Chat2Map: Efficient Scene Mapping from Multi-Ego Conversations

Sagnik Majumder[1,2*] Hao Jiang[2] Pierre Moulon[2] Ethan Henderson[2]
Paul Calamia[2] Kristen Grauman[1,3] Vamsi Krishna Ithapu[2]
[1]UT Austin [2]Reality Labs Research, Meta [3]FAIR

Abstract

Can conversational videos captured from multiple egocentric viewpoints reveal the map of a scene in a cost-efficient way? We seek to answer this question by proposing a new problem: efficiently building the map of a previously unseen 3D environment by exploiting shared information in the egocentric audio-visual observations of participants in a natural conversation. Our hypothesis is that as multiple people ("egos") move in a scene and talk among themselves, they receive rich audio-visual cues that can help uncover the unseen areas of the scene. Given the high cost of continuously processing egocentric visual streams, we further explore how to actively coordinate the sampling of visual information, so as to minimize redundancy and reduce power use. To that end, we present an audio-visual deep reinforcement learning approach that works with our shared scene mapper to selectively turn on the camera to efficiently chart out the space. We evaluate the approach using a state-of-the-art audio-visual simulator for 3D scenes as well as real-world video. Our model outperforms previous state-of-the-art mapping methods, and achieves an excellent cost-accuracy tradeoff. Project: http://vision.cs.utexas.edu/projects/chat2map.

1. Introduction

The spatial layout of the environment around us is fundamental to understanding our physical context. By representing the walls, furniture, and other major structures in a space, *scene maps* ground activity and objects in a persistent frame of reference, facilitating high-level reasoning for many downstream applications in augmented reality (AR) and robotics. For example, episodic memory [18, 30] aims to relocalize lost objects observed in first-person video (*where are my keys?*); floorplan estimation [10, 45, 53] aims to chart out the area and shapes of complex buildings; navigating agents try to discover routes in unfamiliar spaces [4, 11, 60].

While traditional computer vision approaches for mapping (e.g., visual SLAM) are highly effective when extensive exposure to the environment is possible, in many real-world scenarios only a fraction of the space is observed by the camera. Recent work shows the promise of sensing 3D spaces with both sight and sound [8, 14, 26, 28, 59]: listening to echoes bounce around the room can reveal the depth and shape of surrounding surfaces, and even help extrapolate a floorplan beyond the camera's field of view or behind occluded objects [59].

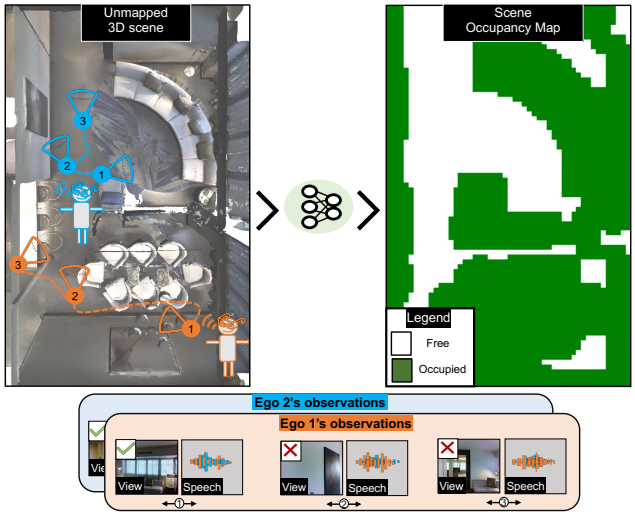


Figure 1. Given egocentric audio-visual observations from multiple people wearing AR glasses and moving and conversing (left), we aim to accurately map the scene (right). To mitigate cost, our model receives audio continuously but learns to selectively employ the ego cameras only when the visual input is expected to be informative.

While we are inspired by these advances, they also have certain limitations. Often systems will emit sounds (e.g., a frequency sweep) into the environment to ping for spatial information [1, 14, 15, 24, 28, 44, 59, 69], which is intrusive if done around people. Furthermore, existing audio-visual models assume that the camera is *always on* grabbing new frames, which is wasteful if not intractable, particularly on lightweight, low-power computing devices in AR settings.

We introduce Chat2Map, a new scene mapping task

*Work done during an internship at Reality Labs Research, Meta

aimed at eliminating these challenges. In the proposed setting, multiple people converse as they move casually through the scene while wearing AR glasses equipped with an egocentric camera, microphones, and potentially other sensors (e.g., for odometry).[1] Given their egocentric audio-visual data streams, the goal is to infer the ground-plane occupancy map for the larger environment around them. See Figure 1.

We observe that audio-visual data from the egos' interactions will naturally reflect scene structure. First, as they walk and talk, their movements reveal spaces like corridors, doorways, and large rooms, in both modalities. Second, the speech captured by the device-wearer's cameras and microphones can be localized to different speakers, which, compared to active sound emission, is non-intrusive.

To realize this vision, we develop a novel approach to efficient scene mapping from multi-ego conversations. Our approach has two key elements: a shared scene mapper and a visual sampling policy. For the former, we devise a transformer-based mapper that incorporates the multiple data streams to infer a map beyond the directly observed areas, and, most importantly, that enables communication among the egos about their observations and states in the 3D space to improve mapping accuracy. For the latter, our idea is to relax the common assumption of an "always-on" camera, and instead *actively select* when to sample visual frames from any one of the ego cameras. Intuitively, certain regions where the egos move will be more or less important for mapping (e.g., corners of the room, doors). We train a sampling policy with deep reinforcement learning that activates the visual feed only when it is anticipated to complement the continuous audio feed. This is a cost-conscious approach, mindful that switching on a camera is much more power consuming than sensing audio with microphones [2].

We demonstrate our approach using a state-of-the-art audio-visual simulator for 3D scenes as well as some real-world video input. We can successfully map an unfamiliar environment given only partial visibility via multiple conversing people moving about the scene. Compared to sampling all visual frames, our model reduces the visual processing by 87.5% while the mapping accuracy declines marginally ($\sim 9\%$).

2. Related Work

Visual scene mapping. Past works tackle scene mapping using 3D Manhattan layouts [20, 73, 80, 85, 86], detailed floorplans [10, 45, 53, 71, 78], occupancy [23, 39, 52, 61, 67, 68], and semantic maps [51]. Manhattan layouts include structured outputs like scene boundaries [73, 85, 86], corners [85, 86], and floor/ceilings [80, 86], but do not generalize to unseen environment regions. Floorplan estimation methods use dense scans of 3D scenes to predict geometric (walls, exterior/ interior) and semantic layouts (room type, object type, etc.), rely on extensive human walkthroughs with RGB-D [10, 45] or 3D point cloud [53, 71] scans, and are usually limited to polygonal layouts [10, 45, 53, 71, 78]. Occupancy maps traditionally rely on wide field-of-view (FoV) LiDAR scanners [62] or evaluate on simple 2D environments wihtout non-wall obstacles [23, 39, 68, 68]. More recent methods [4, 5, 11, 60] train an embodied agent to explore and build topdown maps of more complex scenes using RGB-D. On the contrary, our method uses both vision and audio from the observations of a group of conversing people for mapping. Rather than steer the camera of a robot to map the scene, our task requires processing passive video from human camera wearers.

Audio-visual scene mapping. To our knowledge, the only prior work to translate audio-visual inputs into a general (arbitrarily shaped) floorplan maps is AV-Floorplan [59]. Unlike AV-Floorplan, our method maps from speech in natural human conversations, which avoids emitting intrusive frequency sweep signals to generate echoes. In addition, a key goal of our work is to reduce mapping cost by skipping redundant visual frames. Our experiments demonstrate the benefits of our model design over AV-Floorplan [59].

Audio(-visual) spatial understanding. More broadly, beyond the mapping task, various methods leverage audio for geometric and material information about the 3D scene and its constituent objects. Prior work relies on acoustic reflections to estimate the shape of an objects [44]. Echolocation is used in robotics to estimate proximity to surrounding surfaces [1, 15, 24, 69]. Together, vision and audio can better reveal the shape and materials of objects [54, 65, 84], self-supervise imagery [28], and improve depth sensing [40, 81]. Recent work exploits correlations between spatial audio and imagery to reason about scene acoustics [7, 49] or aid active embodied navigation [6, 9, 19, 27, 83] and source separation [47, 48]. No prior work intelligently captures images during conversations to efficiently map a scene.

Multi-agent spatial understanding. There is existing work [17, 33, 35, 36, 57] in the visual multi-agent reinforcement learning (MARL) community that learns collaborative agents for performing tasks like relocating furniture [35, 36], playing 3D multi-player games [34], coordinated scene exploration [33], or multi-object navigation [57]. In such settings, the collaborative agents actively interact with the environment to learn a shared scene representation for successfully completing their task. In contrast, we aim to learn a shared geometric map of a 3D scene given *passive* observations that come from the trajectories chosen by a group of people involved in a natural conversation.

Efficient visual sampling in video. Efficient visual sampling has been studied in the context of video recogni-

[1] Throughout, we call each person participating in the conversation an "ego" for short.

tion [29, 42, 43, 79, 82] and summarization [12, 72] with the goal of selectively and smartly processing informative frames, which can both reduce computational cost and improve recognition performance. More closely related to our approach are methods that use audio for the decision-making [29, 42, 56]. Different from the above, we use efficient visual sampling in the context of mapping scenes. Furthermore, in our case an online sampling decision needs to be made at every step *before* looking at the current visual frame (or frames from future steps).

