

Multimodal Spatial Language Maps for Robot Navigation and Manipulation

International Journal of Robotics Research
 XX(X):1–24
 ©The Author(s) 2024
 Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
 DOI: 10.1177/ToBeAssigned
www.sagepub.com/



Chenguang Huang¹, Oier Mees², Andy Zeng³ and Wolfram Burgard¹

Abstract

Grounding language to a navigating agent's observations can leverage pretrained multimodal foundation models to match perceptions to object or event descriptions. However, previous approaches remain disconnected from environment mapping, lack the spatial precision of geometric maps, or neglect additional modality information beyond vision. To address this, we propose multimodal spatial language maps as a spatial map representation that fuses pretrained multimodal features with a 3D reconstruction of the environment. We build these maps autonomously using standard exploration. We present two instances of our maps, which are visual-language maps (VLMaps) and their extension to audio-visual-language maps (AVLMaps) obtained by adding audio information. When combined with large language models (LLMs), VLMaps can translate natural language commands into open-vocabulary spatial goals (e.g., “in between the sofa and TV”) directly localized in the map, and be shared across different robot embodiments to generate tailored obstacle maps on demand. Building upon the capabilities above, AVLMaps extend VLMaps by introducing a unified 3D spatial representation integrating audio, visual, and language cues through the fusion of features from pretrained multimodal foundation models. This enables robots to ground multimodal goal queries (e.g., text, images, or audio snippets) to spatial locations for navigation. Additionally, the incorporation of diverse sensory inputs significantly enhances goal disambiguation in ambiguous environments. Experiments in simulation and real-world settings demonstrate that our multimodal spatial language maps enable zero-shot spatial and multimodal goal navigation and improve recall by 50% in ambiguous scenarios. These capabilities extend to mobile robots and tabletop manipulators, supporting navigation and interaction guided by visual, audio, and spatial cues. Code and videos are available at <https://mslmaps.github.io>.

Keywords

Robot Navigation, Scene Representations, Large Language Models, Audio Language Models

1 Introduction

People are excellent navigators of the physical world due in part to their remarkable ability to build cognitive maps (McNamara et al. 1989) that form the basis of spatial memory (Chun and Jiang 1998; Newman et al. 2007) to (i) localize landmarks at varying ontological levels, such as a book; on the shelf; in the living room, or to (ii) determine whether the layout permits navigation between two points. Meanwhile, humans exhibit a remarkable ability to integrate and leverage multiple sensing modalities to efficiently move around in the physical world. Our actions are driven by a myriad of sensory cues: the sound of glass breaking might signal a dangerous situation, the microwave might buzz to indicate it is done, or a dog might bark to draw our attention. Acoustic signals particularly represent a valuable complementary form of information, also evident by the utility that they provide for the visually impaired, who may rely on them for navigation. Research in cognitive science also suggests that children understand and integrate information from different sensing modalities into spatial cognitive maps (Kording et al. 2007).

Classic methods for robot navigation (Thrun et al. 1998; Endres et al. 2012) build geometric maps for path planning. Although some previous extensions parse goals from templated natural language commands (Tellec et al.

2011; MacMahon et al. 2006), they struggle to generalize to unseen instructions. Learning methods directly optimize for navigation policies grounded in language end-to-end (commands to actions) (Anderson et al. 2018b, 2021) but require copious amounts of data. Recent works demonstrate that multimodal foundation models (Radford et al. 2021; Li et al. 2021) pretrained on Internet-scale data (e.g., images and their captions) can be used out-of-the-box to ground language to the visual observations of a navigating agent, without additional data collection or model fine-tuning. These models enable mobile robots to handle new instructions that specify unseen object goals and can be combined with exploration algorithms to search for the first instance of any object (CoW) (Gadre et al. 2023) or traverse object-centric landmarks in graphs (LM-Nav) (Shah et al. 2023). While promising, these methods predominantly

¹University of Technology Nuremberg, Germany

²UC Berkeley, USA

³Google Research, USA

Corresponding author:

Chenguang Huang, University of Technology Nuremberg, Department of Computer Science and Artificial Intelligence, Artificial Intelligence and Robotics Lab, Ulmenstraße 52i, 90461 Nuremberg, Germany
 Email: chenguang.huang@utn.de

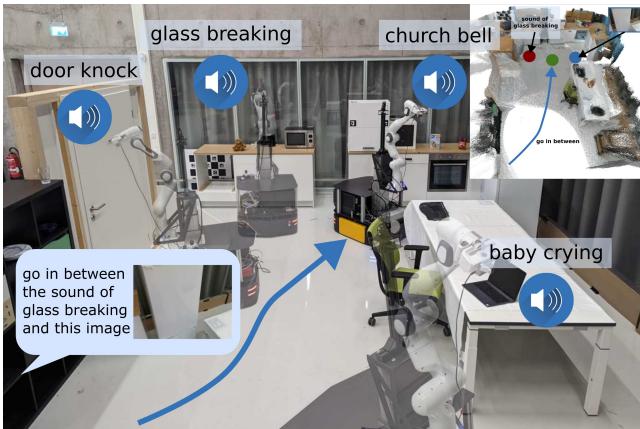


Figure 1. AVLMaps provide an open-vocabulary 3D map representation for storing cross-modal information from audio, visual, and language cues. When combined with large language models, AVLMaps consumes multimodal prompts from audio, vision, and language to solve zero-shot spatial goal navigation by effectively leveraging complementary information sources to disambiguate goals.

use vision language models (VLMs) as critics to match image observations to object goal descriptions, but remain disjoint from the mapping of the environment, lacking the fine-grained spatial precision of classic geometric maps. Furthermore, they neglect the great potential of information in other sensing modalities such as audio. Therefore, these methods struggle to (i) localize spatial goals e.g., “in between the sofa and the TV”, to (ii) build persistent representations that can be shared across different embodiments, e.g., mobile robots, drones, or to (iii) localize multimodal goals e.g., “the sound of the baby crying”, or an image of a refrigerator. How to best spatially anchor various sensing modalities, including visual and audio signals, in ways that enable effective data-efficient cross-modal reasoning for downstream robotics tasks, remains a relatively open question.

In this work, we address that question by introducing Multimodal Spatial Language Maps, a general mapping framework that is (i) spatial, (ii) multimodal, (iii) reusable across different robot embodiments, and (iv) readily extensible to additional sensing modalities in the future. At the heart of our approach are two concrete map instances: Visual-Language Maps (VLMaps) and their multimodal extension, Audio-Visual-Language Maps (AVLMaps). We begin by evaluating VLMaps, which fuse pretrained visual-language features from image observations with a 3D reconstruction of the environment. This fusion makes VLMaps both spatial-preserving, enabling localizing queries like “in between the sofa and the TV” in the map, and reusable across embodiments, since the same voxelized map can generate tailored obstacle grids for different robots by defining different sets of obstacle categories. VLMaps can be built from a robot’s video stream using standard exploration strategies. When coupled with a large language model (LLMs) in a Socratic fashion (Zeng et al. 2023), they translate long-horizon natural-language commands into sequences of open-vocabulary, spatially grounded goals.

Subsequently, we extend VLMaps to Audio-Visual-Language Maps, AVLMaps, a unified 3D spatial map representation for storing cross-sensing information from

audio, visual, and language modalities. By introducing a new modality, audio, we extrapolate the capabilities of our framework to the multimodal setting, showing its extensibility to additional sensing modalities. AVLMaps can be built from image and audio observations captured during reconstruction, by computing dense pre-trained features from open-vocabulary multimodal foundation models trained on Internet-scale data (Li et al. 2022; Ghiasi et al. 2022; Guzhov et al. 2022) and fusing them into a shared 3D voxel grid representation. Beyond VLMaps, AVLMaps:

- allow for landmarks (or areas and regions of interest) indexing in the environment via open-vocabulary *multimodal queries* (e.g., abstract textual descriptions, images, or audio snippets), enabling downstream applications including multimodal goal-driven navigation, without domain-specific model finetuning.
- include audio information, which allows robots to more often correctly disambiguate goal locations using sound (e.g., “go to the table where you heard coughing” in environments where there are multiple tables, etc).
- extend the spatial characteristic of VLMaps to the multimodal domain, enabling zero-shot *multimodal spatial* goal localization, e.g., “Go in between the {image of a refrigerator} and the sound of breaking glass” as in Fig. 1.

Extensive experiments in both simulated and real-world settings demonstrate that our VLMaps enable more effective long-horizon language-conditioned spatial goal navigation than baseline alternatives, such as CoW (Gadre et al. 2023) and LM-Nav (Shah et al. 2023). They can be shared across different robot embodiments to generate tailored obstacle maps for efficient, embodiment-specific path planning. Building on this, AVLMaps further extend these capabilities to navigating to goal locations specified by, e.g., natural language descriptions of sounds or visual landmarks – and notably, can disambiguate multiple possible goal locations using multimodal information, (using object semantics to pinpoint one of the multiple possible sound goals, or using vision to pinpoint one of the multiple possible locations where similar objects were found) quantitatively better than unimodal baseline alternatives by up to 50% in top-1 recall in ambiguous scenarios. This article expands our previous work (Huang et al. 2023b,a) by expanding our evaluation to demonstrate that AVLMaps’ capabilities continue to naturally improve with better-performing pre-trained audio-language foundation models such as AudioCLIP (Guzhov et al. 2022) and CLAP (Elizalde et al. 2023) and that the achieved multimodal disambiguation capabilities also translate to challenging robot manipulation tasks. AVLMaps are simple and effective in leveraging multiple multimodal foundation models together in tandem to reach broader language-driven robot navigation capabilities, but are also not without limitations – we discuss these and avenues for future work. The code is available at <https://mslmaps.github.io/>.

2 Related Work

Semantic Mapping. In recent years, the synergy of the traditional SLAM techniques and the advancements in

vision-based semantic understanding has led to augmenting 3D maps with semantic information (Salas-Moreno et al. 2013; McCormac et al. 2017). Stemming from the intuition of augmenting 3D points in the map with 2D segmentation results, previous works focus on either abstracting the map at object-level with a pose graph (McCormac et al. 2018) or an octree (Xu et al. 2019) or modeling the dynamics of objects in the map (Runz et al. 2018). Despite lifting the 3D reconstruction to a semantic level, these methods are restricted to a predefined set of semantic classes. Recent works like LM-Nav (Shah et al. 2023), CoW (Gadre et al. 2023), VLMaps (Huang et al. 2023b), NLMap-SayCan (Chen et al. 2023a), OpenScene (Peng et al. 2023), or CLIP-Fields (Shafiullah et al. 2023) have shown that integrating visual-language features, generated by either pre-trained or fine-tuned models, into a topological graph or an occupancy map enables open-vocabulary object indexing with natural language, freeing the maps from fixed-size semantic categories. Recent approaches also investigate other open-vocabulary map representations such as NeRF (Engelmann et al. 2024; Kerr et al. 2023; Kim et al. 2024), Gaussian Splatting (Qin et al. 2024; Zuo et al. 2024), and Scene Graphs (Gu et al. 2024; Werby et al. 2024). However, these works focus on visual perception to map and move through an environment, overlooking complementary sources of information such as acoustic signals. In contrast, AVLMaps integrate audio, visual, and language cues into a 3D map, equipping the agent with the ability to navigate to multiple types of multimodal goals and effectively disambiguate goals.

Vision and Language Navigation. Recently, also Vision-and-Language Navigation (VLN) has received increased attention (Anderson et al. 2018b; Krantz et al. 2020). Further work has focused on learning end-to-end policies that can follow route-based instructions on topological graphs of simulated environments (Anderson et al. 2018b; Fried et al. 2018; Guhur et al. 2021). However, agents trained in this setting do not have low-level planning capabilities and rely heavily on the topological graph, limiting their real-world applicability (Anderson et al. 2021). Moreover, despite extensions to continuous state spaces (Krantz et al. 2020, 2021; Hong et al. 2022), most of these learning-based methods are data-intensive. The recent success of large pretrained vision and language models (Radford et al. 2021; Brown et al. 2020) has spurred a flurry of interest in applying their zero-shot capabilities to open-vocabulary object navigation (Shah et al. 2023; Gadre et al. 2023). LM-Nav (Shah et al. 2023) combines three pre-trained models to navigate via a topological graph in the real world. CoW (Gadre et al. 2023) performs zero-shot language-based object navigation by combining CLIP-based (Radford et al. 2021) saliency maps and traditional exploration methods. However, both methods are limited to navigating to object landmarks and are less capable of understanding finer-grained queries, such as “to the left of the chair” and “in between the TV and the sofa”. In contrast, our method, VLMaps, enables spatial language indexing beyond object-centric goals and can generate open-vocabulary obstacle maps. Our extension, AVLMaps, further enables multimodal spatial concept indexing such as “between the image of a refrigerator and the sound of breaking glass”.

Multimodal Navigation. Recent advances in simulation applications (Savva et al. 2019; Kolve et al. 2017; Chen et al. 2020; Gan et al. 2020a) have boosted research on multimodal navigation in two distinct directions: (i) vision-and-language navigation (VLN) (Anderson et al. 2018b; Krantz et al. 2020) where an agent needs to follow a natural language instruction towards the goal with visual input, and (ii) audio-visual navigation (AVN) (Chen et al. 2020) in which an agent should navigate to the sound source based on information from a binaural sensor and vision. Despite different degrees of success in both directions (Fried et al. 2018; Guhur et al. 2021; Chen et al. 2021a; Younes et al. 2023; Gan et al. 2020b, 2022), less attention has been paid to solving the navigation problem involving vision, language, and audio at the same time. The most relevant concept to our knowledge is from AVLEN (Paul et al. 2022), which extends the AVN with a further query step, introducing a language instruction that helps with navigating to the sound source. In addition, most of the existing methods on AVN focus on approaching the sound without understanding its semantics. In our work, we propose a method to integrate both visual and sound semantics into the same map, enabling a robot to navigate to multimodal goals specified with either goal image or natural language like “go to the sound of baby crying”, “go to the table” or multimodal prompts such as “go to the {image of a table} where the sound of the microwave was heard”.

Pre-trained Zero-shot Models in Robotics. Recent trends have shown that pre-trained foundation models (Radford et al. 2021; Brown et al. 2020; Liang et al. 2023a) serve as powerful tools for robotic tasks. Most works exploit the inherent perception and reasoning abilities of Vision Language Models (VLMs) or Large Language Models (LLMs) trained with cloud-sourced data to boost the performance of robot tasks including object detection and segmentation (Kamath et al. 2021; Gu et al. 2021; Li et al. 2021), robot manipulation (Shridhar et al. 2022; Liang et al. 2023b; Mees et al. 2022b,a, 2023; Rosete-Beas et al. 2022; Zawalski et al. 2024; Chen et al. 2024a; Zhou et al. 2024), and navigation (Shah et al. 2023; Gadre et al. 2023; Chen et al. 2023b; Huang et al. 2023b; Hirose et al. 2024; Gu et al. 2024; Werby et al. 2024). With access to a growing volume of robot control data annotated with semantic language labels, recent approaches have advanced the training of a Vision-Language-Action (VLA) model, bridging the gap between visual-language comprehension and low-level robot control within a unified foundational model (Brohan et al. 2023; Zitkovich et al. 2023; O’Neill et al. 2024; Kim et al. 2025; Octo Model Team et al. 2024; Doshi et al. 2024). Despite promising results from previous methods, little effort has been made to exploit the audio-language pre-trained models (ALMs) (Guzhov et al. 2022; Elizalde et al. 2023) in robotic tasks. In this work, we leverage the foundation models focusing on different modalities, e.g. audio, language, and vision, to create a mapping pipeline to understand multimodal information in the scene and achieve robot navigation given language, image, or audio description queries. Concurrent work ConceptFusion (Jatavallabhula et al. 2023) demonstrates that audio can be used as queries to index locations in a visual-language map. However, it doesn’t support integrating audio information from the

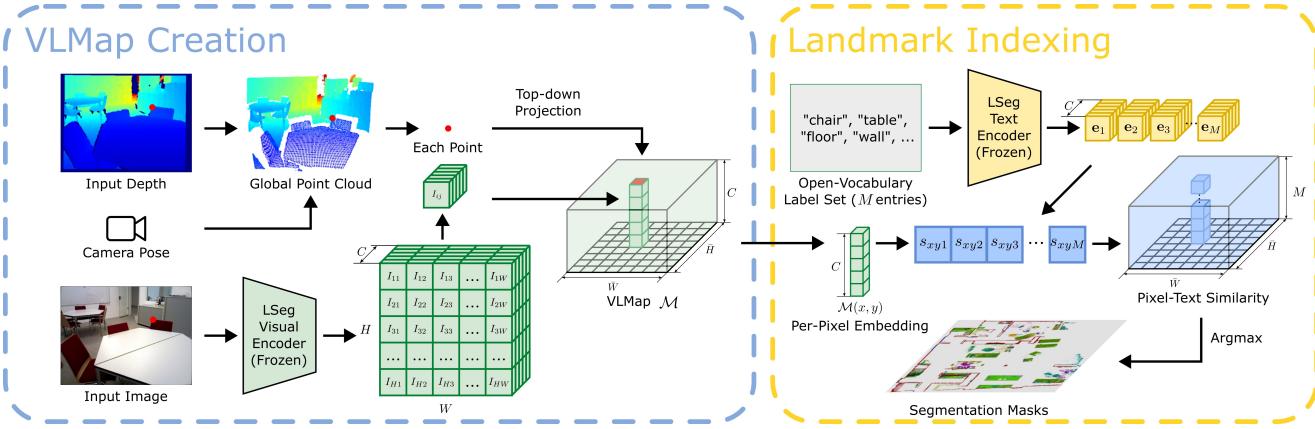


Figure 2. The creation and language-conditioned indexing of a VLMap. A VLMap is created by fusing pretrained visual-language features into the reconstruction of the environment to enable visual-spatial-language-based reasoning. By providing a list of open-vocabulary labels, we retrieve segmentation masks for semantic classes required by downstream applications.

observation data into the representation of the scene as we do in this work.

