Audio Visual Language Maps for Robot Navigation

Chenguang Huang
University of Freiburg, Germany

Oier Mees
University of Freiburg, Germany

Andy Zeng
Google Research, USA

Wolfram Burgard
University of Technology Nuremberg, Germany

Abstract

While interacting in the world is a multi-sensory experience, many robots continue to predominantly rely on visual perception to map and navigate in their environments. We propose AVLMaps, a 3D spatial map representation that stores cross-modal information from audio, visual, and language cues. AVLMaps fuse features from pre-trained multimodal foundation models into a centralized voxel grid. This enables robots to index goals in the map based on multimodal queries, such as textual descriptions, images, or audio snippets of landmarks. AVLMaps allow for zero-shot multimodal goal navigation and perform better than alternatives in ambiguous scenarios. These capabilities extend to mobile robots in the real world. Videos and code are available at https://avlmaps.github.io[1].

1. Introduction

Humans are adept at using multiple senses to navigate the world [9], but robots mostly rely on visual perception. To address this limitation, we propose AVLMaps, a 3D spatial map that integrates audio, visual, and language information. AVLMaps can be built using pre-trained multimodal models [6, 7, 10] and can index landmarks based on open-vocabulary queries. The system enables language-driven navigation and can disambiguate multiple goal locations using multimodal information, outperforming baseline alternatives by up to 50% in recall. A key aspect of AVLMaps is that they extend prior multimodal mapping representations [3, 8, 19] to 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. Additionally, when paired with large language models (LLMs) we show that AVLMaps enable 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.

2. Method

We aim to create an audio-visual-language map that can directly localize objects, areas, audio and visual goals using natural language or target images. We propose AVLMaps by combining 3D reconstruction libraries with pre-trained visual-language and audio-language models. We also suggest a cross-modal reasoning approach to disambiguate locations referring to targets from different modalities. Fig. 2 shows the system pipeline.

[1]The link to the arXiv version of our full paper is https://arxiv.org/pdf/2303.07522.pdf

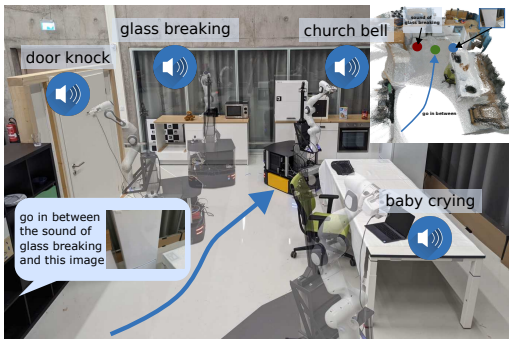

Figure 1. **AVLMaps** provide an open-vocabulary 3D map representation for storing cross-modal information from audio, visual, and language cues. When combined with large language models, AVLMaps consumes multimodal prompts from audio, vision and language to solve zero-shot spatial goal navigation by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

2.1. Building an Audio Visual Language Map

Given an RGB-D video stream with an audio track and odometry information, we utilize four modules to build a multimodal features database as AVLMaps.

Visual Localization Module. The module localizes a query image in a map using a hierarchical scheme [16]. It computes global [1] and local descriptors [4, 17] for all images in the RGB stream for localizing query images during inference. More details are shown in Appendix, Sec. A.

Object Localization Module. The Object Localization Module uses an open-vocabulary segmentation method (*e.g.* LSeg [10] or OpenSeg [6]) to generate pixel-level features from the RGB image and associates them with back-projected depth pixels in 3D reconstruction. During inference, it encodes a target text query [15], computes the cosine similarity scores between all point-wise and language features, and selects the top-scoring points in the map as the indexing result.

Area Localization Module. We propose building a sparse topological CLIP features map [20] to recognize coarse visual concepts like "kitchen area". During inference, given a language concept, we compute the language features with the CLIP language encoder [15] and image-to-language cosine similarity scores to predict the location with confidence values.