3. Chat2Map Task Formulation

We propose a novel task: efficient and shared mapping of scenes from multi-ego conversations.

Without loss of generality, we consider two egos, E_1 and E_2, each wearing AR glasses equipped with an RGB-D camera and a multi-channel microphone array. The egos have a conversation and move around in an unmapped 3D environment. Each conversation is T steps long. At each step t, the ego E_i's glasses receives an observation $\mathcal{O}_{i,t} = (\mathcal{V}_{i,t}, \mathcal{S}_{i,t}, P_{i,t}, \mathcal{S}^{'}_{i,t}, P^{'}_{i,t})$. $\mathcal{V}_{i,t}$ is the 90° FOV RGB-D image and $\mathcal{S}_{i,t}$ is the speech waveform uttered by E_i, as observed from its pose $P_{i,t} = (x_{i,t}, y_{i,t}, \theta_{i,t})$, where $(x_{i,t}, y_{i,t})$ denotes its location and $\theta_{i,t}$ denotes its orientation in the 3D scene. $\mathcal{S}^{'}_{i,t}$ is the speech of the other ego $E^{'}_i$ (the other person involved in the conversation), as perceived by E_i (note, the voice sounds different depending on the listener position), and $P^{'}_{i,t}$ is $E^{'}_i$'s pose relative to E_i. Modern AR glasses, like Bose Frames or Facebook Aria already support capturing such multi-sensory observations, making it possible to have a real-world instantiation of our task.

Given the real-time observation stream $\mathcal{O}$ for the egos, where $\mathcal{O} = \{\mathcal{O}_{i,t} : i = 1, \ldots, 2, t = 1 \ldots T\}$ and a total budget of visual frames B, we aim to learn a model that can accurately estimate the top-down occupancy map M of the scene without exceeding the visual budget. We assume the first visual frames (at $t = 0$) for both egos to be observed by the model. Thus we aim to learn a policy that samples B frames from $2 * (T - 1)$ choices—which are not considered a batch, but rather unfold in sequence—and a mapper that predicts the scene map given the sampled frames. Recall that our goal is to build a model that samples the expensive visual frames only when absolutely needed for scene mapping. This is captured by the constraint $1 \leq B << 2 * (T - 1)$.

There are three important aspects to our task. First, it requires learning from both vision and audio. While the visual signal carries rich information about the local scene geometry, there can be a high amount of redundancy in the visual feed captured during a conversation (*e.g.*, the egos may visit the same location more than once or change their viewpoint only marginally). Second, not only does the long-range nature of audio help uncover the global scene properties [21, 59] like shape and size—beyond what's visible in images—we can also exploit audio to undersample the visual frames, thereby reducing the cost of capturing and processing sensory inputs for mapping. Third, shared mapping of a scene implies jointly leveraging the complementary information in the audio (speech) from self and other egos, and the synergy of the audio-visual cues from multiple egos. These insights form the basis of our key hypothesis in this task—selectively sampling visual frames during a conversation involving egos that share information with each other can facilitate efficient mapping of a scene.

4. Approach

We solve the task by learning a model that estimates the scene map given the egos' audio-visual observations and also sequentially decides when to sample visual frames for mapping given the audio stream, ego poses, and previously sampled frames, if any. Here, "sampling" refers to *individually* deciding for *each* ego whether to use its camera or not to capture the visuals at every step of its trajectory in the scene. The sampling is preemptive in nature, *i.e.* the policy selects or skips a frame *without capturing it first*.

Our model has two main components (see Fig. 2): **(1)** a shared scene mapper, and **(2)** a visual sampling policy. At every step t, the shared mapper has two functions. First, it estimates the map of a previously unseen environment by exploiting the shared spatial cues in the audio-visual observations of the two egos. Second, it informs the policy about the utility of sampling a certain visual frame. Guided by the mapper, the policy samples only the most informative visual frames that can boost mapping significantly over using just audio. Note that, unlike the visuals, we observe audio continuously as it is less resource-intensive vis-a-vis storage and power requirements for processing [2].

We learn our task through the synergy of the mapper and the policy, such that under the constraint of a limited visual budget B, our model implicitly understands which visual frames are critical for mapping.

First, we describe the steps involved to prepare our model inputs (Sec. 4.1). Next, we introduce our visual sampling policy (Sec. 4.2) and shared scene mapper (Sec. 4.3). Finally, we present model training details (Sec. 4.4). Through the rest of the text, we use separate notations to distinguish the egos' observations $\mathcal{O}$ (*i.e.* what the egos *receive* from the *environment*) from our model inputs O (*i.e.* what we *capture* and *feed* to our model for efficient mapping).

4.1. Model input preparation

We prepare our model inputs by separately preprocessing the visual and audio modalities. If our policy decides to sample an image $\mathcal{V}$, we transform it into $V = (V^R, V^M)$. V^R denotes the normalized RGB image with pixel values $\in [0, 1]$. V^M denotes the 90° FoV topdown occupancy map created by projecting the depth image. To do the depth

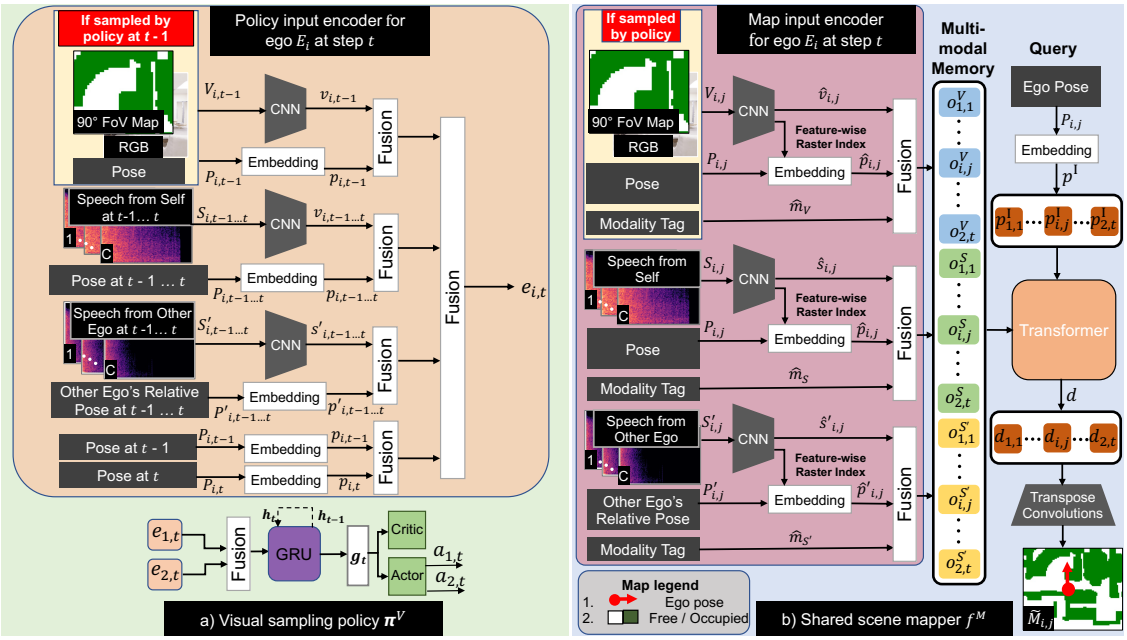


Figure 2. Our model has two main components: a) a visual sampling policy (left), and b) a shared scene mapper (right). At each step, our policy receives the current audio along with the previous audio(-visual) observations for the egos and decides for each ego individually whether to capture its visual frame at the current step. As per the policy predictions, the shared mapper conditionally uses the current visual frame(s) and audio along with the past audio(-visual) observations to predict the occupancy map of the scene, a ground-plane map showing where obstacles and freespace are (shown in green and white).

projection, we first backproject it into the world coordinates using the camera's intrinsic parameters to compute the local visible scene's 3D point cloud. Next, we project these points to obtain a two-channel binary topdown map of size $h \times w \times 2$, where the first channel of the map reveals occupied/free areas, and the second channel reveals seen/unseen areas. If our policy skips $\mathcal{V}$, we set V^R and V^M to all-zero matrices of the appropriate size.

For a speech waveform $\mathcal{S}$, we calculate the short-time Fourier transform (STFT) magnitude spectrogram denoted by S of size $F \times \mathcal{T} \times C$, where F, $\mathcal{T}$, and C are the number of frequency bins, time windows, and ambisonic microphone channels, respectively. Lastly, we normalize each pose $P_{i,t}$ to be relative to $P_{1,1}$. See Sec. 5 and Supp. Sec. 7.6 for more details.