3 Method

Our goals are two-fold: (i) build a visual language map representation that synergizes the open-vocabulary capability of vision-language models and the spatial characteristic of geometric maps, and (ii) extend such a map to a multimodal spatial language map representation, in which object landmarks (“sofa”), areas (“kitchen”), audio semantics (“the sound of a baby crying”), or visual goals can be directly localized using natural language or a target image. To achieve the first goal, we propose VLMaps, which can be constructed with pre-trained visual-language models by consuming RGB-D streaming data and camera odometry. Such a map allows for spatial goal indexing like “to the left of the sofa”, and can dynamically adapt to different embodiments, enabling users to freely define obstacle categories for the robot to generate a customized occupancy map for path planning. For the second goal, we propose a more general representation, AVLMaps, which takes VLMaps as one sub-module and extends it to consume multimodal data during mapping and support multimodal concept indexing, such as audio, images, and language. We also propose a cross-modal reasoning method to disambiguate locations referring to targets from different modalities (“the sound of brushing teeth near the sink”, or “the table near this image: {image}”). In the following subsections, we first start with (i) how to build a VLMap by integrating visual-language features into spatial map location (Sec. 3.1), (ii) how to use this map to localize open-vocabulary landmarks (Sec. 3.2), (iii) how we can build open-vocabulary obstacle maps from a list of obstacle categories for different robot embodiments (Sec. 3.3), and (iv) how we can use this map for spatial goal navigation with the help of an LLM (Sec. 3.4). Later, we consider VLMaps in a broader context, and demonstrate (v) how to extend it to a multimodal map that integrates audio, language, and visual information in the map and use it for localizing different targets (Sec. 3.5), (vi) how to disambiguate goal locations with multimodal information (Sec. 3.6), and (vii) how such a multimodal map can be used with large language models

(LLMs) for multimodal goal navigation, without additional data collection or model fine-tuning (Sec. 3.7). We show the pipeline of building a VLMap in Fig. 2, and later show the system pipeline of our multimodal map representation in Fig. 7.

3.1 Building a Visual-Language Map

The key idea behind VLMaps is to fuse pretrained visual-language features with a 3D reconstruction. We achieve this by computing dense pixel-level embeddings from an existing visual-language model (over the video feed of the robot) and by back-projecting them onto the 3D surface of the environment (captured from depth data used for reconstruction with visual odometry). The overview of VLMaps creation is shown on the left of Fig. 2.

In our work, we utilize LSeg (Li et al. 2021) as the visual-language model, a language-driven semantic segmentation model that segments the RGB images based on a set of free-form language categories. The LSeg visual encoder maps an image such that the embedding of each pixel lies in the CLIP feature space. In our approach, we fuse the LSeg pixel embeddings with their corresponding 3D map locations. In this way, without explicit manual segmentation labels, we incorporate a powerful language-driven semantic prior that inherits the generalization capabilities of VLMs. The only assumption we make is access to odometry, which is readily available from RGB-D SLAM systems and enables us to build a map from sequences of RGB-D images.

Formally, we define VLMap as $\mathcal{M} \in \mathbb{R}^{\bar{H} \times \bar{W} \times C}$, where \bar{H} and \bar{W} represent the size of the top-down grid map, and C represents the length of the VLM embedding vector for each grid cell. Together with the scale parameter s (meters per pixel), a VLMap \mathcal{M} represents an area with size $s\bar{H}$ meters \times $s\bar{W}$ meters. To build the map, for each RGB-D frame, we back-project all the depth pixels $\mathbf{u} = (u, v)$ to form a local depth point cloud that we transform to the world frame, $\mathbf{P}_k = D(\mathbf{u})K^{-1}\tilde{\mathbf{u}}$ and $\mathbf{P}_W = T_{Wk}\mathbf{P}_k$ where $\tilde{\mathbf{u}} = (u, v, 1)$, $K \in \mathbb{R}^{3 \times 3}$ is the intrinsic matrix of the depth camera, $D(\mathbf{u}) \in \mathbb{R}$ is the depth value of the pixel \mathbf{u} , T_{Wk} is the transformation from the world coordinate frame to the k-th camera frame, $\mathbf{P}_k \in \mathbb{R}^3$ is the 3D point position in the k-th frame, and $\mathbf{P}_W \in \mathbb{R}^3$ is the 3D point position in the

world coordinate frame. We then project the point \mathbf{P}_W to the ground plane and get the pixel \mathbf{u} 's corresponding position on the grid map,

$$p_{map}^x = \left\lfloor \frac{\bar{H}}{2} + \frac{P_W^x}{s} + 0.5 \right\rfloor, p_{map}^y = \left\lfloor \frac{\bar{W}}{2} - \frac{P_W^z}{s} + 0.5 \right\rfloor \quad (1)$$

where p_{map}^x and p_{map}^y represent the coordinates of the projected point in the map \mathcal{M} .

Once we build the grid map, we apply LSeg's visual encoder $f(\mathcal{I}) : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H \times W \times C}$ to the RGB image \mathcal{I}_k and generate the pixel-level embedding $\mathcal{F}_k \in \mathbb{R}^{H \times W \times C}$. Given the RGB-D registration, we project each image pixel \mathbf{u} 's embedding $\mathbf{q} = \mathcal{F}_k(\mathbf{u}) \in \mathbb{R}^C$ to its corresponding grid cell location (p_{map}^x, p_{map}^y) in the top-down grid map. Intuitively, there exist multiple 3D points projecting to the same grid location in the map. Thus, we average their embeddings, $\mathcal{M}(p_{map}^x, p_{map}^y) = \frac{1}{n} \sum_{i=1}^n \mathbf{q}_i$ where $\mathcal{M}(p_{map}^x, p_{map}^y) \in \mathbb{R}^C$ represents the map features at the grid position (p_{map}^x, p_{map}^y) , n represents the total number of points projecting to the grid location (p_{map}^x, p_{map}^y) , and $\mathbf{q}_i \in \mathbb{R}^C$ denotes the corresponding pixel embedding of each point. We note that these n points might not only come from a single frame, but also from points from multiple frames. Therefore, the resulting features contain the averaged embeddings from multiple views of the same object.

3.2 Localizing Open-Vocabulary Landmarks

We now describe how to localize landmarks in VLMaps with free-form natural language. The overview of the indexing process is shown on the right of Fig. 2. Formally, we define the input language list as $\mathcal{L} = [\mathbf{l}_0, \mathbf{l}_1, \dots, \mathbf{l}_M]$ where \mathbf{l}_i represents the i -th category in text form, and M represents the number of categories defined by the user. Some examples of the input language list are [“chair”, “sofa”, “table”, “other”] or [“furniture”, “floor”, “other”]. As Li *et al.* (Li *et al.* 2021), we apply the pre-trained CLIP text encoder (Radford *et al.* 2021) to convert such list of texts into a list of vector embeddings $[\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_M]$, $\mathbf{e} \in \mathbb{R}^C$, which are organized into an embedding matrix $E \in \mathbb{R}^{M \times C}$, where each row of the matrix represents the embedding of a category. The map embeddings \mathcal{M} are also flattened into a matrix $Q \in \mathbb{R}^{\bar{H}\bar{W} \times C}$, where each row represents the embedding of a pixel in the top-down grid map. We then compute the pixel-to-category similarity matrix $S = Q \cdot E^T$, where $S \in \mathbb{R}^{\bar{H}\bar{W} \times M}$. Each element S_{ij} in the matrix stores the similarity value between a pixel and a text category, indicating how likely this pixel belongs to the class. By applying the argmax operator along the row direction to S and reshaping the resulting vector to shape $\bar{H} \times \bar{W}$, we get the final segmentation result $R \in \mathbb{R}^{\bar{H} \times \bar{W}}$. Each element R_{ij} represents the label index of the input language list \mathcal{L} at the grid map location (i, j) . With the final resulting matrix R , we compute the most related language-based category for every pixel in the grid map.

3.3 Generating Open-Vocabulary Obstacle Maps

Building a VLMap enables us to generate obstacle maps that inherit the open-vocabulary nature of the VLMs used (LSeg

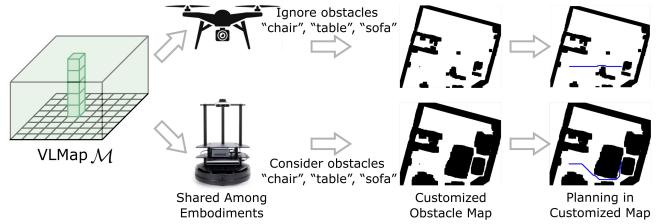


Figure 3. The overview of building customized obstacle maps for different robot embodiments. By specifying different obstacle categories in natural language for different embodiments, different obstacle maps can be built to ensure the most efficient path planning for different embodiments.

and CLIP). Specifically, given a list of obstacle categories described with natural language, we can localize those obstacles at runtime to generate a binary map for collision avoidance and/or shortest path planning, as is shown in Fig. 3. A prominent use case for this is sharing a VLMap of the same environment between different robots with different embodiments (i.e., cross-embodiment problem (Zakka *et al.* 2022; Ganapathi *et al.* 2022)), which may be useful for multi-agent coordination (Wu *et al.* 2021). For example, a large mobile robot may need to navigate around a table (or other large furniture), while a drone can directly fly over it. By simply providing two different lists of obstacle categories – one for the large mobile robot (that contains “table”), and another for the drone (that does not), we can generate two distinct obstacle maps for the two robots to use respectively, sourced on-the-fly from the same VLMap.

To do so, we first extract an obstacle map $\mathcal{O} \in \{0, 1\}^{\bar{H} \times \bar{W}}$ where each projected position of the depth point cloud in the top-down map is assigned 1, and otherwise 0. To avoid points from the floor or the ceiling, points P_W are filtered out depending on their height,

$$\mathcal{O}_{ij} = \begin{cases} 1, & t_1 \leq P_W^y \leq t_2 \text{ and } p_{map}^x = i \text{ and } p_{map}^y = j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where $t_1, t_2 \in \mathbb{R}$ are the lower and upper thresholds for the y -component (we define y axis to be along the height direction) of the point P_W . Second, to obtain obstacle maps tailored to a certain embodiment, we define a list of potential obstacle categories $\mathcal{L}_{obs} = [\mathbf{l}_{obs0}, \mathbf{l}_{obs1}, \dots, \mathbf{l}_{obsM}]$, where \mathbf{l}_{obsi} represents the i -th obstacle category in language, and M represents the total number of obstacle categories defined by the user. We then apply the open-vocabulary landmark indexing introduced in Sec. 3.2 and obtain segmentation masks for all defined obstacles. For a specific embodiment k , we choose a subset of classes out of the whole potential obstacle list \mathcal{L}_{obs} and take the union of their segmentation masks to get the obstacles mask $\tilde{\mathcal{O}}_{emb_k}$. We ignore false predictions of obstacles on floor region in $\tilde{\mathcal{O}}_{emb_k}$ by taking the intersection with \mathcal{O} to get the final obstacle map \mathcal{O}_{emb_k} .

3.4 Zero-Shot Spatial Goal Navigation from Language

In this section, we describe our approach to long-horizon (spatial) goal navigation, given a set of landmark descriptions specified by natural language instructions such as

```
move first to the left side of the counter, then
move between the sink and the oven, then move back
and forth to the sofa and the table twice
```

Notably different from prior work (Gadre et al. 2023; Shah et al. 2023), VLMaps allow us to reference precise spatial goals such as: “in between the sofa at the TV” or “three meters to the east of the chair.” Specifically, we use a large language model (LLM) to interpret the input natural language commands and break them down into subgoals (Ahn et al. 2022; Shah et al. 2023; Zeng et al. 2023). In contrast to prior work, which may reference these subgoals with language and map to low-level policies with semantic translation (Huang et al. 2022a) or affordances (Ahn et al. 2022; Huang et al. 2022b; Zeng 2019), we leverage the code-writing capabilities of LLMs to generate executable Python robot code (Liang et al. 2023b; Mees et al. 2023; Chen et al. 2021b; Brown et al. 2020) that can (i) make precise calls to parameterized navigation primitives, and (ii) perform arithmetic when needed. The generated code can directly be executed on the robot with the built-in Python `exec` function.

Note that recent works (Liang et al. 2023b; Mees et al. 2023; Chen et al. 2021b; Brown et al. 2020) have shown that code-writing language models (e.g., Codex (Chen et al. 2021b)) trained on billions of lines of code from Github can be used to synthesize new simple Python programs from docstrings. In this work, we re-purpose these models for mobile robot planning by priming them with several input examples of natural language commands (formatted as comments) paired with corresponding robot code (via few-shot prompting). The robot code can express functions or logic structures (if-then-else statements or for/while loops) and parameterize API calls (e.g., `move_to(target_name)` or `turn(degrees)`). The full list is available in Table 1 that map to spatial behaviors specified by the language commands. The full prompt is shown in Fig. 4.

Spatial goals are defined as positions around the reference object based on spatial descriptions. For example, “in the middle of the counter and the fridge”, or “to the left of the sofa” etc. Traditional object goal navigation methods either directly retrieve the target object’s location on the map and plan to it (Shah et al. 2023; Gadre et al. 2023) or are trained to approach objects as a reactive system (Chaplot et al. 2020). These methods fall short in reaching spatial goals since these goal locations are free space rather than retrievable locations on semantic maps. However, when we know the reference position, those spatial locations can be computed with simple offsets. The navigation primitive functions (APIs) being called by the language model (e.g., `move_to_left('counter')`) use a pre-generated VLMAP to localize the coordinates of the open-vocabulary landmarks (“counter”) in the maps (described in Sec. 3.2) modified with predefined scripted offsets (to define “left”). We then navigate to these coordinates using an off-the-shelf navigation stack (Quigley 2009) that takes as input the embodiment-specific obstacle map (generated using the same VLMAP, with the process described in Sec. 3.3). Some examples of spatial goal navigation in the real world is shown in Fig. 5.

At test time, the LLM is prompted with context examples (in gray) in Fig. 4 as well as commands (in green) in

```
# move a bit to the right of the fridge
robot.move_to_right('refrigerator')

# move in between the couch and bookshelf
robot.move_in_between('couch', 'bookshelf')

# face the toilet
robot.face('toilet')

# move to the west of the chair
robot.move_west('chair')

# turn right 20 degrees
robot.turn(20)

# find any chairs in the environment
robot.move_to_object('chair')

# with the television on your left
robot.with_object_on_left('television')

# move forward for 3 meters
robot.move_forward(3)

# move right 2 meters
robot.turn(90)
robot.move_forward(2)

# move back and forth to the chair and table
# 3 times
pos1 = robot.get_pos('chair')
pos2 = robot.get_pos('table')
for i in range(3):
    robot.move_to(pos1)
    robot.move_to(pos2)

# move 3 meters south of the chair
robot.move_south('chair')
robot.face('chair')
robot.turn(180)
robot.move_forward(3)

# turn west
robot.turn_absolute(-90)

# turn east
robot.turn_absolute(90)

# turn south
robot.turn_absolute(180)

# turn north
robot.turn_absolute(0)

# turn east and then turn left 90 degrees
robot.turn_absolute(90)
robot.turn(-90)

# navigate to 3 meters right of the table
robot.move_to_right('table')
robot.face('table')
robot.turn(180)
robot.move_forward(3)
```

Figure 4. The full context prompt (prompt in gray) VLMAP used for achieving spatial goal navigation tasks in the experiments.