Audio Localization Module. The audio localization module partitions the audio clip from the audio stream input into several segments using silence detection and computes audio-lingual features for each segment with AudioCLIP [7]. During inference, given a language description, it computes matching scores

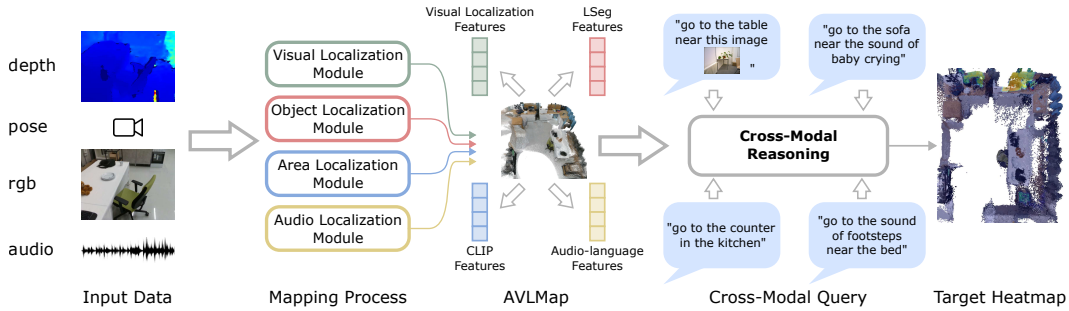

Figure 2. System overview. AVLMaps are constructed from RGB-D, audio, and odometry inputs, converting raw data into visual localization features, visual-language features, and audio-language features. During inference time, each module's output is unified with cross-modal reasoning, allowing users to query spatial location with multimodal information.

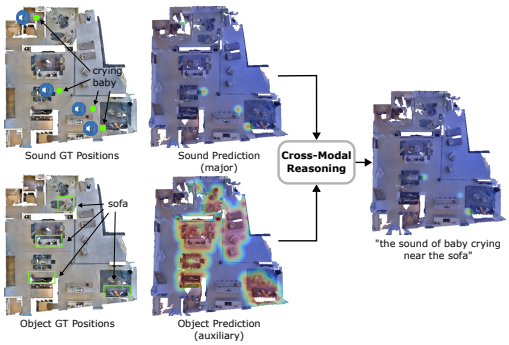

Figure 3. The key idea of cross-modal reasoning is converting the prediction from different modalities into heatmaps, and then fusing them with element-wise multiplication, effectively using complementary multimodal information to resolve ambiguous prompts.

between the language and all audio segments. The odometry associated with the top-scoring segment is the predicted location.

2.2. Cross-Modality Reasoning

A key advantage of our method is its capability to disambiguate goals with additional information, even from different modalities. Each localization module returns a heatmap with probabilities for each voxel position in the map based on the distance to the target location. Cross-modal reasoning is performed by computing the element-wise multiplication of all heatmaps for several queries referring to different modalities as in Fig. 3, and the predicted location is extracted from the highest probability position on the target heatmap. The detailed formulation of the cross-modality reasoning is in the Appendix, Sec. B.

2.3. Multimodal Goal Navigation from Language

We present a multimodal goal navigation approach that utilizes large language models (LLMs) to interpret natural language descriptions of targets from different modalities and plan paths to them. Our approach unifies various navigation tasks by using LLMs to synthesize API calls and executable python code [8,11,12]. To generate heatmaps indicating target locations, we implement two interfaces with different decay rates. Additionally, we support image prompts by adding image paths to language queries. Prompt examples is listed in Appendix, Sec. C.

3. Experiments

3.1. Multimodal Ambiguous Goal Navigation

We conducted experiments to test our method with ambiguous goal navigation tasks, requiring reasoning across different modalities to localize the targets. We compared our method to two single-modality baselines (VLMaps [8] and AudioCLIP [7]) and a multimodal baseline using VLMaps for object localization and wav2clip [22] for audio localization. The results in Tab. 1 showed that AVLMaps had 24.2% and 2.1% higher success rate for ambiguous sound and object goals, respectively, compared to the multimodal baseline.

Tasks	No. Subgoals in a Row		Sound	Object
	1	2	Goals	Goals
VLMaps [8]	-	-	-	27.1
AudioCLIP [7]	-	-	16.9	-
VLMaps + wav2clip	22.0	12.7	22.0	53.4
VLMaps + AudioCLIP (Ours)	**46.2**	**28.6**	**46.2**	**55.5**

Table 1: 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.