4.2. Visual sampling policy

At every step t, our visual sampling policy π^V (Fig. 2 left) receives $O^\pi(t)$ as input and makes the decision to either capture or skip the visual frame $\mathcal{V}_{i,t}$ for each ego E_i. $O^\pi(t)$ comprises the visual cue from the last step along with the speech cues and the poses from the current step and the last step for both egos. Formally, $O^\pi(t) = \{O_i^\pi(t) : i = 1 \dots 2\}$, where $O_i^\pi(t) = \{V_{i,t-1}, S_{i,j}, P_{i,j}, S'_{i,j}, P'_{i,j} : j = t-1 \dots t\}$. The policy first uses an encoder network to generate a multi-modal embedding of $O^\pi(t)$, and then passes the embedding to a policy network that makes a sampling decision per ego. At $t = 1$, as per our problem definition (Sec. 3), the policy always chooses to sample the visual frames for both egos, i.e., the cameras are initially on.

Multi-modal policy embedding. To process ego E_i's visual input $V_{i,t-1}$ from the last step, we encode the RGB image $V^R_{i,t-1}$ and map $V^M_{i,t-1}$ with separate CNNs. We then concatenate the two features to generate the visual embedding $v_{i,t-1}$. To encode the pose inputs $\{P_{i,t-1}, P'_{i,t-1}, P_{i,t}, P'_{i,t}\}$, we use a linear layer and generate pose embeddings $\{p_{i,t-1}, p'_{i,t-1}, p_{i,t}, p'_{i,t}\}$. We process the speech inputs $\{S_{i,t-1}, S_{i,t-1}, S_{i,t}, S'_{i,t}\}$ using another CNN and create speech embeddings $\{s_{i,t-1}, s'_{i,t-1}, s_{i,t}, s'_{i,t}\}$. Next, we fuse the visual, speech and pose embeddings using linear layers (see Fig. 2 left for details) to obtain the multi-modal policy embedding $e_{i,t}$ for E_i. Finally, we fuse the policy embeddings for the two egos, $e_{1,t}$ and $e_{2,t}$ with a linear layer to produce the multi-modal policy embedding e_t.

The visual, audio, and pose inputs carry complementary cues required for efficient visual sampling. Whereas the pose inputs from the last and current steps explicitly reveal the viewpoint change between the steps, the previous and current speech inputs provide information about the changes in the local and global scene structures as a function of the previously sampled visual inputs, which together suggest the value of sampling a visual frame at the current step. Furthermore, guided by our training reward (below in Sec. 4.4), the previously observed visual frames and audio together enable

our policy to anticipate the current frames and skip them if they are deemed redundant, thereby improving mapping accuracy for a low visual budget.

Policy network. The policy network consists of a GRU that estimates an updated history h_t along with the current state representation g_t, using the fused embedding e_t and the history of states h_{t-1}. An actor-critic module takes g_t and h_{t-1} as inputs and predicts a policy distribution $\pi_\theta(a_{i,t}|g_t, h_{t-1})$ per ego along with the value of the state $H_\theta(g_t, h_{t-1})$ (θ are policy parameters). The policy samples an action $a_{i,t} \in \{0, 1\}$ for every E_i. $a_{i,t} = 1$ corresponds to selecting $\mathcal{V}_{i,t}$, $a_{i,t} = 0$ otherwise.

4.3. Shared scene mapper

Whereas $O^\pi(t)$ denotes our policy input (Sec. 4.2), $O^M(t)$ denotes the input to our shared scene mapper f^M at step t, such that $O^M(t) = \{(V_{i,j}, S_{i,j}, S'_{i,j}, P_{i,j}, P'_{i,j}) : i = 1 \dots 2, j = 1 \dots t\}$. f^m starts by embedding each component of $O^M(t)$ using a separate network. This is followed by a multi-modal memory that stores the embeddings since the start of the episode. Finally, a transformer [76] predicts an estimate $\tilde{M}(t)$ of the scene map conditioned on the multi-modal memory and the egos' poses in the episode.

Multi-modal mapper embedding. For the visual input $V_{i,j}$, we encode $V^R_{i,j}$ and $V^M_{i,j}$ using separate CNNs and do a channel-wise concatenation to get visual features $\hat{v}_{i,j}$. Similarly speech is encoded using separate CNNs to get $\hat{s}_{i,j}$ and $\hat{s}'_{i,j}$. Each of $\hat{v}$, $\hat{s}$ and $\hat{s}'$ is of size $4 \times 4 \times 1024$.

For both vision and speech, we compute two positional embeddings, p^{I} and p^{II}. They encode the pose of the egos in the 3D space, and the index of each 1024-dimensional feature in the visual or speech features in the raster order respectively. Whereas p^{I} helps discover spatial cues as a function of the egos' location in the 3D scene, p^{II} enables our model to attend to different modalities in a more fine-grained manner. For both, we compute an 8-dimensional sinusoidal positional encoding [76] and then pass it through a linear layer to obtain a 1024-dimensional embedding. For p^{II}, we additionally repeat this process for every feature index in the raster order. Lastly, we reshape p^{I} and add it with p^{II} to produce $4 \times 4 \times 1024$-dimensional positional embeddings, $\hat{p}_{i,j}$ for $\hat{v}_{i,j}$ and $\hat{s}_{i,j}$, and $\hat{p}'_{i,j}$ for $\hat{s}'_{i,j}$.

Following [49], we also learn an embedding $\hat{m}_{i,j} \in \{\hat{m}_V, \hat{m}_S, \hat{m}_{S'}\}$ to capture different modality types, where $\hat{m}_V$ represents vision, and $\hat{m}_S$ and $\hat{m}_{S'}$ represent the speech from self and that of the other ego, respectively. The modality-based embeddings help our model differentiate between different modalities and better map the scene by learning complementary spatial cues from them.

Multi-modal memory. For the visual input $V_{i,j}$, we add its embedding $\hat{v}_{i,j}$ with its positional embedding $\hat{p}_{i,j}$ and modality embedding $\hat{m}^V_{i,j}$, and flatten the sum to get a 16×1024-dimensional embedding. Similarly, we fuse the speech embeddings by taking their sum and flattening it. This generates a multi-modal memory of fused embeddings o, such that $o = \{o^V_{1,1}, \dots, o^V_{2,t}, o^S_{1,1}, \dots, o^S_{2,t}, o^{S'}_{1,1}, \dots, o^{S'}_{2,t}\}$.

Occupancy prediction. To predict the underlying scene occupancy, we first use a transformer encoder [76] to attend to the embeddings in o and capture short- and long-range correlations within and across modalities using a stack of self-attention layers. This generates an audio-visual representation that models the spatial layout of the 3D scene.

Next, we use a transformer decoder [76] to perform cross-attention on the audio-visual representation of the scene conditioned on the embedding $\hat{p}_{i,j}$ for every pose $P_{i,j}$ in $O^M(t)$ and generate an embedding $d_{i,j}$ for the pose. Finally, we upsample $d_{i,j}$ using a multi-layer network U comprising transpose convolutions and a sigmoid layer at the end to predict an estimate $\tilde{M}_{i,j}$ of the ground-truth local 360° FoV map for the pose, $M_{i,j}$. Both $M_{i,j}$ and its estimate $\tilde{M}_{i,j}$ are two-channel binary occupancy maps of size $H \times W$. To obtain the estimated map $\tilde{M}(t)$ for the scene, we register each prediction $\tilde{M}_{i,j}$ onto a larger shared map using the pose $P_{i,j}$ and threshold the final shared map at 0.5 (see Supp. Sec. 7.6 for map registration details). Importantly, the shared map allows communication between both egos' data streams for more informed mapping and sampling, as we show in results.

4.4. Model training

Policy training. We propose a novel dense RL reward to train policy π^V:

$$r(t) = \Delta Q(t) - \eta * \rho(t).$$

$\Delta Q(t)$ measures the improvement in mapping from taking actions $\{a_{i,t} : i = 1 \dots 2\}$ over not sampling any visual frame at step t. $\rho(t)$ is a penalty term to discourage sampling a frame from the same pose more than once, which we weight by η. We define $\Delta Q(t)$ as

$$\Delta Q(t) = Q\big(\tilde{M}(t) \mid O^M(t)\big) - Q\big(\tilde{M}(t) \mid (O^M(t) \setminus V_t)\big),$$

where Q is a map quality measure, $Q(X|Y)$ represents the quality of map estimate X given inputs Y, and $(O^M(t) \setminus V_t)$ denotes the mapper inputs devoid of any visual frame for the current step. We define $\rho(t)$ as

$$\rho(t) = \sum_{i=1\dots2} a_{i,t} * \mathbb{1}(V_{i,t} \in O^M(t-1)),$$

where the indicator function checks if $V_{i,t}$ was used in mapping before. While $\Delta Q(t)$ incentivizes sampling frames that provide a big boost to the mapping accuracy over skipping them, $\rho(t)$ penalizes wasting the visual budget on redundant sampling, thereby maximizing mapping performance within

the constraints of a limited budget. We set $\rho = 0.03$ in all our experiments and define Q as the average F1 score over the occupied and free classes in a predicted occupancy map.

We train π^V with Decentralized Distributed PPO (DD-PPO) [77]. The DD-PPO loss consists of a value loss, policy loss and an entropy loss to promote exploration (see Supp. Sec. 7.8.4 for details).

Mapper training. At each step t, we train the shared mapper f^m with a loss $\mathcal{L}^M(t)$, such that

$$\mathcal{L}^M(t) = \frac{1}{2 \times t} \sum_{i=1...2} \sum_{j=1...t} \text{BCE}(\tilde{M}_{i,j}, M_{i,j}),$$

where $\text{BCE}(\tilde{M}_{i,j}, M_{i,j})$ is the average binary cross entropy loss between $\tilde{M}_{i,j}$ and $M_{i,j}$.