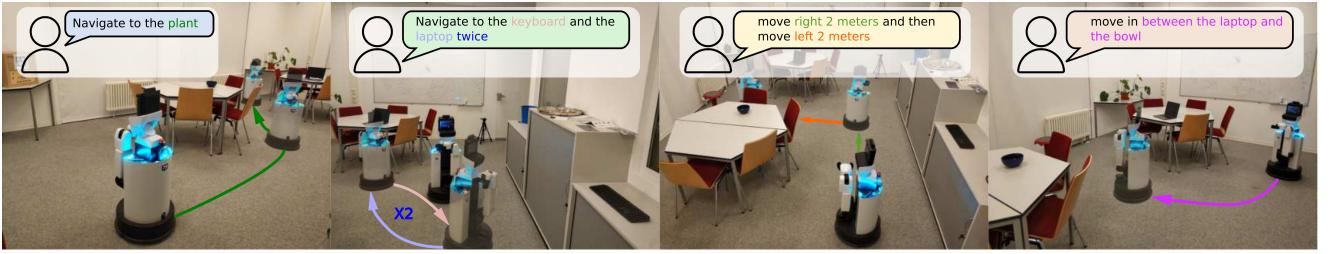


Figure 5. VLMaps enable a robot to perform complex zero-shot spatial goal navigation tasks given natural language commands, without additional data collection or model finetuning.

APIs	Functions
<code>move_to(pos)</code>	move to a position on the map.
<code>move_to_left(object_name)</code>	move to the left side of the nearest front object.
<code>move_to_right(object_name)</code>	move to the right side of the nearest front object.
<code>with_pos_on_left(object_name)</code>	turn until the object is on the robot’s left side.
<code>with_pos_on_right(object_name)</code>	turn until the object is on the robot’s right side.
<code>move_in_between(object_a, object_b)</code>	move in between two objects.
<code>face(object_name)</code>	turn until the robot’s front is pointing to the object.
<code>turn(angle)</code>	turn right a certain angle. If the angle value is negative, turn left.
<code>turn_absolute(angle)</code>	turn to absolute angle. 0 is north, 90 is east, -90 is west, 180 is south.
<code>move_north(object_name)</code>	move to the north side of the object.
<code>move_south(object_name)</code>	move to the south side of the object.
<code>move_east(object_name)</code>	move to the east side of the object.
<code>move_west(object_name)</code>	move to the west side of the object.
<code>move_forward(dist)</code>	move forward “dist” meters.

Table 1: Navigation API library for spatial goal navigation in VLMaps.

```
# move first to the left side of the counter, then
# move between the sink and the oven, then move back
# and forth to the sofa and the table twice
robot.move_to_left('counter')
robot.move_in_between('sink', 'oven')
pos1 = robot.get_pos('sofa')
pos2 = robot.get_pos('table')
for i in range(2):
    robot.move_to(pos1)
    robot.move_to(pos2)

# move 2 meters north of the laptop, then move 3
# meters rightward
robot.move_north('laptop')
robot.face('laptop')
robot.turn(180)
robot.move_forward(2)
robot.turn(90)
robot.move_forward(3)
```

Figure 6. The query and the generated results from the LLM for spatial goal navigation tasks. During the query, the context prompt in Fig. 4 and the input task commands are prompted to the LLM together. The input task commands are in green and generated outputs are highlighted

Fig. 6. It can autonomously re-compose API calls to generate new robot code that not only references the new landmarks mentioned in the language commands (as comments), but also can chain together new sequences of API calls to follow unseen instructions accordingly. The inference process is shown in Fig. 6, and generated outputs are highlighted.

3.5 Building an Audio Visual Language Map

VLMaps provide an intuitive interface for humans to issue natural language commands to robots, enabling open-vocabulary spatial goal navigation. However, interacting with the world is inherently a multimodal experience. Since VLMaps primarily rely on visual information for map construction, it is important to consider other complementary sources of information that can enhance navigation. Motivated by this, we broaden the scope of VLMaps and propose a more general framework, AVLMaps, which enables robots to integrate and interpret multimodal information, including audio, images, and language, within a unified representation (as illustrated in Fig. 7). In this paper, we present the extension to audio and image modalities as a case study, showcasing the modularity and extensibility of our multimodal spatial language maps. The AVLMaps framework is flexible and designed to accommodate future integration of additional sensing modalities, such as temperature, tactile feedback, or magnetic fields. The key idea behind AVLMaps is to combine visual localization features, pre-trained visual-language features, and audio-language features with a 3D reconstruction. Given an RGB-D video stream with an audio track and odometry information, we utilize four modules to build a multimodal features database. Given a specific query, each module returns predicted spatial locations on the map in the form of 3D voxel heatmaps. A heatmap can be denoted as $\mathcal{H} \in [0, 1]^{\bar{H} \times \bar{W} \times \bar{Z}}$, where \bar{H} , \bar{W} and \bar{Z} represent the size of the voxel map and the value in each element represents the probability of being

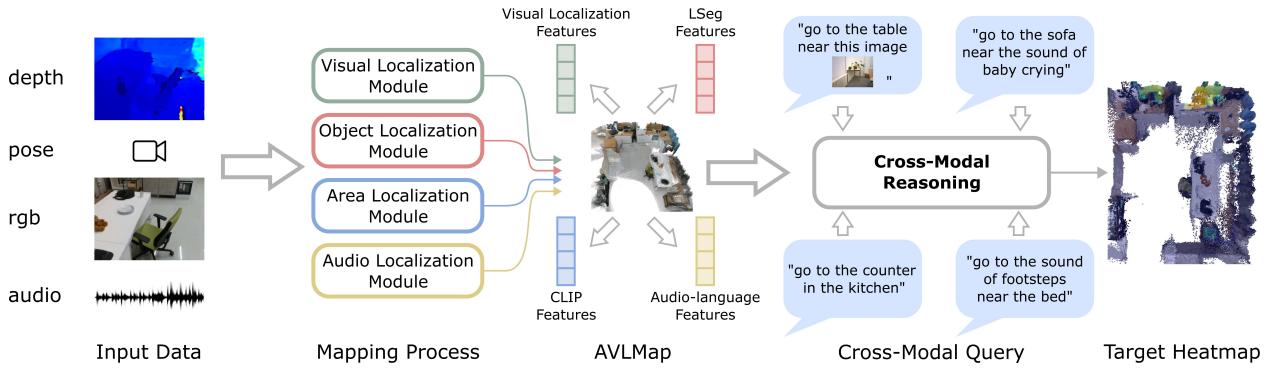


Figure 7. System overview. AVLMaps are constructed from RGB-D, audio, and odometry inputs, converting raw data into visual localization features, visual-language features, and audio-language features. During inference time, each module’s output is unified with cross-modal reasoning, allowing users to query spatial location with multimodal information.

the target position. $\mathbf{p} = (x, y, z)^T, \{x, y, z \in \mathbb{Z} \mid 1 \leq x \leq H, 1 \leq y \leq W, 1 \leq z \leq Z\}$ is a voxel position in the map \mathcal{H} .

Visual Localization Module. The overview of the module is shown in Fig. 8. The main purpose of this module is to localize a query image in our map. To achieve this goal, we follow a hierarchical localization scheme (Sarlin et al. 2019, 2020). We first compute the NetVLAD (Arandjelovic et al. 2016) global descriptors and SuperPoint (DeTone et al. 2018) local descriptors for all images from the RGB stream during exploration and store them with the corresponding depth and odometry as a reference database. During inference, we compute the global and local descriptors for the query image in the same manner. By searching the nearest neighbor of the query NetVLAD features in the reference database, we can find a reference image as our candidate. Next, we use SuperGLUE (Sarlin et al. 2020) to establish key point correspondences between the query image and the reference image we retrieve with NetVLAD from the database with their local SuperPoint features. With registered depth, we back-project the reference image’s key points into the 3D space and obtain the 3D-2D correspondences for the query key points. In the end, we can apply the Perspective-n-Point method (Fischler and Bolles 1981) to estimate the query camera pose relative to the reference camera, and thus obtain the global camera pose with the odometry of the reference camera.

In the visual localization module, the predicted global camera location is denoted as $\mathbf{p}_v = (x_v, y_v, z_v)^T$. In the heatmap \mathcal{H}_v , we define the probability at \mathbf{p}_v as 1.0, and the probability linearly decays around this location according to the distance on the top-down map:

$$\mathcal{H}_v(\mathbf{p}) = \max(1.0 - \epsilon \cdot dist_{xy}(\mathbf{p}, \mathbf{p}_v), 0) \quad (3)$$

$$dist_{xy}(\mathbf{p}, \mathbf{q}) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (4)$$

where ϵ is the decay rate, and $dist_{xy}(\mathbf{p}, \mathbf{q})$ denotes the distance between 3D vectors \mathbf{p} and \mathbf{q} on the xy -plane.

Object Localization Module. The overview of the object localization module is shown in Fig. 9. This module is abstracted from the VLMaps we introduced in Sec. 3.1, Sec. 3.2, and Sec. 3.3 with some minor changes. The key idea is to exploit an open-vocabulary segmentation method (e.g., LSeg (Li et al. 2022) or OpenSeg (Ghiasi et al. 2022))

for pixel-level feature generation from the RGB image and to associate these features with the back-projected depth pixels in the 3D reconstruction. Different from Sec. 3.1, we don’t project the 3D points into the top-down plane to create a 2D grid map but maintain a 3D voxel map where each voxel is associated with a visual-language feature. When there are multiple points projected into the same voxel, we store their mean features at the voxel. The inference process is similar to Sec. 3.2. We define a list of categories in natural language and encode them with the language encoder. We compute the cosine similarity scores between all voxel-wise features and language features and use an argmax operator to select the top-scoring voxels for a certain category in the map. Depending on the application, the top-scoring 3D voxel points for a certain category can be used as the target point cloud for manipulation tasks or can be projected onto a top-down map for navigation purposes.

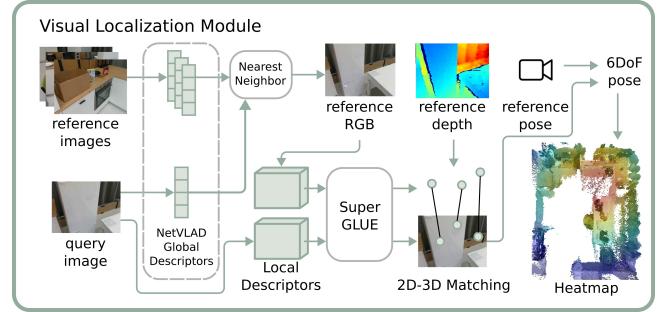


Figure 8. The overview of the visual localization module. We follow the hierarchical localization scheme by using NetVLAD and SuperGLUE to localize the query image’s location before generating a heatmap as the indexing result.

The object localization results are a list of points, denoted as $\{\mathbf{p}_{oi} = (x_{oi}, y_{oi}, z_{oi}) \mid i = 1, \dots, N\}$ where N is the total number of points for the target object. We define the probabilities for all these locations as 1.0 in heatmap \mathcal{H}_o , and the probability linearly decays around these locations based on the Euclidean distance:

$$d_{min}(\mathbf{p}) = \min\{dist(\mathbf{p}, \mathbf{p}_{oi}) \mid i = 1, \dots, N\} \quad (5)$$

$$\mathcal{H}_o(\mathbf{p}) = \max(1.0 - \epsilon \cdot d_{min}(\mathbf{p}), 0) \quad (6)$$

where $d_{min}(\mathbf{p})$ denotes the minimal distance between \mathbf{p} and all object points $\{\mathbf{p}_{oi} \mid i = 1, \dots, N\}$, $dist(\mathbf{p}, \mathbf{q})$ denotes the Euclidean distance between \mathbf{p} and \mathbf{q} .

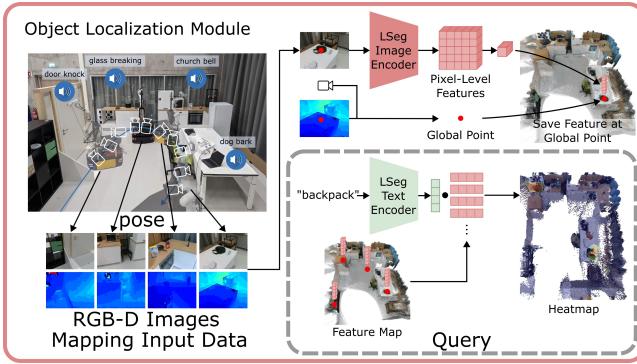


Figure 9. The overview of the object localization module. Similar to Sec. 3.1, during mapping, the RGB images in the exploration video are input to a vision-language model, LSeg (Li et al. 2021), to generate pixel-level features. Corresponding pixels are back-projected with depth images and transformed to locations in the global coordinate frame, where the features are associated with. During inference, the query text is encoded by LSeg’s text encoder, and a dot product with the point-level embeddings generates a score for each point. A heatmap is then created based on point scores and distances.

In AVLMaps, the object localization module is also used to generate a 2D obstacle grid map for path planning, as is shown in Sec. 3.3. Different from the 2D feature grid map of a VLMAP, the visual language map is now in the form of a 3D voxel grid where each occupied voxel is associated with a feature. We prompt the map with a list of free area concepts (e.g., “floor”) and obstacle concepts (e.g., “chair”, “table”, “counter”, and “other”) for a specific embodiment, assigning a score to each concept for every voxel. Each voxel is then labeled with the concept that achieves the highest score, resulting in a 3D semantic voxel map. We merge the voxels labeled with obstacle concepts into a combined obstacle map, and then perform a top-down projection to produce a 2D obstacle grid map. In this grid, all pixel locations corresponding to projected obstacle voxels are marked as occupied, while all others are marked as free and navigable. It is worth noting that beyond object categories, AVLMaps also support the definition of audio or images as obstacles. For example, we can define “the sound of glass breaking”, or a list of images as obstacles, generate their corresponding heatmaps, and treat locations with heat above a certain threshold as obstacles. This capability is useful when the forbidden regions are hard to describe with only object descriptions, like a glass-breaking scene without surrounding objects, or a region specified with a cell phone video.

Area Localization Module. While the object localization module is good at extracting object segments on the map, it falls short of localizing coarser goals such as regions (e.g., “the area of the kitchen”). This is because the visual encoder for generating pixel-aligned features is obtained by fine-tuning a pre-trained model on a segmentation dataset, leading to the notorious catastrophic forgetting effect. Therefore, the visual encoder is better at segmenting common objects while worse at recognizing general visual concepts (Jatavallabhula et al. 2023). To take advantage of both pre-trained and fine-tuned methods, we propose to build an extra sparse topological CLIP features map similar to (Shah et al. 2023).

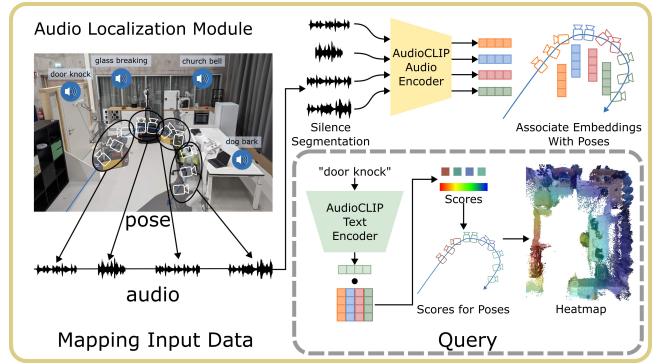


Figure 10. The overview of the audio localization module. During mapping, the exploration video’s audio is segmented by silence and encoded using AudioCLIP’s audio encoder. The resulting embeddings are linked to poses based on their timestamps. During inference, the query text is encoded by AudioCLIP’s text encoder, and a dot product with the pose embeddings generates a score for each pose. A heatmap is then created based on pose scores and distances.