4. Conclusion

In this paper, AVLMaps were introduced as a 3D spatial map representation that can store cross-modal information from audio, visual, and language cues. AVLMaps can be combined with large language models to enable zero-shot spatial goal navigation by effectively leveraging complementary information sources to disambiguate goals. Experiments showed that AVLMaps improved target indexing accuracy compared to baselines, especially in scenarios with ambiguous goals.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[3] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. *arXiv preprint arXiv:2209.09874*, 2022.

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[5] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[6] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021.

[7] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 976–980. IEEE, 2022.

[8] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[9] Konrad P Körding, Ulrik Beierholm, Wei Ji Ma, Steven Quartz, Joshua B Tenenbaum, and Ladan Shams. Causal inference in multisensory perception. *PLoS one*, 2(9):e943, 2007.

[10] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022.

[11] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

[12] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[13] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015.

[14] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[16] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

[17] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[18] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[19] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv: Arxiv-2210.05663*, 2022.

[20] Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. *arXiv preprint arXiv:2207.04429*, 2022.

[21] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[22] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4563–4567. IEEE, 2022.

APPENDIX

A. Visual Localization Module

The Visual Localization Module localizes a query image in a map using a hierarchical scheme [16]. It computes global NetVLAD descriptors [1] and local SuperPoint descriptors [4] for all images in the RGB stream and stores them with corresponding depth and odometry. During inference, it computes descriptors for the query image, finds a reference image using nearest neighbor search with global features, establishes key point correspondences between two images using SuperGLUE [17], backprojects reference key points into 3D space to obtain 3D-2D correspondences for query key points, and estimates query camera pose relative to reference camera using Perspective-n-Point method [5].

B. Cross-Modality Reasoning Formulation

A key advantage of our method is its capability to disambiguate goals with additional information, even from different modalities. Given a specific query, each module introduced in the last section 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 the target position. $\mathbf{p} = (x,y,z)^T, \{x,y,z \in \mathbb{Z} | 1 \leq x \leq \bar{H}, 1 \leq y \leq \bar{W}, 1 \leq z \leq \bar{Z}\}$ is a voxel position in the map $\mathcal{H}$.

Visual Localization Heatmap. 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:

$$\mathcal{H}_v(\mathbf{p}) = \max(1.0 - \epsilon \cdot dist_{xy}(\mathbf{p}, \mathbf{p}_v), 0) \tag{1}$$

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

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 Heatmap. The object localization results are a list of points, denoted as $\{\mathbf{p}_{oi} = (x_{oi}, y_{oi}, z_{oi}) | i = 1,2,...,N\}$ 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:

$$d_{min}(\mathbf{p}) = \min\{dist(\mathbf{p}, \mathbf{p}_{oi}) | i = 1,2,...,N\} \tag{3}$$

$$\mathcal{H}_o(\mathbf{p}) = \max(1.0 - \epsilon \cdot d_{min}(\mathbf{p}), 0) \tag{4}$$

where $d_{min}(\mathbf{p})$ denotes the minimal distance between $\mathbf{p}$ and all object points $\{\mathbf{p}_{oi} | i = 1,2,...,N\}$, $dist(\mathbf{p}, \mathbf{q})$ denotes the Euclidean distance between $\mathbf{p}$ and $\mathbf{q}$.

Area Localization Heatmap. The area localization results are a list of position-confidence pairs, denoted as $\{(\mathbf{p}_{ai}, s_{ai}) | i = 1, 2, ..., M\}$ 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:

$$\mathcal{H}_a(\mathbf{p}) = \max(\max\{s_{ai} - \epsilon \cdot dist_{xy}(\mathbf{p}, \mathbf{p}_{ai}) | i = 1,2,...,M\}, 0) \tag{5}$$

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 Heatmap. The audio localization results are similar to those of the area localization module. The position-score pairs are denoted as $\{(\mathbf{p}_{si}, s_{si}) | i = 1,2,...,K\}$ where K is the total number of sound segments in the input video stream. The heatmap $\mathcal{H}_s$ is defined as:

$$\mathcal{H}_s(\mathbf{p}) = \max(\max\{s_{si} - \epsilon \cdot dist_{xy}(\mathbf{p}, \mathbf{p}_{si}) | i = 1,2,...,K\}, 0) \tag{6}$$

