. *arXiv preprint arXiv:2203.07359*, 2022. 5

Matteo Luperto, Marco Maria Ferrara, Giacomo Boracchi, and Francesco Amigoni. Estimating Map Completeness in Robot Exploration. *arXiv preprint arXiv:2406.13482*, 2024. 23

Aravindh Mahendran, Hakan Bilen, João F Henriques, and Andrea Vedaldi. Researchdoom and cocodoom: Learning computer vision with games. *arXiv preprint arXiv:1610.02431*, 2016. 14

Alexei A Makarenko, Stefan B Williams, Frederic Bourgault, and Hugh F Durrant-Whyte. An experiment in integrated exploration. In *IEEE/RSJ international conference on intelligent robots and systems*, volume 1, pp. 534–539. IEEE, 2002. 5

Neil Houlsby Matthias Minderer, Alexey Gritsenko. Scaling open-vocabulary object detection. *NeurIPS*, 2023. 1

Pietro Mazzaglia, Ozan Catal, Tim Verbelen, and Bart Dhoedt. Curiosity-driven exploration via latent bayesian surprise. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7752–7760, 2022. 5

Yash Mehan, Kumaraditya Gupta, Rohit Jayanti, Anirudh Govil, Sourav Garg, and Madhava Krishna. QueSTMaps: Queryable Semantic Topological Maps for 3D Scene Understanding. *arXiv preprint arXiv:2404.06442*, 2024. 14

Jean-Arcady Meyer and David Filliat. Map-based navigation in mobile robots:: Ii. a review of map-learning and path-planning strategies. *Cognitive Systems Research*, 4(4):283–317, 2003. 2

Medhini Narasimhan, Erik Wijmans, Xinlei Chen, Trevor Darrell, Dhruv Batra, Devi Parikh, and Amanpreet Singh. Seeing the un-scene: Learning amodal semantic maps for room navigation. In *ECCV*, pp. 513–529. Springer, 2020. 6, 13

OpenAI. Gpt-4 technical report, 2023. 2, 20, 22

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 5, 15, 22

Ruslan Partsey, Erik Wijmans, Naoki Yokoyama, Oles Dobosevych, Dhruv Batra, and Oleksandr Maksymets. Is mapping necessary for realistic pointgoal navigation? In *CVPR*, pp. 17232–17241, 2022. 2

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, pp. 2778–2787, 2017. 5

Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D scene understanding with open vocabularies. In *CVPR*, 2023. 2, 6, 8, 11, 16, 23

Tianqiang Peng and Fang Li. Bag of visual word model based on binary hashing and space pyramid. In *ICDIP*, volume 10033, pp. 1155–1159. SPIE, 2016. 6

Rolf Pfeifer and Fumiya Iida. Embodied artificial intelligence: Trends and challenges. *Lecture notes in computer science*, pp. 1–26, 2004. 2

Julio A Placed and José A Castellanos. Enough is enough: Towards autonomous uncertainty-driven stopping criteria. *IFAC-PapersOnLine*, 55(14):126–132, 2022. 23

Julio A Placed, Jared Strader, Henry Carrillo, Nikolay Atanasov, Vadim Indelman, Luca Carlone, and José A Castellanos. A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics*, 39(3):1686–1705, 2023. 6

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots. *arXiv preprint arXiv:2310.13724*, 2023. 1

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3D language Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20051–20060, 2024. 16

Ri-Zhao Qiu, Yafei Hu, Ge Yang, Yuchen Song, Yang Fu, Jianglong Ye, Jiteng Mu, Ruihan Yang, Nikolay Atanasov, Sebastian Scherer, et al. Learning generalizable feature fields for mobile manipulation. *arXiv preprint arXiv:2403.07563*, 2024. 16

Peteris Racinskis, Janis Arents, and Modris Greitans. Constructing maps for autonomous robotics: An introductory conceptual overview. *Electronics*, 12(13):2925, 2023. 3

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763. PMLR, 2021. 2, 5, 7, 15, 16, 20

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3D dataset (HM3d): 1000 large-scale 3D environments for embodied AI. In *NeurIPS Datasets and Benchmarks Track (Round 2)*, 2021. 13, 14

Sonia Raychaudhuri, Tommaso Campari, Unnat Jain, Manolis Savva, and Angel X. Chang. MOPA: Modular Object Navigation with PointGoal Agents. *arXiv preprint arXiv:2304.03696*, 2023. 1, 4, 5, 8, 13, 14, 19, 23

Sonia Raychaudhuri, Duy Ta, Katrina Ashton, Angel X. Chang, Jiuguang Wang, and Bernadette Bucher. NL-SLAM for OC-VLN: Natural Language Grounded SLAM for Object-centric VLN, 2024. URL https://arxiv.org/abs/2411.07848. 23