The idea is to compute the CLIP visual features (Radford et al. 2021) for all images from the RGB stream and associate the features with corresponding poses. During inference, given the language concept like “the area of a bedroom”, we compute the language features with the CLIP language encoder and the image-to-language cosine similarity scores. These similarity scores indicate how likely these images match the language description. The odometry together with the score of each image indicates the predicted location with a confidence value.

The area localization results are a list of position-confidence pairs, denoted as $\{(\mathbf{p}_{ai}, s_{ai}) \mid i = 1, \dots, M\}$ where M is the total number of frames in the input RGB-D stream. The scores s_{ai} are normalized between 0 and 1. We define the probability for each point \mathbf{p}_{ai} on the heatmap \mathcal{H}_a as its score s_{ai} , and the probability linearly decays around the point on the xy -plane direction:

$$\bar{\mathcal{H}}_a(\mathbf{p}) = \max_{i=1, \dots, M} \{s_{ai} - \epsilon \cdot dist_{xy}(\mathbf{p}, \mathbf{p}_{ai})\} \quad (7)$$

$$\mathcal{H}_a(\mathbf{p}) = \max(\bar{\mathcal{H}}_a(\mathbf{p}), 0) \quad (8)$$

where the max operator for the curly brackets means taking the highest probability when a location is inside the affected regions for several \mathbf{p}_{ai} .

Audio Localization Module. The overview of the module is shown in Fig. 10. In this module, we utilize the audio information from the input stream. The key idea is to compute the audio-lingual features with audio-language pre-trained models such as wav2clip (Wu et al. 2022), AudioCLIP (Guzhov et al. 2022) or CLAP (Elizalde et al. 2023). We first segment the whole audio clip into several segments with silence detection. Whenever the volume is above a threshold, we mark this time step as the starting point of a segment. Whenever the volume of the sound is not larger than this threshold for a certain duration, we end the segment. In the next step, we compute the audio features for each segment with pre-trained audio-language models and associate the features with the odometry at the specific segment. During inference, given a language description of

the sound, like “the sound of door knocks”, we encode the language into language features and compute the matching scores between the language and all audio segments in the same way as in the object localization module. The odometry associated with the top-scoring segment is the predicted location.

The audio localization results are similar to those of the area localization module. The position-score pairs are denoted as $\{(\mathbf{p}_{si}, s_{si}) \mid i = 1, \dots, K\}$ where K is the total number of sound segments in the input video stream. The heatmap \mathcal{H}_s is defined as:

$$\bar{\mathcal{H}}_s(\mathbf{p}) = \max_{i=1, \dots, K} \{s_{si} - \epsilon \cdot dist_{xy}(\mathbf{p}, \mathbf{p}_{si})\} \quad (9)$$

$$\mathcal{H}_s(\mathbf{p}) = \max(\bar{\mathcal{H}}_s(\mathbf{p}) - \epsilon \cdot dist_{xy}(\mathbf{p}, \mathbf{p}_{si}), 0) \quad (10)$$

3.6 Cross-Modality Reasoning

A key advantage of our method is its capability to disambiguate goals with additional information, even from different modalities. The goal of the cross-modality reasoning method is to output a target location of a specific concept (for example, “the sofa”) given the information of other nearby concepts (for example, “near the sound of glass breaking”). In the last section, we introduced how we generate heatmaps for concepts of different modalities. Given these heatmaps, we want to further narrow down the target location.

Cross-Modal Reasoning. The main idea of our cross-modal reasoning method is shown in Fig. 11. We treat the predictions from four modules as four modalities. When there are several queries referring to different modalities, we compute the respective heatmaps first and then perform element-wise multiplication among all heatmaps:

$$\mathcal{H}_{target} = \mathcal{H}_1 \odot \dots \odot \mathcal{H}_L \quad (11)$$

where \odot is the element-wise multiplication operator, and L is the total number of referred modalities. We extract the position on the target heatmap \mathcal{H}_{target} that has the highest probability as the predicted location.

When we compute the heatmaps, there is always a primary heatmap while others are auxiliary ones. For example, in Fig. 11, in the query “the sound of baby crying near the sofa”, the heatmap for “the sound of baby crying” is the primary heatmap, while the heatmap for “the sofa” is the auxiliary. We set the decay rate for the primary heatmap higher (e.g., 0.1 in this work for voxel map with 0.05 meter voxel size) since we want to know the exact location of the target while tuning the decay rate for the auxiliary heatmap lower (e.g., 0.01) as having a broader effect area to narrow down major targets is desirable. More specifically, a higher decay rate indicates that the relevancy to the concept decreases faster when the location is farther away from the concept locations (heat value decreases 0.1 for every 0.05 meter). When there are multiple concepts or modalities mentioned in the target specification, the map with a high decay rate refers to the target concept we want the robot to get closer to. Those maps with low decay rates serve as constraints to select targets in the main map. From another perspective, the decay rates of multiple maps represent the importance weights we assign

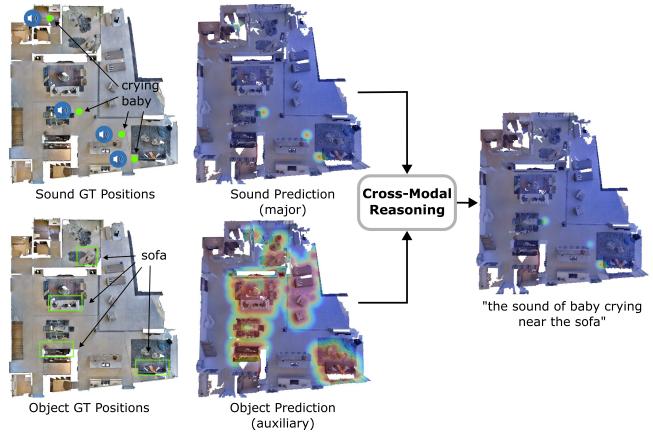


Figure 11. The key idea of cross-modal reasoning is converting the prediction from different modalities into heatmaps, and then fusing them with element-wise multiplication, effectively using complementary multimodal information to resolve ambiguous prompts.

to their corresponding concepts. The higher the decay rate is, the more important that concept is, and we want the robot to get closer to that concept. When the decay rates of two concepts are the same, the two concepts are equally important, and the fusion of the two concepts’ heatmaps will peak at the middle points between those concepts.

It is worth noting that our cross-modal reasoning method is relatively simple and has the underlying assumption that the heatmaps from different modalities are conditionally independent. Despite this simplification, our method proves to be effective across our experiments. Nonetheless, exploring more advanced cross-modal fusion techniques remains an exciting direction for future work.

3.7 Multimodal Goal Navigation from Language

In the setting of multimodal goal navigation from language, the agent is given language descriptions referring to targets from different modalities (e.g., sounds, images and objects), and is required to plan paths to them. While most of the previous navigation methods focus mainly on a specific type of goal, we unify these tasks with the help of large language models (LLMs). Following a similar spirit to Sec. 3.4, we use an LLM to interpret the natural language commands and synthesize API calls combined with simple logic structures in the form of executable python code. In the following, we will detail the code generation process for multimodal goal navigation, including (i) the introduction of the API library as a tool set for the LLM to use during code generation, (ii) the mechanism of spatial goal reasoning, (iii) the way we generate multimodal maps with code, (iv) the method to achieve cross-modal reasoning with code, and (v) the conversion of code to navigation commands that robots receive. At the end, we provide the prompt and an example of inference for our method.

Navigation API library. In Table 2, we listed all the APIs we provide to the LLM for potential usage. Compared to VLMaps’ API library in Table 1, we simplify and adapt the library to more modalities for AVLMaps. We provide a list of examples consisting of language instructions and

APIs	Functions
<code>load_image(path)</code>	load an image as an array from the path.
<code>move_to(pos)</code>	move to a position on the map.
<code>get_major_map(img=None, obj=None, sound=None)</code>	generate a heatmap with high heat decay rate for a specific modality (image, object description, or sound description).
<code>get_map(img=None, obj=None, sound=None)</code>	generate a heatmap with low heat decay rate for a specific modality (image, object description, or sound description).
<code>get_max_pose_3d(heatmap)</code>	retrieve the coordinate in a 3d voxel heatmap which has the maximal heat value.

Table 2: Navigation API library for multimodal goal navigation in AVLMaps.

generated code, which demonstrates the usage of those APIs as a context prompt (as in Fig. 12) to the LLM when we ask it to generate code during the inference time.

```
# move to the middle of the sound of cat meowing and
# the image: /path/to/image.png
img = robot.load_image("/path/to/image.png")
sound_map = robot.get_major_map(sound="cat meowing")
img_map = robot.get_major_map(img=img)
pos1 = robot.get_max_pos_3d(sound_map)
pos2 = robot.get_max_pos_3d(img_map)
pos = (pos1 + pos2) / 2
robot.move_to(pos)

# move to the window next to the sound of
# glass breaking
obj_map = robot.get_major_map(obj="window")
sound_map = robot.get_map(sound="glass breaking")
fuse_map = obj_map * sound_map
pos = robot.get_max_pos_3d(fuse_map)
robot.move_to(pos)

# move to the sound of crying baby next to the
# counter
obj_map = robot.get_map(obj="counter")
sound_map = robot.get_major_map(sound="crying baby")
fuse_map = obj_map * sound_map
pos = robot.get_max_pos_3d(fuse_map)
robot.move_to(pos)

# move to the table next to the sound of
# crying baby and the sound of dog
obj_map = robot.get_major_map(obj="table")
sound_map = robot.get_map(sound="crying baby") /
* robot.get_map(sound="dog")
fuse_map = obj_map * sound_map
pos = robot.get_max_pos_3d(fuse_map)
robot.move_to(pos)

# move to the middle of the table and
# the chair next to the sound of crying baby
obj_map_1 = robot.get_major_map(obj="table")
obj_map_2 = robot.get_major_map(obj="chair")
sound_map = robot.get_map(sound="crying baby")
fuse_map = obj_map_2 * sound_map
pos1 = robot.get_max_pos_3d(obj_map_1)
pos2 = robot.get_max_pos_3d(fuse_map)
pos = (pos1 + pos2) / 2
robot.move_to(pos)
```

Figure 12. The full context prompt (prompt in gray) AVLMaps used for achieving all navigation tasks in the experiments.

```
# move in between the image ./006899.png and the
backpack near the sound of glass breaking
img = robot.load_image("./006899.png")
img_map = robot.get_major_map(img=img)
obj_map = robot.get_major_map(obj="backpack")
sound_map = robot.get_map(sound="glass breaking")
fuse_map = obj_map * sound_map
pos1 = robot.get_max_pos_3d(img_map)
pos2 = robot.get_max_pos_3d(fuse_map)
pos = (pos1 + pos2) / 2
robot.move_to(pos)
```

Figure 13. The query and the generated results from the LLM. During the query, the context prompt in Fig. 12 and the input task commands are prompted to the LLM together. The input task commands are in green and generated outputs are highlighted

Spatial Goal Reasoning. We follow the intuition in Sec. 3.4 that spatial locations can be computed with simple math during code generation. For example, the location of “in between the counter and the fridge” can be obtained by getting the positions of the counter and the fridge respectively and apply an average to the two locations. In the multimodal goal navigation prompt, we reduce the spatial goal API calling examples to lay the focus more on multimodal targets and cross-modal reasoning. In principle, more diverse spatial concept reasoning examples could be integrated into the prompt, as is shown in Sec. 3.4.

Multimodal Heatmap Generation with Code. For heatmap generation, we implement interfaces `get_major_map(obj=None, sound=None, img=None)` and `get_map(obj=None, sound=None, img=None)` (Table 2). They take an object name, a sound name, or an image as input and output heatmaps indicating the locations of targets. These two functions are implemented following the localization modules in Sec. 3.5. The `get_major_map` generates heatmaps with a higher decay rate while `get_map` creates heatmaps with a lower decay rate. To support the image prompt, we add the image path in the language query like “the image /path/to/image.png” and use LLMs to call the image loading APIs.

Cross-Modal Reasoning with Code. As is introduced before, the logic of the cross-modal reasoning is relatively simple, which is just an element-wise multiplication of all relevant heatmaps. In the code, it can be performed with one line of code `fusemap = map1 * map2`.

Navigation Commands from Code. The API `move_to(pos)` takes a 3D voxel grid position as input, projects it onto the 2D grid map. It applies a planning algorithm on the grid map to generate a path on the map. The points on the path are treated as a list of subgoals and are used to generate a list of low-level actions (the action space contains turning left or right 5 degrees and moving forward 0.25 meter) to reach them sequentially. In the end, the list of low-level action commands is executed to finish the navigation instruction.

Prompt and an Inference Example. Since large language models are great few-shot learners (Brown et al. 2020), they are able to learn and imitate internal patterns when several simple examples are provided as context in addition to the direct query. To enable the LLM to understand how to use our APIs, we need to provide a few examples of how to use these APIs to tackle tasks described with language instructions. During the inference, we prompt the full context prompts (in gray) in Fig. 12 together with task command (in green) in Fig. 13 and an example of the generated outputs are highlighted. In our work, we use OpenAI’s text-davinci-003 model as our LLM for all experiments.

4 Experiments

In this section, we aim to evaluate our multimodal spatial map representations in a variety of tasks. More specifically, we address nine key questions: (i) how is VLMaps’ spatial language goals navigation performance compared to recent open-vocabulary navigation baselines (Sec. 4.1), (ii) whether VLMaps with their capacity to specify open-vocabulary obstacle maps can provide utility in improving the navigation efficiency of different robot embodiments (Sec. 4.2), (iii) how AVLMaps enable a robot to navigate to multimodal goals, including sound, image, and object queries (Sec. 4.4), (iv) how our cross-modality reasoning approach helps a robot to disambiguate goals with multimodal information (Sec. 4.5), (v) how the performance of AVLMaps scales with recent advanced foundation models specialized in different modalities (Sec. 4.7), and (vi) how AVLMaps’ multimodal indexing and reasoning capabilities translate to real-world environments, empowering robots with diverse embodiments to perform tasks that demand comprehensive multimodal understanding, such as mobile navigation (Sec. 4.8) and table-top manipulation with multimodal prompts (Sec. 4.9).

4.1 Zero-Shot Spatial Goal Navigation from Language

Experimental setup. We use the Habitat simulator (Savva et al. 2019) with the Matterport3D dataset (Chang et al. 2017) for the evaluation of multi-object and spatial goal navigation tasks. The dataset contains a large set of realistic indoor scenes that help evaluate the generalization capabilities of navigating agents. To evaluate the creation of open-vocabulary multi-embodiment obstacle maps, we adopt the AI2THOR simulator due to its support of multiple agent types, such as LoCoBot and drone. In these two environments, the robot is required to navigate in a continuous environment with actions: **move forward 0.05 meters, turn left 1 degree, turn right 1 degree and stop**. For map creation in Habitat, we collect 12,096 RGB-D

frames across ten different scenes and record the camera pose of each frame.

Baselines. We evaluate VLMaps against three baseline methods, all of which utilize visual-language models and are capable of zero-shot language-based navigation:

- LM-Nav (Shah et al. 2023) creates a graph where image observations of an environment are stored as nodes while the proximity between images are represented as edges. By combining GPT-3 and CLIP, it parses language instructions into a list of landmarks and plans on the graph towards corresponding nodes.
- CLIP on Wheels (CoW) (Gadre et al. 2023) achieves language-based object navigation by building a saliency map for the target category with CLIP and GradCAM (Selvaraju et al. 2020). By thresholding the saliency values, it retrieves a segmentation mask for the target object category and then plans the path on the map.
- CLIP-features-based map (CLIP Map) is an ablative baseline that generates a feature map for the environment in a similar way as ours. Instead of using LSeg visual features, it projects the CLIP visual features onto the map averaged across views. Object category masks are generated by thresholding the similarity between map features and the object category features.

For additional context and analysis, we also report results from a system that has access to a ground truth semantic map for navigation (GT Map), to provide a systems-level upper bound on performance.

Tasks Collection. In these experiments, we investigate the performance of VLMaps versus other baselines for zero-shot *spatial* goal navigation from language. Our benchmark consists of 21 trajectories in seven scenes, with manually specified corresponding language instructions for evaluation. Each trajectory contains four different spatial locations as subgoals. Examples of subgoals are “east of the table”, “in between the chair and the sofa”, or “move forward 3 meters”. There are also instructions for the robot to realign itself in reference to nearby objects such as “with the counter on your right”. We only consider a subgoal as having been achieved, when the robot reaches the subgoal location within a range of one meter. To evaluate the long-horizon navigation capabilities of the agents, we compute the success rate (SR) of continuously reaching one to four subgoals in a sequence, shown in Tab. 3. For all map-based methods, including CoW, CLIP Map, ground truth semantic map, and our method, we apply the code generation techniques introduced in Sec. 3.4. For LM-Nav, we simply use the same parsing method in the original paper (Shah et al. 2023) to break down the language instruction into subgoals.