Cross-Modal Reasoning. The main idea of cross-modal reasoning is shown in Fig. 3. 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:

$$\mathcal{H}_{target} = \mathcal{H}_1 \odot \mathcal{H}_2 \odot ... \odot \mathcal{H}_L \tag{7}$$

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 the query "the chair near the sound of crying", the heatmap for "the chair" is the primary heatmap, while the heatmap for "the sound of crying" is the auxiliary. We set the decay rate for the primary heatmap higher (*e.g.* 0.1 in this work) 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 affect area to narrow down major targets is desirable.

C. Multimodal Goal Navigation from Language Prompt Examples

In the setting of multimodal goal navigation from language, the agent is given language descriptions of targets from different modalities (*e.g.* sound, image, and object) 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). Specifically, 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 [8, 11, 12]. For heatmap generation, we implement interfaces ***get_major_map****(obj=None, sound=None, img=None)* and ***get_map****(obj=None, sound=None, img=None)*. They take object name, sound name, or image as input and output heatmaps indicating the locations of targets. The ***get_major_map*** generates heatmaps with higher decay rate while ***get_map*** with lower decay rate. To support image prompt, we add image path in the language query like "the image /path/to/image.png" and use LLMs to call the image loading API. Some examples of prompts and queries are shown below (prompt in gray, input task commands in green, and generated outputs are highlighted):

```
# 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 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)
```

D. Supplement Experiments

D.1. Simulation Setup

Exerimental setup. We use the Habitat simulator [18, 21] with the Matterport3D dataset [2] for the evaluation of multimodal navigation tasks. The dataset contains a large set of apartment-level realistic indoor scenes that help evaluate the generalization capabilities of navigating agents. For mapping purposes, we manually collect RGB-D video streams in the simulator across 10 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 [13], 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 1 meter from the ground truth position, the subgoal is considered successfully finished.

Tasks collection. In multimodal goal navigation tasks in Sec. D.2, 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. D.3, 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) 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. 3.1, we randomly sample starting pose in 10 scenes and treat the visual-object and object-sound cross-modal goals in Sec. D.3 as navigation goals.

D.2. Multimodal Goal Navigation

Sound goal navigation. We first test AVLMaps in sound goal navigation tasks. We collect 200 sequences of sound goals in 10 different scenes. In each sequence, there are 4 sound categories that require the robot to reach. The results are shown in Tab. 2. We generate AudioCLIP [7] 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 link[2]. 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

[2] https://github.com/karolpiczak/ESC-50

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

Table 2: The success rate (%) of sound goal navigation with AVLMaps.

an image and two object categories in the language in one sequence of tasks and asked to navigate to the image goal and two object goals in sequence. In 200 sequences of tasks in 10 scenes, the success rate is reported in Tab. 3. The results show that our method enables the agent to navigate to goals from different modalities.

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

Table 3: 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.

D.3. 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 10 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-1 recall with different distance tolerance is reported in Tab. 4. We can notice that VLMaps [8] struggles to find the goal in the correct region because VLMaps 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.

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 1 location, introducing ambiguity to the localization scenario. The recall is reported in Tab. 5. 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 (%)				Avg. min. distance (m)
	<0.5m	<1m	<1.5m	<2m	
baseline (VLMaps)	5.56	7.78	13.33	17.78	8.22
+ ConceptFusion	12.22	13.33	16.67	21.11	7.60
+ CLIP sparse map (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 4: The recall (%) of area-object cross-modal indexing.

Method	Recall@1 (%)				Avg. 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
VLMaps + wav2clip	24.37	30.25	33.61	38.66	6.27
VLMaps + AudioCLIP (Ours)	**53.78**	**65.55**	**67.23**	**70.59**	**2.74**

Table 5: 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. 6.

Method	Recall@1 (%)				Avg. min. distance (m)
	<0.5m	<1m	<1.5m	<2m	
VLMaps w/o vis loc	7.55	9.43	11.32	11.94	11.22
VLMaps w/ vis loc (Ours)	**62.26**	**66.67**	**70.44**	**72.32**	**3.11**

Table 6: The recall (%) of visual-object cross-modal indexing.