Training curriculum. To train our model, we first pretrain mapper f^m in two phases and then train the policy π^V while keeping f^m frozen. In phase 1, we train f^m without visual sampling, *i.e.* all visual frames are provided at each step. In phase 2, we finetune the pretrained weights of f^m from phase 1 on episodes where we randomly drop views to satisfy the budget B. While phase 1 improves convergence when training with visual sampling, phase 2 helps with reward stationarity when training our RL policy.

5. Experiments

Experimental setup. For our main experiments, we use SoundSpaces [8] acoustic simulations with AI-Habitat [63] and Matterport3D [3] visual scenes. While Matterport3D provides dense 3D meshes and image scans of real-world houses and other indoor scenes, SoundSpaces provides room impulse responses (RIRs) at a spatial resolution of 1m for Matterport3D that model all real-world acoustic phenomena [8]. This setup allows us to evaluate with as many as 83 scenes, split in 56/10/17 for train/val/test, compare against relevant prior work [59, 60] and report reproducible results.

We also collect real-world data in a mock-up apartment due to the absence of a publicly available alternative suited for our task. We capture a dense set of RGB images using a Samsung S22 camera and generate the corresponding depth images using monocular depth estimation [22, 38]. To compute the RIRs, following [25], we generate a sinusoidal sweep sound from 20Hz-20kHz with a loudspeaker at source location, capture it with an Eigenmike at a receiver location, and convolve the spatial sound with the inverse of the sweep sound to retrieve the RIR. All capturing devices are placed at a height of 1.5 m. We generate occupancy maps by back-projecting the depth images (cf. Sec. 4.1) and register them onto a shared topdown map before taking egocentric crops to generate the local occupancy inputs and targets.

Note that *both datasets have real-world visuals* as they are captured in the real environments; SoundSpaces has simulated audio while the apartment data has real-world collected audio RIRs.

Conversation episode. For each episode (both simulation and real), we randomly place the two egos in a scene. Episode length is $T = 16$ and 8 for simulation and real resp. At each step, the egos execute a movement from $\mathcal{A} = \{MoveForward, TurnLeft, TurnRight\}$, where $MoveForward$ moves an ego forward by 1 m, and the $Turn$ actions rotate the ego by $90°$. Further, either of the egos speaks or both speak with equal probability of $\frac{1}{3}$ at every step, i.e., there are no moments of silence. The egos stay between $1 - 3$m from each other so that they don't collide and so that each ego is audible by the other at all times. This results in train/val splits of 1,955,334/100 episodes in simulation, and a simulated/real-world test split of 1000/27 episodes. Visual budget $B = 2$ for our main experiments (see Supp. Sec. 7.3 for $B = 4, 6$ evaluations). Note that these episodes are simply to generate video data; our task requires processing passive video, not controlling embodied agents.

Observations and model output. For the occupancy maps, we generate $31 \times 31 \times 2$-dimensional input maps that cover 3.1×3.1 m^2 [4, 11, 60] in area at a resolution of 0.1 m, and set the local target map size to $H \times W = 6.4 \times 6.4$ m^2 (~ 41 m^2). For speech, we use 100 distinct speakers from LibriSpeech [55], split in 80/11 for *heard/unheard*, where *unheard* speech is only used in testing. We assume access to correct camera poses since modern AR devices are equipped with motion sensors that can robustly estimate relative poses [46]. We test our robustness to ambient sounds that get mixed with the egos' speech, and incorporate odometry noise models [59, 60] (see Supp. Sec. 7.4).

Evaluation settings. We evaluate our model in two settings: 1) *passive mapping*, the mapper has access to all visual frames in an episode (i.e., the camera is always-on), and 2) *active mapping*, where the mapping agent has to actively sample frames to meet the visual budget B. This helps disentangle our modeling contributions—whereas *passive mapping* lets us show improvements in the mapper h^M over existing methods [59, 60], *active mapping* helps demonstrate the benefits of smart visual sampling.

We use standard evaluation metrics [60]: **F1 score** and **IoU** (intersection over union) between the predicted and target scene maps. For both metrics, we report the mean over the free and occupied classes. For *active mapping*, we average the metrics over 3 random seeds. We use the following baselines to compare our model's efficacy.

Passive mapping:

- **All-occupied:** a naive baseline that predicts all locations in its map estimate as occupied
- **Register-inputs:** a naive baseline that registers the input maps onto a shared map and uses it as its prediction

Model	Simulation F1 score ↑	Simulation IoU ↑	Real world F1 score ↑	Real world IoU ↑
All-occupied	63.4	48.8	36.2	23.8
Register-inputs	72.6	60.1	50.8	35.0
OccAnt [60]	74.5	62.7	53.9	38.3
AV-Floorplan [59]	79.3	67.9	54.5	38.7
Ours	**81.8**	**71.4**	**55.5**	**39.2**
Ours w/o vision	72.8	60.3	50.8	35.0
Ours w/o audio	78.1	66.7	54.1	38.0
Ours w/o $E_i^{'}$'s speech	81.5	70.9	55.4	39.1
Ours w/o shared mapping	80.7	70.0	54.9	38.6

Table 1. *Passive mapping* performance (%).

Model	F1 score ↑	IoU ↑
All-occupied	63.4	48.8
Register-inputs	72.6	60.1
OccAnt [60]	74.5	62.7
AV-Floorplan [59]	78.7	67.5
Ours	**81.9**	**71.5**
Ours w/o vision	73.5	61.2
Ours w/o audio	78.1	66.7
Ours w/o $E_i^{'}$'s speech	81.5	70.9
Ours w/o shared mapping	80.0	69.1

Table 2. *Passive mapping* performance (%) with ambient sounds.

- **OccAnt [60]:** a vision-only SOTA model that uses the RGB-D images at each step to anticipate the occupancy of the area around an ego that's outside its visible range.
- **AV-Floorplan [59]:** an audio-visual SOTA model that *passively* predicts the floorplan of a scene using a walk-through in it, where the audio is either self-generated or comes from semantic sources in the scene. We adapt the model for our occupancy prediction task and give it the exact same audio-visual observations as our model.

Active mapping:

- **Random:** an agent that selects visual frames randomly for each ego as long as the budget allows
- **Greedy:** an agent that greedily uses up the visual budget by sampling frames as early as possible
- **Unique-pose:** an agent that samples a frame for every new ego pose in the episode

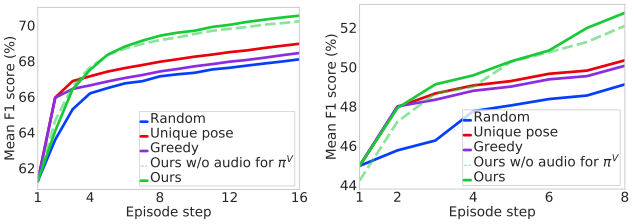
In *active mapping*, we use the model from the second pre-training phase (Sec. 4.4) as the mapper for all models for fair comparison. Thus, any difference in performance is due to the quality of each method's sampling decisions.

See Supp. for all other details like network architectures and training hyperparameters (Sec. 7.8), and baseline implementation (Sec. 7.7).

5.1. Map prediction results

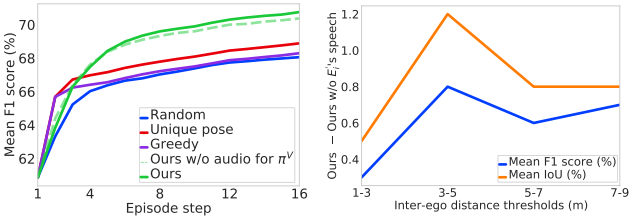
Passive mapping. Table 1 (top) reports the prediction quality of all models in the *passive mapping* setting. Naive baselines (All-occupied, Register-inputs) perform worse than the learned models, showing the complexity of our map prediction task. AV-Floorplan [59] fares the best among all baselines. Its improvement over OccAnt [60] demonstrates the benefits of exploiting the spatial cues in audio for mapping and using an attention-based model to leverage the long- and short-range correlations in the audio-visual inputs.

Our method outperforms all baselines. Its improvement over AV-Floorplan [59] underlines the efficacy of performing attention at different granularities—across modalities, within a single modality and within a single input—guided by our positional and modality type embeddings. It also generalizes to the real-world setting and retains its benefits over the baselines, even without retraining on the real-world data. However, we do observe a drop in performance gains, probably due to the large sim-to-real gap.



(a) Simulation **(b)** Real world

Figure 3. *Active mapping* performance vs. episode step.



(a) Effect of ambient sounds **(b)** Impact of ego $E_i^{'}$'s speech

Figure 4. (a) Effect of ambient environment sounds on *active mapping* (b) Impact of the other ego's speech on *passive mapping* vs. distance between the egos.

Active mapping. Fig. 3 shows the active mapping performance as a function of episode progress. Employing naive heuristics for sampling, like Random or Greedy, isn't enough for high-quality mapping, which emphasizes the high levels of redundancy in the visual frames. Unique-pose improves over both Random and Greedy, showing that sampling diverse viewpoints provides more information about the underlying scene geometry.