Tab. 3 summarizes the zero-shot spatial goal navigation success rates. Our method outperforms other baselines in this task. Different from object navigation tasks where agents only need to approach a certain object type within a range, disregarding the relative spatial shift to the object, the language-based spatial goal navigation tasks require the robot to accurately arrive at the described location in reference to the object. This poses a bigger challenge to

Tasks	No. Subgoals in a Row			
	1	2	3	4
LM-Nav (Shah et al. 2023)	5	5	0	0
CoW (Gadre et al. 2023)	33	5	0	0
CLIP Map	19	0	0	0
VLMaps (ours)	62	33	14	10
GT Map	76	48	33	29

Table 3: The success rate (%) of zero-shot spatial goal navigation with language.

the landmark localization ability of the method. The low localization ability of CoW and CLIP Map leads to their high failure rates in this task.

4.2 Cross-Embodiment Navigation

Experimental Setup. We collect 1,826 RGB-D frames across ten rooms in AI2THOR (Kolve et al. 2017) and build the VLMaps for these scenes. We study the ability of VLMaps to improve navigation efficiency by retrieving different obstacle maps for different embodiments (given the same VLMap) in navigation tasks. We evaluate more than 100 sequences of object subgoals in the AI2THOR simulator. We evaluate VLMaps on both a LoCoBot and a drone to test its capability of generating obstacle maps at runtime for multi-embodiment navigation.

Obstacle Maps Generation. We apply the open-vocabulary obstacle map generation method in Sec. 3.3 to create an obstacle map for the drone (drone map) and one for the LoCoBot (ground map) by defining obstacles for them differently. For the LoCoBot (ground robot), we first define a potential obstacle list as [“chair”, “wall”, “wall above the door”, “table”, “window”, “floor”, “stairs”, “other”] and perform open-vocabulary landmark indexing. Later, we only select the union of the masks for the objects “wall”, “chair”, “table”, “window”, “stairs”, “other” as the obstacle map. For the drone (flying robot), we perform landmark indexing with the potential obstacle list: [“chair”, “sofa”, “wall”, “table”, “counter”, “window”, “floor”, “stairs”, “ceiling lights”, “cabinet”, “counter support”, “other”]. Afterwards, we take union of the masks for [“wall”, “window”, “stairs”, “ceiling lights”, “cabinet”, “other”] to generate the obstacle map.

Baselines. We test the navigation ability of these embodiments with three setups: a LoCoBot with a ground map, a drone with a ground map, and a drone with a drone map.

Metrics. We evaluate the Success Rate (SR) and the Success rate weighted by the (normalized inverse) Path Length (SPL) (Anderson et al. 2018a) defined as: $SPL = \frac{1}{N} \sum_{i=1}^N S_i \frac{l_i}{\max(p_i, l_i)}$ where N is the total number of evaluated tasks, $S_i \in \{0, 1\}$ is the binary indicator of success, l_i denotes the ground truth shortest path length, and p_i denotes the actual path length of the agent in navigation. This metric indicates how efficient the actual path is compared to the ground truth shortest path when the

navigation task is achieved. In our three setups, the ground truth trajectories for the LoCoBot and the drone are planned on floor-level and on height level of 1.7 meters respectively.

The results provided in Tab. 4 show that the average navigation success rates of the ground-map version of the LoCoBot and the drone are similar because the same obstacles map is used for planning. However, there is an obvious gap between their SPL values. This is because when the drone does not have access to a customized obstacle map, it fails to benefit from flying over ground objects to improve the navigation efficiency. In contrast, while achieving similar success rate compared to the drone with a ground map, the drone with a drone map manages to navigate with higher path efficiency, reflected by the increased SPL values. The comparable SPL values for the drone with the drone map and the LoCoBot with the ground map shows that VLMaps help to generalize the navigation efficiency among different embodiments. An example of the multi-embodiment object navigation task is shown in Fig. 14, where by defining a more efficient obstacles map, the drone flies over the sofa and reaches the laptop target directly, while the LoCoBot has to move aside first to avoid colliding with the sofa.

4.3 Multimodal Navigation Simulation Setup

Exerimental setup. We use the Habitat simulator (Savva et al. 2019; Szot et al. 2021) with the Matterport3D dataset (Chang et al. 2017) for the evaluation of multimodal navigation tasks. For mapping purposes, we manually collect RGB-D video streams in the simulator across ten different scenes and add random audio tracks to the videos to simulate the audio sensing modality. All audio comes from the validation fold (Fold-1) of the ESC-50 dataset (Piczak 2015), which contains 50 categories of common sounds. In navigation tasks, the robot has four actions to take: **move forward 0.1 meters**, **turn left 5 degrees**, **turn right 5 degrees**, and **stop**. In sequential goal setting, the robot is required to navigate to a sequence of goals and take the **stop** action when it reaches each subgoal. When the stop position is less than one meter away from the ground truth position, the subgoal is considered successfully finished.

Tasks collection. In multimodal goal navigation tasks in Sec. 4.4, we consider three kinds of goals: image goals, object goals, and sound goals. For image goals, we randomly sample positions and orientations on the top-down map and render images as targets. For object goals, we access the metadata (e.g., bounding boxes and semantics) from the Matterport3D dataset and sample a list of categories in each scene as queries. For sound goals, we randomly sample sound classes of audio merged with the mapping videos as targets, treating the video frame positions as the ground truth.

In cross-modal goal indexing tasks in Sec. 4.5, we collect three types of datasets:

- **Visual-Object cross-modal indexing** We manually select image-object pairs on the top-down map for localizing “an object X near the image Y”.
- **Area-Object cross-modal indexing** We access the region and object metadata (e.g., bounding boxes and semantics)



Figure 14. VLMaps enable different embodiments to define their own obstacle maps for navigation. The left image shows the top-down view of an environment. The middle columns show the observations of agents during navigation. The images on the right demonstrate the obstacles maps generated for different embodiments and the corresponding navigation paths.

Tasks	No. Subgoals in a Row								Indep. Subgoals SR	
	1		2		3		4			
	SR	SPL	SR	SPL	SR	SPL	SR	SPL		
LoCoBot (ground map)	53	49.0	28	17.8	14	6.7	6	2.5	52.3	
Drone (ground map)	53	41.8	28	15.5	14	5.3	6	2.0	53.3	
Drone (drone map)	56	45.4	30	16.3	17	7.0	7	2.5	55.0	

Table 4: The success rate (%) and SPL of multi-embodiment object goal navigation with language.

from the Matterport3D dataset to automatically generate a list of object-region pairs. This dataset is for localizing “an object X in the area of Y”.

- **Object-Sound cross-modal indexing** We manually insert several sounds of the same kind into a scene and select for each sound location a nearby object for disambiguation. The query is “a sound X near the object Y”.

In cross-modal goal navigation in Sec. 4.6, we randomly sample starting pose in ten scenes and treat the visual-object and object-sound cross-modal goals in Sec. 4.5 as navigation goals. For all cross-modal navigation and indexing tasks, we use the prompt introduced in Sec. 3.7 to generate navigation commands.

4.4 Multimodal Goal Navigation

Sound goal navigation. We first test AVLMaps in sound goal navigation tasks. We collect 200 sequences of sound goals in ten different scenes. In each sequence, there are four sound categories that require the robot to reach. The results are shown in Tab. 5. We generate AudioCLIP (Guzhov et al. 2022) features with our audio localization module and match all audio with the target sound category in the embedding space, similar to a text-to-audio retrieval setup. Then, the agent plans a path to the audio position. We tested different ranges of sound categories inserted into the map. The full list of sound categories in each major class can be found in the link¹. The results show that our agent manages to recognize sound goals and navigate with a 77.5% success rate.

Visual and object goals navigation. We then test AVLMaps with visual and object goal navigation tasks. The agent is given an image and two object categories in the language in one sequence of tasks and asked to navigate to the image

Tasks	No. Subgoals in a Row				Independent Subgoals
	1	2	3	4	
Domestic Sound (10 categories)	59.5	33.0	15.5	7.0	62.5
+ Human Sound (20 categories)	69.5	47.0	36.5	23.0	72.38
+ Animal Sound (30 categories)	74.5	58.5	45.5	33.0	77.5

Table 5: The success rate (%) of sound goal navigation with AVLMaps.

goal and two object goals in sequence. The success rate for 200 sequences of tasks in ten scenes is reported in Tab. 6. The results show that our method enables the agent to navigate to goals from different modalities.

Tasks	No. Subgoals in a Row			Independent Subgoals
	1	2	3	
AVLMaps (Ours)	71.5	40.5	25.0	47.4

Table 6: The success rate (%) of multimodal goal navigation with AVLMaps. The agent is required to navigate to one visual goal, and two object goals in sequence.

4.5 Cross-Modal Goal Indexing

When we refer to a goal with language, it is likely that the goal can be found in more than one place in the environment. A major strength of our method is that it can disambiguate goals with multimodal information. In this experiment, we will show the cross-modal goal reasoning capability of AVLMaps.

Area-Object goal indexing. In this setup, we use an area description to disambiguate the object goal. We collected

100 indexing tasks in ten scenes. Each task consists of an object category and a region category (e.g., “living room”, “kitchen”, “dining room”, “bathroom” etc.). The agent needs to predict the correct object location which is inside the region. The top-one recall with different distance tolerance is reported in Tab. 7. We notice that VLMMaps (Huang et al. 2023b) struggles to find the goal in the correct region because VLMMaps integrates visual-language features from the encoder fine-tuned on the instance segmentation dataset, improving its segmentation performance on common objects while dropping its ability to recognize more general concepts like regions. In contrast, ConceptFusion integrates pre-trained CLIP features into the map without fine-tuning, enabling it to recognize general concepts including regions, and thus the indexing results are improved.

Method	Recall@1 (%)				Average min. distance (m)
	<0.5m	<1m	<1.5m	<2m	
baseline (VLMMaps)	5.56	7.78	13.33	17.78	8.22
+ ConceptFusion	12.22	13.33	16.67	21.11	7.60
+ CLIP sparse (Ours)	15.56	24.44	31.11	35.56	6.17
+ GT region map	37.78	44.44	55.56	61.11	2.62

Table 7: The recall (%) of area-object cross-modal indexing.

Object-Sound goal indexing. In this setting, we use object goals to disambiguate sound goals. We collected 119 indexing tasks, each of which consist of a sound category and a nearby object category. Each sound category in a scene can be heard at more than one location, introducing ambiguity to the localization scenario. The recall is reported in Tab. 8. With the combination of object and audio localization modules, our method largely increases the recall rate for localizing the correct sound goal position in ambiguous scenarios.

Method	Recall@1 (%)				Average min. distance (m)
	<0.5m	<1m	<1.5m	<2m	
baseline (wav2clip)	8.40	10.08	10.92	14.29	8.52
baseline (AudioCLIP)	26.05	35.29	36.97	42.01	5.04
VLMMaps + wav2clip	24.37	30.25	33.61	38.66	6.27
VLMMaps + AudioCLIP (Ours)	53.78	65.55	67.23	70.59	2.74

Table 8: The recall (%) of object-sound cross-modal indexing.

Visual-Object goal indexing. In visual-object goal indexing tasks, visual clues are used to resolve ambiguity. Given an object category and an image, our method can localize the correct object near the image position with over 60% of recall for 0.5 meters distance tolerance, as is shown in Tab. 9.

Method	Recall@1 (%)				Average min. distance (m)
	<0.5m	<1m	<1.5m	<2m	
VLMMaps w/o visual module	7.55	9.43	11.32	11.94	11.22
VLMMaps w/ visual module (Ours)	62.26	66.67	70.44	72.32	3.11

Table 9: The recall (%) of visual-object cross-modal indexing.

4.6 Multimodal Ambiguous Goal Navigation

In this part, we test our method with ambiguous goal navigation tasks. We collect 119 sequences of tasks. In each task, the agent is required to navigate to an ambiguous sound goal (i.e., “move to the sound of baby crying near the sofa”) and an ambiguous object goal (i.e., “move to the counter near {image of the kitchen}”) sequentially. The category of the object near the sound and the image taken near the object are provided. These tasks require the agent to reason across different modalities to accurately localize the target. We consider two single-modality baselines: VLMMaps (Huang et al. 2023b) and AudioCLIP (Guzhov et al. 2022) and one multimodal baseline. The multimodal baseline uses VLMMaps as the object localization module, wav2clip (Wu et al. 2022) as the audio localization module and the same visual localization module as our method. The results are shown in Tab. 10. We observe that AVLMaps navigates to cross-modal goals with a 24.2% higher success rate to ambiguous sound goals and with a 2.1% higher success rate to ambiguous object goals compared to the alternative multimodal baseline.

Tasks	No. Subgoals in a Row		Sound Goals	Object Goals
	1	2		
VLMMaps	-	-	-	27.1
AudioCLIP	-	-	16.9	-
VLMMaps + wav2clip	22.0	12.7	22.0	53.4
VLMMaps + AudioCLIP (Ours)	46.2	28.6	46.2	55.5

Table 10: The success rate (%) of multimodal ambiguous goal navigation with AVLMaps. The agent is required to navigate to one ambiguous sound goal and one ambiguous object goal sequentially.

ALM	VLM	No. Subgoals in a Row				Sound	Object
		1	2	3	4		
AudioCLIP	LSeg	71.0	59.0	29.0	17.0	71.0	41.8
CLAP	LSeg	81.0	69.0	36.5	19.0	81.0	52.0
CLAP	SAM+CLIP	80.0	68.5	29.5	12.0	80.0	40.5
CLAP	OVSeg	81.0	69.0	37.0	21.0	81.0	53.5

Table 11: The success rate (%) of multimodal goal navigation with different foundation models. The agent is required to navigate to one sound goal, one visual goal, and two object goals in sequence. The right-most three columns indicate the independent success rate of navigating to specific types of goals.

4.7 Scaling Experiment

Since AVLMaps is highly modular, each module can be upgraded with advanced foundation models that generate similar audio-language features or visual-language features. In this section, we explore whether AVLMaps can evolve with the advancement in foundation model research. We follow similar settings of multimodal goal navigation as in Sec. 4.4 to test the AVLMaps modules supported by different foundation models. In this experiment, the robot needs to navigate to a sound goal, a visual goal, and two object goals in a sequence and we report the in-a-row navigation success rate as well as the success rate for each type of goal. In this experiment, we mainly focus on analyzing how the performance scales with improved Audio Language

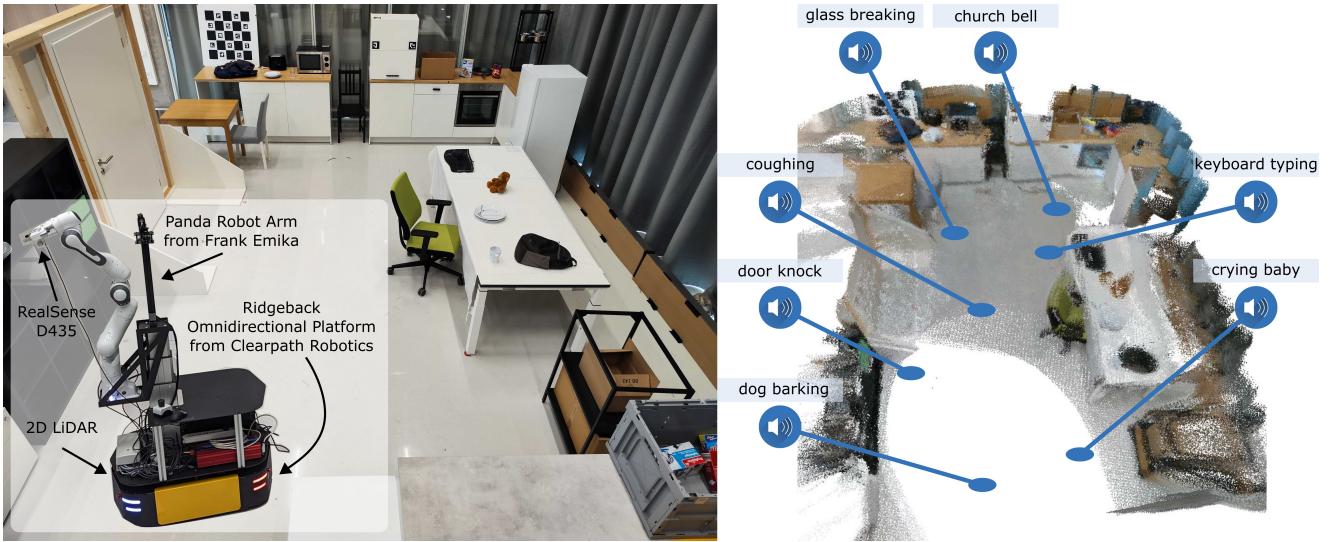


Figure 15. Real-world navigation experiments are conducted in a room with multiple ambiguous goals such as tables, chairs, backpacks, and paper boxes (left). The robot setup is also shown in the left image. We leverage dense SLAM techniques to build a 3D reconstruction of the scene from RGB-D camera data into which we anchor features from multiple foundation models (right). We artificially insert sounds with different semantics at locations shown in the image. Different sounds are played when the robot moves to these locations during mapping. Sounds are sampled from the ESC-50 dataset.