Allen Z Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. Explore until Confident: Efficient Exploration for Embodied Question Answering. *arXiv preprint arXiv:2403.15941*, 2024. 5

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 4, 5, 7, 11

A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone. 3D Dynamic Scene Graphs: Actionable Spatial Perception with Places, Objects, and Humans. In *RSS*, 2020a. 8, 17

Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1689–1696. IEEE, 2020b. 14

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, pp. 2564–2571. IEEE, 2011. 6

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pp. 4938–4947, 2020. 19

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *ICLR*, 2018. 8, 14, 15

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied AI research. In *ICCV*, pp. 9339–9347, 2019. 14

James A Sethian. Fast-marching level-set methods for three-dimensional photolithography development. In *Optical Microlithography IX*, volume 2726, pp. 262–272. International Society for Optics and Photonics, 1996. 5

Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. CLIP-Fields: Weakly supervised semantic fields for robotic memory. In *ICRA*, 2023. 8, 16, 20

Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action. In *CoRL*, pp. 492–504. PMLR, 2023. URL https://openreview.net/forum?id=UW5A3SweAH. 14, 16

Ishneet Sukhvinder Singh, ID Wijegunawardana, SM Bhagya P Samarakoon, MA Viraj J Muthugala, and Mohan Rajesh Elara. Vision-based dirt distribution mapping using deep learning. *Scientific Reports*, 13(1): 12741, 2023. 1

Xu Song, Xuan Liang, and Zhou Huaidong. Semantic mapping techniques for indoor mobile robots: Review and prospect. *Measurement and Control*, pp. 00202940241259903, 2024. 2

Ricardo B Sousa, Héber M Sobreira, and António Paulo Moreira. A systematic literature review on long-term localization and mapping for mobile robots. *Journal of Field Robotics*, 40(5):1245–1322, 2023. 3

Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *NeurIPS*, 2021. 3

Francesco Taioli, Federico Cunico, Federico Girella, Riccardo Bologna, Alessandro Farinelli, and Marco Cristani. Language-enhanced RNR-map: Querying renderable neural radiance field maps with natural language. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4669–4674, 2023. 16

Yujie Tang, Meiling Wang, Yinan Deng, Zibo Zheng, Jingchuan Deng, and Yufeng Yue. OpenIN: Open-Vocabulary Instance-Oriented Navigation in Dynamic Domestic Environments. *arXiv preprint arXiv:2501.04279*, 2025. 14, 17

Sebastian Thrun. Robotic mapping: a survey. In *Robotic mapping: a survey*, 2003. URL https://api.semanticscholar.org/CorpusID:14633188. 3

Sebastian Thrun and Michael Montemerlo. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429, 2006. 14

Sebastian Thrun, Jens-Steffen Gutmann, Dieter Fox, Wolfram Burgard, Benjamin Kuipers, et al. Integrating topological and metric maps for mobile robot navigation: A statistical approach. *AAAI/IAAI*, 9:989–995, 1998. 8, 17

Edward C Tolman. Cognitive maps in rats and men. *Psychological review*, 55(4):189, 1948. 2

Nicola Tomatis, Illah Nourbakhsh, and Roland Siegwart. Combining topological and metric: A natural integration for simultaneous localization and map building. In *Proceedings of the Fourth European Workshop on Advanced Mobile Robots (Eurobot)*. ETH-Zürich, 2001. 8, 17

Brandon Trabucco, Gunnar A Sigurdsson, Robinson Piramuthu, Gaurav S Sukhatme, and Ruslan Salakhutdinov. A simple approach for visual room rearrangement: 3D mapping and semantic search. In *The Eleventh International Conference on Learning Representations*, 2022. 6

Olivier Trullier and Jean-Arcady Meyer. Animat navigation using a cognitive graph. *Biological Cybernetics*, 83(3):271–285, 2000. 2

Manuela Veloso, Joydeep Biswas, Brian Coltin, and Stephanie Rosenthal. Cobots: Robust symbiotic autonomous mobile service robots. In *Twenty-fourth international joint conference on artificial intelligence*. Citeseer, 2015. 1

Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi SM Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. NeSF: Neural semantic fields for generalizable semantic segmentation of 3D scenes. *arXiv preprint arXiv:2111.13260*, 2021. 16

Saim Wani, Shivansh Patel, Unnat Jain, Angel Chang, and Manolis Savva. MultiON: Benchmarking semantic map memory using multi-object navigation. *NeurIPS*, 33:9700–9712, 2020. 3, 4, 8, 13, 14, 19

Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *CVPR*, 2021. 3

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *ICLR*, 2019. 3, 4, 5

Yuchen Wu, Pengcheng Zhang, Meiying Gu, Jin Zheng, and Xiao Bai. Embodied navigation with multi-modal information: A survey from tasks to methodology. *Information Fusion*, pp. 102532, 2024. 3

Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: real-world perception for embodied agents. In *CVPR*, 2018. 14

Bin Xie, Jiale Cao, Jin Xie, Fahad Shahbaz Khan, and Yanwei Pang. SED: A simple encoder-decoder for open-vocabulary semantic segmentation. In *CVPR*, pp. 3426–3436, 2024. 21

Xiuwei Xu, Huangxing Chen, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. EmbodiedSAM: Online Segment Any 3D Thing in Real Time. *arXiv preprint arXiv:2408.11811*, 2024. 16

Karmesh Yadav, Santhosh Kumar Ramakrishnan, John Turner, Aaron Gokaslan, Oleksandr Maksymets, Rishabh Jain, Ram Ramrakhya, Angel X Chang, Alexander Clegg, Manolis Savva, Eric Undersander, Devendra Singh Chaplot, and Dhruv Batra. Habitat challenge 2022. https://aihabitat.org/challenge/2022/, 2022. 1, 3

Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-Matterport 3D Semantics dataset. In *CVPR*, pp. 4927–4936, 2023. 14

Brian Yamauchi. A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA). 'Towards New Computational Principles for Robotics and Automation'*, pp. 146–151, 1997. 5

Kashu Yamazaki, Taisei Hanyu, Khoa Vo, Thang Pham, Minh Tran, Gianfranco Doretto, Anh Nguyen, and Ngan Le. Open-Fusion: Real-time open-vocabulary 3D mapping and queryable scene representation. *arXiv preprint arXiv:2310.03923*, 2023. 11

Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3D-Mem: 3D Scene Memory for Embodied Exploration and Reasoning, 2024. URL https://arxiv.org/abs/2411.17735. 14

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. VLFM: Vision-language frontier maps for zero-shot semantic navigation. Workshop on Language and Robot Learning, CoRL 2023, 2023. Atlanta, GA, USA. 5, 8, 20, 23

Jiazhao Zhang, Chenyang Zhu, Lintao Zheng, and Kai Xu. Fusion-aware point convolution for online semantic 3d scene segmentation. In *CVPR*, pp. 4534–4543, 2020. 16

Jiazhao Zhang, Liu Dai, Fanpeng Meng, Qingnan Fan, Xuelin Chen, Kai Xu, and He Wang. 3D-aware object goal navigation via simultaneous exploration and identification. In *CVPR*, pp. 6672–6682, 2023a. 8, 16

Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. MapNav: A Novel Memory Representation via Annotated Semantic Maps for VLM-based Vision-and-Language Navigation. *arXiv preprint arXiv:2502.13451*, 2025a. 8, 19

Lingfeng Zhang, Yuecheng Liu, Zhanguang Zhang, Matin Aghaei, Yaochen Hu, Hongjian Gu, Mohammad Ali Alomrani, David Gamaliel Arcos Bravo, Raika Karimi, Atia Hamidizadeh, et al. Mem2Ego: Empowering Vision-Language Models with Global-to-Ego Memory for Long-Horizon Embodied Navigation. *arXiv preprint arXiv:2502.14254*, 2025b. 21

Tianyao Zhang, Xiaoguang Hu, Jin Xiao, and Guofeng Zhang. A survey of visual navigation: From geometry to embodied AI. *Engineering Applications of Artificial Intelligence*, 114:105036, 2022. 2

Xuetao Zhang, Yubin Chu, Yisha Liu, Xuebo Zhang, and Yan Zhuang. A novel informative autonomous exploration strategy with uniform sampling for quadrotors. *IEEE transactions on industrial electronics*, 69 (12):13131–13140, 2021. 5

Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514*, 2023b. 7

Ying Zheng, Lei Yao, Yuejiao Su, Yi Zhang, Yi Wang, Sicheng Zhao, Yiyi Zhang, and Lap-Pui Chau. A survey of embodied learning for object-centric robotic manipulation. *arXiv preprint arXiv:2408.11537*, 2024. 3

Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15838–15847, 2021. 16

Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. ESC: Exploration with soft commonsense constraints for zero-shot object navigation. In *ICML*, pp. 42829–42842. PMLR, 2023. 5

Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 11

Fengda Zhu, Yi Zhu, Vincent Lee, Xiaodan Liang, and Xiaojun Chang. Deep learning for embodied vision navigation: A survey. *arXiv preprint arXiv:2108.04097*, 2021. 2