Even though the baselines make progress initially, they flatten quickly and our model eventually outperforms them all, on both real-world and simulated data. This highlights the benefits of learning a smart policy that, given the audio streams and its visual samples from the past, understands the value of sampling a visual frame for mapping by taking

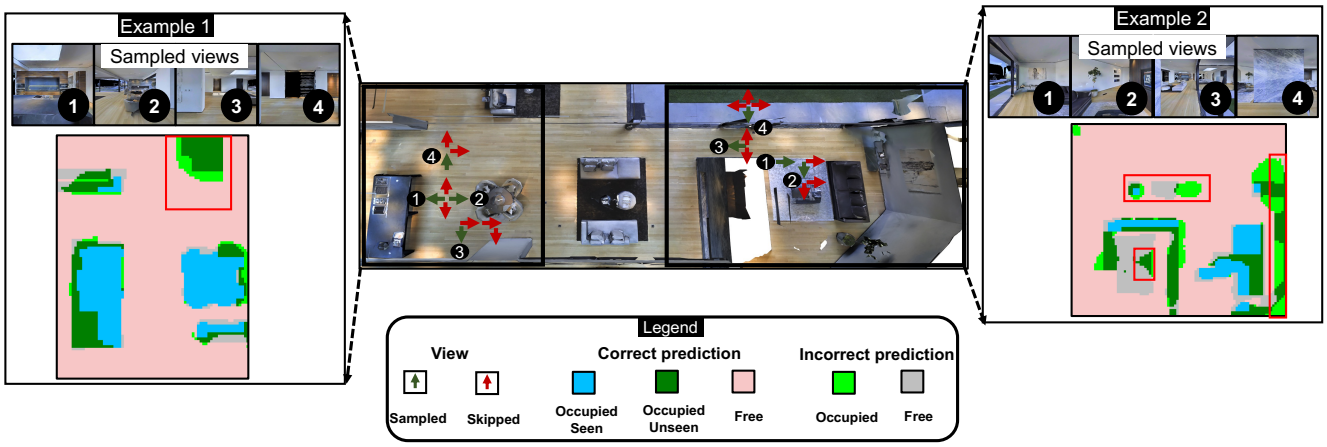


Figure 5. Sample episodes for our *active mapping* model. While our policy samples only the salient visual frames, our mapper can both complete partially seen objects as well as anticipate objects never seen before in the sampled visuals (red boxes on the maps).

cues from our novel reward. Moreover, on the real-world data, we see improved performance margins over the baselines towards end of episodes, showing that our policy can adaptively postpone visual sampling to improve mapping. Owing to our smart sampling, the per-episode reduction in processing for $B = 2$ is 7.2 GFLOPS in simulation and 3.6 GFLOPS for the real-world data.

5.2. Model analysis

Ablations. In Table 1 (bottom), we ablate the components of our model for *passive mapping*. Upon removing audio, our model experiences a large drop in mapping performance, which indicates that our model leverages complementary spatial cues in audio and vision. We also see a drop in the map quality when our model doesn't have access to the speech from the other ego ($E_i^{'}$). This shows that $E_i^{'}$'s speech can better reveal the more global scene geometry than E_i's own speech. Fig. 4b further shows that the impact of the other ego's speech becomes more prominent for larger inter-ego distances ($3 - 5$ m vs. $1 - 3$ m), in which case the two types of speech are dissimilar enough to carry complementary geometric cues, but reduces for even larger distances (5 m or more), in which case $E_i^{'}$ is too far for its speech to carry useful cues about E_i's local scene geometry. Moreover, unlike the ablation that doesn't perform shared mapping, our model benefits significantly from *jointly attending* to the observations of the egos and exploiting the complementary information in them—even though both models use the exact same audio-visual observations, including both speech from self and the other ego.

For *active mapping*, Fig. 3 shows a drop in the mapping performance upon removing audio from the policy inputs. This implies that our policy exploits audio to reason about the level of redundancy in a new visual frame and improve the mapping quality vs. visual budget tradeoff. On the more challenging real-world setting, audio plays an even bigger role, as shown by the larger performance drop in Fig. 3b.

See Supp. for similar results with 1) *unheard* speech (Sec. 7.2), 2) higher values of budget B (Sec. 7.3), 3) sensor noise (Sec. 7.4), and 4) larger target map sizes (Sec. 7.5).

Ambient and background sounds. We also test our model's robustness to ambient and background sounds by inserting a non-speech sound (*e.g.* running AC, dog barking, etc.) at a random location outside the egos' trajectories. Although quite challenging, our model performs better than the baselines for both *passive* (Table 2) and *active mapping* (Fig. 4a). Hence, even without explicit audio separation, our model is able to implicitly ground its audio representations in the corresponding pose features for accurate mapping.

Qualitative results. Fig. 5 shows two successful *active mapping* episodes of our method. Note how our model samples views that tend have to little visual overlap but are informative of the surrounding geometry (both occupied and free spaces). Besides, it is able to complete structures only partially visible in the sampled views, and more interestingly, leverage the synergy of audio and vision to anticipate unseen areas (red boxes on the occupancy maps in Fig. 5).

Failure cases. We notice two common failure cases with *active mapping*: episodes where the people stay at the same location, leading to very few informative visual frames to sample from; and episodes with highly unique visual samples at every trajectory step, in which case each sample is useful and our model behaves similar to Unique-pose or Greedy. For *passive mapping*, our model fails with very complex scenes that commonly have objects in spaces where both vision and audio can't reach (*e.g.* narrow corners)

6. Conclusion

We introduce Chat2Map, a new task aimed at scene mapping using audio-visual feeds from egocentric conversations. We develop a novel approach for Chat2Map comprised of a shared scene mapper and a visual sampling policy based on a novel reinforcement learner that smartly samples the visuals only when necessary. We show promising performance on both simulated data and real-world data from over 80 environments.

References

[1] *Ego-Noise Predictions for Echolocation in Wheeled Robots*, volume ALIFE 2019: The 2019 Conference on Artificial Life of *ALIFE 2022: The 2022 Conference on Artificial Life*, 07 2019. 1, 2

[2] Aaron Carroll and Gernot Heiser. An analysis of power consumption in a smartphone. In *2010 USENIX Annual Technical Conference (USENIX ATC 10)*, 2010. 2, 3

[3] 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. *arXiv preprint arXiv:1709.06158*, 2017. 6

[4] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations*, 2020. 1, 2, 6

[5] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020. 2

[6] Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15516–15525, 2021. 2

[7] Changan Chen, Ruohan Gao, Paul T. Calamia, and Kristen Grauman. Visual acoustic matching. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18836–18846, 2022. 2

[8] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *ECCV*, 2020. 1, 6, 13, 14

[9] Changan Chen, Sagnik Majumder, Ziad Al-Halah, Ruohan Gao, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Learning to set waypoints for audio-visual navigation. In *International Conference on Learning Representations*, 2021. 2

[10] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2661–2670, 2019. 1, 2

[11] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. *arXiv preprint arXiv:1903.01959*, 2019. 1, 2, 6

[12] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 358–373, 2018. 3

[13] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565, 2015. 14

[14] Jesper Christensen, Sascha Hornauer, and Stella Yu. Batvision - learning to see 3d spatial layout with two ears. In *ICRA*, 2020. 1

[15] Jesper Haahr Christensen, Sascha Hornauer, and Stella X. Yu. Batvision: Learning to see 3d spatial layout with two ears. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1581–1587, 2020. 1, 2

[16] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 16

[17] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Michael G. Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. *ArXiv*, abs/1810.11187, 2019. 2

[18] Samyak Datta, Sameer Dharur, Vincent Cartillier, Ruta Desai, Mukul Khanna, Dhruv Batra, and Devi Parikh. Episodic memory question answering. 2022. 1

[19] Victoria Dean, Shubham Tulsiani, and Abhinav Gupta. See, hear, explore: Curiosity via audio-visual association. *Advances in Neural Information Processing Systems*, 33:14961–14972, 2020. 2

[20] Helisa Dhamo, Nassir Navab, and Federico Tombari. Object-driven multi-layer scene decomposition from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5369–5378, 2019. 2

[21] Ivan Dokmanić, Reza Parhizkar, Andreas Walther, Yue M. Lu, and Martin Vetterli. Acoustic echoes reveal room shape. *Proceedings of the National Academy of Sciences*, 110(30):12186–12191, 2013. 3

[22] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 6

[23] Amine Elhafsi, Boris Ivanovic, Lucas Janson, and Marco Pavone. Map-predictive motion planning in unknown environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8552–8558. IEEE, 2020. 2

[24] Itamar Eliakim, Zahi Cohen, Gábor Kósa, and Yossi Yovel. A fully autonomous terrestrial bat-like acoustic robot. *PLoS Computational Biology*, 14, 2018. 1, 2

[25] Angelo Farina. Simultaneous measurement of impulse response and distortion with a swept-sine technique. *Journal of The Audio Engineering Society*, 2000. 6

[26] Chuang Gan, Jeremy Schwartz, Seth Alter, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, Kuno Kim, Elias Wang, Damian Mrowca, Michael Lingelbach, Aidan Curtis,