Models and Vision Language Models. We fixed the visual localization module as NetVLAD and SuperGLUE.

Audio Language Models. We compare the downstream performance of using AudioCLIP (Guzhov et al. 2022) with respect to a more recent CLAP (Elizalde et al. 2023) model.

Vision Language Models. In order to leverage pretrained Vision Language Models, we require the encoder to generate pixel-wise features in CLIP embedding space. We compared LSeg (Li et al. 2021), the VLM used in VLMaps (Huang et al. 2023b), with OVSeg (Liang et al. 2023a) and a method that uses SAM (Kirillov et al. 2023) and CLIP (Radford et al. 2021) introduced in HOV-SG (Werby et al. 2024). The method in HOV-SG first leverage SAM (Kirillov et al. 2023) to generate class-agnostic masks. Each region cropped with a mask and its zero-background version are encoded with a CLIP image encoder and their embeddings are summed with weights. The resulting embedding is assigned to all pixels in the masked region.

Results. We report the results in Table 11. The first two rows compare the success rates of different Audio Language Models (ALMs) using the same VLM. We observe a significant 10% improvement in navigating to sound goals when replacing AudioCLIP with CLAP. This improvement is largely due to CLAP’s more extensive pretraining on larger, more diverse datasets with advanced augmentation techniques (Elizalde et al. 2023). From the second to the final rows, we compared three VLMs that generate dense visual-language features but fixed CLAP as the ALM. However, recent VLMs did not offer clear benefits. In fact, the SAM+CLIP combination significantly reduced performance. Upon closer inspection, we found that while SAM+CLIP performed well in previous work (Werby et al. 2024), it requires extensive hyper-parameter tuning to adapt to different scenes. The quality of SAM’s masks

is highly sensitive to its parameters, and CLIP struggles to interpret masked regions, as highlighted by OVSeg (Liang et al. 2023a). OVSeg addressed this by training learnable mask prompts, and improving 2D semantic segmentation through ensemble techniques. However, these methods are designed to enhance segmentation performance, not visual-language feature generation. Like LSeg, OVSeg is fine-tuned on similar datasets, offering limited improvements in dense pixel-level visual-language features. As a result, AVLMaps’ object localization module, which relies on these features, sees minimal benefit from OVSeg. However, we believe that pixel-level visual language features can be improved with access to more high-quality segmentation datasets in the future, boosting the performance of the object localization module in AVLMaps.

4.8 Real World Experiment for Mobile Robot

To answer the question of how AVLMaps applies to real-world environments and benefits multimodal navigation, we designed a mobile navigation experiment in which the robot must locate sound, image, and object-based goals within an environment containing duplicate objects from certain categories such as chairs, backpacks, shelves and so on. This setup demonstrates how AVLMaps enables the robot to retrieve multimodal concepts and disambiguate goals by leveraging information from additional modalities.

Robot Setup. In the real-world experiment setting, we use a mobile robot equipped with a Ridgeback omnidirectional platform from Clearpath Robotics as the mobile base, and a Panda manipulator from Franka Emika. We mount a RealSense D435 RGB-D camera at the gripper of the Panda manipulator. During the mapping, we run a LiDAR localizer to provide the odometry for the robot base and derive the camera pose through the forward kinematics of the robot arm.

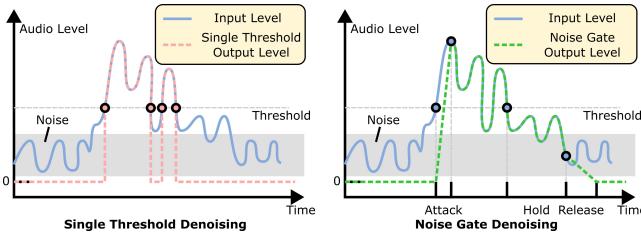


Figure 16. Audio denoising for real-world experiment. We pre-process the audio by applying a noise gate. Simply applying a threshold for filtering out low-level noise might lead to frequent fluctuation of the audio level, leading to fragmentation of the target audio (see the left). A noise gate contains “attack”, “hold”, and “release” phases, which introduce smooth transitions and prevent cutting off short audio signals (see the right).

Environment Setup. We choose a room with multiple ambiguous goals such as tables, chairs, paper boxes, counters, and backpacks, which are shown on the left in Fig. 15. We control the robot in this environment and record RGB-D video. Then we artificially add sounds to the RGB-D video when the robot moves to certain locations. The sound locations are shown on the right in Fig. 15.

Audio Recording and Denoising. To make the experiment more realistic, instead of simply adding clean audio to the video, we recorded environmental noise using a microphone during the robot’s exploration. We then overlaid semantic soundtracks from ESC-50 onto the recorded environmental noise at specific locations. This approach provides an approximation of real-world audio conditions with background noise. We also experimented with playing sounds through a speaker in the environment and recording both the environmental noise and semantic audio simultaneously. However, the mobile robot produced significant vibration noise while moving, which masked the played sounds and rendered them largely inaudible. As a result, we approximated the noisy semantic audio by mixing the semantic soundtracks with the recorded environmental noise.

Since current audio language models are sensitive to noise in input audio, we apply a noise filtering algorithm as a pre-processing step to the audio before sending them to the audio localization module. In our case, we use a noise gate, as shown in Fig. 16. A noise gate applies thresholding with smooth fade-in and fade-out transitions. Instead of simply zeroing out values below the threshold, it includes three phases: attack, hold, and release. When the input audio level exceeds the threshold, the output ramps up linearly from zero to the input level over a period defined by the attack time. Once the input drops below the threshold, the output stays at the input level for the duration of the hold time. If the input remains below the threshold after the hold period, the output level decreases linearly to zero over the release time. This approach effectively reduces environmental noise. We then apply silence segmentation to further extract meaningful audio clips. In our setup, the noise gate threshold is set to -10 dB ($\text{dB} = 20 * \log 10(\text{amplitude})$), with an attack time of 250 ms, hold time of 1000 ms, and release time of 170 ms. The silence segmentation threshold is set to 0.1. As noise cancellation is not the primary focus of this work, these parameters were selected based on empirical experience. In

the future, a more robust approach would involve adding random real-world noise during audio language model pretraining to improve noise tolerance.

Map Building and Navigation. After collecting the data, we run the AVLMaps mapping offline. For navigation tasks, we provide the AVLMaps and the language instruction as input. The robot parses the instruction (Sec. 3.7) and executes the generated Python code for goal indexing and planning. We use the ROS navigation package (Quigley 2009) for global and local planning. We pre-process sound inputs with background noise subtraction to avoid including noise from the robot operation.

Multimodal Spatial Goal Reasoning and Navigation with Natural Language. We design 20 language-based multimodal navigation tasks, asking the robot to navigate to sounds, images, and objects. We report an overall success rate of 50%. We also design an evaluation consisting of ten multimodal spatial goals. The agent needs to reason across object, sound, image and spatial concepts. An example is “navigate in between the backpack near the sound of glass breaking and {the image of a fridge}”. In the end, six out of ten tasks were successfully finished. We show in Fig. 17 the process of resolving ambiguities in the scene. There are different ambiguous objects in the scenes including paper boxes, backpacks, shelves, tables, chairs, and plates. The first and the second columns in Fig. 17 show the ground truth positions of the target objects and sounds. The third and fourth columns show that AVLMaps can accurately localize objects, sounds, and visual goals in the form of 3D heatmaps. The final column shows that our method can correctly narrow down targets in spite of object ambiguities. We can observe from the figure that AVLMaps can accurately localize ambiguous concept with language, audio and image. We observe that the failures come from the composition of the imperfection of different modules. For example, the object localization module (e.g., VLMaps) fails to recognize rare objects like various toys. It also mistakes some shelves for chairs. Similar failures happen in audio localization module. In the second row and the fourth column in Fig. 17, the church bell sound should be at the top-right corner but the module also gives high score for the sound heard at bottom-left.

4.9 Real World Experiment for Table-Top Goal Reaching

In the previous section, we demonstrated how AVLMaps empower a robot to interpret multimodal goals in room-scale mobile navigation tasks. Here, we extend our investigation to assess how AVLMaps benefit a fixed-based manipulator in real-world table-top tasks, which require a more detailed semantic understanding of the scene. In this setup, the robot manipulator must approach multimodal goals with a stricter tolerance for error (within 10 cm). Additionally, we explore AVLMaps’ potential for application across robots with varied embodiments.

Robot setup. We set up a Panda robot arm from Franka Emika on a table and mount a FRAMOS D345e industrial RGB-D camera at the gripper of the manipulator. Before the experiment, we calibrate the extrinsic matrix from the end

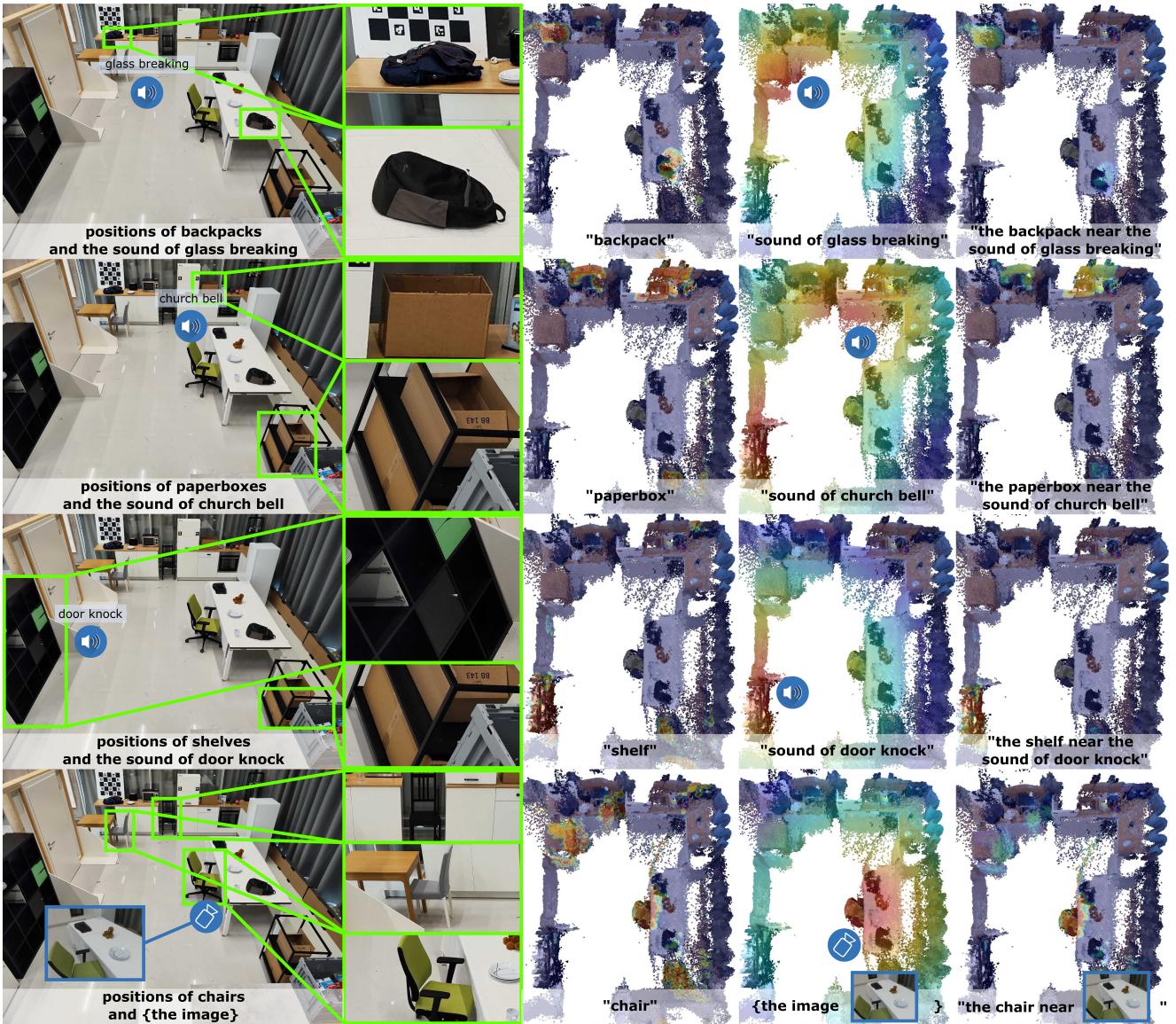


Figure 17. Visualization of example heatmaps in AVLMaps for multimodal goal reasoning for ambiguous object goals. The first column shows the positions of ambiguous objects (green bounding boxes) and the location of a sound (the icon of a speaker) or an image (the icon of a camera), while the second column shows the close-up view of ambiguous objects in the scene. The third column shows the predicted 3D heatmap for the object. The fourth column shows the heatmap for the extra modality, and the fifth column shows the fused heatmap after cross-modal reasoning. Sounds are artificially inserted into the scene for benchmarking and evaluation. The locations of sounds are not sound-source locations but the places where the sounds were heard. The heatmap is shown in the JET color scheme (red means the highest score and blue means the lowest score).

effector coordinate frame to the camera frame. We use an HTC Vive VR controller to teleoperate the robot end effector to collect observation data.

Experiment setup. We set up two tabletop scenes where different objects are lying on the table at random locations. In one scene, we deliberately place duplicate objects on the table to simulate the ambiguity of objects. Subsequently, we teleoperate the robot end effector with a VR controller and collect observations including the RGB, the depth, and the robot end effector poses relative to the robot base coordinate frame. The recording frequency is 30 Hz. Later, we derive the camera poses relative to the base coordinate frame using the end effector pose and the extrinsic matrix obtained during the calibration. For each scene, we first collect a sequence of data for generating maps with the object localization module and the visual localization module. We

then control the robot to different areas on the table and collect an episode of data in each region, to which we later insert a random segment of audio sampled from ESC-50 dataset in a similar way as the mobile robot experiment setting. These observation data augmented with audio are used to generate the audio map with the audio localization module. Thanks to the insights we obtained from the scaling experiments in Sec. 4.7, we use CLAP (Elizalde et al. 2023) and OVSeg (Liang et al. 2023a) as our foundation models for the audio localization module and the object localization module. For the visual localization module, we still use the NetVLAD (Arandjelovic et al. 2016) and SuperGLUE (Sarlin et al. 2020) scheme as in Sec. 3.5.

Tabletop manipulator goal reaching. We randomly moved the robot arm and collected a sequence of RGB images as the visual goals used for querying the AVLMaps created earlier.