D.4. Real World Experiment

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.

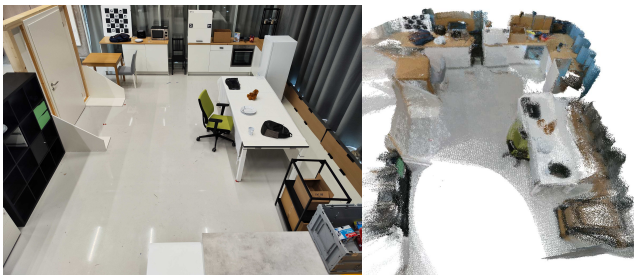

Figure 4. Real-world experiments are conducted in a room with multiple ambiguous goals such as tables, chairs, backpacks, and paper boxes (left). We leverage dense SLAM techniques to build a 3D reconstruction of the scene from RGB-D camera data into which we anchor features from multiple foundation models (right).

Environment setup. We choose a room with multiple ambiguous goals such as tables, chairs, paper boxes, counters, and backpacks which are shown in Fig. 4. 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 in Fig. 5. 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. 2.3) and executes the generated python code for goal indexing and planning. We use the ROS navigation package [14] for global and local planning. To avoid including noise proceeding from the own robots operation, we preprocess sound inputs with background noise substraction.

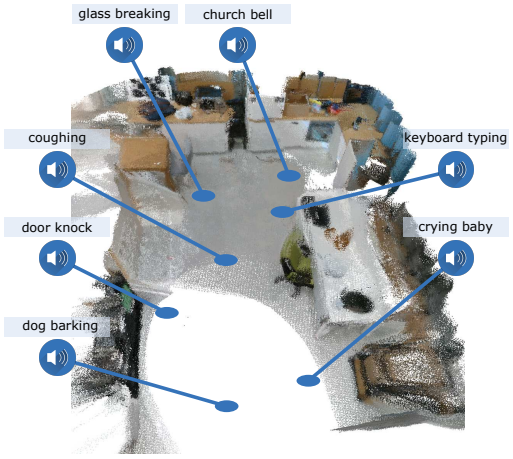

Figure 5. We artificially insert sounds with different semantics at locations shown in the image. Different sounds are played when the robot moves to these locations during mapping. Sounds are sampled from the ESC-50 dataset.

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 10 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, 6 out of 10 tasks were successfully finished. We show in Fig. 6 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. 6 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. 6, 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.

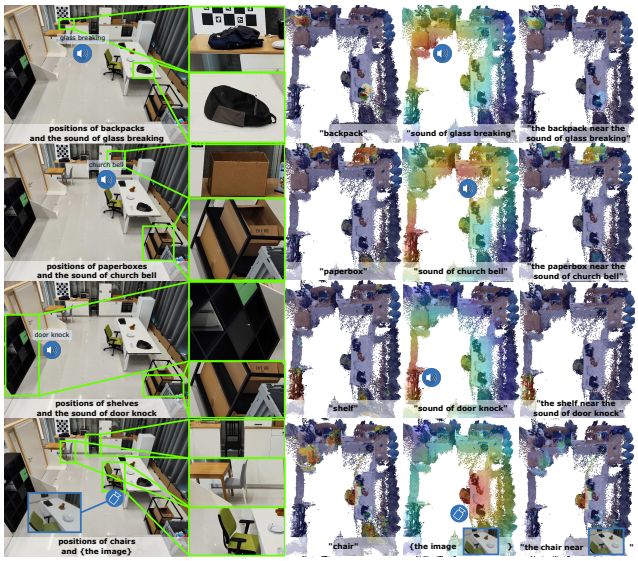

Figure 6. Visualization of example heatmaps in AVLMaps for multimodal goal reasoning for ambiguous object goals. The first column shows the positions of ambiguous objects (green bounding boxes) and the location of a sound (the icon of a speaker) or an image (the icon of a camera). The second column shows the zoom-in 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. The final 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 JET color scheme (red means the highest score and blue means the lowest score).