Kevin T. Feigelis, Daniel M. Bear, Dan Gutfreund, David D. Cox, James J. DiCarlo, Josh H. McDermott, Joshua B. Tenenbaum, and Daniel L. K. Yamins. Threedworld: A platform for interactive multi-modal physical simulation. In *NeurIPS Track on Datasets and Benchmarks*, 2021. 1

[27] Chuang Gan, Yiwei Zhang, Jiajun Wu, Boqing Gong, and Joshua B. Tenenbaum. Look, listen, and act: Towards audio-visual embodied navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9701–9707, 2020. 2

[28] Ruohan Gao, Changan Chen, Ziad Al-Halah, Carl Schissler, and Kristen Grauman. Visualechoes: Spatial image representation learning through echolocation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 658–676, Cham, 2020. Springer International Publishing. 1, 2

[29] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10454–10464, 2020. 3

[30] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 16

[32] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. 15, 16

[33] Shariq Iqbal and Fei Sha. Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning. *arXiv preprint arXiv:1905.12127*, 2019. 2

[34] Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio García Castañeda, Charlie Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364:859 – 865, 2019. 2

[35] Unnat Jain, Luca Weihs, Eric Kolve, Ali Farhadi, Svetlana Lazebnik, Aniruddha Kembhavi, and Alexander G. Schwing. A cordial sync: Going beyond marginal policies for multi-agent embodied tasks. *ArXiv*, abs/2007.04979, 2020. 2

[36] Unnat Jain, Luca Weihs, Eric Kolve, Mohammad Rastegari, Svetlana Lazebnik, Ali Farhadi, Alexander G. Schwing, and Aniruddha Kembhavi. Two body problem: Collaborative visual task completion. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6682–6692, 2019. 2

[37] Hao Jiang, Calvin Murdock, and Vamsi Krishna Ithapu. Egocentric deep multi-channel audio-visual active speaker localization. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10534–10542, 2022. 15

[38] Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18963–18974, 2022. 6

[39] Kapil Katyal, Katie Popek, Chris Paxton, Phil Burlina, and Gregory D. Hager. Uncertainty-aware occupancy map prediction using generative networks for robot navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5453–5459, 2019. 2

[40] Hansung Kim, Luca Remaggi, Philip JB Jackson, Filippo Maria Fazi, and Adrian Hilton. 3d room geometry reconstruction using audio-visual sensors. In *2017 International Conference on 3D Vision (3DV)*, pages 621–629, 2017. 2

[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 16

[42] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6231–6241, 2019. 3

[43] Jintao Lin, Haodong Duan, Kai Chen, Dahua Lin, and Limin Wang. Ocsampler: Compressing videos to one clip with single-step sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13894–13903, 2022. 3

[44] David B. Lindell, Gordon Wetzstein, and Vladlen Koltun. Acoustic non-line-of-sight imaging. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6773–6782, 2019. 1, 2

[45] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floornet: A unified framework for floorplan reconstruction from 3d scans. In *Proceedings of the European conference on computer vision (ECCV)*, pages 201–217, 2018. 1, 2

[46] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel.

Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020. 6

[47] Sagnik Majumder, Ziad Al-Halah, and Kristen Grauman. Move2hear: Active audio-visual source separation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 275–285, 2021. 2

[48] Sagnik Majumder, Ziad Al-Halah, and Kristen Grauman. Active audio-visual separation of dynamic sound sources. In *European Conference on Computer Vision*. Springer, 2022. 2

[49] Sagnik Majumder, Changan Chen, Ziad Al-Halah, and Kristen Grauman. Few-shot audio-visual learning of environment acoustics. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. 2, 5

[50] Vinod Nair and Geoffrey E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814. Omnipress, 2010. 15, 16

[51] Medhini Narasimhan, Erik Wijmans, Xinlei Chen, Trevor Darrell, Dhruv Batra, Devi Parikh, and Amanpreet Singh. Seeing the un-scene: Learning amodal semantic maps for room navigation. In *European Conference on Computer Vision*, pages 513–529. Springer, 2020. 2

[52] Simon T O'Callaghan and Fabio T Ramos. Gaussian process occupancy maps*. *Int. J. Rob. Res.*, 31(1):42–62, jan 2012. 2

[53] Brian Okorn, Xuehan Xiong, and Burcu Akinci. Toward automated modeling of floor plans. In *3D PVT*, 2009. 1, 2

[54] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H. Adelson, and William T. Freeman. Visually indicated sounds. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2405–2413, 2016. 2

[55] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015. 6

[56] Rameswar Panda, Chun-Fu Richard Chen, Quanfu Fan, Ximeng Sun, Kate Saenko, Aude Oliva, and Rogerio Feris. Adamml: Adaptive multi-modal learning for efficient video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7576–7585, 2021. 3

[57] Shivansh Patel, Saim Wani, Unnat Jain, Alexander G. Schwing, Svetlana Lazebnik, Manolis Savva, and Angel X. Chang. Interpretation of emergent communication in heterogeneous collaborative embodied agents. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15993–15943, 2021. 2

[58] Katharine Patterson, Kevin W. Wilson, Scott Wisdom, and John R. Hershey. Distance-based sound separation. In *INTERSPEECH*, 2022. 15

[59] Senthil Purushwalkam, Sebastia Vicenc Amengual Gari, Vamsi Krishna Ithapu, Carl Schissler, Philip Robinson, Abhinav Gupta, and Kristen Grauman. Audio-visual floorplan reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1183–1192, 2021. 1, 2, 3, 6, 7, 13, 14

[60] Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *European Conference on Computer Vision*, pages 400–418. Springer, 2020. 1, 2, 6, 7, 13, 14

[61] Fabio Ramos and Lionel Ott. Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. *The International Journal of Robotics Research*, 35(14):1717–1730, 2016. 2

[62] João Machado Santos, David Portugal, and Rui P. Rocha. An evaluation of 2d slam techniques available in robot operating system. In *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6, 2013. 2

[63] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 6, 14

[64] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *CoRR*, abs/1312.6120, 2014. 16

[65] Carl Schissler, Christian Loftin, and Dinesh Manocha. Acoustic classification and optimization for multi-modal rendering of real-world scenes. *IEEE Transactions on Visualization and Computer Graphics*, 24:1246–1259, 2018. 2

[66] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *CoRR*, 2016. 16

[67] Ransalu Senanayake, Thushan Ganegedara, and Fabio Ramos. Deep occupancy maps: a continuous mapping technique for dynamic environments. In *NIPS 2017 Workshop MLITS*, 2017. 2

[68] Rakesh Shrestha, Fei-Peng Tian, Wei Feng, Ping Tan, and Richard Vaughan. Learned map prediction for enhanced mobile robot exploration. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1197–1204, 2019. 2

[69] Jascha Sohl-Dickstein, Santani Teng, Benjamin M. Gaub, Chris C. Rodgers, Crystal Li, Michael R. DeWeese, and Nicol S. Harper. A device for human ultrasonic echolocation. *IEEE Transactions on Biomedical Engineering*, 62(6):1526–1534, 2015. 1, 2

[70] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 16

[71] Wei Sui, Lingfeng Wang, Bin Fan, Hongfei Xiao, Huaiyu Wu, and Chunhong Pan. Layer-wise floorplan extraction for automatic urban building reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 22(3):1261–1277, 2016. 2

[72] Maitreya Suin and A. N. Rajagopalan. An efficient framework for dense video captioning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):12039–12046, Apr. 2020. 3

[73] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1047–1056, 2019. 2

[74] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. *2015*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2892–2900, 2015. 15, 16

[75] Ryu Takeda, Yoshiki Kudo, Kazuki Takashima, Yoshifumi Kitamura, and Kazunori Komatani. Unsupervised adaptation of neural networks for discriminative sound source localization with eliminative constraint. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3514–3518, 2018. 14

[76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5, 16

[77] Erik Wijmans, Abhishek Kadian, Ari S. Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *ICLR*, 2020. 6, 16

[78] Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. Data-driven interior plan generation for residential buildings. *ACM Trans. Graph.*, 38(6), nov 2019. 2

[79] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1278–1287, 2019. 3

[80] Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3363–3372, 2019. 2

[81] Mao Ye, Yu Zhang, Ruigang Yang, and Dinesh Manocha. 3d reconstruction in the presence of glasses by acoustic and stereo fusion. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4885–4893, 2015. 2

[82] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2678–2687, 2016. 3

[83] Yinfeng Yu, Wenbing Huang, Fuchun Sun, Changan Chen, Yikai Wang, and Xiaohong Liu. Sound adversarial audio-visual navigation. In *International Conference on Learning Representations*, 2022. 2

[84] Zhoutong Zhang, Jiajun Wu, Qiujia Li, Zhengjia Huang, James Traer, Josh H. McDermott, Joshua B. Tenenbaum, and William T. Freeman. Generative modeling of audible shapes for object perception. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1260–1269, 2017. 2

[85] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2051–2059, 2018. 2

[86] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. Manhattan room layout reconstruction from a single 360° image: A comparative study of state-of-the-art methods. *International Journal of Computer Vision*, 129(5):1410–1431, 2021. 2

7. Supplementary Material

In this supplementary material we provide additional details about:

- Video (with audio) for qualitative illustration of our task and qualitative assessment of our map predictions (Sec. 7.1)
- Experiment to show the effect of *unheard* sounds (Sec. 5 in main) on map predictions (Sec. 7.2), as noted in Sec. 5.2 in main
- Experiment to show the impact of the visual budget B (Sec. 3 in main) on mapping quality (Sec. 7.3), as referenced in Sec. 5 and 5.2 in main.
- Experiment to show the effect of sensor noise on mapping accuracy (Sec. 7.4), as mentioned in Sec. 5 and 5.2 in main.
- Experiment to show mapping performance as function of the target map size (Sec. 7.5), as noted in Sec. 5.2 in main.
- Dataset details (Sec. 7.6) in addition to what's provided in Sec. 5 in main.
- Additional baseline details for reproducibility (Sec. 7.7), as referenced in Sec. 5 in main.
- Architecture and training details (Sec. 7.8), as noted in Sec. 5 in main.