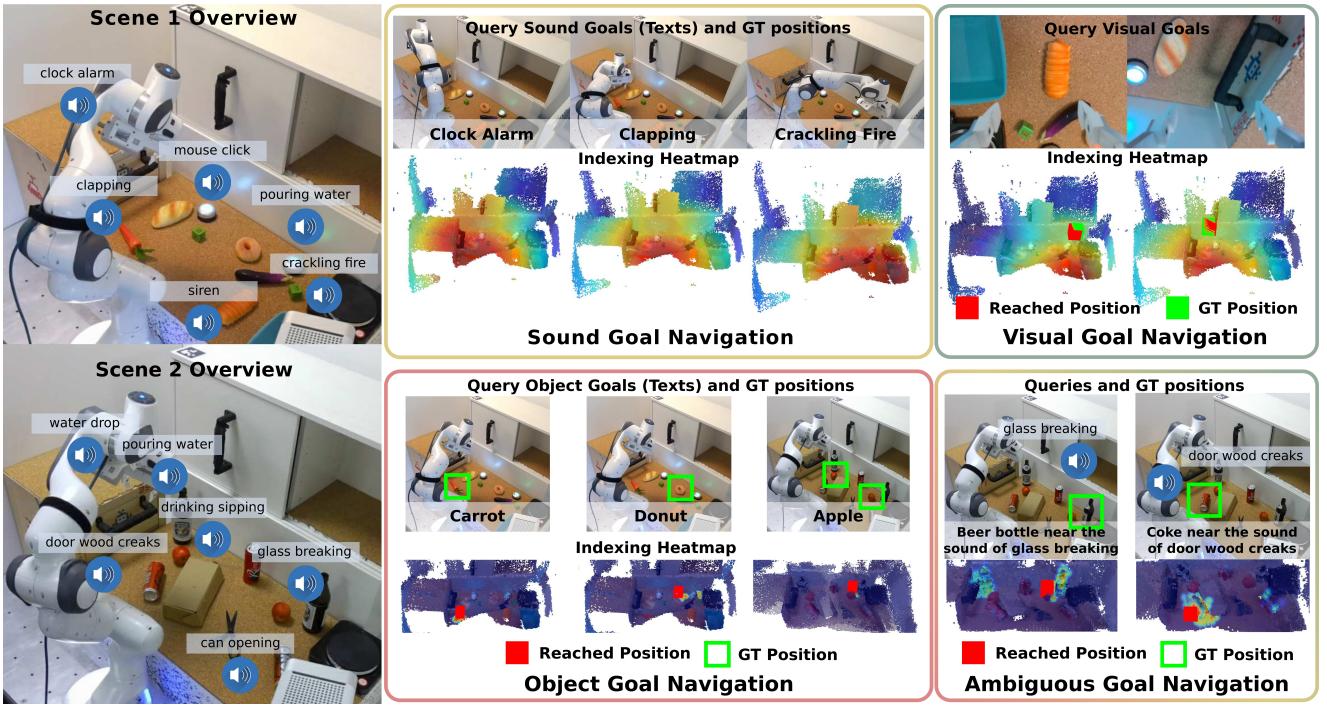


Figure 18. Visualization of tabletop goal-reaching experiments. We set up two tabletop scenes and inserted sounds to different locations in the observation data (left). We show the indexing heatmap results of sound goals, visual goals, object goals, and ambiguous goal reaching results on the right. In addition, for visual goals, object goals, and ambiguous goals, we show the ground truth target locations (green cubes or green boxes) as well as the reached positions (red cubes). The heatmap is shown in the JET color scheme (red means the highest score and blue means the lowest score).

During inference, we prompted the robot with randomly selected visual goals (sampled from the collected images), sound categories (matching the inserted audio types), and object categories (language descriptions of objects on the table), instructing it to approach the target region. If the final position of the robot’s end effector was within 10 cm of the ground truth, the trial was considered successful. We tested 20 visual goals, twelve sound goals, and 13 object goals. The robot successfully reached 100% of the visual and sound goals, and nine out of 13 object goals. Additionally, we tested ten ambiguous goals, such as “the light near the sound of clock alarm” or “the apple near the image {path/to/image}” and the robot successfully approached nine out of ten. In summary, AVLMaps demonstrated excellent performance in a tabletop setting, successfully navigating to multimodal goals, including ambiguous ones. The results are shown in Fig. 18.

5 Limitations and Discussions

While our multimodal spatial language maps approach is versatile in terms of navigating to various spatially grounded concepts and is effective in the disambiguation of duplicate goals with extra information, it does have certain limitations. In this section, we thoroughly discuss the major drawbacks of our method, along with potential directions for extension and improvement. Through this analysis, we aim to offer insights to the community and inspire future research.

Transient Sound Reaction. One of the main challenges with our AVLMaps is that sound is inherently transient, while the map relies on pre-exploration and offline mapping to embed the information for navigation tasks. Even if sounds

are associated with specific locations during the exploration phase, those sounds might disappear or shift to new locations by the time of the inference and navigation. Therefore, AVLMaps struggles to support reactive navigation towards transient sound goals. Previous work on vision and audio navigation (Chen et al. 2020; Younes et al. 2023; Gan et al. 2020b) has focused on enabling robots to respond to transient sounds using biaural and visual observations. However, these methods are limited to transient goals and have been tested only in simulated environments. We argue that both transient sounds and previously heard sounds are essential for intelligent navigation. Transient sounds serve as important cues for immediate action, while past sounds provide valuable references for narrowing down possible targets based on prior experiences. For example, when instructed to go to “the café where you heard the song”, we can recall the specific café and adjust our navigation accordingly. A robot must be able to understand and navigate toward both kinds of sound goals to achieve human-level intelligence in sound-based navigation. However, no current system integrates these dual capabilities. We believe this gap presents an exciting opportunity for future research.

Sound Localization. While AVLMaps manages to comprehend the semantic meaning of sounds, it struggles to localize the sound sources. In this paper, the audio localization module assumes monaural audio input, lacking in the ability to utilize binaural audio to localize sound sources with triangulation like humans. More specifically, a sound can be heard in all locations inside a room, but we only associate its features with the robot’s current location. Encouragingly, several concurrent works are actively addressing the sounds source localization, including efforts to learn the real-world acoustic

sound field (Chen et al. 2024b), reconstruct the acoustic properties of environments (Wang et al. 2024), and develop more acoustically realistic simulation environments (Chen et al. 2022).

Dynamic Scenes. Another challenge faced by our multi-modal spatial language maps (both VLMaps and AVLMaps) is their inability to handle dynamic scenes. One form of dynamics involves in-view dynamics, such as walking humans and moving articulated objects during exploration. These dynamic entities easily corrupt the object map, leaving behind point artifacts that trace their movement trajectories. The semantic features of the moving objects may be erroneously associated with these artifact points, leading to inaccurate representations of their true locations. Another form of dynamics involves long-term dynamics, such as object relocations between the exploration and inference phases. These relocations can invalidate the pre-built map, necessitating an updating mechanism to ensure accurate navigation. Currently, our multimodal spatial language maps lack mechanisms to address both types of dynamics. To explore potential solutions, we investigated approaches to mitigating the impact of dynamics. Prior works have addressed in-view dynamics by removing or tracking certain classes of semantic masks during mapping (Xu et al. 2019; Runz et al. 2018). For long-term dynamics, recent research has proposed learning-based object association methods to update the locations of relocated objects in the map (Yan et al. 2025; Huang et al. 2025). These solutions could be integrated into our mapping pipeline to enhance its robustness.

Extension to More Modalities. In this paper, we have demonstrated the feasibility of integrating information from multiple modalities into a unified map representation. More broadly, our method provides a framework that can be extended to additional modalities, such as odor, temperature, magnetic fields, infrared imagery, and point clouds. In our implementation, we selected audio, language, and vision because they closely mirror human perceptual capabilities. Viewed from a broader perspective, AVLMaps can be regarded as a case study in building a multi-modal spatial memory system, with inherent flexibility to incorporate more modalities. Extending the system to a new modality X involves three steps: (i) implementing the “X localization module” which includes mapping (embedding and storing the data into a representation) and retrieval (generating a heatmap based on the query data and the map) functions for the new modality, (ii) implementing the `get_major_map(X_input=None)` and `get_map(X_input=None)` to accept a new modality’s data as input and return 3D heatmaps with high and low decay rates as in Sec. 3.7, and (iii) implementing a context prompt as in Fig. 12 for the new modality, including an instruction involving the new modality, and the expected generated code. Although AVLMaps only focuses on audio, language, and vision, its simplicity, flexibility, and scalability open up promising directions for future research.

6 Conclusion

In this paper, we introduced multimodal spatial language maps, a unified mapping framework that is spatial, multimodal, reusable, and extensible. We first introduced a

visual language map representation, VLMaps, that enables robots to navigate to long-horizon spatial goals in a zero-shot manner. By defining the categories where the robot can and can not traverse, our map representation can adaptively generate obstacle maps for different embodiments, allowing for efficient path planning. Subsequently, we further extend our visual-language maps to a multimodal version, AVLMaps, which is a unified 3D spatial map representation for storing cross-modal information from audio, visual, and language cues. AVLMaps retain the spatial and reusable properties of VLMaps while enabling robots to reason over multimodal cues to disambiguate goals using large language models. Experiments in both simulated and real-world environments with different robotic embodiments demonstrate that our multimodal spatial language maps enable zero-shot spatial and multimodal goal navigation, significantly outperforming baselines in navigation success rate and landmark indexing accuracy, especially in scenarios with ambiguous goals. Moreover, extensive experimentation reveals that the performance of multimodal navigation and manipulation tasks scales with the capabilities of the underlying foundation models. At the end of the paper, we also present an in-depth discussion of the limitations and potential directions for future work, to inspire further research in multimodal spatial reasoning for robotics.

Notes

1. <https://github.com/karolpiczak/ESC-50>

References

- Ahn M, Brohan A, Brown N, Chebotar Y, Cortes O, David B, Finn C, Gopalakrishnan K, Hausman K, Herzog A, Ho D, Hsu J, Ibarz J, Ichter B, Irpan A, Jang E, Ruano RMJ, Jeffrey K, Jesmonth S, Joshi NJ, Julian RC, Kalashnikov D, Kuang Y, Lee KH, Levine S, Lu Y, Luu L, Parada C, Pastor P, Quiambao J, Rao K, Rettinghouse J, Reyes DM, Sermanet P, Sievers N, Tan C, Toshev A, Vanhoucke V, Xia F, Xiao T, Xu P, Xu S and Yan M (2022) Do as i can, not as i say: Grounding language in robotic affordances. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Anderson P, Chang A, Chaplot DS, Dosovitskiy A, Gupta S, Koltun V, Kosecka J, Malik J, Mottaghi R, Savva M and Zamir AR (2018a) On evaluation of embodied navigation agents. <https://arxiv.org/abs/1807.06757>.
- Anderson P, Shrivastava A, Truong J, Majumdar A, Parikh D, Batra D and Lee S (2021) Sim-to-real transfer for vision-and-language navigation. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Anderson P, Wu Q, Teney D, Bruce J, Johnson M, Sünderhauf N, Reid I, Gould S and Van Den Hengel A (2018b) Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Arandjelovic R, Gronat P, Torii A, Pajdla T and Sivic J (2016) Netvlad: Cnn architecture for weakly supervised place recognition. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Brohan A, Brown N, Carbajal J, Chebotar Y, Dabis J, Finn C, Gopalakrishnan K, Hausman K, Herzog A, Hsu J, Ibarz J,

- Ichter B, Irpan A, Jackson T, Jesmonth S, Joshi N, Julian R, Kalashnikov D, Kuang Y, Leal I, Lee KH, Levine S, Lu Y, Malla U, Manjunath D, Mordatch I, Nachum O, Parada C, Peralta J, Perez E, Pertsch K, Quiambao J, Rao K, Ryoo MS, Salazar G, Sanketi PR, Sayed K, Singh J, Sontakke S, Stone A, Tan C, Tran H, Vanhoucke V, Vega S, Vuong QH, Xia F, Xiao T, Xu P, Xu S, Yu T and Zitkovich B (2023) RT-1: Robotics Transformer for Real-World Control at Scale. In: *Proc. of Robotics: Science and Systems (RSS)*.
- Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I and Amodei D (2020) Language models are few-shot learners. In: *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*.
- Chang A, Dai A, Funkhouser T, Halber M, Niessner M, Savva M, Song S, Zeng A and Zhang Y (2017) Matterport3D: Learning from RGB-D data in indoor environments. In: *International Conference on 3D Vision (3DV)*.
- Chaplot DS, Gandhi DP, Gupta A and Salakhutdinov RR (2020) Object goal navigation using goal-oriented semantic exploration. In: *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*.
- Chen B, Xia F, Ichter B, Rao K, Gopalakrishnan K, Ryoo MS, Stone A and Kappler D (2023a) Open-vocabulary queryable scene representations for real world planning. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Chen B, Xia F, Ichter B, Rao K, Gopalakrishnan K, Ryoo MS, Stone A and Kappler D (2023b) Open-vocabulary queryable scene representations for real world planning. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Chen C, Jain U, Schissler C, Gari SVA, Al-Halah Z, Ithapu VK, Robinson P and Grauman K (2020) Soundspace: Audio-visual navigation in 3d environments. In: *Proc. of European Conference on Computer Vision (ECCV)*.
- Chen C, Majumder S, Al-Halah Z, Gao R, Ramakrishnan SK and Grauman K (2021a) Learning to set waypoints for audio-visual navigation. In: *Proc. of International Conference on Learning Representations (ICLR)*.
- Chen C, Schissler C, Garg S, Kobernik P, Clegg A, Calamia P, Batra D, Robinson P and Grauman K (2022) Soundspace 2.0: A simulation platform for visual-acoustic learning. *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*.
- Chen M, Tworek J, Jun H, Yuan Q, de Oliveira Pinto HP, Kaplan J, Edwards H, Burda Y, Joseph N, Brockman G, Ray A, Puri R, Krueger G, Petrov M, Khlaaf H, Sastry G, Mishkin P, Chan B, Gray S, Ryder N, Pavlov M, Power A, Kaiser L, Bavarian M, Winter C, Tillet P, Such FP, Cummings D, Plappert M, Chantzis F, Barnes E, Herbert-Voss A, Guss WH, Nichol A, Paino A, Tezak N, Tang J, Babuschkin I, Balaji S, Jain S, Saunders W, Hesse C, Carr AN, Leike J, Achiam J, Misra V, Morikawa E, Radford A, Knight M, Brundage M, Murati M, Mayer K, Welinder P, McGrew B, Amodei D, McCandlish S, Sutskever I and Zaremba W (2021b) Evaluating large language models trained on code. <https://arxiv.org/abs/2107.03374>.
- Chen W, Mees O, Kumar A and Levine S (2024a) Vision-language models provide promptable representations for reinforcement learning. <https://arxiv.org/abs/2402.02651>.
- Chen Z, Gebru ID, Richardt C, Kumar A, Laney W, Owens A and Richard A (2024b) Real acoustic fields: An audio-visual room acoustics dataset and benchmark. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Chun MM and Jiang Y (1998) Contextual cueing: Implicit learning and memory of visual context guides spatial attention. *Cognitive psychology*.
- DeTone D, Malisiewicz T and Rabinovich A (2018) Superpoint: Self-supervised interest point detection and description. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Doshi R, Walke H, Mees O, Dasari S and Levine S (2024) Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Elizalde B, Deshmukh S, Al Ismail M and Wang H (2023) Clap learning audio concepts from natural language supervision. In: *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Endres F, Hess J, Engelhard N, Sturm J, Cremers D and Burgard W (2012) An evaluation of the rgb-d slam system. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Engelmann F, Manhardt F, Niemeyer M, Tateno K, Pollefeys M and Tombari F (2024) OpenNeRF: Open Set 3D Neural Scene Segmentation with Pixel-Wise Features and Rendered Novel Views. In: *Proc. of International Conference on Learning Representations (ICLR)*.
- Fischler MA and Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*.
- Fried D, Hu R, Cirik V, Rohrbach A, Andreas J, Morency LP, Berg-Kirkpatrick T, Saenko K, Klein D and Darrell T (2018) Speaker-follower models for vision-and-language navigation.
- Gadre SY, Wortsman M, Ilharco G, Schmidt L and Song S (2023) Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Gan C, Gu Y, Zhou S, Schwartz J, Alter S, Traer J, Gutfreund D, Tenenbaum JB, McDermott JH and Torralba A (2022) Finding fallen objects via asynchronous audio-visual integration. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Gan C, Schwartz J, Alter S, Schrimpf M, Traer J, Freitas JD, Kubilius J, Bhandwaldar A, Haber N, Sano M, Kim K, Wang E, Mrowca D, Lingelbach M, Curtis A, Feigelis KT, Bear D, Gutfreund D, Cox D, DiCarlo JJ, McDermott JH, Tenenbaum JB and Yamins DLK (2020a) Threedworld: A platform for interactive multi-modal physical simulation. <https://arxiv.org/abs/2007.04954>.
- Gan C, Zhang Y, Wu J, Gong B and Tenenbaum JB (2020b) Look, listen, and act: Towards audio-visual embodied navigation. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Ganapathi A, Florence P, Varley J, Burns K, Goldberg K and Zeng A (2022) Implicit kinematic policies: Unifying joint and cartesian action spaces in end-to-end robot learning. In:

- Proc. of the IEEE International Conference on Robotics & Automation (ICRA).*
- Ghiasi G, Gu X, Cui Y and Lin TY (2022) Scaling open-vocabulary image segmentation with image-level labels. In: *Proc. of European Conference on Computer Vision (ECCV)*.
- Gu Q, Kuwajerwala A, Morin S, Jatavallabhula KM, Sen B, Agarwal A, Rivera C, Paul W, Ellis K, Chellappa R, Gan C, de Melo CM, Tenenbaum JB, Torralba A, Shkurti F and Paull L (2024) Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Gu X, Lin TY, Kuo W and Cui Y (2021) Open-vocabulary object detection via vision and language knowledge distillation. In: *Proc. of International Conference on Learning Representations (ICLR)*.
- Guhr PL, Tapaswi M, Chen S, Laptev I and Schmid C (2021) Airbert: In-domain pretraining for vision-and-language navigation. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Guzhov A, Raue F, Hees J and Dengel A (2022) Audioclip: Extending clip to image, text and audio. In: *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Hirose N, Glossop C, Sridhar A, Shah D, Mees O and Levine S (2024) Lelan: Learning a language-conditioned navigation policy from in-the-wild video. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Hong Y, Wang Z, Wu Q and Gould S (2022) Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Huang C, Mees O, Zeng A and Burgard W (2023a) Audio visual language maps for robot navigation. In: *Proc. of the International Symposium of Experimental Robotics (ISER)*.
- Huang C, Mees O, Zeng A and Burgard W (2023b) Visual language maps for robot navigation. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Huang C, Yan S and Burgard W (2025) Bye: Build your encoder with one sequence of exploration data for long-term dynamic scene understanding. *IEEE Robotics and Automation Letters*.
- Huang W, Abbeel P, Pathak D and Mordatch I (2022a) Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In: *Proc. of the International Conference on Machine Learning (ICML)*.
- Huang W, Xia F, Xiao T, Chan H, Liang J, Florence P, Zeng A, Tompson J, Mordatch I, Chebotar Y, Sermanet P, Jackson T, Brown N, Luu L, Levine S, Hausman K and brian ichter (2022b) Inner monologue: Embodied reasoning through planning with language models. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Jatavallabhula KM, Kuwajerwala A, Gu Q, Omama M, Iyer G, Saryazdi S, Chen T, Maalouf A, Li S, Keetha NV, Tewari A, Tenenbaum J, de Melo C, Krishna M, Paull L, Shkurti F and Torralba A (2023) ConceptFusion: Open-set multimodal 3D mapping. In: *Proc. of Robotics: Science and Systems (RSS)*.
- Kamath A, Singh M, LeCun Y, Synnaeve G, Misra I and Carion N (2021) Mdetr-modulated detection for end-to-end multimodal understanding. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Kerr J, Kim CM, Goldberg K, Kanazawa A and Tancik M (2023) Lerf: Language embedded radiance fields. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Kim CM, Wu M, Kerr J, Goldberg K, Tancik M and Kanazawa A (2024) Garfield: Group anything with radiance fields. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kim MJ, Pertsch K, Karamcheti S, Xiao T, Balakrishna A, Nair S, Rafailov R, Foster EP, Sanketi PR, Vuong Q, Kollar T, Burchfiel B, Tedrake R, Sadigh D, Levine S, Liang P and Finn C (2025) Openvla: An open-source vision-language-action model. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg AC, Lo WY, Dollar P and Girshick R (2023) Segment anything. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Kolve E, Mottaghi R, Han W, VanderBilt E, Weihs L, Herrasti A, Deitke M, Ehsani K, Gordon D, Zhu Y, Kembhavi A, Gupta A and Farhadi A (2017) Ai2-thor: An interactive 3d environment for visual ai. <https://arxiv.org/abs/1712.05474>.
- Körding KP, Beierholm U, Ma WJ, Quartz S, Tenenbaum JB and Shams L (2007) Causal inference in multisensory perception. *PLoS one*.
- Krantz J, Gokaslan A, Batra D, Lee S and Maksymets O (2021) Waypoint models for instruction-guided navigation in continuous environments. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Krantz J, Wijmans E, Majumdar A, Batra D and Lee S (2020) Beyond the nav-graph: Vision-and-language navigation in continuous environments. In: *Proc. of European Conference on Computer Vision (ECCV)*.
- Li B, Weinberger KQ, Belongie S, Koltun V and Ranftl R (2021) Language-driven semantic segmentation. In: *Proc. of International Conference on Learning Representations (ICLR)*.
- Li B, Weinberger KQ, Belongie S, Koltun V and Ranftl R (2022) Language-driven semantic segmentation. In: *Proc. of International Conference on Learning Representations (ICLR)*.
- Liang F, Wu B, Dai X, Li K, Zhao Y, Zhang H, Zhang P, Vajda P and Marculescu D (2023a) Open-vocabulary semantic segmentation with mask-adapted clip. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Liang J, Huang W, Xia F, Xu P, Hausman K, Ichter B, Florence P and Zeng A (2023b) Code as policies: Language model programs for embodied control. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- MacMahon M, Stankiewicz B and Kuipers B (2006) Walk the talk: Connecting language, knowledge, and action in route instructions. *Def.*
- McCormac J, Clark R, Bloesch M, Davison A and Leutenegger S (2018) Fusion++: Volumetric object-level slam. In: *International Conference on 3D Vision (3DV)*.
- McCormac J, Handa A, Davison A and Leutenegger S (2017) Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- McNamara TP, Hardy JK and Hirtle SC (1989) Subjective hierarchies in spatial memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*.

- Mees O, Borja-Diaz J and Burgard W (2023) Grounding language with visual affordances over unstructured data. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Mees O, Hermann L and Burgard W (2022a) What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*.
- Mees O, Hermann L, Rosete-Beas E and Burgard W (2022b) Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*.
- Newman EL, Caplan JB, Kirschen MP, Korolev IO, Sekuler R and Kahana MJ (2007) Learning your way around town: How virtual taxicab drivers learn to use both layout and landmark information. *Cognition*.
- Octo Model Team, Ghosh D, Walke H, Pertsch K, Black K, Mees O, Dasari S, Hejna J, Xu C, Luo J, Kreiman T, Tan Y, Chen LY, Sanketi P, Vuong Q, Xiao T, Sadigh D, Finn C and Levine S (2024) Octo: An open-source generalist robot policy. In: *Proc. of Robotics: Science and Systems (RSS)*.
- O'Neill A, Rehman A, Maddukuri A, Gupta A, Padalkar A, Lee A, Pooley A, Gupta A, Mandlekar A, Jain A, Tung A, Bewley A, Herzog A, Irpan A, Khazatsky A, Rai A, Gupta A, Wang A, Singh A, Garg A, Kembhavi A, Xie A, Brohan A, Raffin A, Sharma A, Yavary A, Jain A, Balakrishna A, Wahid A, Burgess-Limerick B, Kim B, Schölkopf B, Wulfe B, Ichter B, Lu C, Xu C, Le C, Finn C, Wang C, Xu C, Chi C, Huang C, Chan C, Agia C, Pan C, Fu C, Devin C, Xu D, Morton D, Driess D, Chen D, Pathak D, Shah D, Büchler D, Jayaraman D, Kalashnikov D, Sadigh D, Johns E, Foster E, Liu F, Ceola F, Xia F, Zhao F, Stulp F, Zhou G, Sukhatme GS, Salhotra G, Yan G, Feng G, Schiavi G, Berseth G, Kahn G, Wang G, Su H, Fang HS, Shi H, Bao H, Ben Amor H, Christensen HI, Furuta H, Walke H, Fang H, Ha H, Mordatch I, Radosavovic I, Leal I, Liang J, Abou-Chakra J, Kim J, Drake J, Peters J, Schneider J, Hsu J, Bohg J, Bingham J, Wu J, Gao J, Hu J, Wu J, Wu J, Sun J, Luo J, Gu J, Tan J, Oh J, Wu J, Lu J, Yang J, Malik J, Silvério J, Hejna J, Booher J, Tompson J, Yang J, Salvador J, Lim JJ, Han J, Wang K, Rao K, Pertsch K, Hausman K, Go K, Gopalakrishnan K, Goldberg K, Byrne K, Oslund K, Kawaharazuka K, Black K, Lin K, Zhang K, Ehsani K, Lekkala K, Ellis K, Rana K, Srinivasan K, Fang K, Singh KP, Zeng KH, Hatch K, Hsu K, Itti L, Chen LY, Pinto L, Fei Fei L, Tan L, Fan LJ, Ott L, Lee L, Weihs L, Chen M, Lepert M, Memmel M, Tomizuka M, Itkina M, Castro MG, Spero M, Du M, Ahn M, Yip MC, Zhang M, Ding M, Heo M, Sriram MK, Sharma M, Kim MJ, Kanazawa N, Hansen N, Heess N, Joshi NJ, Suenderhauf N, Liu N, Di Palo N, Shafullah NMM, Mees O, Kroemer O, Bastani O, Sanketi PR, Miller PT, Yin P, Wohlhart P, Xu P, Fagan PD, Mitrano P, Sermanet P, Abbeel P, Sundaresan P, Chen Q, Vuong Q, Rafailov R, Tian R, Doshi R, Martín-Martín R, Baijal R, Scalise R, Hendrix R, Lin R, Qian R, Zhang R, Mendonca R, Shah R, Hoque R, Julian R, Bustamante S, Kirmani S, Levine S, Lin S, Moore S, Bahl S, Dass S, Sonawani S, Song S, Xu S, Haldar S, Karamcheti S, Adebola S, Guist S, Nasiriany S, Schaal S, Welker S, Tian S, Ramamoorthy S, Dasari S, Belkhale S, Park S, Nair S, Mirchandani S, Osa T, Gupta T, Harada T, Matsushima T, Xiao T, Kollar T, Yu T, Ding T, Davchev T, Zhao TZ, Armstrong T, Darrell T, Chung T, Jain V, Vanhoucke V, Zhan W, Zhou W, Burgard W, Chen X, Wang X, Zhu X, Geng X, Liu X, Liangwei X, Li X, Lu Y, Ma YJ, Kim Y, Chebotar Y, Zhou Y, Zhu Y, Wu Y, Xu Y, Wang Y, Bisk Y, Cho Y, Lee Y, Cui Y, Cao Y, Wu YH, Tang Y, Zhu Y, Zhang Y, Jiang Y, Li Y, Li Y, Iwasawa Y, Matsuo Y, Ma Z, Xu Z, Cui ZJ, Zhang Z and Lin Z (2024) Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration0. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Paul S, Roy-Chowdhury A and Cherian A (2022) Avlen: Audio-visual-language embodied navigation in 3d environments. In: *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*.
- Peng S, Genova K, Jiang CM, Tagliasacchi A, Pollefeys M and Funkhouser T (2023) Openscene: 3d scene understanding with open vocabularies. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Picczak KJ (2015) Esc: Dataset for environmental sound classification. In: *Proceedings of the 23rd ACM international conference on Multimedia*.
- Qin M, Li W, Zhou J, Wang H and Pfister H (2024) Langsplat: 3d language gaussian splatting. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Quigley M (2009) Ros: an open-source robot operating system. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G and Sutskever I (2021) Learning transferable visual models from natural language supervision. In: *Proc. of the International Conference on Machine Learning (ICML)*.
- Rosete-Beas E, Mees O, Kalweit G, Boedecker J and Burgard W (2022) Latent plans for task agnostic offline reinforcement learning. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Runz M, Buffier M and Agapito L (2018) Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In: *Proc. of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*.
- Salas-Moreno RF, Newcombe RA, Strasdat H, Kelly PH and Davison AJ (2013) Slam++: Simultaneous localisation and mapping at the level of objects. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Sarlin PE, Cadena C, Siegwart R and Dymczyk M (2019) From coarse to fine: Robust hierarchical localization at large scale. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Sarlin PE, DeTone D, Malisiewicz T and Rabinovich A (2020) SuperGlue: Learning feature matching with graph neural networks. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Savva M, Kadian A, Maksymets O, Zhao Y, Wijmans E, Jain B, Straub J, Liu J, Koltun V, Malik J, Parikh D and Batra D (2019) Habitat: A Platform for Embodied AI Research. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D and Batra D (2020) Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*.

- Shafiuallah NMM, Paxton C, Pinto L, Chintala S and Szlam A (2023) CLIP-Fields: Weakly Supervised Semantic Fields for Robotic Memory. In: *Proc. of Robotics: Science and Systems (RSS)*.
- Shah D, Osiński B, Ichter B and Levine S (2023) Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Shridhar M, Manuelli L and Fox D (2022) Cliport: What and where pathways for robotic manipulation. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Szot A, Clegg A, Undersander E, Wijmans E, Zhao Y, Turner J, Maestre N, Mukadam M, Chaplot D, Maksymets O, Gokaslan A, Vondrus V, Dharur S, Meier F, Galuba W, Chang A, Kira Z, Koltun V, Malik J, Savva M and Batra D (2021) Habitat 2.0: Training home assistants to rearrange their habitat. In: *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*.
- Tellex S, Kollar T, Dickerson S, Walter M, Banerjee A, Teller S and Roy N (2011) Understanding natural language commands for robotic navigation and mobile manipulation. In: *Proc. of the National Conference on Artificial Intelligence (AAAI)*.
- Thrun S, Burgard W and Fox D (1998) A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*.
- Wang ML, Sawata R, Clarke S, Gao R, Wu S and Wu J (2024) Hearing anything anywhere. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Werby A, Huang C, Büchner M, Valada A and Burgard W (2024) Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation. In: *Proc. of Robotics: Science and Systems (RSS)*.
- Wu HH, Seetharaman P, Kumar K and Bello JP (2022) Wav2clip: Learning robust audio representations from clip. In: *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Wu J, Sun X, Zeng A, Song S, Rusinkiewicz S and Funkhouser T (2021) Spatial intention maps for multi-agent mobile manipulation. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Xu B, Li W, Tzoumanikas D, Bloesch M, Davison A and Leutenegger S (2019) Mid-fusion: Octree-based object-level multi-instance dynamic slam. In: *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- Yan Z, Li S, Wang Z, Wu L, Wang H, Zhu J, Chen L and Liu J (2025) Dynamic open-vocabulary 3d scene graphs for long-term language-guided mobile manipulation. *IEEE Robotics and Automation Letters*.
- Younes A, Honerkamp D, Welschehold T and Valada A (2023) Catch me if you hear me: Audio-visual navigation in complex unmapped environments with moving sounds. *IEEE Robotics and Automation Letters*.
- Zakka K, Zeng A, Florence P, Tompson J, Bohg J and Dwibedi D (2022) Xirl: Cross-embodiment inverse reinforcement learning. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Zawalski M, Chen W, Pertsch K, Mees O, Finn C and Levine S (2024) Robotic control via embodied chain-of-thought reasoning. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Zeng A (2019) *Learning visual affordances for robotic manipulation*. PhD Thesis, Princeton University.
- Zeng A, Attarian M, Ichter B, Choromanski KM, Wong A, Welker S, Tombari F, Purohit A, Ryoo MS, Sindhwan V, Lee J, Vanhoucke V and Florence P (2023) Socratic models: Composing zero-shot multimodal reasoning with language. In: *Proc. of International Conference on Learning Representations (ICLR)*.
- Zhou Z, Atreya P, Lee A, Walke H, Mees O and Levine S (2024) Autonomous improvement of instruction following skills via foundation models. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Zitkovich B, Yu T, Xu S, Xu P, Xiao T, Xia F, Wu J, Wohlhart P, Welker S, Wahid A, Vuong Q, Vanhoucke V, Tran H, Soricut R, Singh A, Singh J, Sermanet P, Sanketi PR, Salazar G, Ryoo MS, Reymann K, Rao K, Pertsch K, Mordatch I, Michalewski H, Lu Y, Levine S, Lee L, Lee TWE, Leal I, Kuang Y, Kalashnikov D, Julian R, Joshi NJ, Irpan A, Ichter B, Hsu J, Herzog A, Hausman K, Gopalakrishnan K, Fu C, Florence P, Finn C, Dubey KA, Driess D, Ding T, Choromanski KM, Chen X, Chebotar Y, Carbalaj J, Brown N, Brohan A, Arenas MG and Han K (2023) Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: *Proc. of the Conference on Robot Learning (CoRL)*.
- Zuo X, Samangouei P, Zhou Y, Di Y and Li M (2024) Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. *International Journal of Computer Vision*.