7.1. Supplementary video

The supplementary video qualitatively depicts our task, Chat2Map:Efficient Scene Mapping from Multi-Ego Conversations. Moreover, we qualitatively show our model's mapping quality by comparing the predictions against the ground truths and the visual samples chosen by our sampling policy for efficient mapping. Please use headphones to hear the spatial audio correctly. We also demonstrate the acoustically realistic SoundSpaces [8] audio simulation platform that we use for our core experiments. The video is available at `http://vision.cs.utexas.edu/projects/chat2map`.

7.2. Unheard sounds

In Sec. 5.1 in main, we showed results with *heard* sounds (Sec. 5 in main), *i.e.* the *anechoic* speech sounds uttered by the egos are shared between train and test splits. However, due to our use of *unseen* environments in test (Sec. 5 in main), the spatial speech sounds input to our model during test are not heard in training. To make the evaluation even more challenging, we conduct a parallel experiment here, where even the anechoic speech is distinct from what's used in training, which we call as the *unheard* sound setting (Sec. 5 in main).

Model	F1 score ↑	IoU ↑
All-occupied	63.4	48.8
Register-inputs	72.6	60.1
OccAnt [60]	74.5	62.7
AV-Floorplan [59]	79.0	67.7
Ours	**81.6**	**71.1**
Ours w/o vision	72.6	60.1
Ours w/o audio	78.1	66.7
Ours w/o E_i''s speech	81.3	70.7
Ours w/o shared mapping	80.7	70.0

Table 3. *Passive mapping* performance (%) on *unheard* sounds.

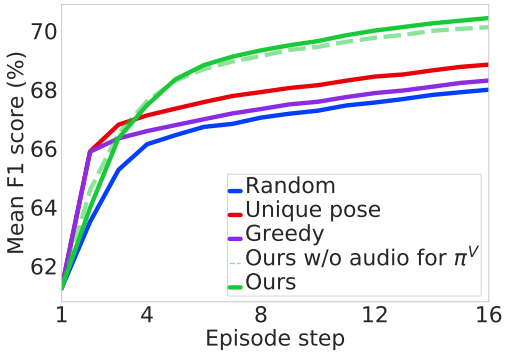


Figure 6. *Active mapping* performance vs. episode step on *unheard* sounds.

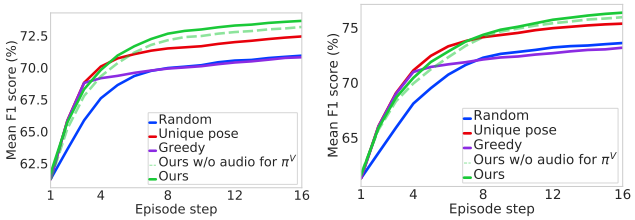
Table 3 shows our *passive mapping* results in the *unheard* sound setting. Our model is able to retain its performance margins over all baselines even in this more challenging scenario.

We notice a similar trend upon evaluating our model for *active mapping* on *unheard* sounds. Fig. 6 shows that our model is able to generalize to novel sounds better than all baselines.

This indicates that both our mapper f^M and visual sampling policy π^V are able to learn useful spatial cues from audio that are agnostic of the speech content and semantics.

7.3. Visual budget value

So far, we have shown *active mapping* results with the visual budget set to $B = 2$ (Sec. 5.1 and Fig. 3 in main). To analyze the effect of larger values of B, we show our *active mapping* performance for $B \in \{4, 6\}$ in Fig. 7. Our model outperforms all baselines even for these larger B values. We also observe that the lower the visual budget, the higher the performance margins are for our model. This shows that our model is particularly more robust to the lack of visuals in extremely low-resource settings.



(a) $B = 4$ **(b)** $B = 6$

Figure 7. *Active mapping* performance vs. episode step with $B \in \{4, 6\}$.

Model	F1 score ↑	IoU ↑
All-occupied	63.0	48.3
Register-inputs	72.3	59.7
OccAnt [60]	74.7	63.0
AV-Floorplan [59]	77.6	65.8
Ours	**79.1**	**68.0**
Ours w/o vision	72.6	60.0
Ours w/o audio	76.7	65.1
Ours w/o $E_i^{'}$'s speech	78.8	67.7
Ours w/o shared mapping	78.5	67.2

Table 4. *Passive mapping* performance (%) with sensor noise.

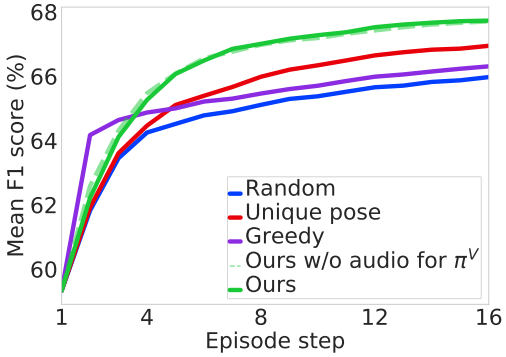


Figure 8. *Active mapping* performance vs. episode step with sensor noise.

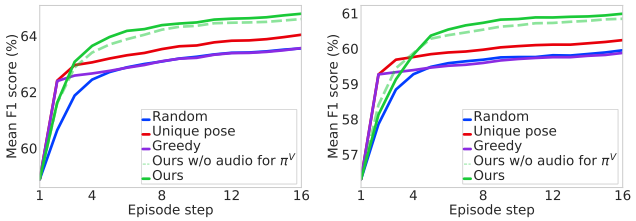
7.4. Sensor noise

Here, we test our model's robustness to sensor noise by adding noise of the appropriate type individually to each sensor. For RGB images, we sample the noise from a Gaussian distribution with a mean of 0 and a standard deviation of 0.2 [60, 63]. For depth, we use the Redwood depth noise model [13, 60, 63], where the amount of noise is higher for higher depth values and vice-versa. Following [60], we sample pose noise from a truncated Gaussian with a mean of 0.025 m and a standard deviation of 0.001 m for the spatial location component of an ego pose $\big((x, y)$ in Sec. 3 in main$\big)$. For orientation θ (Sec. 3 in main), we use another truncated Gaussian with a mean of 0.9° and a standard deviation of 0.057°. Both distributions are truncated at 2 standard deviations. For our multi-channel microphones (Sec. 3 in main), we add a high amount of noise (SNR of 40 dB) [8] using a standard noise model [13, 75].

Table 4 and Fig. 8 report our *passive* and *active mapping* performance, respectively, in the face of sensor noise. In both settings, although our model's performance declines in comparison to the noise-free setting (cf. Table 1 and Fig. 3 in main), it generalizes better than all baselines, thereby underlining the effectiveness of our method.

Model	$H = W = 8$ m F1 score ↑	$H = W = 8$ m IoU ↑	$H = W = 9.6$ m F1 score ↑	$H = W = 9.6$ m IoU ↑
All-occupied	53.5	37.9	46.4	31.2
Register-inputs	65.9	53.4	61.6	49.6
OccAnt [60]	67.8	55.7	63.0	51.3
AV-Floorplan [59]	71.4	59.1	68.7	53.1
Ours	**73.4**	**60.7**	**72.0**	54.4
Ours w/o vision	66.1	53.5	62.6	50.3
Ours w/o audio	71.1	58.1	63.8	51.3
Ours w/o $E_i^{'}$'s speech	73.3	60.5	67.6	54.0
Ours w/o shared mapping	72.9	60.3	68.0	**54.5**

Table 5. *Passive mapping* performance (%) for larger target map sizes.



(a) $H = W = 8$ m **(b)** $H = W = 9.6$ m

Figure 9. *Active mapping* performance vs. episode step for larger target map sizes.

7.5. Target map size

In main (Sec. 5.1), we showed mapping results with $H \times W = 6.4 \times 6.4$ m^2(~ 41 m^2), where H and W denote the height and width of the ground-truth local 360° FoV maps (Sec. 4.3 in main). To analyze the impact of larger target map sizes on the mapping quality, we also test our model with $H \times W \in \{8 \times 8$ m^2(64 m^2), 9.6×9.6 m^2(~ 92 m^2)$\}$. Table 5 and Fig. 9 show the corresponding results for *passive* and *active mapping*, respectively. In both cases, our model outperforms all baselines by a substantial margin, showing that our method is also robust to higher target map sizes.

7.6. Dataset details

Here, we provide additional dataset details. We will release our datasets.

Visual data. All RGB-D images in our experiments have a resolution of 128×128.

To generate the topdown occupancy maps, we threshold the local pointcloud computed from the $90°$ FoV depth images (Sec. 4.1 in main) using a lower and upper height limit of 0.2 and 1.5 m, respectively, such that a map cell is considered occupied if there is a 3D point for it in the 0.2-1.5 m range, and free otherwise.

To generate an estimate of the scene map, we register the estimates of ground-truth local $360°$ FoV maps, $\tilde{M}_{i,j}$s onto a shared scene map $\tilde{M}$ (Sec. 4.3 in main) and maintain a count of the number of updates undergone by every cell in the shared map. To register a local estimate $\tilde{M}_{i,j}$, we first translate and rotate $\tilde{M}_{i,j}$ within $\tilde{M}$ on the basis of its normalized pose $P_{i,j}$. Next, we add $\tilde{M}_{i,j}$ with the corresponding part of $\tilde{M}$ and update the counter for every map cell that's been changed through the registration. We repeat this process for every $\tilde{M}_{i,j}$ in the episode. Finally, we normalize the updated $\tilde{M}$ by dividing each cell in it by its number of updates from the counter, and thresholding at 0.5. In our experiments, $\tilde{M}$ covers a maximum area of 128.4×128.4 m^2.

Audio data. For each conversation episode, we randomly choose 2 speakers from the same split – *heard* or *unheard* (Sec. 5 in main). Starting at a random time in the audio clip for each speaker, we choose contiguous 3s slices from each clip for T steps to use as the anechoic audio for the two egos in the episode, where T denotes the episode length (Sec. 3 in main). Further, we normalize each slice to have the same RMS value of 400 across the whole dataset, where all audio is sampled at 16 kHz and stored using the standard 16-bit integer format.

To generate the spectrograms, we convolve a speech slice with the appropriate 9-channel RIR sampled at 16 kHz and compute its STFT with a Hann window of 31.93 ms, hop length of 8.31 ms, and FFT size of 511 to generate 9-channel magnitude spectrograms, where each channel has 256 frequency bins and 257 overlapping temporal windows. We assume access to detected and separated speech from the egos at all times since on-device microphones of AR glasses can tackle nearby and distant speaker detection [37] and separation [58].

7.7. Baselines

Here, we provide additional implementation details for our *active mapping* baselines for reproducibility (Sec. 5 in main).

- **Random.** At each step t, we generate a random number between 0 and 1 from a uniform distribution. Depending on which quartile of the 0-1 range the random number lies in, we skip visual frames for both egos, sample for just one ego, or sample for both egos.
- **Greedy.** Starting at $t = 2$, we sample visual frames for both egos at every step until we run out of the visual budget B. If the value of B is such that it allows sampling only one visual frame at a certain step (*i.e.* B is odd), we randomly choose the ego for which we sample the frame at that step.
- **Unique-pose.** To implement this baseline, we keep track of the egos' poses during an episode. At any step t, we sample the frame for an ego if it's current pose has never been assumed before by either of the egos in that episode.

7.8. Architecture and training

Here, we provide our architecture and additional training details for reproducibility. We will release our code.

7.8.1 Policy architecture

Visual encoder. To encode local occupancy map inputs, our policy π^V (Sec. 4.2 in main) uses a 6-layer CNN consisting of 5 convolutional (conv.) layers followed by an adaptive average pooling layer. The first three conv. layers use a kernel size of 4 and a stride of 2, while the last two conv. layers use a kernel size of 3 and a stride of 1. All conv. layers use a zero padding of 1, except for the third conv. layer, which uses a zero padding of 2. The number of output channels of the conv. layers are [64, 64, 128, 256, 512], respectively. Each convolution is followed by a leaky ReLU [50, 74] activation with a negative slope of 0.2, and a Batch Normalization [32] of $1e^{-5}$. The adaptive average pooling layer reduces the output of the last conv. layer to a feature of size $1 \times 1 \times 512$.

To encode RGB images (Sec. 4.2 in main), π^V uses a separate CNN with 5 conv. layers and an adaptive average pooling layer. Each conv. layer has a kernel size of 4, stride of 2 and zero padding of 1. The number of output channels are [64, 64, 128, 256, 512], respectively. Similar to the occupancy map encoder, each convolution is followed by a leaky ReLU [50, 74] activation with a negative slope of 0.2 and a Batch Normalization [32] of $1e^{-5}$, and the adaptive average pooling layer reduces the output of the last conv. layer to a feature of size $1 \times 1 \times 512$.

We fuse the occupancy and RGB features by concatenating them and passing through a single linear layer that produces a 512-dimensional visual embedding v (Sec. 4.2 in main).

Speech encoder. The speech encoder (Sec. 4.2 in main) in π^V is a CNN with 5 conv. layers and an adaptive average pooling layer. Each conv. layer has a kernel size of 4, stride

of 2 and a padding of 1, except for the second conv. layer, which has a kernel size of 8, stride of 4 and padding of 3. The number of channels in the CNN are [64, 64, 128, 256, 512], respectively. Similar to the visual encoder, each conv. layer is followed by a leaky ReLU [50, 74] with a negative slope of 0.2 and a Batch Normalization [32] of $1e^{-5}$. The adaptive average pooling layer reduces the output of the last conv. layer to a feature of size $1 \times 1 \times 512$.

Pose encoder. The pose encoder (Sec. 4.2 in main) in π^V is a single linear layer that takes a normalized pose P (Sec. 4.1 in main) as input and produces a 32-dimensional pose embedding.

Fusion layers. We perform linear fusion of the visual, speech and pose embeddings (Sec. 4.2 and Fig. 2 in main) at two levels. The first level has 4 linear layers and the second level has 1 linear layer. Each linear layer produces a 512-dimensional fused feature as its output.

Policy network. The policy network (Sec. 4.2 in main) comprises a one-layer bidirectional GRU [16] with 512 hidden units. The actor and critic networks consist of one linear layer.

7.8.2 Mapper architecture

Visual encoder. To encode local occupancy map inputs, our shared mapper f^M (Sec. 4.3 in main) uses a CNN similar to the one used for encoding occupancy maps in π^V (Sec.), except that it doesn't have a pooling layer at the end. The RGB encoder (Sec. 4.3 in main) in f^M is also similar to the one for π^V, except that it also doesn't have a pooling layer at the end. We fuse the map and RGB features by concatenating them along the channel dimension, and obtain a $4 \times 4 \times 1024$ dimensional feature.

Speech encoder. The speech encoders (Sec. 4.3 in main) in f^M are CNNs with 5 layers that share the architecture with the first 5 conv. layers of the speech encoder in π^V (Sec. 7.8.1), except that the last conv. layer in both encoders has 1024 output channels.

Modality encoder. For our modality embedding $\hat{m}$ (Sec. 4.3 in main), we maintain a sparse lookup table of 1024-dimensional learnable embeddings, which we index with 0 to retrieve the visual modality embedding ($\hat{m}_V$), 1 to retrieve the modality embedding ($\hat{m}_S$) for the speech from self, and 2 to retrieve the modality embedding ($\hat{m}_{S'}$) for the speech from the other ego.

Occupancy prediction network. The transformer [76] (Sec. 4.3 in main) in our occupancy prediction network comprises 6 encoder and 6 decoder layers, 8 attention heads, an input and output size of 1024, a hidden size of 2048, and ReLU [50, 74] activations. Additionally, we use a dropout [70] of 0.1 in our transformer.

The transpose convolutional network U (Sec. 4.3 in main) consists of 6 layers in total. The first 5 layers are transpose convolutions (conv.) layers. The first 4 transpose conv. layers have a kernel size of 4 and stride of 2, and the last transpose conv. layer has a kernel size of 3 and stride of 1. Each transpose conv. has a padding of 1, ReLU [50, 74] activation and Batch Normalization [32]. The number of the output channels for the transpose conv. layers are [512, 256, 128, 64, 2], respectively. The last layer in U is a sigmoid layer (Sec. 4.3 in main), which outputs the map estimates.

7.8.3 Parameter initialization

We use the Kaiming-normal [31] weight initialization strategy to initialize the weights of all our network modules, except for the pose encoding layers and fusion layers, which are initialized with Kaiming-uniform [31] initialization, and the policy network, which is initialized using the orthogonal initialization strategy [64]. We switch off biases in all network modules, except for the policy network where we set the biases initially to 0.

7.8.4 Training hyperparameters.

Policy training. To train our policy π^V using DD-PPO [77] (Sec. 4.4 in main), we weight the action loss by 1.0, value loss by 0.5, and entropy loss by 0.05. We train our policy on 8 Nvidia Tesla V100 SXM2 GPUs with Adam [41], an initial learning rate of $1e^{-4}$ and 8 processes per GPU for 8.064 million policy prediction steps. Among other policy training parameters, we set the clip parameter value to 0.1, number of DD-PPO epochs to 4, number of mini batches to 1, max gradient norm value to 0.5, reward discount factor γ to 0.99, and the value of λ in the generalized advantage estimation [66] formulation for DD-PPO to 0.95.

Mapper training. We train our shared scene mapper f^M (Sec. 4.3 in main) with a binary cross entropy loss (Sec. 4.4 in main) on 4 Nvidia Quadro RTX 6000 GPUs until convergence by using Adam [41], an initial learning rate of $1e^{-4}$ and a batch size of 24.